Computational Homework 1

ECON 330 (Theory of Income)
Fall 2019
Alvin U. Lumbanraja

A Theoretical Benchmark

Notes:
Throughout this exercise, I use the following assumption based on the information from the problem sets

Table 1: Parameters

Parameter	β	α	δ	z	k_0
Values	0.98	1/3	1	1	$0.05k_{ss}$

1. Given the functional form of the utility function, we know that:

   - For every $k \in K$, we know that the feasible set is non-empty
   - Value function is monotone, concave, and differentiable
   - The policy function $g(k)$ is single-valued

   By substituting k_{t+1} as k', substituting the value of c_t as a function of k_t and k_{t+1}, we can define the Bellman equation for this problem as

$$\begin{aligned} v(k) &= \max_{k' \in \Gamma(k)} \{F(k,k') + \beta v(k')\} \\ &= \max_{k' \in \Gamma(k)} \{\ln[zk^\alpha + (1-\delta)k - k'] + \beta v(k')\} \quad \text{(Generalized form)} \quad (1) \\ &= \max_{k' \in \Gamma(k)} \{\ln[k^{\frac{1}{3}} - k'] + \beta v(k')\} \quad \text{(with specified parameters)} \quad (2) \end{aligned}$$

   Let us also define the feasibility constraint Γ in terms of k and k'

$$\begin{aligned} \Gamma(k) &\equiv [0, (zk_t^\alpha + (1-\delta)k_t] \\ &\equiv [0, k^{\frac{1}{3}}] \quad (3) \end{aligned}$$

   The capital is fully depreciated and owing to the fact that the function is in the log form, we can rule out corner solution, both on the lower bound and upper bound (both $c_t = k_t$, which means $c' = k'^{\frac{1}{3}} = 0$ and $c_t = 0$ are not feasible)

2. Let $g(k)$ denotes the optimal choice for k'. The first order condition for this problem is as follows:

$$\begin{aligned} 0 &= \frac{\partial}{\partial k'} \ln[zk^\alpha + (1-\delta)k - g(k)] + \beta \frac{\partial}{\partial k} v(g(k)) \\ &= \frac{\partial}{\partial k'} \ln[k^{\frac{1}{3}} - g(k)] + 0.98 \frac{\partial}{\partial k} v(g(k)) \\ &= -\frac{1}{zk^\alpha + (1-\delta)k - g(k)} + \beta v'(g(k)) \leq 0 \quad \text{(Generalized form)} \quad (4) \\ &= -\frac{1}{(k^{\frac{1}{3}} - g(k))} + 0.98 \frac{\partial}{\partial k} v(g(k)) \quad \text{(with specified parameters)} \quad (5) \end{aligned}$$

The envelope condition is as follows:

$$\frac{\partial}{\partial k}v(k) = \frac{\partial}{\partial k}\ln[zk^{\alpha} + (1-\delta)k - g(k)] \tag{6}$$

$$v'(k) = \frac{\alpha z k^{\alpha-1} + (1-\delta)}{zk^{\alpha} + (1-\delta)k - g(k)} \quad \text{(Generalized form)} \tag{7}$$

$$= -\frac{1}{3k^{\frac{2}{3}}(k^{\frac{1}{3}} - g(k))} \quad \text{(with specified parameters)} \tag{8}$$

The Euler equation is as follows:

$$\begin{aligned}
0 &= \frac{\partial}{\partial k'}\ln[zk^{\alpha} + (1-\delta)k - g(k)] + \beta\frac{\partial}{\partial k}\ln[zg(k)^{\alpha} + (1-\delta)g(k) - g(g(k))] \\
&= \frac{\partial}{\partial k'}\ln[k^{\frac{1}{3}} - g(k)] + 0.98\frac{\partial}{\partial k}\ln[g(k)^{\frac{1}{3}} - g(g(k))] \\
&= -\frac{1}{zk^{\alpha} + (1-\delta)k - g(k)} + \beta\Big[\frac{\alpha z g(k)^{\alpha-1} + (1-\delta)}{zg(k)^{\alpha} + (1-\delta)g(k) - g(g(k))}\Big] && (9) \\
&= -\frac{1}{(k^{\frac{1}{3}} - g(k))} + \frac{0.98}{3g(k)^{\frac{2}{3}}(k^{\frac{1}{3}} - g(k))} && (10)
\end{aligned}$$

The optimal policy function can be recursively written as:

$$\begin{aligned}
\frac{1}{k^{\frac{1}{3}} - g(k)} &= \frac{0.98}{3g(k)^{\frac{2}{3}}(g(k)^{\frac{1}{3}} - g(g(k)))} \\
g(k)^{\frac{1}{3}} - g(g(k)) &= \frac{0.98(k^{\frac{1}{3}} - g(k))}{3g(k)^{\frac{2}{3}}} \\
g(g(k)) &= g(k)^{\frac{1}{3}} - \frac{0.98(k^{\frac{1}{3}} - g(k))}{3g(k)^{\frac{2}{3}}} \\
g(k) &= k^{\frac{1}{3}} - \frac{0.98(k_{t-1}^{\frac{1}{3}} - k)}{3k^{\frac{2}{3}}} \\
&= k^{\frac{1}{3}} - \frac{0.98(k_{t-1}^{\frac{1}{3}} - k)}{3k^{\frac{2}{3}}} && (11)
\end{aligned}$$

It may be difficult to show algebraically how $g(k)$ reacts to change in k_0. However, by concavity of the return function F, we know that approximate policy functions $\{g(k)\}$ converges pointwise. Therefore, the sequence of policy function $\{g(k)\}$ will increase if k_0 increase when $k_0 < k_s s$, and vice-versa.

3. Using the Euler equation, let us define steady state level capital where $k = g(k) = k_{ss}$

$$k_{ss} = \left[\frac{1 - \beta(1-\delta)}{\beta\alpha z}\right]^{\frac{1}{\alpha-1}} \tag{12}$$

$$k_{ss} = \Big(\frac{3}{0.98}\Big)^{\frac{3}{2}} \approx 0.1867 \tag{13}$$

4. Let us assume that the value function $v(k)$ equals to certain value of the functional form $a\log(k) + b$. We then know that:

$$v'(k) = \frac{a}{k}$$

Notice that this is similar to the envelope condition. Recall the general form of for this problem

$$v'(k) = \frac{\alpha z k^{\alpha-1} + (1-\delta)}{zk^{\alpha} + (1-\delta)k - g(k)}$$

Setting the value equal, we obtain

$$\frac{a}{k} = \frac{\alpha z k^{\alpha-1} + (1-\delta)}{zk^\alpha + (1-\delta)k - g(k)} \tag{14}$$

By exploiting the fact that $\delta = 1$, we can further reduce the equation () to

$$\frac{a}{k} = \frac{\alpha z k^{\alpha-1}}{zk^\alpha - g(k)} = \frac{\alpha z}{zk - \frac{g(k)}{k^{\alpha-1}}} \tag{15}$$

However, for the equation above to be consistent, we then need the equation above to be in the form of

$$\frac{a}{k} = \frac{\alpha z}{zk - \frac{g(k)}{k^{\alpha-1}}} = \frac{\alpha z}{(z-\gamma)k} \tag{16}$$

So that

$$\gamma k = \frac{g(k)}{k^{\alpha-1}} \tag{17}$$

$$g(k) = \gamma k^\alpha \tag{18}$$

By plugging the value to our FOC, we obtain

$$\gamma = \frac{\beta \alpha z^2}{z - \gamma + \beta \alpha z}$$

$$(\gamma - z)(\gamma - \alpha\beta z) = 0 \tag{19}$$

The only logical value for γ is therefore $\gamma = \alpha\beta z$. Therefore, we may compute the following

$$a = \frac{\alpha}{1-\alpha\beta} \tag{20}$$

$$g(k) = \alpha\beta z k^\alpha \tag{21}$$

$$v(k) = \frac{\alpha}{1-\alpha\beta}\ln(k) + b \tag{22}$$

We are thus interested in knowing the value of b. By setting the b equal to the value function

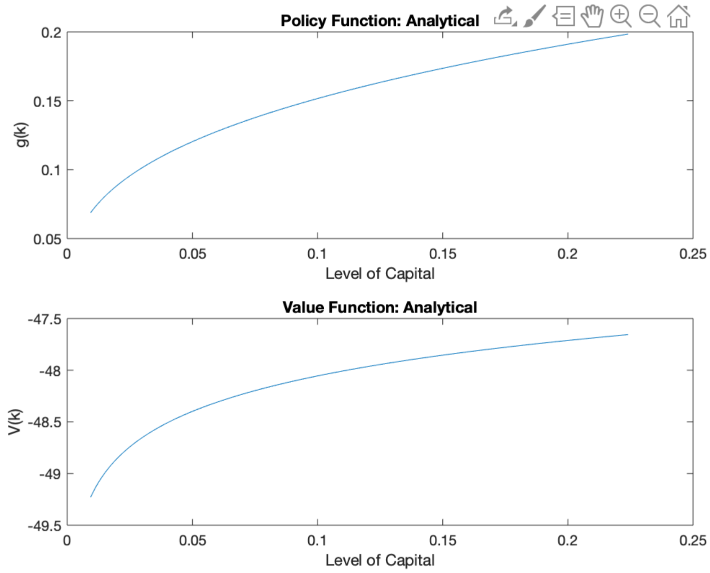
$$\begin{aligned} a\ln(k) + b &= \ln[zk^\alpha + (1-\delta)k - k'] + \beta(\ln k' + b) \\ b &= \frac{\ln[zk^\alpha + (1-\delta)k - k'] + \beta(a) - a\ln(k)}{1-\beta} \\ &= \frac{\ln[zk^\alpha + (1-\delta)k - k'] + \beta(a) - a\ln(k)}{1-\beta} \\ &= \frac{\ln(z(1-\alpha\beta)) + \frac{\alpha\beta}{1-\alpha\beta}\ln(\alpha\beta)}{1-\beta} \end{aligned} \tag{23}$$

We thus can conclude that a and b can indeed be constant, and the value function is in the form of

$$v(k) = \frac{\alpha}{1-\alpha\beta}\ln(k) + \frac{\ln(z(1-\alpha\beta)) + \frac{\alpha\beta}{1-\alpha\beta}\ln(\alpha\beta)}{1-\beta} \tag{24}$$

We can plot analytically the value of $v(k)$ and $g(k)$ as follows:

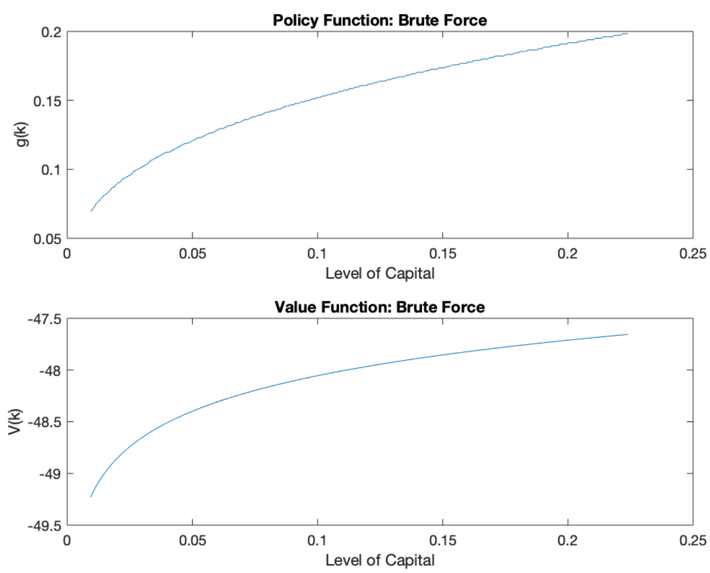
Figure 1: Analytical Simulation



B Brute Force Value Function Iteration

The amount of time it takes to run this algorithm (measured from starting the grid of capital until the end) is 5.0543s. The result (shown below) is similar to (A)

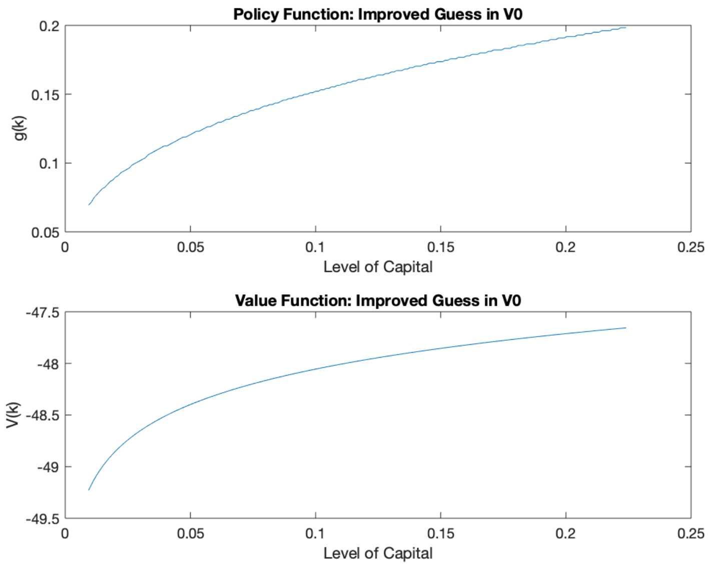
Figure 2: Brute Force Value Function Simulation



C Improved Guess in V_0

The amount of time it takes to run this algorithm (measured from starting the grid of capital until the end) is 0.1018s. The result (shown below) is similar to (A). The interpretation for the V_0 guess is that the value function should be proportional to the value of the return function.

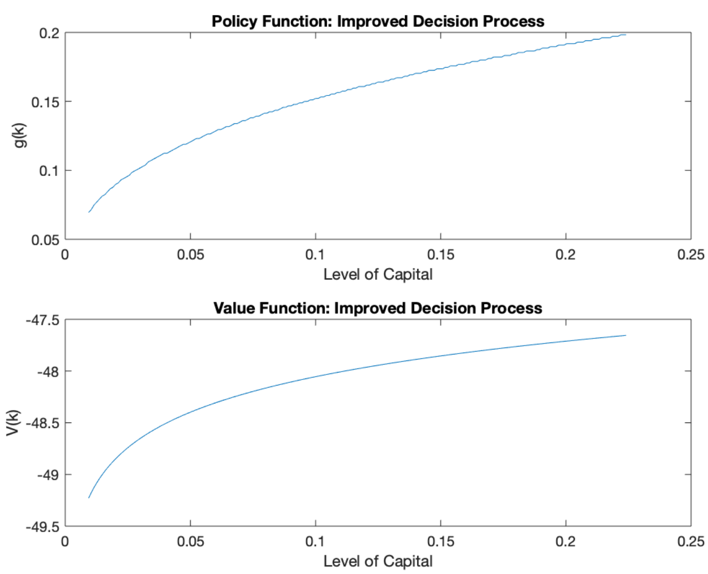
Figure 3: Improved Guess in V_0 Simulation



D Improved Decision Process

The amount of time it takes to run this algorithm (measured from starting the grid of capital until the end) is 2.4361s. The result (shown below) is similar to (A). Compared to the brute force, this algorithm is significantly faster (almost 52% faster), but still slower than educated guess of the value function.

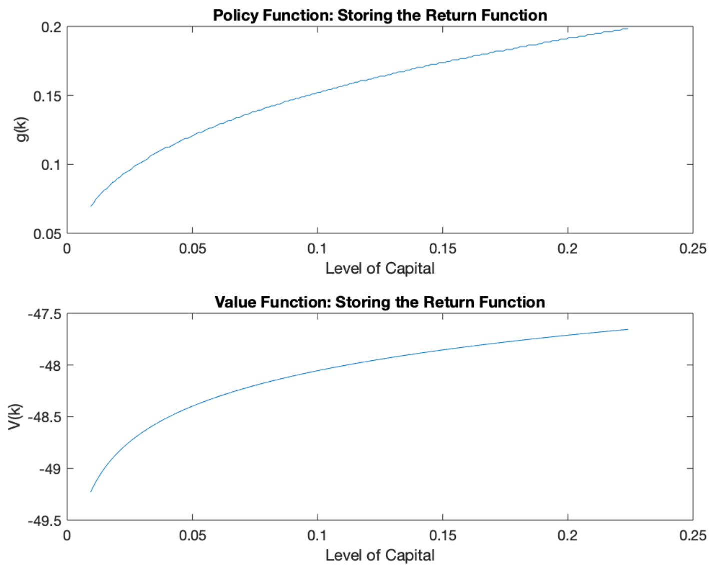
Figure 4: Improved Decision Process



E Storing the Return Function

The amount of time it takes to run this algorithm (measured from starting the grid of capital until the end) is 0.1967s. The result (shown below) is similar to (A). Compared to the brute force, this algorithm is significantly faster (more than 96% faster), but still slower than educated guess of the value function.

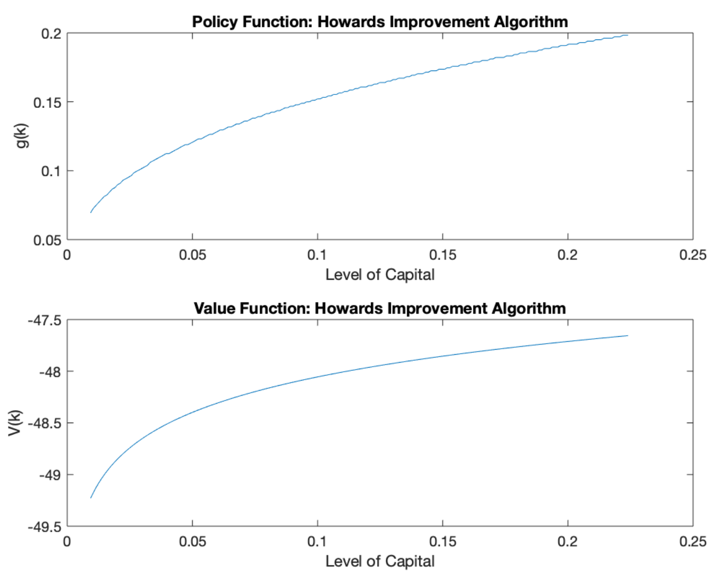
Figure 5: Storing the Return Function



F Howard's Improvement Algorithm

The amount of time it takes to run this algorithm (measured from starting the grid of capital until the end) is 0.2791s. The result (shown below) is similar to (A). Compared to the brute force, this algorithm is significantly faster (more than 94% faster), but still slower than educated guess of the value function.

Figure 6: Howard's Improvement Algorithm



G Combining Methods and Concluding Notes

The amount of time it takes to run this algorithm (measured from starting the grid of capital until the end) is 0.0714s. The result (shown below) is similar to (A). Compared to the brute force, this algorithm is significantly faster (more than 98% faster). This is the only algorithm that is faster than educated guess of the value function.

Figure 7: Howard's Improvement Algorithm

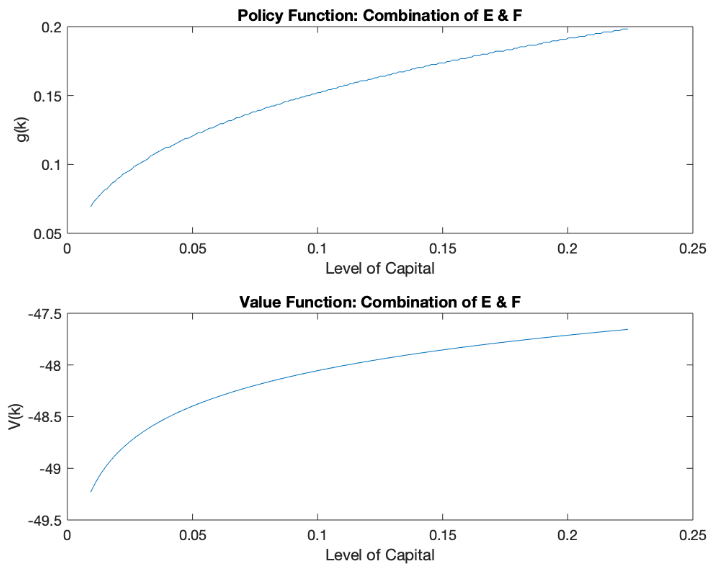


Table 2: Algorithm Performance

Parameter	Iteration	Time
B. Brute Force	570	5.0543
C. Improved Guess in V_0	7	0.1018
D. Improved Decision Process	570	2.4361
E. Storing the Return Function	570	0.1967
F. Howard's Improvement Algorithm	23	0.2791
G. Combination of E & F	23	0.0714

Computation done on 2017 MacBook Pro 13", 3.1 GHz Intel Core i5

The algorithm performance highly depends on (1) the initial guess on the optimal value of capital and (2) the efficiency of algorithm. Having more efficient algorithm can significantly reduce the needed iteration; by not performing unnecessary looping and/or performing maximization every time we perform iteration, we can cut down significantly on the need for computational power. However, the most significant time saving is provided by having an educated guess prior to performing algorithm. Even with the most inefficient algorithm, having a good guess on the value function will cut down the necessary iteration significantly.