## 2 Infinite-Horizon Representative Agent Model

Premise

1. There is one representative agent who lives forever. This is justified when you think of agents as households. Members are periodically born and die, but households sustain and, therefore, become infinite.

2. There is no retirement, as agents live forever. They provide capital and labor and consume goods every period.

3. Population is normalized to 1.

### 2.1 Ramsey Model

Household problem

$$\max_{c_t, k_{t+1}} \sum_{t=0}^{\infty} \beta^t u(c_t).$$

subject to

$$\begin{aligned} c_t + k_{t+1} &= f(k_t) + (1-\delta)k_t, \\ k_0 &> 0. \end{aligned}$$

**Exercise 4.** Solve for the FOC (in this case, the FOC is the Euler)

1. With Lagrangian

$$\mathcal{L} = \sum_{t=0}^{\infty} \beta^t u(c_t) + \sum_{t=0}^{\infty} \lambda_t \left[f(k_t) + (1-\delta)k_t - k_{t+1} - c_t\right]$$

Grouping all the summation and rewriting the Lagrangian:

$$\mathcal{L} = \sum_{t=0}^{\infty} \left[\beta^t u(c_t) + \lambda_t \left(f(k_t) + (1-\delta)k_t - k_{t+1} - c_t\right)\right].$$

The term inside the sum is optimized at each $t$ following the modified Lagrangian:

$$\mathcal{L} = \beta^t u(c_t) + \lambda_t \left(f(k_t) + (1-\delta)k_t - k_{t+1} - c_t\right).$$

FOC: (Note that $k_{t+1}$ appears twice at time $t$ and $t+1$)

$$(c_t): \frac{\partial \mathcal{L}}{\partial c_t} = \beta^t u'(c_t) - \lambda_t = 0. \tag{14}$$

$$(k_{t+1}): \frac{\partial \mathcal{L}}{\partial k_{t+1}} = -\lambda_t + \lambda_{t+1}\left[f'(k_{t+1}) + (1-\delta)\right] = 0. \tag{15}$$

By virtue of Eq. (14), we see that:

$$\beta^{t+1} u'(c_{t+1}) = \lambda_{t+1}.$$

Plugging back to Eq. (15) and rearranging give us the Euler equation:

$$\frac{u'(c_t)}{u'(c_{t+1})} = \beta\left[f'(k_{t+1}) + (1-\delta)\right].$$

For sufficient, the following transversality condition holds [2]

$$\lim_{T\to\infty} \beta^T u'(c_T)k_{T+1} = 0.$$

The intuition of the transversality condition is partly that "there are no savings in the last period." But as there is no "last period" in an infinite horizon environment, we take the limit as time goes to infinity. Put differently, the present value of the capital in an infinitely distant future must be zero.

2. With Bellman equation

Assume that we have found the optimized sequence of capital holding $\{k_t\}_{t=0}^{\infty}$, then the value of lifetime utility associated with that optimum is

$$V(k_t) = \max_{\{c_t,k_{t+1}\}_{t=0}^{+\infty}} \sum_{t=0}^{+\infty} \beta^t u(c_t).$$

where

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

We can write it in recursive form

$$V(k_t) = \max_{k_{t+1}} \left[u(f(k_t) - k_{t+1} + (1-\delta)k_t) + \beta V(k_{t+1})\right].$$

Choosing $c_t$ is the same as choosing $k_{t+1}$. Maximizing the Value function wrt. the control variable $k_{t+1}$:

$$\begin{aligned}
&\frac{\partial V(k_t)}{\partial k_{t+1}} = 0 \\
&\Leftrightarrow \frac{\partial u(k_{t+1})}{\partial k_{t+1}} + \beta \frac{\partial V(k_{t+1})}{\partial k_{t+1}} = 0 \\
&\Leftrightarrow -u'(c_t) + \beta \frac{\partial V(k_{t+1})}{\partial k_{t+1}} = 0.
\end{aligned}$$

By the Envelope theorem, we obtain the Benveniste-Scheinkman Equation

$$\frac{\partial V(k_t)}{\partial k_t} = (f'(k_t) + 1 - \delta)u'(f(k_t) - k_{t+1} + (1-\delta)k_t),$$

Forwarding 1 period

$$\begin{aligned}
\frac{\partial V(k_{t+1})}{\partial k_{t+1}} &= (f'(k_{t+1}) + 1 - \delta)u'(f(k_{t+1}) - k_{t+2} + (1-\delta)k_{t+1}) \\
&= (f'(k_{t+1}) + 1 - \delta)u'(c_{t+1}).
\end{aligned}$$

Derive the Euler equation relating the dynamics of the choice variable.

$$\frac{u'(c_t)}{u'(c_{t+1})} = \beta(f'(k_{t+1}) + (1-\delta)).$$

Assume the following functional form

$$\begin{aligned}
u(c_t) &= \ln c_t, \\
f(k_t) &= k_t^{\alpha}.
\end{aligned}$$

[2] to see why, you can visit: https://economics.stackexchange.com/questions/15290/transversality-condition-in-neoclassical-growth-model

**Exercise 5** (Solve for the steady state)**.** At the steady states

$$c_t = c_{t+1} = \bar{c},$$
$$k_t = k_{t+1} = \bar{k}.$$

Using the Euler, prove that the steady state is

$$\bar{k} = \left(\frac{\alpha\beta}{1-(1-\delta)\beta}\right)^{1/(1-\alpha)}, \tag{16}$$
$$\bar{c} = \left(\frac{1-\beta[1-(1-\alpha)\delta]}{\alpha\beta}\right)\left(\frac{\alpha\beta}{1-(1-\delta)\beta}\right)^{1/(1-\alpha)}.$$

Thus far, we have only solved the Euler equation and the steady states. However, the solution is a sequence of $\{c_t, k_t\}_{t=0}^{\infty}$ that solves the lifetime utility. To derive the sequence, we need to pin down the policy function

$$k_{t+1} = h(k_t).$$

The mission is to estimate this $h(\cdot)$ function. Below, we introduce some solution methods to find such a function.

**Parameters**

| Parameters | Value |
|---|---|
| $\beta$ | 0.99 |
| $\alpha$ | 0.3 |
| $\delta$ | 0.1 |

## 2.2 Method of Undetermined Coefficients

(Also known as Guess and Verify) First, we need to guess the functional form of the Value function using some parameters. Assume that

$$V = a + b\ln(k).$$

with $a$ and $b$ is yet undetermined coefficients. We can rewrite the recursive form as

$$\max_{k'} \ln(f(k) - k' + (1-\delta)k) + \beta(a + b\ln k').$$

The FOC wrt $k'$ is

$$k' = \frac{\beta b}{1+\beta b}k^{\alpha}.$$

Plugging back to the value function to derive $a$ and $b$. This method is very limited as the guess must be correct, and the function is analytically differentiable. We derive below some alternatives. The first is a local solution called the perturbation method, and the second is a global solution method known as value function and policy function iteration.

**Exercise 6.** Write a program that solves and plots this policy function.

## 2.3 Pertubation Methods: Linear Approximation

In previous methods, the exact solution to the policy function could be obtained. However, this is not always feasible. In some cases, approximation is preferred as it provides faster computation, especially in models with stochastic elements.

### Taylor Approximation

Consider the following case system

$$\begin{aligned} x_{t+1} &= f(x_t, y_t), \\ y_{t+1} &= g(x_t, y_t). \end{aligned} \tag{17}$$

Consider the linear dynamics in $\mathbb{R}^2 \to \mathbb{R}^2$. Given the initial state $(x_0, y_0)$. Assume that

$$\begin{aligned} \bar{x} &= f(\bar{x}, \bar{y}), \\ \bar{y} &= g(\bar{x}, \bar{y}). \end{aligned} \tag{18}$$

be the steady state $(\bar{x}, \bar{y})$ of the system (17). The first-order Taylor expansion of $f(\cdot)$ around a steady state:

$$f(x, y) - f(\bar{x}, \bar{y}) \approx f'_x(\bar{x}, \bar{y})(x - \bar{x}) + f'_y(\bar{x}, \bar{y})(y - \bar{y}).$$

Similarly for $g(\cdot)$:

$$g(x, y) - g(\bar{x}, \bar{y}) \approx g'_x(\bar{x}, \bar{y})(x - \bar{x}) + g'_y(\bar{x}, \bar{y})(y - \bar{y}).$$

From (17),(18), we can write them in matrix form [3] as

$$\begin{pmatrix} x_{t+1} - \bar{x} \\ y_{t+1} - \bar{y} \end{pmatrix} = \underbrace{\begin{pmatrix} f'_x(\bar{x}, \bar{y}) & f'_y(\bar{x}, \bar{y}) \\ g'_x(\bar{x}, \bar{y}) & g'_y(\bar{x}, \bar{y}) \end{pmatrix}}_{\mathbf{J}} \begin{pmatrix} x_t - \bar{x} \\ y_t - \bar{y} \end{pmatrix}. \tag{19}$$

where, as we all know by now, $\mathbf{J}$ is the Jacobian matrix. The system has been "linearized" and can be analyzed similarly to the linear case.

### Linear Approximation of Saddle Path

From the resource constraint and Euler equation

$$\begin{aligned} k_{t+1} + c_t &= f(k_t) + (1 - \delta)k_t, \\ u'(c_t) &= \beta u'(c_{t+1})[f'(k_{t+1}) + (1 - \delta)]. \end{aligned}$$

At the steady state

$$\begin{aligned} \bar{c} &= f(\bar{k}) - \delta\bar{k}, \\ 1/\beta &= f'(\bar{k}) + (1 - \delta). \end{aligned}$$

We can write the behavior of variables near the steady state as

$$\begin{pmatrix} k_{t+1} - \bar{k} \\ c_{t+1} - \bar{c} \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} k_t - \bar{k} \\ c_t - \bar{c} \end{pmatrix}. \tag{20}$$

[3] This is called to "linearize" around the steady state

We want to estimate $A, B, C, D$. Near the steady state $(\bar{c}, \bar{k})$ we have

$$\begin{aligned} c_t &= \bar{c} + (c_t - \bar{c}), \\ f(k_t) + (1-\delta)k_t &\approx f(\bar{k}) + (1-\delta)\bar{k} + (f'(\bar{k}) + (1-\delta))(k_t - \bar{k}), \\ k_{t+1} &\approx \bar{k} + (k_t - \bar{k}). \end{aligned}$$

Substituting these into the resource constraint

$$k_{t+1} + \bar{c} + (c_t - \bar{c}) = f(\bar{k}) + (1-\delta)\bar{k} + (f'(\bar{k}) + (1-\delta))(k_t - \bar{k}).$$

Rearranging

$$(k_{t+1} - \bar{k}) = (1/\beta)(k_t - \bar{k}) - (c_t - \bar{c}). \tag{21}$$

Next, we need one more equation containing $c_{t+1} - \bar{c}$, let us use the **Euler equation** and log-linearize it

$$\ln u'(c_t) - \ln u'(c_{t+1}) - \ln \beta = \ln[f'(k_{t+1}) + (1-\delta)]. \tag{22}$$

At the steady state, since $c_{t+1} = c_t = \bar{c}$

$$\ln u'(\bar{c}) - \ln u'(\bar{c}) - \ln \beta = \ln[f'(\bar{k}) + (1-\delta)]. \tag{23}$$

Subtract Eq.(23) from (22) to obtain

$$(\ln u'(c_t) - \ln u'(\bar{c})) - (\ln u'(c_{t+1}) - \ln u'(\bar{c})) = \ln[f'(k_{t+1}) + (1-\delta)] - \ln[f'(\bar{k}) + (1-\delta)]. \tag{24}$$

Near the steady state $(\bar{c}, \bar{k})$ we have

$$\begin{aligned} \ln u'(c_{t+1}) - \ln u'(\bar{c}) &\approx \frac{u''(\bar{c})}{u'(\bar{c})}(c_{t+1} - \bar{c}), \\ \ln u'(c_t) - \ln u'(\bar{c}) &\approx \frac{u''(\bar{c})}{u'(\bar{c})}(c_t - \bar{c}), \\ \ln[f'(k_{t+1}) + (1-\delta)] - \ln[f'(\bar{k}) + (1-\delta)] &\approx [\ln(f'(\bar{k}) + (1-\delta))]'(k_{t+1} - \bar{k}) \\ &= \frac{f''(\bar{k})}{f'(\bar{k}) + (1-\delta)}(k_{t+1} - \bar{k}) \\ &= \beta f''(\bar{k})(k_{t+1} - \bar{k}). \end{aligned}$$

Plugging back to (24) yields

$$\frac{u''(\bar{c})}{u'(\bar{c})}(c_t - \bar{c}) - \frac{u''(\bar{c})}{u'(\bar{c})}(c_{t+1} - \bar{c}) = \beta f''(\bar{k})(k_{t+1} - \bar{k}). \tag{25}$$

Rearranging

$$-\beta f''(\bar{k})(k_{t+1} - \bar{k}) - \frac{u''(\bar{c})}{u'(\bar{c})}(c_{t+1} - \bar{c}) = -\frac{u''(\bar{c})}{u'(\bar{c})}(c_t - \bar{c}). \tag{26}$$

Combining (21) and (26) yields the system

$$\begin{aligned} (k_{t+1} - \bar{k}) + 0 \cdot (c_{t+1} - \bar{c}) &= (1/\beta)(k_t - \bar{k}) - (c_t - \bar{c}), \\ -\beta f''(\bar{k})(k_{t+1} - \bar{k}) - \frac{u''(\bar{c})}{u'(\bar{c})}(c_{t+1} - \bar{c}) &= 0 \cdot (k_t - \bar{k}) - \frac{u''(\bar{c})}{u'(\bar{c})}(c_t - \bar{c}). \end{aligned}$$

Write this in matrix form

$$\begin{pmatrix} 1 & 0 \\ -\beta f''(\bar{k}) & -\frac{u''(\bar{c})}{u'(\bar{c})} \end{pmatrix} \begin{pmatrix} k_{t+1} - \bar{k} \\ c_{t+1} - \bar{c} \end{pmatrix} = \begin{pmatrix} 1/\beta & -1 \\ 0 & -\frac{u''(\bar{c})}{u'(\bar{c})} \end{pmatrix} \begin{pmatrix} k_t - \bar{k} \\ c_t - \bar{c} \end{pmatrix}.$$

To transform it into a form similar to (20), we premultiply both sides with the inverse of the first matrix and obtain

$$\begin{aligned} \begin{pmatrix} k_{t+1} - \bar{k} \\ c_{t+1} - \bar{c} \end{pmatrix} &= \begin{pmatrix} 1 & 0 \\ -\beta f''(\bar{k}) & -\frac{u''(\bar{c})}{u'(\bar{c})} \end{pmatrix}^{-1} \begin{pmatrix} 1/\beta & -1 \\ 0 & -\frac{u''(\bar{c})}{u'(\bar{c})} \end{pmatrix} \begin{pmatrix} k_t - \bar{k} \\ c_t - \bar{c} \end{pmatrix} \\ &= \underbrace{\begin{pmatrix} 1/\beta & -1 \\ \frac{u'(\bar{c}) f''(\bar{k})}{u''(\bar{c})} & 1 + \beta \frac{u'(\bar{c}) f''(\bar{k})}{u''(\bar{c})} \end{pmatrix}}_{\mathbf{J}} \begin{pmatrix} k_t - \bar{k} \\ c_t - \bar{c} \end{pmatrix}. \end{aligned}$$

This Jacobian $\mathbf{J}$ has two eigenvalues $\lambda_1, \lambda_2$ satisfying

$$\begin{aligned} \det \mathbf{J} &= 1/\beta = \lambda_1 \lambda_2, \\ tr\mathbf{J} &= 1 + \frac{1}{\beta} + \frac{\beta u'(\bar{c}) f''(\bar{k})}{u''(\bar{c})} = \lambda_1 + \lambda_2 = \triangle. \end{aligned}$$

Since $u'' < 0, f'' < 0$, we can say that

$$|1 + \det \mathbf{J}| = 1 + 1/\beta < tr\mathbf{J} = 1 + 1/\beta + \frac{\beta u'(\bar{c}) f''(\bar{k})}{u''(\bar{c})}.$$

The steady state is a saddle point, implying that $\lambda_1 < 1 < \lambda_2$. The smaller root is stable, while the bigger root is unstable. We can extract $\lambda_1 = \triangle - \lambda_2$. Then

$$\begin{aligned} \det \mathbf{J} &= \lambda_2 (\triangle - \lambda_2) \\ \iff \triangle &= \frac{\det \mathbf{J}}{\lambda_2} + \lambda_2. \end{aligned}$$

Hence, $\lambda_1, \lambda_2$ are the solutions of the following quadratic function

$$\phi(\lambda) = \lambda^2 - \triangle \lambda + 1/\beta.$$

The eigenvector associated with the eigenvalues are

$$(\lambda_1) : \begin{pmatrix} v_{11} \\ v_{12} \end{pmatrix}, \qquad (\lambda_2) : \begin{pmatrix} v_{21} \\ v_{22} \end{pmatrix}$$

Stacking them in a new matrix

$$V = \begin{pmatrix} v_{11} & v_{21} \\ v_{12} & v_{22} \end{pmatrix},$$

with its inverse matrix

$$V^{-1} = \frac{1}{v_{11} v_{22} - v_{12} v_{21}} \begin{pmatrix} v_{22} & -v_{21} \\ -v_{12} & v_{11} \end{pmatrix}.$$

The eigen diagonal matrix is

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

Then, we can eigendecompose the matrix $\mathbf{J}$ to

$$\mathbf{J} = V\Lambda V^{-1}.$$

The system is now written as

$$\begin{pmatrix} k_{t+1} - \bar{k} \\ c_{t+1} - \bar{c} \end{pmatrix} = V\Lambda V^{-1} \begin{pmatrix} k_t - \bar{k} \\ c_t - \bar{c} \end{pmatrix}.$$

Iterating forward, starting from initial $c_0, k_0$ implies

$$\begin{pmatrix} k_t - \bar{k} \\ c_t - \bar{c} \end{pmatrix} = V\Lambda^t V^{-1} \begin{pmatrix} k_0 - \bar{k} \\ c_0 - \bar{c} \end{pmatrix}.$$

Writing out explicitly

$$\begin{aligned} k_t - \bar{k} &= v_{21} \frac{v_{22}(c_0 - \bar{c}) - v_{12}(k_0 - \bar{k})}{v_{11}v_{22} - v_{12}v_{21}} \lambda_1^t - v_{22} \frac{v_{21}(c_0 - \bar{c}) - v_{11}(k_0 - \bar{k})}{v_{11}v_{22} - v_{12}v_{21}} \lambda_2^t, \\ c_t - \bar{c} &= v_{11} \frac{v_{22}(c_0 - \bar{c}) - v_{12}(k_0 - \bar{k})}{v_{11}v_{22} - v_{12}v_{21}} \lambda_1^t - v_{12} \frac{v_{21}(c_0 - \bar{c}) - v_{11}(k_0 - \bar{k})}{v_{11}v_{22} - v_{12}v_{21}} \lambda_2^t. \end{aligned}$$

If $\lambda_2$ is the unstable root, set

$$c_0 - \bar{c} = \frac{v_{11}}{v_{21}}(k_0 - \bar{k}).$$

will neutralize the unstable root (explosive dynamics). That's why consumption is also called the jump variable. Plugging it back to the initial consumption yields

$$c_t - \bar{c} = v_{11} \frac{v_{22}\frac{v_{11}}{v_{21}} - v_{12}}{v_{11}v_{22} - v_{12}v_{21}} \lambda_1^t (k_0 - \bar{k}) = \frac{v_{11}}{v_{21}} \lambda_1^t (k_0 - \bar{k}).$$

and so the capital accumulation

$$k_t - \bar{k} = v_{21} \frac{v_{22}\frac{v_{11}}{v_{21}} - v_{12}}{v_{11}v_{22} - v_{12}v_{21}} \lambda_1^t (k_0 - \bar{k}) = \lambda_1^t (k_0 - \bar{k}).$$

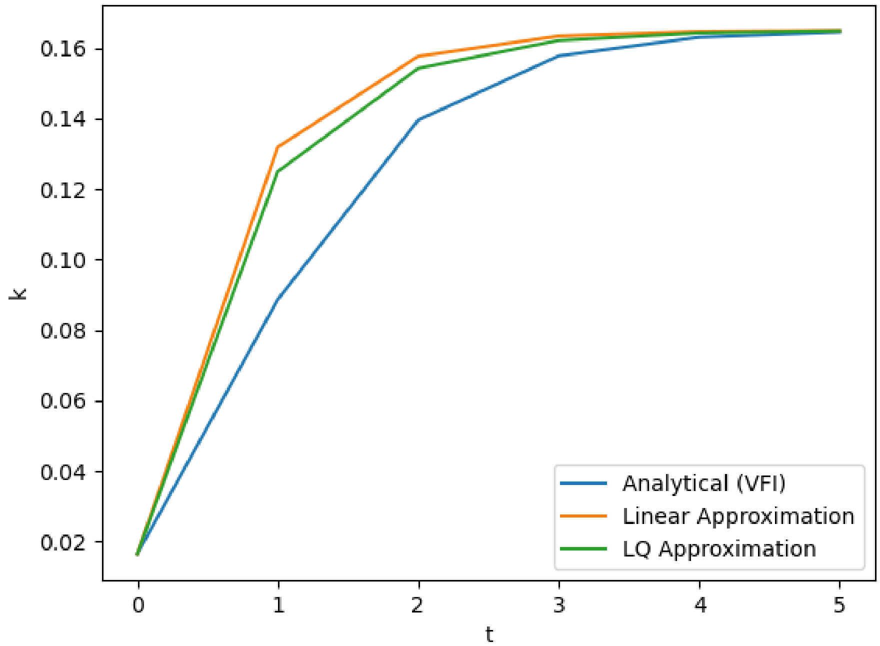
Hence, the solution of $k_t$ is

$$k_{t+1} - \bar{k} = \lambda_1(k_t - \bar{k}).$$

where $\lambda_1$ is the stable root.

> **Exercise 7.** Write a program that solves the policy function using this linear approximation.

There are also other perturbation methods, one commonly used is the LQ method. It was proposed by Hansen and Prescott (1995). L means linearity of constraint, and Q means quadratic utility function. We must look for a linear law of motion and put all remaining nonlinear relations into the current consumption. We do not study that method here, but here is a comparison for accuracy. You can see that the equilibrium path is slightly "off" from the true solution, i.e., "perturbed" as it is an approximation, which is why it is called a perturbation method.




### 2.4 Value Function Iteration

The above methods are called local methods as they try to approximate the policy function point-by-point. In this section, we use one global method called Value Function Iteration. First, we do it by hand. You will then see how the algorithm works. Doing it by hand also delivers an analytical solution so you can check with the method of undetermined coefficients. Later, we write a code that does the iteration for us.

First, drop $t$ notation for short and use a "prime" to denote variables at $t+1$. The true value function is the limit of the following

$$V^{s+1}(k) = \max_{k'} u(f(k) - k' + (1-\delta)k) + \beta V^s(k').$$

1. Initial guess: $V^0 = 0$.
2. $s = 1$: with $V^0$ known, we find $k'$ that maximizes $V^1(k)$, then substitute this $k'$ back to $V^1$ to derive the value of $V^1$.
3. $s = 2$: with $V^1$ known previously, we find $k'$ that maximizes $V^2(k)$, then substitute this $k'$ back to $V^2$ to derive the value of $V^2$.
4. iterate as many as you can until you see the pattern.
5. take the limits of $s \to \infty$.

The method works because Ramsey's value function is a contraction mapping (Acemoglu, 2008, p.190-194). You should be able to derive

$$k' = \alpha\beta k^\alpha.$$

Now, we use a computer to perform this iteration based on the following procedure.

1. Choose a grid that must contain the steady state. The grid should contain a steady state. The steady-state satisfies

$$f'(\bar{k}) = \frac{1}{\beta} \Rightarrow \bar{k}.$$

We also want to find the maximal sustainable capital stock (consume nothing)

$$f(\hat{k}) = \hat{k} \Rightarrow \hat{k}.$$

The minimum grid point should be larger than 0, and the maximum grid point minimum than $\hat{k}$. We generate $n$ points equally spaced on this grid, indexed by $i$.

2. Initiate an array of initial guesses

$$\text{(naive) } V^0 = 0,$$
$$\text{(smart) } V^0 = \frac{u(\bar{c})}{1-\beta}.$$

3. For each point $k(i)$, find a $k(j)$ on the grid that maximizes[4]

$$V_i^{s+1} = \sup_{k_j \in kgrid} u(f(k_i) - k_j) + \beta V_j^s.$$

substitute $k_j$ back to $V_i^{s+1}$ to derive the value $V^1$.

Update $V^0$ to $V^1$, and store the value of optimal $k_j$.

4. For each iteration $s$, check the error $|V^0 - V^1|$. If it is smaller than $tol$, stop. Otherwise, go back to step 3.

**Exercise 8.** Write a program to solve the model based on value function iteration.

A good reference source: https://www.eco.uc3m.es/~jrincon/Teaching/Master/SDDP.pdf#page=8.70. Check page 21.

The sample code is found in the Appendix. In the sample code, we use naive guesses. In your implementation, try to use smart guesses.

The value function iteration is a slow process, as the convergence rate is $\beta$. There are other iterative methods that produce faster convergence, such as policy function iteration or Euler equation iteration. You can see some examples of such algorithms in the Appendix.

[4] or use the Binary Search algorithm. Basically, it searches on the grid and returns the index of the point that maximizes the objective function.