Computational Economics and Finance

Solving the Stochastic Growth Model with Non-Negative Investment via Value Function Iteration

Dr. Maximilian Ulrich Werner

Authors (Student-ID):	Albert Anguera Sempere (19-771-930)
	David Annoni (17-453-010)
	Jan Bauer (18-764-571)
	Mariia Bogdanova (19-760-925)
	Johannes Cordier (19-606-037)

Wirtschaftswissenschaftliche Fakultät
Universität Zürich
October 3, 2021

Contents

1 Introduction

Our task was to solve the stochastic growth model with non-negative investment via value function iteration. First, a brief description of the model will be given, followed by the applied methods carried out for the computational analyses. The economic as well as the computational aspects of the kink complete the report.

Value function iteration is one of the most important and known methods in order to solve Dynamic General Equilibrium models. In this paper, we use the value function iteration (VFI) to solve the stochastic growth model with non-negative investment constraint, that is described by Heer and Maussner in their book "Dynamic General Equilibrium Modeling" [2].

First, we will introduce the standard Ramsey economy and discuss the underlying idea of it. Secondly, we will move from the deterministic to the stochastic growth model with non-negative investments and solve the problem via VFI. The next chapter explains some of the relevant functions that have been used for the simulation of the mentioned model. Lastly, we will discuss the computational as well as the economic aspects of the kink.

2 The Model

2.1 The Ramsey Economy

The question "How much of its income should a nation save?" [1] made Frank Ramsey develop and propose a dynamic economic model in 1928. The model uses an agent to build a policy for choosing the consumption for a given state (in this case, given parameters capital stock and productivity shock). The policy is constructed in a way to maximize the future expected value of the utility function. The standard Ramsey Model looks as follows:

$$Y_t = F(A_t, K_t)$$

where the production output in period t is denoted by Y_t and A_t,K_t denote labor and capital, respectively. Each period, the agents have to decide on how much to produce, to consume and to save for future production. The capital stock for the next period is denoted by K_{t+1}, and consumption is denoted by C_t. The agents' consumption and investment are restricted by:

$$C_t + K_{t+1} \leq Y_t$$

The agent aims to maximize its utility function $U(C_0, C_1, ..., C_T)$, where T denotes the agent's time planning horizon. Additionally, the resources available are restricted and dependent on depreciation δ.

$$Y_t + (1-\delta)K_t \geq C_t + K_{t+1}$$

Finally, we can describe the standard Ramsey economy and state the infinite-horizon deterministic Ramsey problem as basis for our stochastic growth model for $t = 0, ..., T$ and a given K_0.

$$f(K_t) := F(A_t, K_t) + (1-\delta)K_t$$

$$\max_{(C_0, C_1, ..., C_T)} U(C_0, C_1, ..., C_T)$$

$$s.t.$$

$$\begin{aligned} K_{t+1} + C_t &\geq f(K_t), \\ 0 &\geq C_t, \\ 0 &\geq K_{t+1}, \end{aligned}$$

2.2 From the deterministic to the stochastic model

Based on the model described above, we find our general maximization problem for the infinite-horizon stochastic growth model for $t = 0, ..., T$. This set up follows the model presented in [2]. Expanding this model by a non-negative investment constraint $0 \leq K_{t+1} - (1-\delta)K_t = I_t \; \forall t \in T$, implies that the capital stock will not decrease.This yields the following planning problem with K_0 and Z_0 given:

$$\max_{C_0} E_0 \left[\sum_{t=0}^{\infty} \beta^t \frac{C_t^{1-\eta}}{1-\eta} \right], \eta > 0, \beta \in (0,1),$$

$$s.t.$$

$$\begin{aligned} K_{t+1} + C_t &\leq Z_t K_t^{\alpha} + (1-\delta)K_t, \alpha \in (0,1), \\ Z_t &= Z_{t-1}^{\varrho} e^{\epsilon_t}, \varrho \in (0,1), \epsilon \; N(0, \sigma^2), \\ 0 &\leq C_t, \\ 0 &\leq K_{t+1} - (1-\delta)K_t. \end{aligned}$$

To solve our problem, we can abstract from the given maximization problem and its constraints. In order to achieve this, the Bellman equation needs to be solved. The value function that solves it, has the following expression:

$$v(K, Z) = \max_{K' \in B_{K,Z}} u(Zf(K) + (1-\delta)K - K') + \beta E\left[v(K', Z')|Z\right] \tag{2.1}$$

where B is the set of possible future values for the capital stock, K':

$$B_{K,Z} = [K' : (1-\delta) \geq K' \geq Z(f(K) + (1-\delta)K]$$

The implementation starts with the set up of the state space for the stock of capital and the level of productivity $\{K_t, Z_t \in \mathbb{R}^2\}$. We can say that the elements of a stochastic process are independent and identically distributed (i.i.d), if the probability distribution is the same for each member of the process Z_t and independent of the realizations of other members of the process $Z_{t+s}, s \neq 0$. It is assumed that a stochastic process has the Markov property only if the probability distribution of Z_{t_1} is dependent only upon the realization of Z_t [2]. In the model, the authors use the first order auto-regressive process, referred to as AR(1) later on. It is considered that the first order process plays a prominent role in the development of stochastic Ramsey models, because any higher order auto-regressive processes can be reduced to first order vector auto-regressive processes.

A Markov chain is used to approximate the AR(1) processed in 2 steps:

1. Computation of the the discrete approximation of the realizations.
2. Computation of the transition matrix $P = (p_{ij})$.

This approach forces z_t to be realized on a finite grid of points.

The next step was to calculate the boundaries for the stock of capital, which will be used later on in the model. In a first step, we calculate the stationary capital stock:

$$K_{min} = \left[\frac{\alpha \beta z_{min}}{1 - \beta(1-\delta)} \right]^{\frac{1}{1-\alpha}}$$

$$K_{max} = \left[\frac{\alpha \beta z_{max}}{1 - \beta(1-\delta)} \right]^{\frac{1}{1-\alpha}}$$

Once one has computed the grids, the conditional expectation in Bellman equation stated in equation (2.1), can be expressed in discrete form where, given the matrix P of probability transitions of the Markov process and the set of possible capital stocks in a given time k, B_{ij}, the Bellman equation of the discrete valued problem looks like:

$$v_{ij} = \max_{K_k \in B_{ij}} u(Z(K) + (1-\delta)K - K') + \beta \sum_{l=1}^{m} p_{jl} v_{kl}$$

In the discrete form, the value function is no longer a function but a matrix. By interpolation, we use the value function matrix to build a continuous function $\hat{\phi}(K)$ that approximates the right hand-side of the Bellman equation.

$$\hat{\phi}(K) = u(Zf(K) + (1-\delta)K - K') + \beta \sum_{l=1}^{m} p_{jl} \hat{v}_l(K) \tag{2.2}$$

Given that expression, we will use a maximization procedure to find the optimal capital stock.
Finally, to measure the performance of our method we use the residuals of the Euler equation:

$$C^{-\eta} = \beta E \Big\{ \Big[(C')^{-\eta} \Big(1 - \delta + \alpha (e^{\rho \ln Z + \sigma \epsilon'})(K')^{\alpha - 1} \Big) \Big] \Big| Z \Big\}$$

Here the expected value of the right hand-side expression is computed using Gauss-Hermite four point integration formula.

3 Solving the stochastic growth model with non-negative investment constraint

3.1 The method

To solve the given stochastic growth model with non-negative investment, we are required to use value function iteration. For the computational part we have been following the reference code *Ramsey3d.g* provided in the book [2]. In simple words, one has to optimize the value function and then extract the optimal policy function, instead of directly iterating over and improving the policy function [6]. The description of the code will be provided as subsections, due to the limitation of pages only the most relevant files will be explained. For more information the provided MATLAB code tends to be as self-explanatory as possible, so in case of doubts refer to those documents.

3.1.1 Group_project.m

This is the main function of the code. For this method to work, the values for certain parameters must be given. The values used in our problem are listed in table 3.1. However, other combinations of parameters can be used. The first six elements ($\alpha - \rho$) are already described in the model chapter. Regarding the other parameters, grid-size is the size of the discretization grid for the productivity shock and this is linked to n_k, which represents the number of points we want on the grid. $kmin_g$ and $kmax_g$ are the boundaries of the grid for the capital shock, while $kmin_e$ and $kmax_e$ are used to compute the boundaries for the capital that we will consider as the limit for accept the next capital value as a valid solution.

Table 3.1: Program parameters

Variable	Value	Specification
α	0.27	Elasticity of production with respect to capital
β	0.994	Discount factor
σ	0.05	Standard deviation of the innovations of the productivity shock
δ	0.011	Depreciation rate of capital
η	2	Elasticity of marginal utility
ρ	0.9	Auto-regressive parameter
grid-size	5.5	Grid width of the productivity shock
kmin_g	0.6	Lower boundary of the capital shock grid
kmax_g	1.4	Upper boundary of the capital shock grid
kmin_e	0.8	Lower bound for the Euler equation residuals
kmax_e	1.2	Upper bound for the Euler equation residuals
nobs_e	200	Number of residuals to be computed

This are the main parameters one can play with. However there are even more such as VI_EPS for changing the stopping criteria, VI_MAX for the maximum number of iterations, etc. Regarding these others parameters, we have fixed them to be:

- For the interpolation steps we used linear interpolation (VI_IP=1).
- The number of grid points for the productivity shock was set to 9 and 20 in two 4-case iterations (Vector iterations see below).
- Vector iterations (nk) were set to 5, 10, 20, 50 (and 250).
- To find the maximum value of the value function we use the binary search algorithm.

The first step of the program computes the probability of the transitions for each state given by the grid of the productivity shock to another state of the same grid. Afterwards, the minimum and maximum values for the productivity shock are computed in order to obtain maximum and minimum values of the capital's shock grid.
The second section computes the stationary solution of deterministic model and defines the *nvec* vector which contains a list of the values for the iterations over the value function. After initializing the policy and the value function, we start a loop over the *nvec*. Inside of it, we will compute the value and policy functions using the code from *SolveVIS.m*, once one has the optimal policy function, one knows which action to take in the consumption level given a state of a given productivity shock and capital stock. Eventually, one computes the residuals from the Euler equation using our policy function and print the maximum error obtained on a separated text file.

3.1.2 SolveVIS.m

This function is designed to compute the value and policy function. To do so, it will use the previously created grids, the matrix of the Markov transition probabilities and the initial policy function (which is a matrix of zeros for the first iteration).

First, the algorithm tries all combinations of Z and K in the following way. For each value of Z_j and each value of K_i with $j \in [1, m]$ and $i \in [1, n]$, we find the index k^* that maximizes:

$$w_k = u(Z_j, K_i, K_k) + \beta \sum_{l=1}^{m} p_{jl} v_{kl}^0 \tag{3.1}$$

Where v_{kl}^0 is the initialized value for the policy function. If $k^* = 1$ evaluate the function $\hat{\phi}$ defined by equation (2.1) at a point close to K_1. If this returns a smaller value than K_1, set $\tilde{K}$=K_1, else we find the maximizer $\tilde{K}$ of $\hat{\phi}$ in the interval $[K_1, K_2]$ using the Golden Section Search algorithm (in our code called *GSS.m*), we store that value in h_{ij}^1, our policy function matrix (the procedure is identical if $k^* = n$). If k^* is not 1 or n, then we find the maximizer $\tilde{K}$ of $\hat{\phi}$ in the interval $[K_{k^*-1}, K_{k^*+1}]$, and set $h_{ij}^1 = \tilde{K}$ and $v_{ij}^1 = \hat{\phi}(\tilde{K})$. Finally, we check for the convergence criteria:

$$\max_{i=1,\ldots,n;j=1,\ldots,m} |v_{ij}^1 - v_{ij}^0| \leq \epsilon(1-\beta)$$

Where epsilon is given by the variable VI_EPS.
If it is fulfilled, the algorithm stops, if not $V^0 = V^1$, $H^0 = H^1$ and repeat the procedure again from where we computed a new value and policy function (the maximum numbers the procedure can be repeated is given by VI_Max).

3.1.3 rf.m

In this function we handle the non-negative investment constraint. Before we start to calculate the each period consumption, we check whether the following condition holds:

$$K_2 \geq (1-\delta) * K_1; \tag{3.2}$$

If it does hold, we further check if C<0. If it is, we place a missing value for consumption as we don't want negative consumption. If it does not hold, we move on to the following condition:

$$\text{if} \quad \eta = 1: \quad C = ln(C); \tag{3.3}$$

$$\text{else} \quad C = Z \cdot K_1^{\alpha} + (1 - \delta) \cdot K_1 - K_2; \tag{3.4}$$

3.1.4 Others

The most important functions have been explained or mentioned in the previous sections, however some of the core functions have been skipped for summarizing purposes. The following functions have also been used and can be checked by viewing the respective Matlab files: rhs_bellman, PF, MachEps, LIP, BLIP, GSS, GetRhs, Euler, VI_valuefunction.

3.2 Solutions

The following tables show the running time, the calculated Euler error and the number of iterations for the specified environment. The computation was done with an Intel Core I7 9700K processor at an average working speed of 3.90 GHz. Sigma was set to $\sigma = 0.05$, with all other parameters unchanged. Our Markov chain is given by 9 and 20 points, respectively. Both times on a grid size of 5.5. Depending on the definition of sigma, the policy function hits the lower and upper bound of the predefined grid. More specified, if $sigma \leq 0.059$, the policy function hits its boundaries.

Table 3.2: The table shows, for a number of grid points for the productivity shock equal to 9, the computational times, error and iterations obtained for different values for the number of iterations of the value function

Results for 9 grid points - productivity shock			
nk	Runtime min:sec:milsec	Error	Iterations
5	00:52:27	8.825842^{-03}	974
10	01:30:13	1.629677^{-02}	816
20	04:51:03	1.347048^{-02}	671
50	10:08:85	4.993371^{-03}	576

Table 3.3: The table shows, for a number of grid points for the productivity shock equal to 20, the computational times, error and iterations obtained for different values for the number of iterations of the value function

Results for 20 grid points - productivity shock			
nk	Runtime min:sec:milsec	Error	Iterations
5	02:10:89	1.075766^{-02}	954
10	05:50:29	2.687904^{-02}	791
20	11:31:05	1.944151^{-02}	646
50	24:56:83	9.624254^{-03}	557

Table 3.4: The table shows, for a number of grid points for the productivity shock equal to 31 and a grid size of 9 the computational times, error and iterations obtained for different values for the number of iterations of the value function

Results for 31 grid points - productivity shock			
nk	Runtime min:sec:milsec	Error	Iterations
5	03:47:21	4.785267^{-02}	1101
10	06:55:29	6.593558^{-02}	944
20	17:12:13	5.061405^{-02}	958
250	243:44:97	5.061626^{-02}	949

Figure 3.1 and 3.2 plots the consumption levels (Z-axis) on total factor productivity (Y-axis) and capital (X-axis). It is clearly visible that there is a kink in the policy function for consumption. Furthermore, we can see that with increasing levels of capital input we have increasing levels of consumption. The plots were created using the last iteration from table 3.7.

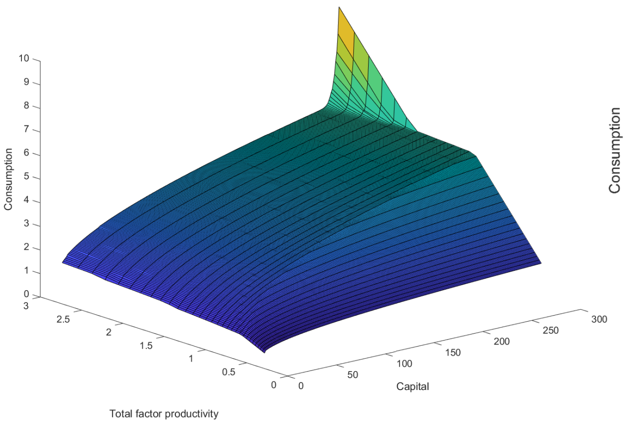

Figure 3.1: Policy function for consumption

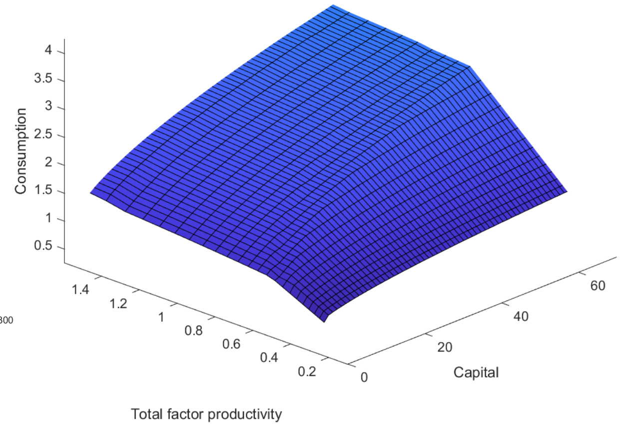

Figure 3.2: Policy function for consumption (scaled)

Figures 3.3-3.6 display the impact of an increase in the productivity shock. We started with $\sigma = 0.0072$ in Figure 3.3 and notice, that the non-negative investment constraint does not bind. As we increase our value for σ, we can see that the constraint becomes binding at a certain level for K. Since our grid_size is rather small, the constraint becomes binding in a close neighbourhood of total factor productivity at 1. By increasing the values, one can get a better picture (see Figure 3.1). In the end, when $\sigma = 0.05$, which means a large productivity shock, we can clearly see the kink when the constraint becomes binding.

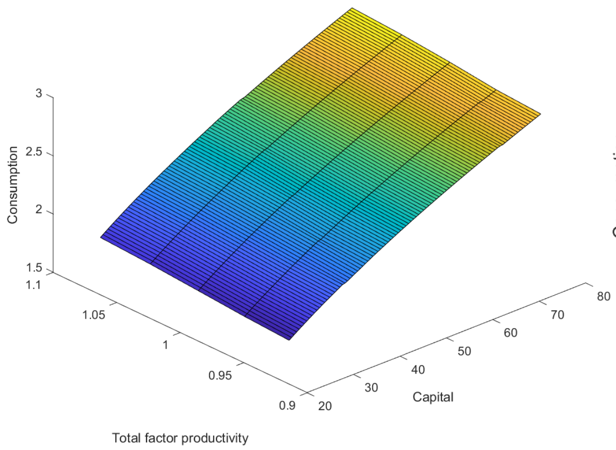

Figure 3.3: sigma=0.0072

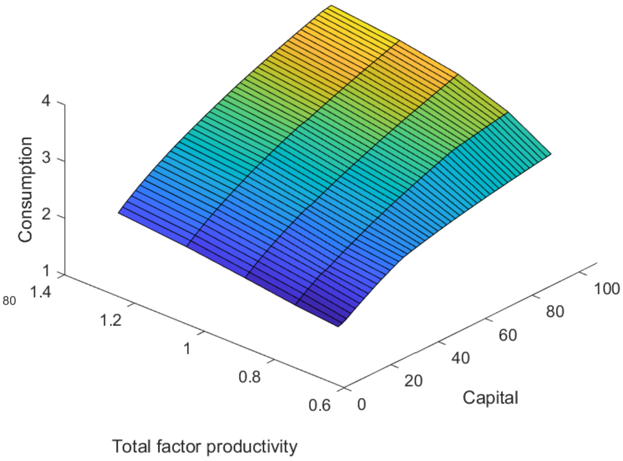

Figure 3.4: sigma=0.03

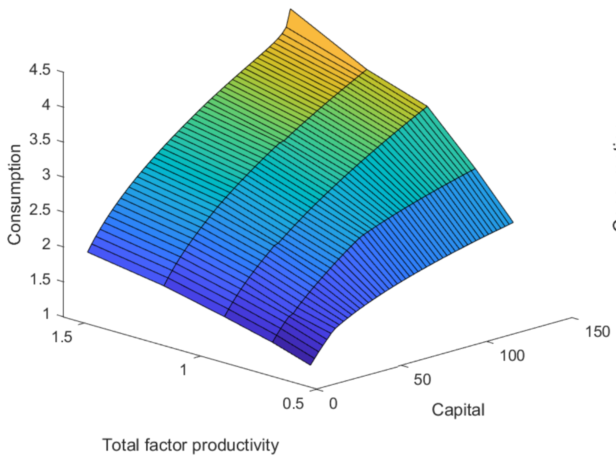

Figure 3.5: sigma=0.045

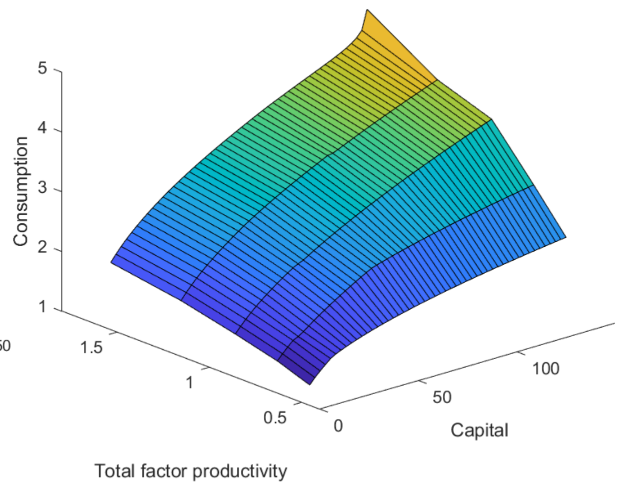

Figure 3.6: sigma=0.05

4 The kink

4.1 Computational aspects

The kink generally represents a point of a function where it is not differentiable, which is a challenging problem in computational economics. Depending on the method of choice, this can cause several problems. E.g., when we are dealing with first order conditions, such as Euler, the method will fail. Furthermore, the kink is a challenge since the approximation will not adequately represent the kink, which in return can lead to false outcomes as the original function can't be represented. This means the approximation error around the kink is significantly large and lead to a poor approximation [4]. Inevitably, it becomes computationally intense and challenging since derivative-free methods result in an increase in the economy's size as we hit the binding constraints. Taking into account the non negativity constraint and endogenous labor supply, the kink cannot be predicted but has to be approximated iteratively. This leads to the difficulties just described [3].

4.2 Economic aspects

Such constraints most frequently arise from endogenous reasons like government tax. The resulting kinks can lead to unexpected economic effects that can induce exactly the opposite of interest [5]. In our stochastic growth model, the kink represents the point in the policy function where the non-negative investment condition becomes *just* binding. This means that the constraint is binding and the associated multiplier is zero. Non negative investment means, that the capital stock will continuously increase, which results in a positive successive increment. Or in other words, consumption cannot exceed production. So, depending on the total factor productivity, the agent would like to smooth consumption or not. This decision is defined around the kink. If we are above the threshold value for a given point K, the constraint does not bind, which means our consumption function is not affected by the non-negative investment constraint. Below the given point K, the agent would like to smooth consumption - which is intuitive - assuming the agent wants to consume at a constant level.

Literature references

[1] Frank Ramsey. "A Mathematical Theory of Saving". In: *The Economic Journal* 38.152 (Dec. 1928), pp. 543–559. ISSN: 0013-0133. DOI: 10.2307/2224098. eprint: https://academic.oup.com/ej/article-pdf/38/152/543/27609734/ej0543.pdf. URL: https://doi.org/10.2307/2224098.

[2] B. Heer and A. Mauner. *Dynamic General Equilibrium Modeling: Computational Methods and Applications*. 2nd ed. 2009. 2nd printing. Springer Publishing Company, Incorporated, 2009. ISBN: 364203148X.

[3] B. Heer and A. Maussner. "Value Function Iteration as a Solution Method for the Ramsey Model". In: *CESifo Group Munich, CESifo Working Paper Series* 231 (Apr. 2008). DOI: 10.1515/jbnst-2011-0404.

[4] K. Judd. *Numerical methods in economics*. Cambridge, Massachusetts: MIT Press, 1998.

[5] R. Moffitt. "The Econometrics of Kinked Budget Constraints". In: *Journal of Economic Perspectives* 4 (1990), pp. 119–139.

[6] E. Sims. "Notes on Value Function Iteration". In: (2011).