

**FUNDAÇÃO GETULIO VARGAS
ESCOLA DE PÓS-GRADUAÇÃO EM
ECONOMIA**

BRUNO SULTANUM TEIXEIRA

THE AIYAGARI MODEL WITH LIQUIDITY SHOCK

Rio de Janeiro, 2010

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

BRUNO SULTANUM TEIXEIRA

THE AIYAGARI MODEL WITH LIQUIDITY SHOCK

Dissertação submetida à Escola de
Pós-Graduação em economia como
requesito parcial para a obtenção
do grau de Mestre em Economia.

Orientador: RICARDO DE OLIVEIRA CAVALCANTI

Rio de Janeiro, 2010

The Aiyagari Model with Liquidity Shock*

A Dissertation Presented to the Faculty of Getulio Vargas Foundation in Candidacy for
the Degree of Master in Economics

Bruno Sultanum Teixeira[†]

Adviser: Ricardo de Oliveira Cavalcanti[‡]

June 24, 2010

Abstract

A version of the Aiyagari (1994) model with a liquidity shock is developed in this work. The model has Huggett (1993) and Aiyagari (1994) as particular cases, but the general one allows for two assets in the economy, a liquid and an illiquid one. Using two different assets implies in two returns clearing the market, so the computational strategy used by Aiyagari and Hugget does not work here. Therefore, Scarf's triangulation algorithm replaces it. Our computational experiment shows that the equilibrium return of the liquid asset is smaller than the return of the illiquid one. In addition, poor people carry relatively more of the liquid asset, which is a source of inequality that is not present in the Aiyagari's work.

Keywords: Liquidity; self-insurance; income inequality.

*I would like to make a especial reference to my adviser Ricardo de Oliveira Cavalcanti who helped me with this work as well as with my training in Economics research. I am also grateful to Jefferson Bertolai and Evelyn Pereira Nunes for helpful comments.

[†]Graduate student at The Graduate School of Economics - Getulio Vargas Foundation.

[‡]Professor of Economics at The Graduate School of Economics - Getulio Vargas Foundation.

Resumo

Neste trabalho é desenvolvida uma versão do modelo de Aiyagari (1994) com choque de liquidez. Este modelo tem Huggett (1993) e Aiyagari (1994) como casos particulares, mas esta generalização permite dois ativos distintos na economia, um líquido e outro ilíquido. Usar dois ativos diferentes implica em dois retornos afetando o “market clearing”, logo, a estratégia computacional usada por Aiyagari e Hugget não funciona. Consequentemente, a triangulação de Scarf substitui o algoritmo. Este experimento computacional mostra que o retorno em equilíbrio do ativo líquido é menor do que o retorno do ilíquido. Além disso, pessoas pobres carregam relativamente mais o ativo líquido, e essa desigualdade não aparece no modelo de Aiyagari

Palavras-Chave: Liquidez; seguro pessoal; desigualdade de renda.

Contents

1	Introduction	7
2	The Model	8
2.1	Environment and Arrangement	8
2.2	The Consumer's Problem	9
2.3	Equilibrium	10
2.4	Limit Cases	11
3	Numerical Method	12
4	Numerical Exercise	16
4.1	Parametrization	16
4.2	Results	16
5	Conclusion	21
A	The Existence Problem	23
A.1	The Consumer's Problem	23
A.2	The Firm's Problem	24
A.3	Equilibrium	24
A.4	Existence Results	25
B	The Algorithm	30
B.1	Evaluating the Stationary Demand	30
B.1.1	Interpolation	30
B.1.2	Optimization	32
B.1.3	Evaluating K_q^d and X_q^d	33
B.2	Scarf's Method	33
C	The Code	36

List of Figures

1	Primitive set $\{q^{n_1}, q^{n_2}, q^{n_3}\}$	13
2	Searching the Primitive Set	14
3	Individuals without the liquidity constraint.	18
4	Individuals with liquidity constraint.	19
5	The distribution of assets among people.	20

1 Introduction

In this work a computational model with heterogeneous agents, built on the Aiyagari (1994) paper, is developed. The model has Huggett (1993) and Aiyagari (1994) as particular cases. If people only save in the form of liquid assets (“*Money*” or Bonds), market clearing takes the shape of Hugget’s model. If they save in the form of capital, then it becomes the Aiyagari’s model. In order to produce a role for both assets, I let shocks block the capacity to sell or buy capital in the spirit of Kiyotaki and Moore (2001) (KM). Thus, when they save on both assets, we have a kind of KM problem¹.

Many difficulties appeared while making it into a computational model. First, because there are two returns clearing the market, the computational strategy used by Aiyagari or Hugget does not work here. So I use Scarf’s triangulation algorithm to find the equilibrium invariant distribution of the model. A different problem coming from the same source is that in Hugget or Aiyagari’s model, there is only one asset, and thus one variable to be chosen, whereas mine has two, capital which suffers from the liquidity shock and an asset which does not. It implies a bigger dimensionality that can be very costly in terms of computing time. To solve this problem I work with lotteries, allowing individuals to choose combinations of the grid points, so I reduce the number of points without loss in the accuracy of estimations.

The computational experiment shows that the equilibrium return of the liquid asset is smaller than the return of capital. Another characteristic found in equilibrium is that poor people carry more of the liquid asset. The effect of the wealth on the portfolio of people seems to be an important issue, and it cannot be studied in Kiyotaki-Moore since with the strong structure imposed by them the portfolio of individuals does not change with the wealth. Therefore, it represents an advantage of the model proposed here over KM.

This work has five sections. In section two I present the model and give the definition of equilibrium. This section is divided in two parts, in the first the environment and the arrangement of the economy are given, in the second part the definition of equilibrium is made with a brief discussion about it. Section three discusses the numerical method used and section four shows the main results. In section five I conclude the work. In addition we have two appendixes, the first discusses technical problems on the existence of an equilibrium and convergence of the algorithm, and the second one describes in details the algorithm.

¹However, this model does not have aggregate shocks, like the original problem of KM.

2 The Model

2.1 Environment and Arrangement

The time is discrete and infinite, beginning at date 0, and there is one perishable consumption good per date.

Individuals

I consider a continuum of individuals with measure one, each with a utility function given by

$$\mathbb{E}_t \sum_{s=t}^{\infty} \beta^{s-t} u(c_s) \quad (1)$$

for a consumption path $\{c_s\}_{s=t}^{\infty}$. Where $u : \mathbb{R} \rightarrow \mathbb{R}$ is a strictly increasing, bounded and strictly concave function. In each period t individuals receive an idiosyncratic shock of productivity $\theta_t \in \Theta \equiv \{\theta_h, \theta_l\}$, where $\theta_h > \theta_l > 0$ and $\{\theta_t\}_t$ follows a Markov process with transition matrix given by Q , which has only one stationary distribution. Individuals offer inelastically their labor, therefore, those with high productivity offer θ_h and the ones with low productivity offer θ_l .

A person can carry capital, $k_t \in K \equiv [0, \bar{k}]$, or a risk-free asset, $x_t \in X \equiv [\underline{x}, \bar{x}]$, with $\underline{x} \leq 0$. The capital pay a return of r_k and the risk-free asset is negotiated at the price q_x and pays one unit of consumption good in the following period.

Firms

There is an infinite number of firms that rent capital and labor from individuals in order to produce consumption goods. The technology is given by a Cobb-Douglas production function with parameter $\alpha \in (0, 1)$ and the capital depreciates in $\delta \in (0, 1)$ during the production. I assume that the firms pay the depreciation of the capital. Then, if the firm rents an amount k of capital it uses it in the production; pays the depreciation, δk ; and the return, $r_k k$; and so, the individuals receive $k + r_k k$.

In addition, I assume that all markets are competitive. Therefore, the firms are price takers, and we denote by w the wage. The problem of the firm in any date is

$$\max_{(k,l) \in K \times \mathbb{R}_+} \{k^\alpha l^{1-\alpha} - (r_k + \delta)k - wl\}. \quad (2)$$

We can anticipate the consequences of a competitive market and establish the following relations:

$$k^s(q) := \min \left\{ (r_k + \delta)^{\frac{1}{\alpha-1}}, \frac{\bar{k}}{\bar{\theta}} \right\} \quad (3)$$

$$w(q) := k^s(q)^\alpha - (r_k + \delta)k^s(q), \quad (4)$$

where $q = (r_k, q_x, w)$ is the vector of prices, $k^s(q)$ is the asset-labor ratio, and $\bar{\theta}$ is the mean of the stationary distribution of θ_t . The first equation comes from the first order condition of the firm while the second one comes from the null profit in equilibrium.

Liquidity Constraints

In this work the liquidity shocks are idiosyncratic and characterized by a stochastic process $\{\iota_t\}_t$, with $\iota_t \in \{0, 1\} \forall t$, such that ι_t are *i.i.d.* each following a Bernoulli distribution with parameter $\mu \in [0, 1]$. That is, in each period, with probability μ , the individual has access to an one-to-one technology between capital and consumption goods, and with a probability $1 - \mu$ he does not access this technology. It characterizes the lack of liquidity in capital.

2.2 The Consumer's Problem

Let $S = K \times X \times \Theta \times \{0, 1\}$ be the state space of the economy for one consumer and consider Σ the borel σ -field of S . Given a vector of prices, $q = (r_k, q_x, w)$, for each state $s \in S$, the individual needs to choose the optimal choice of capital and risk-free assets to carry for the next period. Define the correspondence $\Gamma : S \times \mathbb{R}^3 \rightarrow K \times X \times \mathbb{R}_+$ as the set of feasible choices of the person who faces the prices q and the state s . That is, given s and q , the correspondence $\Gamma(s; q)$ is the set of all vectors $(k', x', c) \in K \times X \times \mathbb{R}_+$ such that

i - $c + k' + q_x x' \leq w\theta + (1 + r_k)k + x$; and

ii - the individual is restricted by his liquidity shock, that is, $k' = k$ if $\iota = 0$;

Given this definition, the consumer's problem in date 0 for an individual who faces a state s_0 is

$$\max \left\{ \mathbb{E} \sum_{t=0}^{\infty} \beta^t u(c_t); (k_{t+1}, x_{t+1}, c_t) \in \Gamma(k_t, x_t, \theta_t, \iota_t; q) \forall t \geq 0 \right\} \quad (5)$$

where $q = (r_k, q_x, w)$ denotes the vector of prices.

The Recursive Formulation

From now on, I will always talk about the recursive formulation of the consumer's problem. As usually done in the recursive format, I denote by y the variable in period t and by y' the variable in $t + 1$. Using this notation, the functional equation which describes the individual problem is

$$V_q(s) = \max \left\{ u(c) + \beta \mathbb{E} [V_q(s')] ; (k', x', c) \in \Gamma(s; q) \right\} \quad (6)$$

with $q = (r_k, q_x, w)$ denoting the vector of prices.

2.3 Equilibrium

The concept of equilibrium that I use here is very similar to those used in Huggett (1993) and Aiyagari (1994). The idea comes from Bewley (1986) who suggests that, with a continuum of people, if there are no aggregate shocks, the prices can be constant in spite of the shocks each one receives. For this concept of equilibrium, it is necessary to look for the stationary distribution of the aggregates, and they need to be in accordance with the prices. Then we have the following definition of equilibrium.

Definition 1 *An equilibrium in this economy is given by:*

i - a vector of prices $q = (r_k, q_x, w) \in \mathbb{R}^3$ with $w = w(q)$;

ii - a measurable policy function for (6) given q , denoted by

$$(g, h, c) : S \rightarrow K \times X \times \mathbb{R}_+;$$

iii - a transition function $P : S \times \Sigma \rightarrow [0, 1]$ in accordance with: the policy functions (g, h) ; the transition matrix for labor shocks, Q ; and the probability of the liquidity shocks, π ;

iv - a distribution over the probability space (S, Σ) , Ψ , which is stationary in respect to the transition function P , and such that:

- the aggregate demand for assets set to zero, $\int_S h(s)d\Psi = 0$;*
- defining $K = \int_S g(s)d\Psi$ and $\bar{\theta} = \int_S \varphi_\theta(s)d\Psi$ where φ_θ is the projection of S on Θ . We have that $(K, \bar{\theta})$ solve the problem (2); and*
- the market of consumption goods is clearing. Thus,*

$$\int_S c(s)d\Psi + \delta K = K^\alpha \bar{\theta}^{1-\alpha}.$$

2.4 Limit Cases

The model presented here preserves the same aspects of the ones in Huggett (1993) and Aiyagari (1994). In addition, for some parameters, the models are the same. In the following paragraphs I discuss these possibilities.

Aiyagari

Suppose that $K = [0, \infty)$, $X = [\underline{x}, \infty)$ and $\mu = 1$. That is, there is no limit in the accumulation of capital and people never face a liquidity constraint. But if there is no liquidity constraint, the capital and the risk-free asset are substitute, which implies $q_x = 1/(1 + r_k)$ and we can look only to $z = k + q_x x$. So the set $\Gamma(s; q)$ is equivalent to the set, $\tilde{\Gamma}(z; \theta; q)$, of all vectors $(z', c) \in [\underline{x}, \infty) \times \mathbb{R}_+$ such that

$$pc + z' \leq w\theta + (1 + r_k)z.$$

Thus, the consumer problem becomes

$$\begin{aligned} & \max \left\{ \mathbb{E} \sum_{t=0}^{\infty} \beta^t u(c_t) \right\} \\ & \text{s.t. } (z_{t+1}, c_t) \in \tilde{\Gamma}(z_t, \theta; q) \quad \forall t \end{aligned} \tag{7}$$

which is exactly the same consumer problem presented in Aiyagari (1994). It is straightforward to see that the problem of the firm is also equivalent.

Hugget

Take $X = [\underline{x}, \infty)$, the value of π equal to one and α equal to zero. Therefore, the technology function will be $F(K, L) = L$ which implies that the capital rented by the firms in equilibrium will be zero and the wage will be equal to the price of consumption. Thus, the consumer problem becomes

$$\begin{aligned} & \max \left\{ \mathbb{E} \sum_{t=0}^{\infty} \beta^t u(c_t) \right\} \\ & \text{s.t. } c_t + q_x x_{t+1} \leq \theta_t + x_t \end{aligned} \tag{8}$$

So, the problem of the consumer here is equivalent to the one found in Huggett (1993). In fact the entire model is the same.

3 Numerical Method

As we discussed above, the equilibrium we need to find here is similar to the one in Aiyagari (1994), so our numerical strategy will be close to his. As in his problem, there is a one-to-one function between the wage and the return of capital. It comes from the hypothesis of competitive markets. Thus, the problem is only to find the returns of assets. The challenge here is that we have two assets and so two equilibrium returns, while he has only one.

Let us summarize the usual method used to obtain the equilibrium return of Aiyagari's model.

1. Start with a guess r_1 very close to $1/\beta - 1$.
2. Find the value function of individuals and evaluate the stationary demand of asset from them, $K^d(r_1)$, given the return r_1 . It will imply a new return r_0 , that will be the return which makes the firms offer a quantity of asset $K^s(r_0) = K^d(r_1)$.
3. Start the bisection method on the interval $[r_0, r_1]$. So, guess $r = (r_0 + r_1)/2$.
4. Evaluate the demand of assets from individuals, $K^d(r)$, and the offer of assets from the firms, $K^s(r)$.
5. If $EK(r) = K_r^d - K_r^s > 0$, make $r_1 = r$ and come back to 3.
6. Else if $EK(r) = K_r^d - K_r^s < 0$, make $r_0 = r$ and come back to 3.
7. Else if $EK(r) = K_r^d - K_r^s = 0$, finish the algorithm.

This algorithm explores the fact that there is only one return to be found using a bisection method. In our case there are two returns, therefore, bisection can not be used. We propose to use the Scarf's algorithm to find the equilibrium. Looking to prices instead of returns, we have here three prices to be determined, so, I use Scarfs' method to find the equilibrium price in the three dimensional simplex.

Let us define

$$q_k = \frac{p}{1 + r_k}.$$

where p is the price of the consumption good. And, instead of normalizing the price of the consumption good as 1, let us normalize to have the prices in the simplex. That is,

$$p + q_k + q_x = 1.$$

Consider the simplex $\Delta = \{(p, q_k, q_x) \in \mathbb{R}_+; p + q_k + q_x = 1\}$. Let $\{q^i\}_{i=1}^n$ be a list of vectors such that $q^1 = (0, 1.1, 1.1)$, $q^2 = (1.2, 0, 1.2)$, $q^3 = (1.3, 1.3, 0)$ and the others q^i belong to the interior of Δ . For each vector q^i , $i = 1, 2, 3$ give the label $l(q^i) = i$. And for q^i with $i = 4, \dots, n$, give the label

$$l(q^i) = \begin{cases} 1 & ; EK(q^i), EX(q^i) \leq 0 \\ 2 & ; EK(q^i) > 0 \wedge EK(q^i) \geq EX(q^i) \\ 3 & ; EX(q^i) > 0 \wedge EX(q^i) > EK(q^i) \end{cases}$$

where $EK(q) = K_q^d - K_q^s$ and $EX(q) = X_q^d$; K_q^d and X_q^d are the stationary demand from individuals of capital; and $K_q^s = \bar{\theta}k^s(q)$ is the demand of capital to rent. Scarf (1967) shows that it is possible to find a set, $\{q^{n_1}, q^{n_2}, q^{n_3}\}$, in which the label of each point in the set is different among them², as in Figure 1. The possibility to find this set characterizes the main idea of Scarf's algorithm.

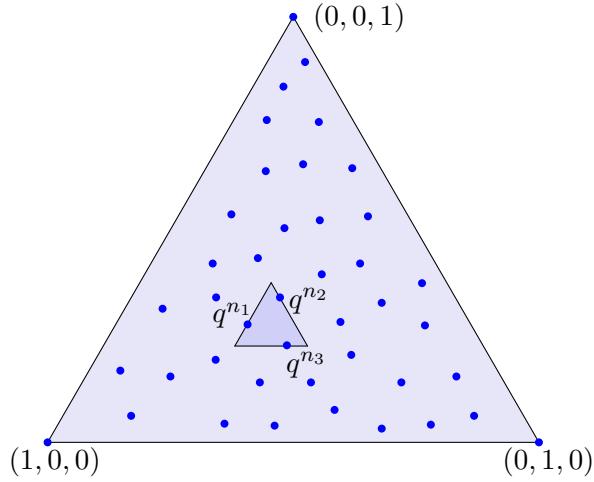


Figure 1: Primitive set $\{q^{n_1}, q^{n_2}, q^{n_3}\}$.

Note that in the point q^{n_1} we have $EK(q^{n_1}) > 0$ and in the point q^{n_2} we have $EK(q^{n_2}) \leq 0$. Therefore, if the function $EK(\cdot)$ is continuous and the points q^{n_1} and q^{n_2} are close enough we will have $EK(q^{n_1}) \approx EK(q^{n_2}) \approx 0$. The same argument can be used to the excess of demand in asset x , since $EX(q^{n_1}) > 0$ and $EX(q^{n_3}) \leq 0$ imply that $EX(q^{n_1}) \approx EX(q^{n_3}) \approx 0$. In addition, aggregating the budget constraint of the individuals we have that

²The conditions of our problem imply that these points will belong to the interior of the simplex for a rich enough grid. See proposition 6 on appendix A for details.

$$C_q^d + K_q^d + q_x X_q^d = w(q)\bar{\theta} + (1 + r_k)K_q^d + X_q^d.$$

Using equation (4) we have that

$$C_q^d + K_q^d + q_x X_q^d = \left(\frac{K_q^s}{\bar{\theta}} \right)^\alpha \bar{\theta} - (r_k + \delta)K_q^s + (1 + r_k)K_q^d + X_q^d.$$

By $EK(q^{n_1}) \approx EK(q^{n_2}) \approx 0$ and $EX(q^{n_1}) \approx EX(q^{n_3}) \approx 0$, we can conclude that

$$C_{q^{n_1}}^d + \delta K_{q^{n_1}}^s \approx (K_{q^{n_1}}^s)^\alpha \bar{\theta}^{1-\alpha}.$$

It shows how this algorithm can be used to find an equilibrium price, or at least approximate one. As the list $\{q^i\}_{i=1}^n$ grows, with those points well distributed in the simplex, the points $\{q^{n_1}, q^{n_2}, q^{n_3}\}$ become closer to each other. Thus, the approximations $EK(q^{n_1}) \approx EK(q^{n_2}) \approx 0$ and $EX(q^{n_1}) \approx EX(q^{n_3}) \approx 0$ can be made as good as we want them to be. Now, it is left to us to establish how to search for this set of three points.

Searching the Primitive Set

We start the algorithm with three points $\{q^2, q^3, q^j\}$, where q^j is the vector in the family $\{q^i\}_{i=4}^n$ with the largest first coordinate. Figure 2 shows the subsimplex, S_1 , generated by this primitive set.

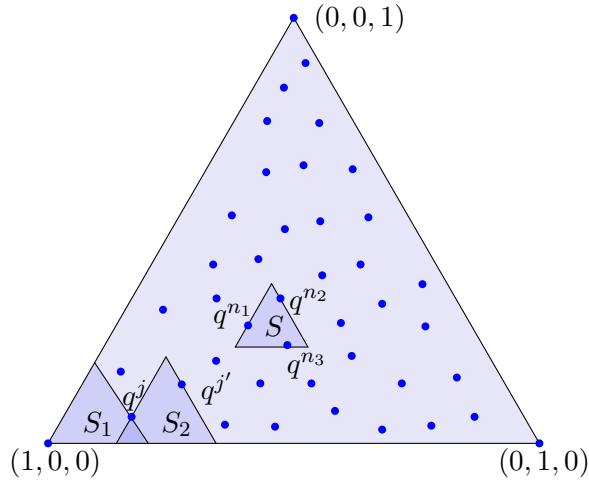


Figure 2: Searching the Primitive Set

Consider $\{2, 3, l(q^j)\}$ the set of labels of each point in $\{q^2, q^3, q^j\}$. So, if $l(q^j) = 1$ the algorithm is finished. If not, for example if $l(q^j) = 3$, we replace the vector in the primitive

set that have this label, in this case q^3 , by another vector that keeps the set a primitive one. In Figure 2, this replacement introduced the point $q^{j'}$ and generated the subsimplex S_2 . This process of replacement will, in a finite number of steps, converge to a primitive set, as $\{q^{n_1}, q^{n_2}, q^{n_3}\}$ in Figure 2, with the desired property³.

³See Scarf (1973).

4 Numerical Exercise

In order to better understand what happens in our economy, I simulate it with parameters that are common in the literature of general equilibrium with idiosyncratic shocks. The reader should note that it is not a calibration exercise.

4.1 Parametrization

The economy presented here has twelve parameters. These parameters are listed below with a brief description of them, and the specification of each value.

Preferences - The utility function used here is the CRRA utility. Thus, there are two parameters of preference, the discount rate β , and the constant relative risk aversion coefficient σ . I set the value of β at 0.96 and the risk aversion at 3.

Technology - Since I assume a Cobb-Douglas format for the production function, there are two parameters of technology. The first one is the depreciation rate δ , which I assume to be 0.08. The second one is the coefficient of the Cobb-Douglas α , I define this value at 0.36.

Productivity Process - The productivity process is completely characterized by four parameters, θ_l , θ_h , Q_{ll} and Q_{hh} . Where $Q_{ll} = P(\theta_{t+1} = \theta_l | \theta_t = \theta_l)$ and $Q_{hh} = P(\theta_{t+1} = \theta_h | \theta_t = \theta_h)$. I normalize $\theta_h = 1$ and put θ_l as ten percent of θ_h . In addition, I set $Q_{ll} = 0.5$ and $Q_{hh} = 0.925$.

Liquidity Process - The liquidity process is characterized only by the parameter of the Bernoulli distribution μ , I set this value to be 0.5.

Limits to carry Assets - A person can carry capital, $k_t \in K \equiv [0, \bar{k}]$, or a risk-free asset, $x_t \in X \equiv [\underline{x}, \bar{x}]$, with $\underline{x} \leq 0$. I set the value of \bar{k} at 25 and \underline{x} , \bar{x} at -4 and 25 respectively.

4.2 Results

The table below summarizes the equilibrium prices found with the parameters described in section 4.1. Here the return r_x is defined as

$$r_x = \frac{1}{q_x} - 1,$$

and the equilibrium price vector denoted by $q^* = (r_k^*, q_x^*, w^*)$.

q_x^*	r_k^*	r_x^*	w^*
0.32948	3.59%	3.55%	0.413

Table 1: Equilibrium Prices.

Note that the return of the capital is greater than the risk-free asset. It indicates that there is a liquidity premium, the return of the illiquid asset is 0.04% greater than the return of the liquid one. The intuition of this phenomenon is that each individual receives an idiosyncratic productivity shock, thus his savings have the additional purpose of being precautionary. But, since using capital to consumption can be blocked by the liquidity shock, this asset does not accomplish this goal very well. So individuals are willing to pay a liquidity premium in order to protect themselves against the productivity shock.

Let us remember what happens in a model with a single liquid asset as in Aiyagari's or Hugget's model. In these models the individual who suffers a positive productivity shock saves resources for periods when he receives the negative one. He does it because there is no insurance in the economy so he needs to self insure himself. This behavior implies that in equilibrium the return of the asset needs to be smaller than in the economy with perfect insurance. In Aiyagari's case it is equivalent to saying that we have over accumulation of capital. What this model is saying is that the capital being not a liquid asset goes against this result. That is, the absence of liquidity is a force that makes people reduce their demand of capital, thus the tendency of over accumulation of capital becomes weaker.

Below we study the policy function of the individuals in order to better understand what is happening in our economy.

The Policy Function

Consider $(g, h) : S \rightarrow K \times X$ the policy function of the individuals given the equilibrium vector price q^* . That is, given the state $(k, x, \theta, \iota) \in S$, the values $g(k, x, \theta, \iota)$ and $h(k, x, \theta, \iota)$ are the choice of capital and risk-free asset, respectively, when the vector price is q^* . Figures 3 and 4 show the behavior of these functions. Note that the policy function is not defined at the point $(k, x) = (0, -4)$. It occurs because $(0, -4)$ is not a feasible point or $(0, -4) \notin S_q$, where S_q is the set of the feasible amounts of assets⁴. It means that, if the individual chooses

⁴See proposition 1 on appendix A for details on this set.

$(k, x) = (0, -4)$, with a strictly positive probability he will have a negative consumption, which is not allowed.

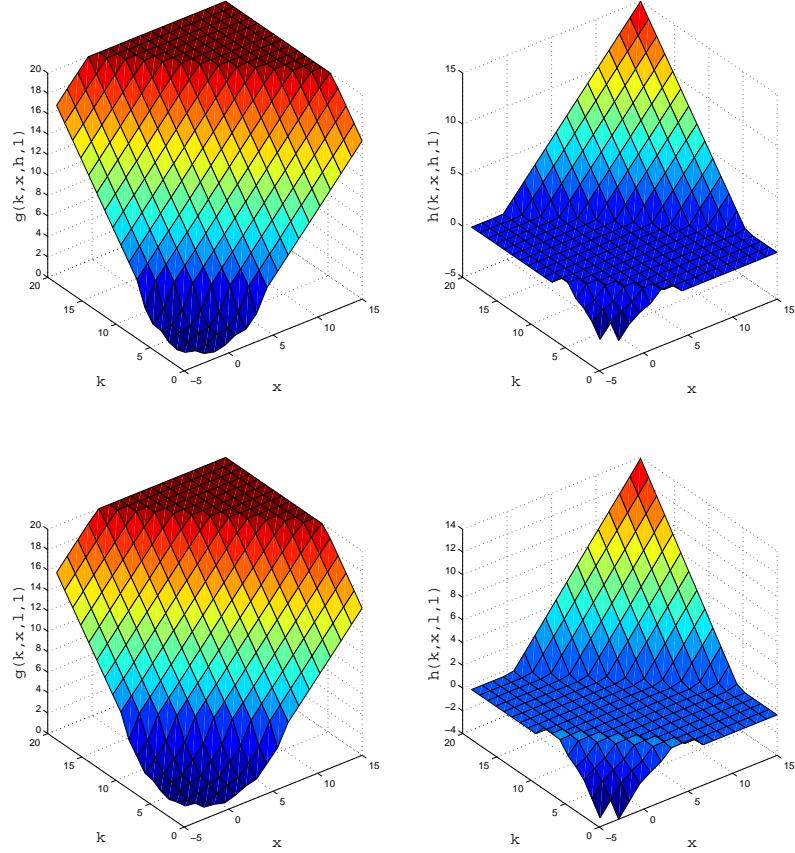


Figure 3: Individuals without the liquidity constraint.

We can see by Figure 3 that very poor individuals, those with (k, x) near to $(0, \underline{x})$, carry a little amount of both assets when they are not blocked by the liquidity shock. When increasing the wealth of the individuals, the first effect is to increase the amount of the liquid asset that they demand more than the amount of capital. It occurs because a poor person has a low consumption, thus the marginal utility of the consumption is high and so this individual is willing to pay more of the liquidity premium in order to protect himself against the negative productivity shock. On the other hand, while the wealth increased, this effect became weaker. Thus, individuals start to reduce their demand of the liquid asset and increase their demand of the illiquid one.

Another important point we learn from the policy function of individuals without the

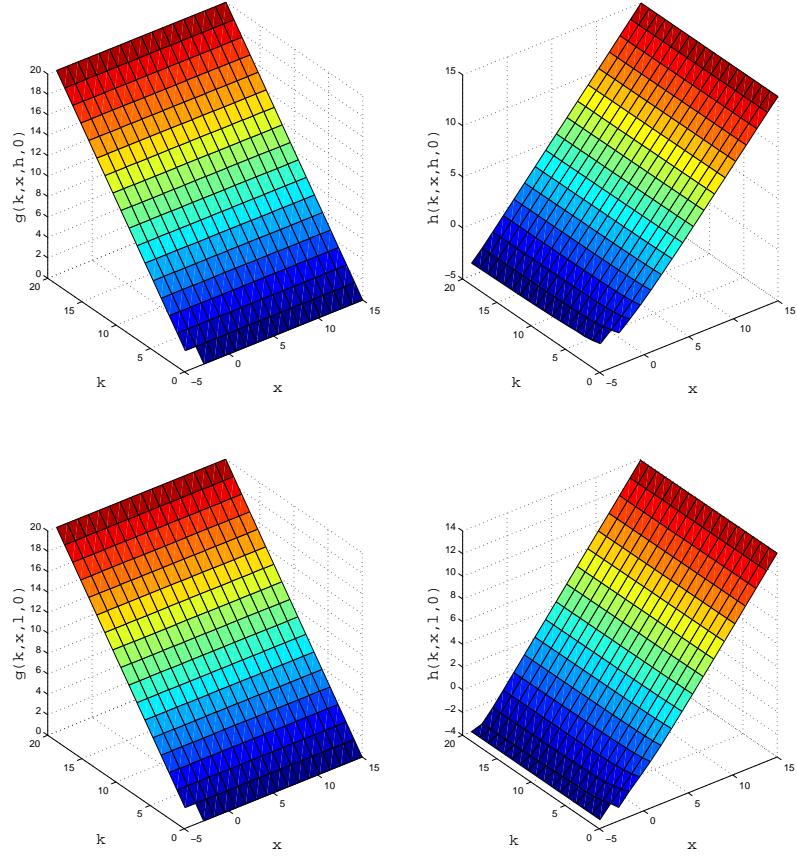


Figure 4: Individuals with liquidity constraint.

liquidity constraint is that people do not explore the difference in the returns of assets to the most. That is, they could carry $x = \underline{x}$ in order to increase their investment in capital which pays a higher return. They do not do this because, in the case of a negative productivity shock and a negative liquidity shock, with the borrowing constraint binded they could not smooth their consumption. Note in Figure 4 that individuals with a negative productivity and liquidity shock reduce their amount of asset x . Since they are blocked to change their amount of k , reducing x is the only way they have to smooth consumption.

The Stationary Distribution

An important equilibrium object to be studied here is the stationary distribution of assets among people in the economy, and Figure 5 gives this distribution, f . By the graph we can see that the distribution concentrates in two groups. The first one, A, carries a quantity

around 1 of the risk-free asset and around 2 of capital. While the second group, B, carries less of the liquid asset, something around -0.5 , but carry more than 4 of capital. Thus, the individuals of the group A are poorer than the ones from group B, since the wealth of individuals is given by $w\theta + (1 + r_k)k + x$.

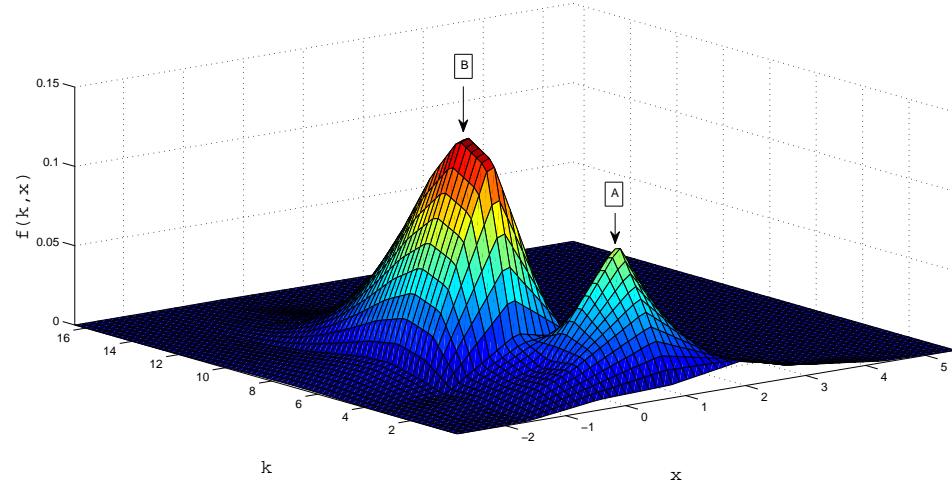


Figure 5: The distribution of assets among people.

As we can see in the policy functions, the poor carry more of the liquid asset than of the illiquid one, this characterizes the group A. But while the wealth increased, individuals started to reduce their demand of the liquid asset and increased their demand of capital. So, they migrate to group B. It is important to note that it can be a source of inequality in the economy, since the poor invests less in the asset with the higher return than the rich.

This new source of inequality could be an answer to Aiyagari (1994). He found that his model cannot replicate the observed degrees of inequality, only the relative ones. He wrote:

“Lorenz curves and Gini coefficients show that the model does generate empirically plausible relative degrees of inequality. Consumption exhibits the least inequality, followed by net income, gross income and then capital, and saving exhibits the greatest inequality. However, the model cannot generate the observed degrees of inequality.”

Aiyagari (1994), page 681.

Therefore, our model indicates that liquidity could be the source of inequality that was missing in Aiyagari's paper. In order to verify this assertion, this model needs to be calibrated.

5 Conclusion

In this work a computational model with heterogeneous agents was built on the Aiyagari (1994) model. The model has Huggett (1993) and Aiyagari (1994) as particular cases, however the computational strategy used by them does not work here. So I use Scarf's triangulation algorithm to find the equilibrium invariant distribution of the model.

The computational experiment shows that the equilibrium return of the liquid asset is smaller than the return of the illiquid one. In some way this result goes against the result of over accumulation of capital found in Aiyagari. The lack of liquidity is a force that makes people reduce their demand of capital if capital is an illiquid asset, thus, the tendency to over accumulation described by him becomes weaker.

Another characteristic found in equilibrium is that poor individuals carry more of the liquid asset than the rich one. The effect of the liquidity on the returns of assets and the portfolio seems to be an important issue, since it can have implications on income inequality.

Aiyagari (1994) does not differentiate liquidity across assets, and his model cannot replicate the observed degrees of inequality in the economy. The lack of inequality found could be accounted for models with differences on liquidity across assets.

Therefore, a future work should be to calibrate this model in order to verify if it can really account for the lack of inequality found in Aiyagari (1994).

References

- Aiyagari, S. Rao**, “Uninsured idiosyncratic risk and aggregate saving,” *The Quarterly Journal of Economics*, 1994, pp. 659–684.
- Bewley, T.**, “Stationary monetary equilibrium with a continuum of independently fluctuating consumers,” *Contributions to Mathematical Economics in Honor of Gerard Debreu*. Amsterdam: North Holland, 1986.
- Cavalcanti, Ricardo O.**, “On the smallest concave interpolant on the plane,” *Ph.D Thesis(University of Minnesota, Minneapolis, MN)*, 1997.
- Huggett, M.**, “The risk-free rate in heterogeneous agent incomplete insurance economies,” *Journal of Economic Dynamics and Control*, 1993, 17 (5/6), 953–970.
- Kiyotaki, N. and J. Moore**, “Liquidity, business cycles and monetary policy,” *Clarendon Lecture, London School of Economics*, 2001.
- Lagarias, J.C., J.A. Reeds, M.H. Wright, and P.E. Wright**, “Convergence properties of the Nelder-Mead simplex method in low dimensions,” *SIAM Journal on Optimization*, 1999, 9 (1), 112–147.
- Scarf, H.**, “The approximation of fixed points of a continuous mapping,” *SIAM Journal on Applied Mathematics*, 1967, 15 (5), 1328–1343.
- , “The Computation of Economic Equilibria (with collaboration of T. Hansen), Cowles Foundation Monograph No. 24,” 1973.
- Stokey, N.L., R.E. Lucas, and E.C. Prescott**, *Recursive methods in economic dynamics*, Harvard University Press, 1989.

A The Existence Problem

Before we start to analyze the existence problem, let us define some objects necessary to this purpose. From now on I work with the variable $(1 + r_k)k$ instead of k . But I still denote this new variable by k . In addition I define

$$q_k = p/(1 + r_k), \quad q_x = pq_x \quad \wedge \quad w = pw;$$

where p is the consumption price.

Let us see the firms' and consumer's problem after this change.

A.1 The Consumer's Problem

Let $S = K \times X \times \Theta \times \{0, 1\}$ be the state space of the economy for one consumer and consider Σ the borel σ -field of S . Define the correspondence $\Gamma : S \times \mathbb{R}^4 \rightarrow K \times X \times \mathbb{R}_+$ as the set of feasible choices of the person who faces the prices $q = (p, q_k, q_x, w)$ and the state s . That is, given s and q , the correspondence $\Gamma(s; q)$ is the set of all vectors $(k', x', c) \in K \times X \times \mathbb{R}_+$ such that

i - $pc + q_k k' + q_x x' \leq w\theta + p(k + x)$; and

ii - the individual is restricted by his liquidity shock, that is, $k' = k$ if $\iota = 0$;

Given this definition, the consumer's problem in date 0 for an individual who faces a state s_0 is

$$\max \left\{ \mathbb{E} \sum_{t=0}^{\infty} \beta^t u(c_t); (k_{t+1}, x_{t+1}, c_t) \in \Gamma(k_t, x_t, \theta_t, \iota_t; q) \forall t \geq 0 \right\} \quad (9)$$

where $q = (p, q_k, q_x, w)$ denotes the vector of prices.

The Recursive Formulation

The functional equation which describes the individual problem is

$$V_q(s) = \max \left\{ u(c) + \beta \mathbb{E} [V_q(s')]; (k', x', c) \in \Gamma(s; q) \right\} \quad (10)$$

with $q = (p, q_k, q_x, w)$ denoting the vector of prices.

A.2 The Firm's Problem

It is assumed that all markets are competitive. Therefore, the firms are price takers, and we denote by p and w the consumption good price and the wage, respectively. So, the problem of the firm in any date is

$$\max_{(k,l) \in K \times \mathbb{R}_+} \left\{ p \left(\frac{q_k}{p} k \right)^\alpha l^{1-\alpha} - [p - (1 - \delta)q_k]k - wl \right\}. \quad (11)$$

We can anticipate the consequences of a competitive market and establish the following relations:

$$k^s(q) := \min \left\{ \frac{p}{q_k} \left(\frac{p - (1 - \delta)q_k}{\alpha q_k} \right)^{\frac{1}{\alpha-1}}, \bar{\theta} \right\} \quad (12)$$

$$w(q) := p \left(\frac{q_k}{p} k^s(q) \right)^\alpha - (p - (1 - \delta)q_k)k^s(q), \quad (13)$$

where $k^s(q)$ is the asset-labor ratio, and $\bar{\theta}$ is the mean of the stationary distribution of θ_t .

A.3 Equilibrium

We have the following definition of equilibrium.

Definition 2 *An equilibrium in this economy is given by:*

i - a vector of prices $q = (p, q_k, q_x) \in \mathbb{R}^3$ and $w = w(q)$;

ii - a measurable policy function for (10) given q , denoted by

$(g, h, c) : S \rightarrow K \times X \times \mathbb{R}_+$;

iii - a transition function $P : S \times \Sigma \rightarrow [0, 1]$ in accordance with: the policy functions (g, h) ; the transition matrix for labor shocks, Q ; and the probability of the liquidity shocks, π ;

iv - a distribution over the probability space (S, Σ) , Ψ , which is stationary in respect to the transition function P , and such that:

- the aggregate demand for assets set to zero, $\int_S h(s)d\Psi = 0$;

- defining $K = \int_S g(s)d\Psi$ and $\bar{\theta} = \int_S \varphi_\theta(s)d\Psi$ where φ_θ is the projection of S on Θ .

We have that $(K, \bar{\theta})$ solve the problem (11); and

- the market of consumption goods is clearing. Thus,

$$\int_S c(s)d\Psi + \delta \frac{q_k}{p} K = \left(\frac{q_k}{p} K \right)^\alpha \bar{\theta}^{1-\alpha}.$$

A.4 Existence Results

It is easy to see that there is no equilibrium with non-positive prices. Thus, we will search for the equilibrium only in the set

$$\overset{\circ}{\Delta} := \left\{ (p, q_k, q_x) \in \mathbb{R}_{++}^3; p + q_k + q_x = 1 \right\}.$$

Now let us look at what is feasible for the individuals to choose⁵. Given the price $q \in \overset{\circ}{\Delta}$, we say that $(k, x) \in K \times X$ is feasible if there is a measurable function $(g, h, c) : S \rightarrow K \times X \times \mathbb{R}_+$ such that the stochastic process $\{s_t\}_{t=0}^\infty$ generated by (g, h) , with $s_0 = (k, x, \theta_l, 0)$ satisfies $(g, h, c)(s_t) \in \Gamma(s_t, q, w(q))$, $\forall t$ and all realizations of $\{s_\tau\}_{\tau=0}^t$ with positive measure.

Define S_q ⁶ as the set of all points in $s \in S$ such that (k, x) is feasible given the price $q \in \overset{\circ}{\Delta}$. Now we have the following proposition.

Proposition 1 Define the set $\hat{S}_q := \{s \in S; w(q)\theta_l + (p - q_k)k + (p - q_x)x \geq 0\}$, thus we have that $S_q = \hat{S}_q$.

Proof. First let us see that $\hat{S}_q \subset S_q$. It is easy to see that, given $s \in \hat{S}_q$, the function (g, h, c) defined as $g(s) = k$, $h(s) = x$ and $c(s) = (w(q)\theta_l + (p - q_k)k + (p - q_x)x)/p$ satisfies the conditions in the definition of S_q . Thus s belongs to S_q . Now take $s \in S_q$ and suppose by way of contradiction that $s \notin \hat{S}_q$. But we know that

$$\begin{aligned} 0 &\leq w(q)\theta_l + (p - q_k)k + px - q_xh(s) \\ &= w(q)\theta_l + (p - q_k)k + (p - q_x)x + q_x(x - h(s)). \end{aligned}$$

Since $s_0 \notin \hat{S}_q$ the first part $w(q)\theta_l + (p - q_k)k + (p - q_x)x < 0$. Thus $x - h(s) = \epsilon > 0 \Rightarrow h(s) = x - \epsilon$, from what we conclude that $x_{t+1} \leq x_0 - t\epsilon$ for a sequence of shocks $\{\theta_l, 0\}_{\tau=0}^t$. So for a sequence of shocks $\{\theta_l, 0\}_{\tau=0}^t$ big enough, $x_{t+1} \notin X$ or $c_t < 0$. Contradiction with $s \in S_q$, and we can conclude that $s \notin \hat{S}_q$. ■

Hence, I can define the correspondence

$$\begin{aligned} \tilde{\Gamma}_q : S_q &\rightarrow K \times X \times \mathbb{R}_+ \\ s &\mapsto \{(k, x, c) \in \Gamma(s, p, q_k, q_x, w(p, q_k, q_x)); (k, x, \theta, \iota) \in S_q\}. \end{aligned}$$

⁵For more details on feasibility of a plan in a dynamic program see Stokey et al. (1989).

⁶This set can be understood as an extension of the set $[\phi_{w,r}, \infty)$ in Aiyagari (1994).

We can easily note that the correspondence $\tilde{\Gamma}_q$ is non-empty, convex, continuous and compact valued.

The next proposition establishes the existence of a solution for the functional equation, which defines the recursive problem. In addition, some properties of the solution and the argmax are established.

Proposition 2 *Given $q \in \overset{\circ}{\Delta}$ and the operator*

$$T \cdot W(s) = \max \left\{ u(c) + \beta \mathbb{E} [W(s')] ; (k', x', c) \in \tilde{\Gamma}(s; q) \right\}, \quad (14)$$

there is a singleton bounded, continuous function $V_q : S_q \rightarrow \mathbb{R}$, fixed point of T , which is concave in (k, x) ; for any point $W \in \mathcal{C}(S_q)$, the sequence $\{T^n \cdot W\}_n$ converges to V_q ; and defining the correspondence $(G, H, C) : S \rightarrow K \times X \times \mathbb{R}_+$ as the argmax of the problem (14), it is upper semi continuous (u.s.c.).

Proof. Since u and $\tilde{\Gamma}$ are continuous and $\tilde{\Gamma}$ is non-empty with compact values, we have by Berge's Maximum Theorem that $T \cdot W$ is continuous. Moreover, u bounded assures us that $T \cdot W$ is bounded. Hence, T maps $\mathcal{C}(S_q)$ in itself. Using now the Blackwell's sufficient conditions and that $\beta \in (0, 1)$ we have that T defines a contraction mapping on the space $\mathcal{C}(S_q)$ with modulus β . Thus, by the Contraction Mapping Theorem⁷, there is a singleton function $V_q : S \rightarrow \mathbb{R}$ which solves $T \cdot V_q = V_q$ and for all $W \in \mathcal{C}(S_q)$ the sequence $\{T^n \cdot W\}_n$ converges to V_q . To see the concavity note that for all $W \in \mathcal{C}(S_q)$ concave, we have $T \cdot W$ concave since the maximum operator preserves the concavity. Therefore, taking V_q as the limit of a sequence $\{T^n \cdot W\}_n$ of concave functions we have that V_q is concave. For the second part, Berge's Theorem also guarantees that

$$(G, H, C) \equiv \text{argmax} \left\{ u(c) + \beta \mathbb{E} [V_q(s')] ; (k', x', c) \in \tilde{\Gamma}(s; q) \right\}$$

is an u.s.c. correspondence, which concludes the proof. ■

The next proposition establishes the relation between the original problem and the recursive one.

Proposition 3 *Consider $q \in \overset{\circ}{\Delta}$ and define the value function, $V^* : S \rightarrow \mathbb{R}$, as*

$$V_q^*(s_0) = \sup \left\{ \mathbb{E} \sum_{t=0}^{\infty} \beta^t u(c_t) ; (k_{t+1}, x_{t+1}, c_t) \in \Gamma(s_t, q) \forall t \geq 0 \right\}. \quad (15)$$

⁷Note that $\mathcal{C}(S_q)$ is a complete metric space with the sup norm. Thus, the Contraction Mapping Theorem can be applied.

So, $V_q^* = V_q$ and any plan generated by (G, H, C) attain the maximum in the problem (15).

Proof. For this proof, only verifying the conditions from Theorem 9.2 of Stokey et al. (1989) is necessary. The assumptions 9.1 and 9.2 are easy to see since S is a subset of the Euclidean space and the function u is bounded. The conditions over V come from it belonging to $\mathcal{C}(S)$, which concludes the proof. ■

The value function V_q is not necessarily strictly concave in (k, x) , hence, it is not easy to establish that (G, H, C) is in fact a continuous function instead of a correspondence. The next proposition shows this result.

Proposition 4 *Given $q \in \overset{\circ}{\Delta}$, there are continuous functions $(g, h, c) : S \rightarrow K \times X \times \mathbb{R}_+$ such that, for all $s \in S$, we have $G(s) = \{g(s)\}$, $H(s) = \{h(s)\}$ and $C(s) = \{c(s)\}$.*

Proof. Since (G, H, C) is u.s.c. we only need to show that the solution of the problem

$$\max \left\{ u(c) + \beta \mathbb{E} [V(s', q)] ; (k', x', c) \in \tilde{\Gamma}(s_0; q) \right\}$$

is unique for any $s_0 \in S$. Assume by way of contradiction that there are two solutions $(k', x', c) \neq (\tilde{k}', \tilde{x}', \tilde{c})$ which solve the problem above for a given s_0 . And considerer $\{c_t\}_{t=0}^\infty$ and $\{\tilde{c}_t\}_{t=0}^\infty$ their respectively consumption plans, hence

$$V(s) = \mathbb{E} \sum_{t=0}^{\infty} u(c_t) = \mathbb{E} \sum_{t=0}^{\infty} u(\tilde{c}_t).$$

Note that, $\{c_t\}_{t=0}^\infty \neq \{\tilde{c}_t\}_{t=0}^\infty$ implies a contradiction since, by the convexity of $\tilde{\Gamma}$, the plan $\{\alpha c_t + (1 - \alpha) \tilde{c}_t\}_{t=0}^\infty$ belongs the budget set $\forall \alpha \in (0, 1)$ and, for u being strictly concave, it generates a higher utility. Let us use this fact to establish the desired result. It is easy to see that $q_k k' + q_x x' \neq q_k \tilde{k}' + q_x \tilde{x}'$ implies $c_0 \neq \tilde{c}_0$, then we have a contradiction.

Now assume that $k' + x' \neq \tilde{k}' + \tilde{x}'$, without loss of generality $k' + x' > \tilde{k}' + \tilde{x}'$, and $\{c_t\}_{t=0}^\infty = \{\tilde{c}_t\}_{t=0}^\infty$. Note that the plan $c_t^* = \tilde{c}_t \forall t \neq 1$; $c_1^* = \tilde{c}_1$ if $\iota = 0$ and $c_1^* = \tilde{c}_1 + k' + x' - \tilde{k}' - \tilde{x}'$ if $\iota = 1$ is necessarily feasible and implies a higher utility, which is a contradiction. Thus, it is necessary that $q_k k' + q_x x' = q_k \tilde{k}' + q_x \tilde{x}'$ and $k' + x' = \tilde{k}' + \tilde{x}'$. But, since $(k', x') \neq (\tilde{k}', \tilde{x}')$, it implies $q_k = q_x$.

In order to analyze the case when $q_k = q_x$, let us look at the following problem

$$\max \left\{ \mathbb{E} \sum_{t=0}^{\infty} \beta^t u(c_t) \right\}$$

$$s.t. \begin{cases} pc_t + q_k z_{t+1} = w(q)\theta_t + p z_t \quad \forall t \geq 0, \\ b(q) \leq z_t \leq \bar{k} + \bar{x} \quad \forall t \geq 0, \text{ and} \\ z_0 = k_0 + x_0. \end{cases}$$

Where, $b(q) = \max\{\underline{x}; -w(q)\theta_l/(p - q_k)\}$ if $p > q_k$, and $b(q) = \underline{x}$ if $p \leq q_k$. Now consider $\{c_t^*, z_t^*\}_{t=0}^\infty$ the solution of this problem⁸. Without loss of generality assume that $k' > \tilde{k}'$. If $p/q_k \geq 1/\beta$ a sequence of shocks $\theta_t = \theta_h$ and $\iota_t = 0$ long enough implies $z^* > \bar{x} + k' > \bar{x} + \tilde{k}'$, hence $\{c_t\}_{t=0}^\infty \neq \{\tilde{c}_t\}_{t=0}^\infty$. Besides, if $p/q_k < 1/\beta$ a sequence of shocks $(\theta_l, 0)$ big enough implies $z^* < \underline{x} + \tilde{k}' < \underline{x} + k'$, thus $\{c_t\}_{t=0}^\infty \neq \{\tilde{c}_t\}_{t=0}^\infty$.

Summarizing, since $\{c_t^*\}_{t=0}^\infty$ attains the optimal in the problem without the liquidity constraint it generates a utility strict greater than $\{c_t\}_{t=0}^\infty$ or $\{\tilde{c}_t\}_{t=0}^\infty$, and $k' \neq \tilde{k}'$ implies that restrictions are not binding for the same event, so we have that $\{c_t\}_{t=0}^\infty \neq \{\tilde{c}_t\}_{t=0}^\infty$. Therefore, we have a contradiction and conclude that the problem has a unique solution. ■

Proposition 5 Given $q \in \overset{\circ}{\Delta}$ consider $P_q : S_q \times \Sigma \rightarrow [0, 1]$ a transition function generated by (g_q, h_q) as in Theorem 9.13 of Stokey et al. (1989). Then, there exists a probability measure, Ψ_q , over (S, Σ) which is invariant in relation to P_q .

Proof. See Theorems 9.14 and 12.10 of Stokey et al. (1989). ■

The next proposition gives conditions under which there is an equilibrium. The proof is standard in the literature and most of the arguments can be found in Scarf (1973). The advantage of this proof is that the computational method comes directly from this demonstration.

Proposition 6 If the functions:

$$\begin{aligned} k^d(q) &= \int_S g(s, q) \Psi_q(ds) / \int_S \varphi_\theta(s, q) \Psi_q(ds) \\ x^d(q) &= \int_S h(s, q) \Psi_q(ds) / \int_S \varphi_\theta(s, q) \Psi_q(ds) \end{aligned}$$

are continuous in q , then there is an equilibrium.

Proof. Consider the functions $Ek(q) := k^d(q) - k^s(q)$ and $Ex(q) := x^d(q)$. Let $\{q^i\}_{i=1}^n$ be a list of vectors such that $q^1 = (0, 1.1, 1.1)$, $q^2 = (1.2, 0, 1.2)$, $q^3 = (1.3, 1.3, 0)$ and

⁸This problem was well studied in many works, like Aiyagari (1994), without the limit $\bar{k} + \bar{x}$. The asserts made here come directly from these works.

$q^i \in \overset{\circ}{\Delta} \forall i = 4, \dots, n$. For each vector q^i , $i = 4, \dots, n$, give a label

$$lq^i = \begin{cases} 1 & ; Ek(q^i), Ex(q^i) \leq 0 \\ 2 & ; Ek(q^i) > 0 \wedge Ek(q^i) \geq Ex(q^i) , \\ 3 & ; Ex(q^i) > 0 \wedge Ek(q^i) < Ex(q^i) \end{cases}$$

and for $i = 1, 2$ and 3 define $lq^i = i$.

From Scarf (1967) there are $q_n^{i_1}, q_n^{i_2}, q_n^{i_3} \in \{q^i\}_{i=1}^n$ such that $lq_n^{i_1} = 1$, $lq_n^{i_2} = 2$ and $lq_n^{i_3} = 3$.

Now let us add points in the set $\{q^i\}_{i=1}^n$ in a way that $\sup\{|q^i - q^j|; i \neq j\} \rightarrow 0$. It implies that, passing to a subsequence if necessary, $q_n^{i_1}, q_n^{i_2}, q_n^{i_3} \rightarrow \bar{q}$. Let us see that $\bar{q} \in \overset{\circ}{\Delta}$. Suppose by way of contradiction that $\bar{q} \notin \overset{\circ}{\Delta}$, then $\bar{p} = 0$ or $\bar{q}_k = 0$ or $\bar{q}_x = 0$. If $\bar{p} = 0$ then $p_n^{i_2} \rightarrow 0$ and $p_n^{i_2} \rightarrow 0$, but it implies that $Ek(q_n^{i_2}) \leq 0$ or $Ex(q_n^{i_3}) \leq 0$ for n big enough, a contradiction. If $\bar{q}_k = 0$ with $\bar{p} \neq 0$ then $q_{k_n}^{i_1} \rightarrow 0$ and $q_{k_n}^{i_3} \rightarrow 0$ which implies that $Ek(q_n^{i_1}) \geq 0$ or $Ek(q_n^{i_3}) \geq Ex(q_n^{i_3})$ for n big enough, another contradiction. And if $\bar{q}_x = 0$ with $\bar{p} \neq 0$ then $q_{x_n}^{i_1} \rightarrow 0$ and $q_{x_n}^{i_2} \rightarrow 0$ which implies $Ex(q_n^{i_1}) > 0$ or $Ek(q_n^{i_2}) \geq Ex(q_n^{i_2})$ for n big enough, also a contradiction. Thus, $\bar{q} \in \overset{\circ}{\Delta}$.

Now, by the continuity of $Ek(\cdot)$ and $Ex(\cdot)$ we have

$$Ek(q_n^{i_2}) > 0 \wedge Ek(q_n^{i_2}) \geq Ex(q_n^{i_2}) \implies Ek(\bar{q}) \geq Ex(\bar{q}), 0;$$

$$Ex(q_n^{i_3}) > 0 \wedge Ex(q_n^{i_3}) > Ek(q_n^{i_3}) \implies Ex(\bar{q}) \geq Ek(\bar{q}), 0; \text{ and}$$

$$Ek(q_n^{i_1}), Ex(q_n^{i_1}) \leq 0 \implies Ex(\bar{q}), Ek(\bar{q}) \leq 0.$$

Thus, $Ek(\bar{q}) = Ex(\bar{q}) = 0$. It is straightforward to show that $Ek(\bar{q}) = Ex(\bar{q}) = 0$ implies that $(\bar{q}, w(\bar{q}))$ is an equilibrium price. ■

B The Algorithm

The algorithm we propose to use here can be divided in two parts. The first one is to evaluate the stationary demand for the assets, and the second is to apply the Scarf's method in order to find the equilibrium. Let us see these issues in detail.

B.1 Evaluating the Stationary Demand

In order to evaluate K_q^d and X_q^d we need first to find the value function, V_q , such that

$$V_q(s) = \max \left\{ u(c) + \beta \mathbb{E} [V_q(s')] ; (k', x', c) \in \tilde{\Gamma}(s; q) \right\}. \quad (16)$$

By proposition (2), we know that for any function W , the sequence $\{T^n \cdot W\}_n$ converges to V_q . This result comes from the Contraction Mapping Theorem (CMT).

Applying the CMT to find the value function is usual in the literature and there is nothing new about it. On the other hand, the optimization here has two variables, k and x , which creates others problems. First, the dimensionality of the problem grows up, so an optimization in the grid, for example, become very expensive in terms of computational time and memory. Our solution to this problem is to use an interpolation method. It permits us to use few points in the grid without a big loss in terms of accuracy of the solution. The interpolation we use is described below.

B.1.1 Interpolation

The interpolation method is necessary to evaluate the value function given by the equation (16). Since we know by proposition (2) that this function is concave, a desired property of the interpolation we want to use here is that it preserves the concavity of the original function. The interpolation proposed by Cavalcanti (1997) has this property. Besides, it has the important appeal of being “*cheap*” to compute, that is, it is described by few arithmetic operations. I will give here only an outline of the algorithm proposed.

Consider a family of points $G \subset \mathbb{Z}^2$ and assume that there is a concave function $v : \mathbb{R}^2 \rightarrow \mathbb{R}$, which we only know the values $v(g)$ for $g \in G$. Our goal with the interpolation is to establish a function $\tilde{v} : \mathbb{R}^2 \rightarrow \mathbb{R}$ such that $\tilde{v}(g) = v(g) \forall g \in G$.

Let x be a point in the \mathbb{R}^2 . We say that x is feasible if there are $(g^1, g^2, g^3)' \in G$ and $(w_1, w_2, w_3)' \in \mathbb{R}^3$ such that

$$w_1 + w_2 + w_3 = 1 \quad \wedge \quad x = w_1 g^1 + w_2 g^2 + w_3 g^3.$$

So, the value $\tilde{v}(x)$ will be

$$w_1 v(g^1) + w_2 v(g^2) + w_3 v(g^3).$$

The key here is to find the “right” triple of (g^1, g^2, g^3) in order to preserve the concavity of the original function. Consider, for a positive integer $a \in \mathbb{N}$, the value $mod(a) \in \{1, 2, 3\}$ is the modulo three of a . Given a feasible x , the algorithm works as follows.

1. Define the following matrices:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

2. Let $\langle x \rangle$ denote the integer part of x and $\tilde{x} = (\tilde{x}_1, \tilde{x}_2)' = x - \langle x \rangle$. Thus:

- If $\tilde{x}_1 + \tilde{x}_2 < 1$ define

$$\begin{bmatrix} g^1 \\ g^2 \\ g^3 \end{bmatrix} = \begin{bmatrix} \langle x \rangle + (1, 0)' \\ \langle x \rangle + (0, 1)' \\ \langle x \rangle + (0, 0)' \end{bmatrix} \quad \text{and} \quad w = L^{-1} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ 1 \end{bmatrix}.$$

- Otherwise

$$\begin{bmatrix} g^1 \\ g^2 \\ g^3 \end{bmatrix} = \begin{bmatrix} \langle x \rangle + (1, 0)' \\ \langle x \rangle + (0, 1)' \\ \langle x \rangle + (1, 1)' \end{bmatrix} \quad \text{and} \quad w = U^{-1} \begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \\ 1 \end{bmatrix}.$$

3. Let $a = 1, b = 1$ and $c = 1$.

4. Let $z = g^{mod(a+1)} + g^{mod(a+2)} - g^a$.

5. If $v(g^{mod(a+1)}) + v(g^{mod(a+2)}) \geq v(g^a) + v(z)$ go to 8.

6. Store a in b . Let $m = \operatorname{argmin}\{w_i; i \neq a\}$, $g^m = z$ and pick $c \in \{1, 2, 3\}$ with $c \notin \{a, m\}$.

7. Store $w_a + w_m$ in w_a and $w_c - w_m$ in w_c . Come back to 4.

8. Store $mod(a+1)$ in a .

9. If $a = b$ or $a = c$ finish the algorithm. Otherwise, go to 4.

Note that most calculations above are very simple, which guarantees that it is in fact “cheap” to evaluate this interpolation.

A limitation of this method is that, since it is linear, we loose all pieces of information about the second derivate of the function. It could be a problem because most optimization algorithms use it, but I use the Nelder-Mead simplex method which does not use the second derivate in the optimization.

B.1.2 Optimization

Let us consider an equally spaced grid K_g for $K \equiv [0, \bar{k}]$ and X_g for $X \equiv [\underline{x}, \bar{x}]$, each grid with n_k and n_x points respectively. Thus, a function $W : S \rightarrow \mathbb{R}$ can be understood as an array of dimension $n_k \times n_x \times 2 \times 2$, where $S = K \times X \times \Theta \times \{0, 1\}$. The points outside this grid will be evaluated using the interpolation explained above.

Remember that we only need to define the function W in the feasible set, that is, in the set S_q described in proposition (1). Therefore, the value of W in $s \in S/S_q$ will be $-\infty$, which guarantees that the optimum will not be attained in any point outside S_q . Define the sets $N_k = \{1, \dots, n_k\}$ and $N_x = \{1, \dots, n_x\}$. Thus, for each point $(i, j, r, \iota) \in N_k \times N_x \times \{l, h\} \times \{0, 1\}$, we need to solve the problem

$$\max \left\{ u(c) + \beta \mathbb{E} [W(s')] ; (k', x', c) \in \tilde{\Gamma}(K_g(i), X_g(j), \theta_r, \iota; q) \right\}.$$

Note that since the function u is strictly increasing, the restriction

$$pc + q_k k' + q_x x' \leq w\theta_r + p(K_g(i) + X_g(j))$$

will always be binded. It implies that our problem is in fact to find the point $(k', x') \in K \times X$ with

$$(k', x', c) \in \tilde{\Gamma}(K_g(i), X_g(j), \theta_r, \iota; q), \text{ where } c = \frac{w\theta_r + p(K_g(i) + X_g(j)) - q_k k' - q_x x'}{p}.$$

In order to solve this program I use the simplex optimization method of Lagarias et al. (1999). This is a direct method that does not use numerical or analytic gradients. Therefore, the interpolation not preserving the second derivate is not a problem for this algorithm.

The optimization problem above has an optimal point, (k', x') , in the 2 dimensional space, a simplex in this space is a triangle. At each step of the search for (k', x') , a new point in

or near the current simplex is generated. The function value at the new point is compared with the function's values at the vertices of the simplex and, usually, one of the vertices is replaced by the new point, giving a new simplex. This step is repeated until the diameter of the simplex is less than the specified tolerance.

A limitation of this method is that it may only give local solutions. But by proposition (4) we know that the solution is unique, thus, it is not a problem to us.

B.1.3 Evaluating K_q^d and X_q^d

The last step to evaluate K_q^d and X_q^d is to build a transition function, P_q , among the states and to find the stationary distribution of this transition. The transition can be made using the Theorem 9.13 of Stokey et al. (1989). To find the stationary distribution, start with a initial distribution, Φ_0 , then make

$$\Phi_1(A) = \int_{z \in S_q} P_q(z, A) d\Phi_0 \quad \forall A \in \Sigma,$$

where Σ is the Borel σ -field of S_q . If $\Phi_1 = \Phi_0$ we have our stationary distribution, if not, store Φ_1 in Φ_0 and repeat the process above. Iterate this step until it converges. With the stationary distribution, Φ_q , the stationary demands, K_q^d and X_q^d , can be evaluated integrating the policy function of (16). That is,

$$K_q^d = \int_{S_q} g_q(s) d\Psi_q \quad \wedge \quad X_q^d = \int_{S_q} h_q(s) d\Psi_q,$$

where (g_q, h_q) comes from the solution of (16).

B.2 Scarf's Method

In this section, I describe the code for the Scarf's algorithm⁹. Consider the simplex

$$\Delta = \{(p, q_k, q_x)' \in \mathbb{R}_+; p + q_k + q_x = 1\}$$

and a list of vectors, $\{q^i\}_{i=1}^n$, such that

$$q^1 = (0, 1.1, 1.1)', q^2 = (1.2, 0, 1.2)', q^3 = (1.3, 1.3, 0)'$$

and the others q^i belong to the interior of Δ ¹⁰. For each vector q^i , give the label $l^i = l(q^i)$ as described in section 3. The idea here is to find a triple, $\{q^{n_1}, q^{n_2}, q^{n_3}\}$, such that $l(q^{n_i}) = i$ for $i = 1, 2, 3$, but first let us see some definitions.

⁹For details see Scarf (1973).

¹⁰The list $\{q^i\}_{i=4}^n$ can be randomly generated using the usual methods to simulate random variables.

Definition 3 A set $S \subset \Delta$ is called a subsimplex with the same orientation of Δ if there are a_1, a_2 and a_3 in \mathbb{R}_+ such that

$$\sum_{i=1}^3 a_i < 1 \quad \wedge \quad S = \{q \in \Delta; q_i \geq a_i, i = 1, 2, 3\}.$$

Definition 4 Given $\{q^i\}_{i=1}^n$ as described above, we say that $\{q^{j_1}, q^{j_2}, q^{j_3}\}$ is a primitive set of $\{q^i\}_{i=1}^n$ if defining $a_i = \min\{q_i^{j_1}, q_i^{j_2}, q_i^{j_3}\}$, for $i = 1, 2, 3$, and

$$S = \{q \in \Delta; q_i \geq a_i, i = 1, 2, 3\}.$$

The subsimplex S has the same orientation of Δ and there is no point in the list $\{q^i\}_{i=1}^n$ that is an interior point of it.

The Scarf's algorithm gives a method to find a primitive set, $\{q^{j_1}, q^{j_2}, q^{j_3}\}$, such that $l(q^{n_i}) = i$ for $i = 1, 2, 3$. An outline of the solution is given below.

1. Start the algorithm with a matrix $X = (q^2, q^3, q^j)$, where q^j is the vector in the family $\{q^i\}_{i=4}^n$ with the largest being the first coordinate.
2. Define $M = (l^2, l^3, l^j)$. If $l^j = 1$, go to step 5, otherwise let a be equal to 3.
3. If $M(a) = M(\text{mod}(a + 1))$ store $\text{mod}(a + 1)$ in a . Or, if $M(a) = M(\text{mod}(a + 2))$ store $\text{mod}(a + 2)$ in a .
4. Replace the column a in the matrix X by another in the list $\{q^i\}_{i=1}^n$, say $q^{j'}$, but keeping $\{X(:, 1), X(:, 2), X(:, 3)\}$ a primitive set. Store $l^{j'}$ in $M(a)$.
5. If M has three different elements, stop the algorithm. If not, go back to 3.

An important question here is when the replacement step can be carried out. We can not always guarantee that there is an unique option of replacement, but if there are two options, the algorithm may not converge. Let us see which properties are necessary so this method works.

Proposition 7 Assume that $\{q^i\}_{i=1}^n$ satisfies

$$q_l^i \neq q_l^j \text{ for all } i \neq j \text{ and } l = 1, 2, 3.$$

Then, given a primitive set $\{q^{j_1}, q^{j_2}, q^{j_3}\}$, we can always replace one of its element keeping the set a primitive set. In addition, the replacement is unique and given by the following rule.

1. Suppose you want to replace the element q^j . Define l such that

$$q_l^j = \min\{q_l^{j_1}, q_l^{j_2}, q_l^{j_3}\}.$$

2. Let q^{j_α} be the vector in $\{q^{j_1}, q^{j_2}, q^{j_3}\}$ with the second smallest coordinate being l . Let m be such that

$$q_m^{j_\alpha} = \min\{q_m^{j_1}, q_m^{j_2}, q_m^{j_3}\}.$$

3. Replace the vector q^j by the vector in $\{q^i\}_{i=1}^n$ with the largest coordinate being m , say $q^{j'}$, satisfying

- $q_s^{j'} > \min\{q_s^{j_1}, q_s^{j_2}, q_s^{j_3}\}$, for $s \neq l, m$; and
- $q_l^{j'} > q_l^{j_\alpha}$.

Therefore, if the list $\{q^i\}_{i=1}^n$ satisfies the conditions of the proposition 7, the Scarf's algorithm can be carried out without problems and will always converge to a primitive set with the desired property.

C The Code

I have faced two main problems while implementing the algorithm proposed above. First, solving the consumer problem is troublesome because the objective function is almost flat. Thus, the optimization can stop before the argmax is reached, since the increments in the objective function are very small. It is a serious problem because the argmax is important to evaluate the stationary demand, therefore, without precision on it the entire algorithm can fail. The solution to this problem is to consider only increments in the argmax itself to conclude the optimization. Note that it raises the computing time of the optimization.

The second problem found is that the cost of the second dimension is not small. That is, while the Aiyagari or Hugget model can be computed in few minutes, this code needs at least three weeks to run. I have not been able to reduce this time yet, but I believe it can be done using parallel computing methods.

The matlab code used to compute the equilibrium in this work is given below.

```

1 % ===== %
2 % === This program solves the Computable Kiyotaki-Moore Model === %
3 % ===== %
4 tic
5 disp(' ')
6 disp('Solving the Computable Kiyotaki-Moore Model.')
7 % ===== %
8 %% ===== User Definition Area %% %
9 % ===== %
10 %% Primitive's Parameters.
11 alpha = .36; % Elasticity of output to input of capital.
12 beta = 0.96; % Discount factor.
13 delta = 0.08; % Depreciation rate.
14 prob = [.925, .075; .5, .5]^6;
15 % prob(i,j)=Prob(A(t+1)=Aj|A(t)=Ai) IT NEEDS TO BE CONSISTENT WITH STATES.
16 mu = .50; % Probability of the liquidity shock.
17 aval = [1,.1]; % Labor shock(1=employed, .1=unemployed).
18 sigma = 3; % Coefficient of relative risk-aversion.
19 center = [.34087 .3294 .32974];
20 % ===== %
21 %% Grid Parameters.

```

```

22 mink      = 1e-4;                      % Minimum value of the capital grid.
23 maxk      = 20;                        % Maximum value of the capital grid.
24 nk        = 18;                         % Number of grid points for capital.
25 minx      = -4;                        % Minimum value of the asset grid.
26 maxx      = 14;                        % Maximum value of the asset grid.
27 nx        = 16;                         % Number of grid points for asset.
28 % ===== %
29 %% Optimization's Parameters.
30 tol       = 1e-5;                      % Tolerance parameter for optimization.
31 tolV     = 1e-2;                        % Tolerance parameter value function.
32 maxI     = 1e+4;                        % Maximum number of iterations.
33 options = optimset('Display','off','MaxFunEvals',maxI,...
34      'MaxIter',maxI,'TolFun',1e-8*tol,'TolX',tol);
35 % ===== %
36 %% ===== %
37 % ===== Nothing below needs to be changed by the user ===== %
38 % ===== %
39 %% Finding the ergodic probabilities of the transition probability matrix.
40 plr = (1/2)*ones(2,1); dp=1;
41 while dp>1e-15
42     plrl = (plr'*prob)'; dp=max(abs(plrl-plr)); plr=plrl;
43 end
44 empl = aval*plr; IL=inv([1,0,0;0,1,0;1,1,1]); IU=inv([1,0,1;0,1,1;1,1,1]);
45 % ===== %
46 %% Computing the utility function.
47 if sigma==1
48     u=@(c) log(c); du=@(c) 1/c;
49 else
50     u=@(c) (c^(1-sigma)-1)/(1-sigma); du=@(c) c^(-sigma);
51 end
52 % ===== %
53 %% Initial guess for the value and policy functions.
54 V=ones(nk,nx,2,2);
55 V(:,:,1,1)=log(linspace(.1,maxk,nk)'*linspace(.1,maxk,nx));
56 V(:,:,1,2)=V(:,:,1,1); V(:,:,2,1)=V(:,:,1,1); V(:,:,2,2)=V(:,:,1,1); TV=V;
57 EV1=mu*(prob(1,1)*V(:,:,1,1)+prob(1,2)*V(:,:,2,1))+...
58     (1-mu)*(prob(1,1)*V(:,:,1,2)+prob(1,2)*V(:,:,2,2));
59 EV2=mu*(prob(2,1)*V(:,:,1,1)+prob(2,2)*V(:,:,2,1))+...
60     (1-mu)*(prob(2,1)*V(:,:,1,2)+prob(2,2)*V(:,:,2,2));

```

```

61 g=ones(nk,nx,2,2); h=ones(nk,nx,2,2); save valuefunction V TV EV1 EV2 g h
62 % ===== %
63 %% Defining values for Scarf's algorithm.
64 s=0;
65 if s==1
66     load initial
67 else
68     Na=3; Ka=3e3; XX=zeros(3,3+Ka); XX(:,1:3)=[0 2 3; 1.1 0 3; 1.1 2 0];
69     XX(:,4:Ka+3)=rand(3,Ka);
70     for i=4:3+Ka
71         XX(:,i)=XX(:,i)/(ones(1,3)*XX(:,i));
72     end
73     [b j]=max(XX(1,4:3+Ka)); PS=[2 3 j]; I0=[2 3 1];
74     X=[XX(:,2) XX(:,3) XX(:,j+3)]; save initial XX X I0 PS Ka
75 end
76 counterS=0; jout1=0; M=[1 1 1]; History=zeros(Ka,4);
77 % ===== %
78 %% Implementing Scarf's algorithm.
79 while (M(1)==M(2) || M(1)==M(3) || M(2)==M(3)) && counterS<maxI
80     counterS=counterS+1;
81     if counterS==1
82         jout=3; M=[2 3 0];
83     else
84         istar=I0(jout);
85         if jout==1
86             if X(istar,2)<X(istar,3);
87                 jalpha=2; j=3;
88             else
89                 jalpha=3; j=2;
90             end
91         elseif jout==2
92             if X(istar,1)<X(istar,3);
93                 jalpha=1; j=3;
94             else
95                 jalpha=3; j=1;
96             end
97         else
98             if X(istar,1)<X(istar,2);
99                 jalpha=1; j=2;

```

```

100
101         else
102             jalpha=2; j=1;
103         end
104
105         ialpha=I0(jalpha); I0(jalpha)=istar; a=[0;0;0];
106
107         for k=1:3+Ka
108             if XX(I0(j),k)>X(I0(j),j) && XX(istar,k)>X(istar,jalpha) ...
109                 && XX(ialpha,k)≥a(ialpha)
110                 a=XX(:,k); nj=k;
111             end
112         end
113
114         X(:,jout)=a; I0(jout)=ialpha; PS(jout)=nj;
115     end
116
117     % ===== %
118     P=X(:,jout); p=P(1); qk=P(2); qx=P(3);
119     if p>(1-Δ)*qk
120         kd=min(p*((p-(1-Δ)*qk)/(alpha*qk^alpha))^(1/(alpha-1)),...
121             p*maxk/(qk*aval(2)));
122     else
123         kd=maxk/aval(2);
124     end
125
126     w=p*(qk*kd/p)^(alpha)-(p-(1-Δ)*qk)*kd;
127
128     kgrid=linspace(mink,maxk,nk)'; xgrid=linspace(minx,maxx,nx)';
129
130
131
132
133
134
135
136
137
138

```

```

139         VV1(i,j)=CI(EV1,kk(i),xx(j),kgrid,xgrid,nx,nk,IL,IU);
140         VV2(i,j)=CI(EV2,kk(i),xx(j),kgrid,xgrid,nx,nk,IL,IU);
141     end
142 end
143 for i0=1:nk
144     for j0=1:nx
145         if w*aval(2)+(p-qk)*kgrid(i0)+(p-qx)*xgrid(j0)>0
146             [g11 h11 TV11 g12 h12 TV12]=...
147                 solve(VV1,EV1,kgrid(i0),xgrid(j0),...
148                     g(i0,j0,1,1),h(i0,j0,1,1),aval(1),...
149                     i0,w,p,qk,qx,beta,u,kgrid,xgrid,kk,xx,nkk,nxx,...
150                     options,aval,nx,nk,IL,IU);
151             [g21 h21 TV21 g22 h22 TV22]=...
152                 solve(VV2,EV2,kgrid(i0),xgrid(j0),g(i0,j0,2,1),...
153                     h(i0,j0,2,1),aval(2),...
154                     i0,w,p,qk,qx,beta,u,kgrid,xgrid,kk,xx,nkk,nxx,...
155                     options,aval,nx,nk,IL,IU);
156             Tg(i0,j0,:,:)=[g11 g12; g21 g22];
157             Th(i0,j0,:,:)=[h11 h12; h21 h22];
158             TV(i0,j0,:,:)=[TV11 TV12; TV21 TV22];
159         else
160             TV11=-1e20; TV12=TV11; TV21=TV11; TV22=TV11;
161             g11=kgrid(1); g21=g11; g12=g11; g22=g11;
162             h11=0; h21=h11; h12=h11; h22=h11;
163             Tg(i0,j0,:,:)=[g11 g12; g21 g22];
164             Th(i0,j0,:,:)=[h11 h12; h21 h22];
165             TV(i0,j0,:,:)=[TV11 TV12; TV21 TV22];
166         end
167     end
168 end
169 errorV=max(max(max(max(abs(TV-V))))); V=TV; g=Tg; h=Th;
170 disp(errorV)
171 EV1=mu*(prob(1,1)*V(:,:,1,1)+prob(1,2)*V(:,:,2,1))+...
172     (1-mu)*(prob(1,1)*V(:,:,1,2)+prob(1,2)*V(:,:,2,2));
173 EV2=mu*(prob(2,1)*V(:,:,1,1)+prob(2,2)*V(:,:,2,1))+...
174     (1-mu)*(prob(2,1)*V(:,:,1,2)+prob(2,2)*V(:,:,2,2));
175 end
176 save valuefunction V TV EV1 EV2 g h
177 % ===== %

```

```

178 %% Evaluating the excess demand.
179 % Initial guess.
180 d=zeros(nk*nx,1);
181 for l=1:nk*nx
182     j=floor((l-1)/nk)+1; i=l-(j-1)*nk;
183     if w*aval(2)+(p-qk)*kgrid(i)+(p-qx)*xgrid(j)>0
184         d(l)=1;
185     end
186 end
187 d=d/(ones(1,nk*nx)*d);
188 % Transition Matrix.
189 T=zeros(nk*nx,nk*nx);
190 for l=1:nk*nx
191     for m=1:nk*nx
192         j0=floor((l-1)/nk)+1; i0=l-(j0-1)*nk;
193         j1=floor((m-1)/nk)+1; i1=m-(j1-1)*nk;
194         [w11 G11]=CIweight(EV1,g(i0,j0,1,1),h(i0,j0,1,1),...
195             kgrid,xgrid,nx,nk,IL,IU);
196         [w12 G12]=CIweight(EV1,g(i0,j0,1,2),h(i0,j0,1,2),...
197             kgrid,xgrid,nx,nk,IL,IU);
198         [w21 G21]=CIweight(EV2,g(i0,j0,2,1),h(i0,j0,2,1),...
199             kgrid,xgrid,nx,nk,IL,IU);
200         [w22 G22]=CIweight(EV2,g(i0,j0,2,2),h(i0,j0,2,2),...
201             kgrid,xgrid,nx,nk,IL,IU);
202         if w*aval(2)+(p-qk)*kgrid(i0)+(p-qx)*xgrid(j0)>0 && ...
203             w*aval(2)+(p-qk)*kgrid(i1)+(p-qx)*xgrid(j1)>0
204             T(l,m)=plr(1)*mu*(w11(1)*(G11(1,1)==i1)*(G11(2,1)==j1)+...
205                 w11(2)*(G11(1,2)==i1)*(G11(2,2)==j1)+...
206                 w11(3)*(G11(1,3)==i1)*(G11(2,3)==j1))+...
207                 plr(1)*(1-mu)*(w12(1)*(G12(1,1)==i1)*(G12(2,1)==j1)+...
208                     w12(2)*(G12(1,2)==i1)*(G12(2,2)==j1)+...
209                     w12(3)*(G12(1,3)==i1)*(G12(2,3)==j1))+...
210                     plr(2)*mu*(w21(1)*(G21(1,1)==i1)*(G21(2,1)==j1)+...
211                         w21(2)*(G21(1,2)==i1)*(G21(2,2)==j1)+...
212                         w21(3)*(G21(1,3)==i1)*(G21(2,3)==j1))+...
213                         plr(2)*(1-mu)*(w22(1)*(G22(1,1)==i1)*(G22(2,1)==j1)+...
214                             w22(2)*(G22(1,2)==i1)*(G22(2,2)==j1)+...
215                             w22(3)*(G22(1,3)==i1)*(G22(2,3)==j1));
216     end

```

```

217      end
218  end
219 % Evaluating the stationary distribution.
220 error=1; counter=0;
221 while error>1e-12 && counter<maxI
222     counter=counter+1; d1=(d'*T)';
223     error=max(abs(d1-d)); d=d1; %disp(error)
224 end
225 % Evaluating the demand.
226 kk=zeros(nk*nx,1); xx=zeros(nk*nx,1);
227 for l=1:nk*nx
228     j0=floor((l-1)/nk)+1; i0=l-(j0-1)*nk;
229     [w11 G11]=CIweight(EV1,g(i0,j0,1,1),h(i0,j0,1,1),...
230         kgrid,xgrid,nx,nk,IL,IU);
231     [w12 G12]=CIweight(EV1,g(i0,j0,1,2),h(i0,j0,1,2),...
232         kgrid,xgrid,nx,nk,IL,IU);
233     [w21 G21]=CIweight(EV2,g(i0,j0,2,1),h(i0,j0,2,1),...
234         kgrid,xgrid,nx,nk,IL,IU);
235     [w22 G22]=CIweight(EV2,g(i0,j0,2,2),h(i0,j0,2,2),...
236         kgrid,xgrid,nx,nk,IL,IU);
237 if w*aval(2)+(p-qk)*kgrid(i0)+(p-qx)*xgrid(j0)>0
238     kk(l)=mu*plr(1)*(w11(1)*kgrid(G11(1,1))+...
239         w11(2)*kgrid(G11(1,2))+w11(3)*kgrid(G11(1,3)))+...
240         mu*plr(2)*(w21(1)*kgrid(G21(1,1))+...
241             w21(2)*kgrid(G21(1,2))+w21(3)*kgrid(G21(1,3)))+...
242             (1-mu)*plr(1)*(w12(1)*kgrid(G12(1,1))+...
243                 w12(2)*kgrid(G12(1,2))+w12(3)*kgrid(G12(1,3)))+...
244                 (1-mu)*plr(2)*(w22(1)*kgrid(G22(1,1))+...
245                     w22(2)*kgrid(G22(1,2))+w22(3)*kgrid(G22(1,3)));
246     xx(l)=mu*plr(1)*(w11(1)*xgrid(G11(2,1))+...
247         w11(2)*xgrid(G11(2,2))+w11(3)*xgrid(G11(2,3)))+...
248         mu*plr(2)*(w21(1)*xgrid(G21(2,1))+...
249             w21(2)*xgrid(G21(2,2))+w21(3)*xgrid(G21(2,3)))+...
250             (1-mu)*plr(1)*(w12(1)*xgrid(G12(2,1))+...
251                 w12(2)*xgrid(G12(2,2))+w12(3)*xgrid(G12(2,3)))+...
252                 (1-mu)*plr(2)*(w22(1)*xgrid(G22(2,1))+...
253                     w22(2)*xgrid(G22(2,2))+w22(3)*xgrid(G22(2,3)));
254 end
255 end

```

```

256 EK=empl*d-d'*kk; EX=-d'*xx; History(counterS,:)=[qk qx EK EX];
257 disp([' [EK EX]= ', num2str([EK EX])))
258 if EK>0 && EX>0
259     M(jout)=1;
260 elseif EK<EX
261     M(jout)=2;
262 else
263     M(jout)=3;
264 end
265 if jout==1
266     if M(jout)==M(2)
267         jout=2;
268     elseif M(jout)==M(3)
269         jout=3;
270     end
271 elseif jout==2
272     if M(jout)==M(1)
273         jout=1;
274     elseif M(jout)==M(3)
275         jout=3;
276     end
277 else
278     if M(jout)==M(1)
279         jout=1;
280     elseif M(jout)==M(2)
281         jout=2;
282     end
283 end
284 end
285 % ===== %
286 rk=(p-qk)/qk; rx=(p-qx)/qx; save final p qk qx rk rx EK EX History
287 %% ===== %
288 % ====== Print out results ===== %
289 % ===== %
290 disp('COMPUTATION OF STATIONARY DISTRIBUTION IS COMPLETED.')
291 disp('ELAPSED TIME:')
292 toc
293 disp('RESULTS')
294 disp(['RETURN OF ASSET K : ', num2str(rk)])

```

```

295 disp(['EXCESS DEMAND FOR K          : ', num2str( EK )])
296 disp(['RETURN OF ASSET X           : ', num2str( rx )])
297 disp(['EXCESS DEMAND FOR X         : ', num2str( EX )])
298 % ===== %

```

```

1 function [Vkx]=CI(V,k,x,kgrid,xgrid,nx,nk,IL,IU)
2
3 % This function interpolates the function V, in the grid (kgrid,xgrid), in
4 % order to find the value of V in the point (k,x), Vkx.
5
6 if k<kgrid(1) || k>kgrid(nk) || x<xgrid(1) || x>xgrid(nx)
7     Vkx=-1e100;
8 else
9     zero=min(xgrid(2)-xgrid(1),kgrid(2)-kgrid(1))/5;
10    if abs(kgrid(nk)-k)<=zero
11        k1=k-zero;
12    elseif abs(k-kgrid(1))<=zero
13        k1=k+zero;
14    else
15        k1=k;
16    end
17    if abs(xgrid(nx)-x)<=zero
18        x1=x-zero;
19    elseif abs(xgrid(1)-x)<=zero
20        x1=x+zero;
21    else
22        x1=x;
23    end
24    i0=find(max(kgrid-k1,0),1)-1;
25    j0=find(max(xgrid-x1,0),1)-1;
26    ik=(k-kgrid(i0))/(kgrid(i0+1)-kgrid(i0));
27    jx=(x-xgrid(j0))/(xgrid(j0+1)-xgrid(j0));
28    if ik+jx<1
29        G=[i0+1 i0 i0;j0 j0+1 j0];
30        w=IL*[ik; jx; 1];
31        VG=[V(i0+1,j0); V(i0,j0+1); V(i0,j0)];
32    else
33        G=[i0+1 i0 i0+1;j0 j0+1 j0+1];

```

```

34         w=IU*[ik; jx; 1];
35         VG=[V(i0+1,j0); V(i0,j0+1); V(i0+1,j0+1)];
36     end
37
38     flag1=0; a=1; b=1; c=1;
39     while flag1==0
40         z=G(:,modulo3(a+1))+G(:,modulo3(a+2))-G(:,a);
41         if 1≤z(1) && z(1)≤nk && 1≤z(2) && z(2)≤nx
42             Vz=V(z(1),z(2));
43             if VG(modulo3(a+1))+VG(modulo3(a+2))<Vz+VG(a)
44                 b=a;
45                 if w(modulo3(a+1))≤w(modulo3(a+2))
46                     m=modulo3(a+1);
47                     G(:,m)=z; VG(m)=Vz; c=modulo3(a+2);
48                     w(a)=w(a)+w(m); w(c)=w(c)-w(m);
49                 else
50                     m=modulo3(a+2);
51                     G(:,m)=z; VG(m)=Vz; c=modulo3(a+1);
52                     w(a)=w(a)+w(m); w(c)=w(c)-w(m);
53                 end
54             else
55                 a=modulo3(a+1);
56                 if a==b || a==c
57                     flag1=1;
58                 end
59             end
60         else
61             a=modulo3(a+1);
62             if a==b || a==c
63                 flag1=1;
64             end
65         end
66     end
67     Vkx=w'*VG;
68 end
69 end

```

```

1  function [w G]=CIweight(V,k,x,kgrid,xgrid,nx,nk,IL,IU)
2
3  % This function interpolates the function V, in the grid (kgrid,xgrid), in
4  % order to find the points of the grid, G, and the weight of each point, w,
5  % in the interpolation.
6
7  zero=min(xgrid(2)-xgrid(1),kgrid(2)-kgrid(1))/3;
8  if abs(kgrid(nk)-k)≤zero
9      k1=k-zero;
10 elseif abs(k-kgrid(1))≤zero
11     k1=k+zero;
12 else
13     k1=k;
14 end
15 if abs(xgrid(nx)-x)≤zero
16     x1=x-zero;
17 elseif abs(xgrid(1)-x)≤zero
18     x1=x+zero;
19 else
20     x1=x;
21 end
22 i0=find(max(kgrid-k1,0),1)-1;
23 j0=find(max(xgrid-x1,0),1)-1;
24 ik=(k-kgrid(i0))/(kgrid(i0+1)-kgrid(i0));
25 jx=(x-xgrid(j0))/(xgrid(j0+1)-xgrid(j0));
26 if ik+jx<1
27     G=[i0+1 i0 i0;j0 j0+1 j0];
28     w=IL*[ik; jx; 1];
29     VG=[V(i0+1,j0); V(i0,j0+1); V(i0,j0)];
30 else
31     G=[i0+1 i0 i0+1;j0 j0+1 j0+1];
32     w=IU*[ik; jx; 1];
33     VG=[V(i0+1,j0); V(i0,j0+1); V(i0+1,j0+1)];
34 end
35
36 flag1=0; a=1; b=1; c=1;
37 while flag1==0
38     z=G(:,modulo3(a+1))+G(:,modulo3(a+2))-G(:,a);
39     if 1≤z(1) && z(1)≤nk && 1≤z(2) && z(2)≤nx

```

```

40      Vz=V(z(1),z(2));
41      if VG(modulo3(a+1))+VG(modulo3(a+2))<Vz+VG(a)
42          b=a;
43          if w(modulo3(a+1))≤w(modulo3(a+2))
44              m=modulo3(a+1);
45              G(:,m)=z; VG(m)=Vz; c=modulo3(a+2);
46              w(a)=w(a)+w(m); w(c)=w(c)-w(m);
47          else
48              m=modulo3(a+2);
49              G(:,m)=z; VG(m)=Vz; c=modulo3(a+1);
50              w(a)=w(a)+w(m); w(c)=w(c)-w(m);
51          end
52      else
53          a=modulo3(a+1);
54          if a==b || a==c
55              flag1=1;
56          end
57      end
58  else
59      a=modulo3(a+1);
60      if a==b || a==c
61          flag1=1;
62      end
63  end
64 end
65 end

```

```

1 function [gf hf TVf gnf hnf TVnf]=solve(VV,V,k,x,gk,hx,l,i0nf,w,p,qk,qx, ...
2 beta,u,kgrid,xgrid,kk,xx,nkk,nxx,options,aval,nx,nk,IL,IU)
3
4 % This function solves the consumer problem in the state (k,x) with the
5 % productivity shock l when he receives the negative and positive
6 % liquidity shock.
7
8 RR=-1e20*ones(size(VV));
9 for i=1:nkk
10    for j=1:nxx
11        if w*l+p*(k+x)-qk*kk(i)-qx*xx(j)>0 && ...

```

```

12           w*aval(2)+(p-qk)*kk(i)+(p-qx)*xx(j)>0
13           RR(i,j)=u((w*l+p*(k+x)-qk*kk(i)-qx*xx(j))/p);
14       end
15   end
16 end
17 TT=RR+beta*VV; [T b]=max(TT,[],2); [TVf, fi]=max(T,[],1);
18 gf=kk(fi); hf=xx(b(fi)); gnf=k; hnf=xx(b(i0nf));
19
20 TVnf0=ones(1,1); hnf0=TVnf0; guess=hnf;
21 F=@(y) Func(V,k,x,gnf,y,l,u,w,p,qk,qx,beta,aval,kgrid,xgrid,nx,nk,IL,IU);
22
23 for t=1:1
24     y=fminsearch(F,guess(t),options);
25     hnf0(t)=y; cnf=(w*l+p*(k+x)-qk*gnf-qx*hnf0(t))/p;
26     Vnf=CI(V,gnf,hnf0(t),kgrid,xgrid,nx,nk,IL,IU);
27     TVnf0(t)=u(cnf)+beta*Vnf;
28 end
29 [TVnf jnf]=max(TVnf0); hnf=hnf0(jnf);
30
31 TVf0=ones(3,1); gf0=TVf0; hf0=gf0; guess=[gf hf; gnf hnf; gk hx];
32 F=@(y) ...
33     Func(V,k,x,y(1),y(2),l,u,w,p,qk,qx,beta,aval,kgrid,xgrid,nx,nk,IL,IU);
34
35 for t=1:3
36     y=fminsearch(F,guess(t,:),options);
37     gf0(t)=y(1); hf0(t)=y(2); cf=(w*l+p*(k+x)-qk*gf0(t)-qx*hf0(t))/p;
38     Vf=CI(V,gf0(t),hf0(t),kgrid,xgrid,nx,nk,IL,IU);
39     TVf0(t)=u(cf)+beta*Vf;
40 end
41 [TVf jf]=max(TVf0); gf=gf0(jf); hf=hf0(jf);
42 end

```

```

1 function Fv=Func(V,k,x,k0,x0,l,u,w,p,qk,qx,beta,aval,kgrid,xgrid, ...
2 nx,nk,IL,IU)
3
4 % This function gives the value of the objective function in the
5 % consumer problem.
6

```

```

7  if w*l+p*(k+x)-qk*k0-qx*x0>0 && w*aval(2)+(p-qk)*k0+(p-qx)*x0>0
8      Fv=-u((w*l+p*(k+x)-qk*k0-qx*x0)/p) - ...
9          beta*CI(V,k0,x0,kgrid,xgrid,nx,nk,IL,IU);
10 else
11     Fv=1e20;
12 end
13 end

```

```

1 function [mod]=modulo3(a)
2 if a==4
3     mod=1;
4 elseif a==5
5     mod=2;
6 elseif a==6
7     mod=3;
8 else
9     mod=a;
10 end
11 end

```

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)

[Baixar livros de Literatura de Cordel](#)

[Baixar livros de Literatura Infantil](#)

[Baixar livros de Matemática](#)

[Baixar livros de Medicina](#)

[Baixar livros de Medicina Veterinária](#)

[Baixar livros de Meio Ambiente](#)

[Baixar livros de Meteorologia](#)

[Baixar Monografias e TCC](#)

[Baixar livros Multidisciplinar](#)

[Baixar livros de Música](#)

[Baixar livros de Psicologia](#)

[Baixar livros de Química](#)

[Baixar livros de Saúde Coletiva](#)

[Baixar livros de Serviço Social](#)

[Baixar livros de Sociologia](#)

[Baixar livros de Teologia](#)

[Baixar livros de Trabalho](#)

[Baixar livros de Turismo](#)