Two Assets and Kinked Adjustment Costs

Greg Kaplan, Benjamin Moll and Gianluca Violante

This note is based on Kaplan, Moll and Violante (2016). It describes a consumption-saving problem with two assets and kinked adjustment costs, and the algorithm for solving this problem numerically. The code is `two_asset_kinked.m` with subroutines `two_asset_kinked_cost.m` and `two_asset_kinked_cost.m`.

Note that this is just an example and *not* the code used in Kaplan, Moll and Violante (2016). For that code, see here `http://www.princeton.edu/~moll/HANK_replication.zip` (it's in Fortran).

1 Model Setup

Households solve the following problem:

$$\begin{aligned}\max_{\{c_t,d_t\}_{t\geq 0}} \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_t) dt \quad \text{s.t.}\\ \dot{b}_t &= (1-\xi) w z_t + r^b(b_t) b_t - d_t - \chi(d_t, a_t) - c_t \\ \dot{a}_t &= r^a a_t + \xi w z_t + d_t \\ z_t &= \text{Poisson process with intensities } \lambda(z, z') \\ a_t &\geq 0, \quad b_t \geq \underline{b}\end{aligned}$$

Here a_t denotes illiquid assets, b_t liquid assets, c_t consumption, z_t idiosyncratic productivity, d_t deposits, and χ the transaction cost function. The wage is denoted by w, the return on illiquid assets is r^a and the return on liquid assets is r^b. Finally, we assume that a fraction ξ of income is automatically deposited in the illiquid account (e.g. capturing automatic payroll deductions into a 401(k) account).

We assume the following functional form for the adjustment cost function:

$$\chi(d, a) = \chi_0 |d| + \frac{\chi_1}{2} \left(\frac{d}{a}\right)^2 a \tag{1}$$

with $\chi_0, \chi_1 > 0$. The adjustment cost function looks as in Figure 1. Note in particular the kink at $d = 0$. The two components of the adjustment cost functions have different

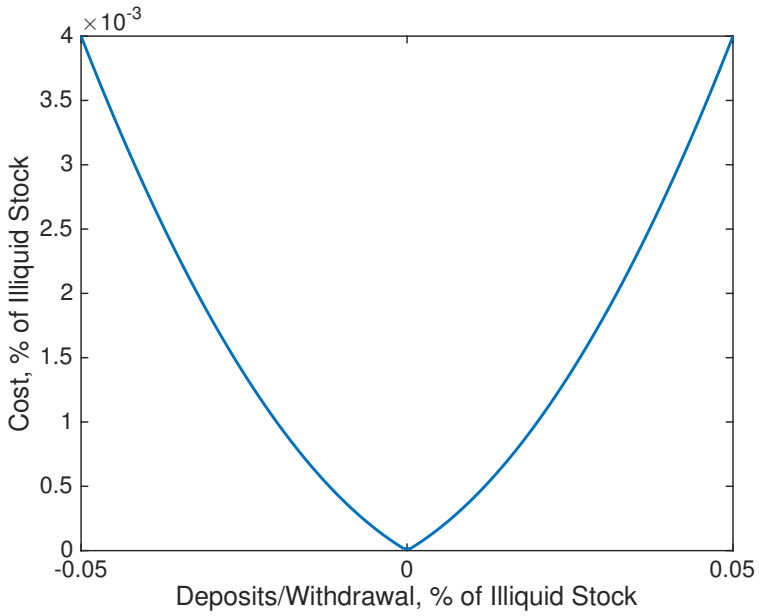

Figure 1: Adjustment Cost Function

implications for household behavior: the kinked cost component implies inaction, and the convex component implies finite deposit rates.

2 Recursive Formulation

The HJB equation is

$$\begin{aligned}\rho V(a,b,z) = \max_c \; & u(c) + V_b(a,b,z)((1-\xi)wz + r^b(b)b - d - \chi(d,a) - c) \\ & + V_a(a,b,z)(r^a a + \xi wz + d) \\ & + \sum_{z'} \lambda(z,z')(V(a,b,z') - V(a,b,z))\end{aligned} \tag{2}$$

The first-order conditions are

$$\begin{aligned} u'(c) &= V_b(a,b,z) \\ V_b(a,b,z)(1+\chi_d(d,a)) &= V_a(a,b,z)\end{aligned}$$

We have

$$\chi_d(d,a) = \begin{cases} \chi_0 + \chi_1 d/a, & d > 0 \\ -\chi_0 + \chi_1 d/a, & d < 0 \end{cases}$$

Given that $\chi_d(d,a) = \chi_0 + \chi_1 d/a$ if $d > 0$ and $\chi_d(d,a) = -\chi_0 + \chi_1 d/a$ if $d < 0$, conditional

on paying the fixed cost, optimal deposits are given by

$$d = \left(\frac{V_a}{V_b} - 1 + \chi_0\right)^- \frac{a}{\chi_1} + \left(\frac{V_a}{V_b} - 1 - \chi_0\right)^+ \frac{a}{\chi_1} \tag{3}$$

where we use the short-hand notation $x^+ = \max\{x, 0\}$ and $x^- = \min\{x, 0\}$ for any scalar x. Note in particular that $d = 0$ if $-\chi_0 < \frac{V_a}{V_b} - 1 < \chi_0$. Intuitively, households deposit into the illiquid account when the marginal value of illiquid wealth is large relative to that of liquid wealth and withdraw in the opposite case. Households do not deposit or withdraw when the marginal values of the two assets are relatively similar.

Assumption 1: $r^a < 1/\chi_1$. If this assumption were violated, households would accumulate illiquid wealth to infinity. To see this, consider (3). As illiquid wealth $a \to \infty$, its marginal value $V_a(a, b, z) \to 0$. Therefore, for large a, deposits are negative and behave like $d \sim -a/\chi_1$. Similarly, the drift of illiquid wealth is given by

$$\dot{a} = w\xi z + r^a a + d \sim r^a a + d \sim (r^a - 1/\chi_1)a$$

where the first $\approx$ uses the fact that as $a \to \infty$. If Assumption 1 were violated i.e. $r^a - 1/\chi_1 > 0$ this drift would be necessarily positive and households would accumulate to infinity. Therefore Assumption 1 is necessary to rule out this behavior.

3 Numerical Solution

See `two_asset_kinked.m` with subroutines `two_asset_kinked_cost.m` and `two_asset_kinked_cost.m`. To solve the HJB equation (2) we use an upwind finite difference method along the lines of Achdou et al. (2014). We use an implicit upwind finite difference method. For simplicity we here set $\xi = 0$, i.e. all labor income is paid into the liquid asset. Our upwind method splits the drift of b, $wz_k + r^b(b)b - d - \chi(d, a) - c$ into two parts $s^c = wz_k + r^b(b)b - c$ and $s^d = -d - \chi(d, a)$ and upwinds these separately.

We denote grid points by $b_i, i = 1, ..., I, a_j, j = 1, ..., J$ and $z_k, k = 1, ..., K$ and

$$V_{i,j,k} = V(a_j, b_i, z_k)$$

We use non-equispaced grids and denote $\Delta b_i^+ = b_{i+1} - b_i$ and $\Delta b_i^- = b_i - b_{i-1}$ and so on. We approximate derivatives for a and b with either a forward or a backward-difference

approximation

$$V_b(a_j, b_i, z_k) \approx V^F_{b,i,j,k} = \frac{V_{i+1,j,k} - V_{i,j,k}}{\Delta b_i^+} \tag{4}$$

$$V_b(a_j, b_i, z_k) \approx V^B_{b,i,j,k} = \frac{V_{i,j,k} - V_{i-1,j,k}}{\Delta b_i^-} \tag{5}$$

and similarly for V_a. The discretized version of (2) is

$$\begin{aligned}\frac{V^{n+1}_{i,j,k} - V^n_{i,j,k}}{\Delta} + \rho V^{n+1}_{i,j,k} = u\left(c^n_{i,j,k}\right) + V^{n+1}_{b,i,j,k} s^{b,n}_{i,j,k} + V^{n+1}_{a,i,j,k}(r^a a_j + d^n_{i,j,k}) \\ + \sum_{k' \neq k}^{K} \lambda_{k,k'}(V^{n+1}_{i,j,k'} - V^{n+1}_{i,j,k})\end{aligned} \tag{6}$$

$$s^n_{i,j,k} = w z_k + r^b(b_i) b_i - d^n_{i,j,k} - \chi(d^n_{i,j,k}, a_j) - c^n_{i,j,k} \tag{7}$$

$$u'(c^n_{i,j,k}) = V^n_{b,i,j,k} \tag{8}$$

$$V^n_{b,i,j,k}(1 + \chi_d(d^n_{i,j,k}, a_j)) = V^n_{a,i,j,k} \tag{9}$$

where $V^{n+1}_{b,i,j,k}$ is either the forward difference approximation in (4) or the backward difference approximation in (5), and similarly for $V^{n+1}_{a,i,j,k}$. Given a guess for the value function $V^n_{i,j,k}$ and a choice which finite difference approximation to use, (8) to (9) implicitly define optimal choices $c^n_{i,j,k}$ and $d^n_{i,j,k}$.

We use an upwind method to choose when to use backward and forward difference approximations. In the a-dimension, we use a forward difference approximation whenever the drift of a is positive and a backward approximation otherwise. In the b-dimension, we additionally make use of the fact that the HJB equation features a lot of linearity. As mentioned, we split the drift of b, $s_{i,j,k}$ into two terms and upwind each term separately. To this end, define $c^B_{i,j,k}$ to be optimal consumption calculated using the backward difference approximation with respect to b, $V^B_{b,i,j,k}$, and $c^F_{i,j,k}$ to be optimal consumption calculated using the forward difference approximation, $V^F_{b,i,j,k}$. Similarly define

$$s^{c,B}_{i,j,k} = w z_k + r^b(b_i) b_i - c^{B,n}_{i,j,k} \tag{10}$$

$$s^{c,F}_{i,j,k} = w z_k + r^b(b_i) b_i - c^{F,n}_{i,j,k} \tag{11}$$

We then approximate

$$V(a_j, b_i, z_k) s^c(a_i, b_j, z_k) \approx V^{B,n+1}_{b,i,j,k}(s^{c,B}_{i,j,k})^- + V^{F,n+1}_{b,i,j,k}(s^{c,F}_{i,j,k})^+$$

Here and elsewhere we use the short-hand notation $x^+ = \max\{x, 0\}$ and $x^- = \min\{x, 0\}$ for

any scalar x.

Define $d^{BB}_{i,j,k}, d^{BF}_{i,j,k}, d^{FB}_{i,j,k}, d^{FF}_{i,j,k}$ in an analogous fashion. For example $d^{BF}_{i,j,k}$ are optimal deposits calculated using the backward difference approximation with respect to b, $V^{B}_{b,i,j,k}$ and the forward difference approximation with respect to a, $V^{F}_{a,i,j,k}$, i.e. $d^{BF}_{i,j,k}$ satisfies

$$V^{B}_{b,i,j,k}(1 + \chi_d(d^{BF}_{i,j,k}, a_j)) = V^{F}_{a,i,j,k}$$

Further define

$$d^{B}_{i,j,k} = (d^{BF}_{i,j,k})^{+} + (d^{BB}_{i,j,k})^{-} \tag{12}$$

$$d^{F}_{i,j,k} = (d^{FF}_{i,j,k})^{+} + (d^{FB}_{i,j,k})^{-} \tag{13}$$

$$s^{d,B}_{i,j,k} = -d^{B,n}_{i,j,k} - \chi(d^{B,n}_{i,j,k}, a_j) \tag{14}$$

$$s^{d,F}_{i,j,k} = -d^{F,n}_{i,j,k} - \chi(d^{F,n}_{i,j,k}, a_j) \tag{15}$$

$$d_{i,j,k} = d^{B}_{i,j,k}\mathbf{1}_{\{s^{d,B}_{i,j,k}<0\}} + d^{F}_{i,j,k}\mathbf{1}_{\{s^{d,F}_{i,j,k}>0\}} \tag{16}$$

Then our upwind finite difference approximation is given by

$$\begin{aligned}
\frac{V^{n+1}_{i,j,k} - V^{n}_{i,j,k}}{\Delta} + \rho V^{n+1}_{i,j,k} = u\left(c^{n}_{i,j,k}\right) & \\
+ V^{B,n+1}_{b,i,j,k}(s^{c,B}_{i,j,k})^{-} + V^{F,n+1}_{b,i,j,k}(s^{c,F}_{i,j,k})^{+} & \\
+ V^{B,n+1}_{b,i,j,k}(s^{d,B}_{i,j,k})^{-} + V^{F,n+1}_{b,i,j,k}(s^{d,F}_{i,j,k})^{+} & \\
+ V^{B,n+1}_{a,i,j,k} d^{-}_{i,j,k} + V^{B,n+1}_{a,i,j,k}(d^{+}_{i,j,k} + r^{a} a_j) & \\
+ \sum_{k' \neq k}^{K} \lambda_{k,k'}(V^{n+1}_{i,j,k'} - V^{n+1}_{i,j,k}) &
\end{aligned} \tag{17}$$

Importantly, the scheme satisfies the Barles-Souganidis monotonicity condition. See Achdou et al. (2017) for a definition.

To update V^{n+1} given V^{n} requires solving a system of linear equations. As in the algorithm for the one-asset case described in Achdou et al. (2017), (17) can be written in matrix notation as

$$\frac{1}{\Delta}(V^{n+1} - V^{n}) + \rho V^{n+1} = u^{n} + (\mathbf{A}^{n} + \mathbf{\Lambda})V^{n+1} \tag{18}$$

Here V^{n}, V^{n+1} and u^{n} are vectors of length $I \times J \times K$ and $\mathbf{A}^{n}$ and $\mathbf{\Lambda}$ are matrices of size $(I \times J \times K) \times (I \times J \times K)$. $\mathbf{A}^{n}$ has a similar structure as in Achdou et al, and $\mathbf{\Lambda}$ summarizes the stochastic process for income. When the number of income grid points K is not too large (e.g. with a two-state Poisson process $K = 2$), (18) can be solved very efficiently using Matlab's sparse matrix routines.

4 Handling Many Income Grid Points

The algorithm described above works well with a small number of income grid points K. However, it runs into computational difficulties when there are a large number of income grid points (say $K = 30$). This is because with large K, the matrix $\mathbf{A}^n + \mathbf{\Lambda}$ starts has a large "bandwidth" and so sparse matrix routines become slow. The case with a large number of income grid points K can be handled as follows: instead of using a finite difference scheme that is implicit in all of (a, b, z), use a scheme that is *implicit* in the a- and b-dimensions and *explicit* in the z-dimension. To this end write (17) as

$$\begin{aligned}\frac{V_{i,j,k}^{n+1} - V_{i,j,k}^{n}}{\Delta} + \rho V_{i,j,k}^{n+1} = u\left(c_{i,j,k}^{n}\right) \\ + V_{b,i,j,k}^{B,n+1}(s_{i,j,k}^{c,B})^{-} + V_{b,i,j,k}^{F,n+1}(s_{i,j,k}^{c,F})^{+} \\ + V_{b,i,j,k}^{B,n+1}(s_{i,j,k}^{d,B})^{-} + V_{b,i,j,k}^{F,n+1}(s_{i,j,k}^{d,F})^{+} \\ + V_{a,i,j,k}^{B,n+1} d_{i,j,k}^{-} + V_{a,i,j,k}^{B,n+1}(d_{i,j,k}^{+} + r^{a} a_{j}) \\ + \sum_{k' \neq k} \lambda_{k,k'}(V_{i,j,k'}^{n} - V_{i,j,k}^{n})\end{aligned} \tag{19}$$

Note in particular the $n+1$ superscripts on the parts of the HJB equation capturing movements in a and b (the scheme is implicit in the a- and b-dimensions) and n superscripts on those parts capturing movements in y (the scheme is explicit in the y-dimension). In matrix notation

$$\frac{1}{\Delta}(V^{n+1} - V^{n}) + \rho V^{n+1} = u^{n} + \mathbf{A}^{n} V^{n+1} + \mathbf{\Lambda} V^{n} \tag{20}$$

Given that we use an explicit scheme in the z-dimension and hence V^n in $\mathbf{\Lambda}V^n$ has a n-subscript, the problem (20) can be broken up into K smaller problems and we can take advantage of parallelization to solve it. In particular, (18) is equivalent to

$$\frac{1}{\Delta}(V_{k}^{n+1} - V_{k}^{n}) + \rho V_{k}^{n+1} = u_{k}^{n} + \mathbf{A}_{k}^{n} V_{k}^{n+1} + \sum_{k' \neq k} \lambda_{k,k'}(V_{k'}^{n} - V_{k}^{n}), \quad k = 1, ..., K \tag{21}$$

where V_k^n, V_k^{n+1} and u_k^n are K vectors of length $I \times J$ and $\mathbf{A}_k^n$ are K matrices of size $(I \times J) \times (I \times J)$. Given that the scheme is explicit in the z-dimension, the updating/time steps Δ cannot be "too large" (CFL condition). Finally, we note that the "implicit-explicit scheme" in (21) is also very amenable to parallelization (though in practice, the code is fast enough without implementing this).

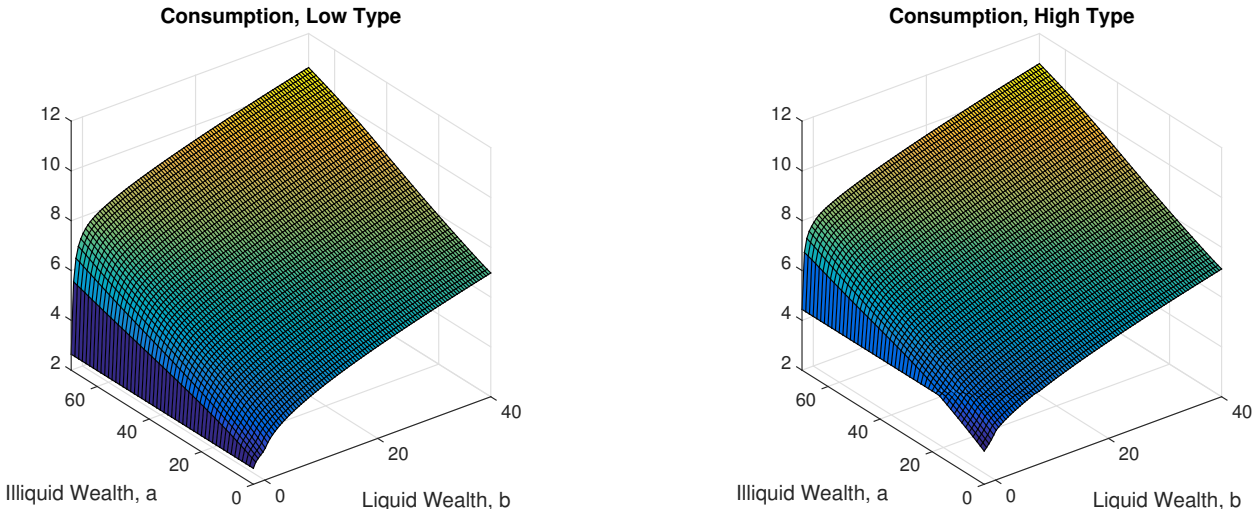

Figure 2: Consumption

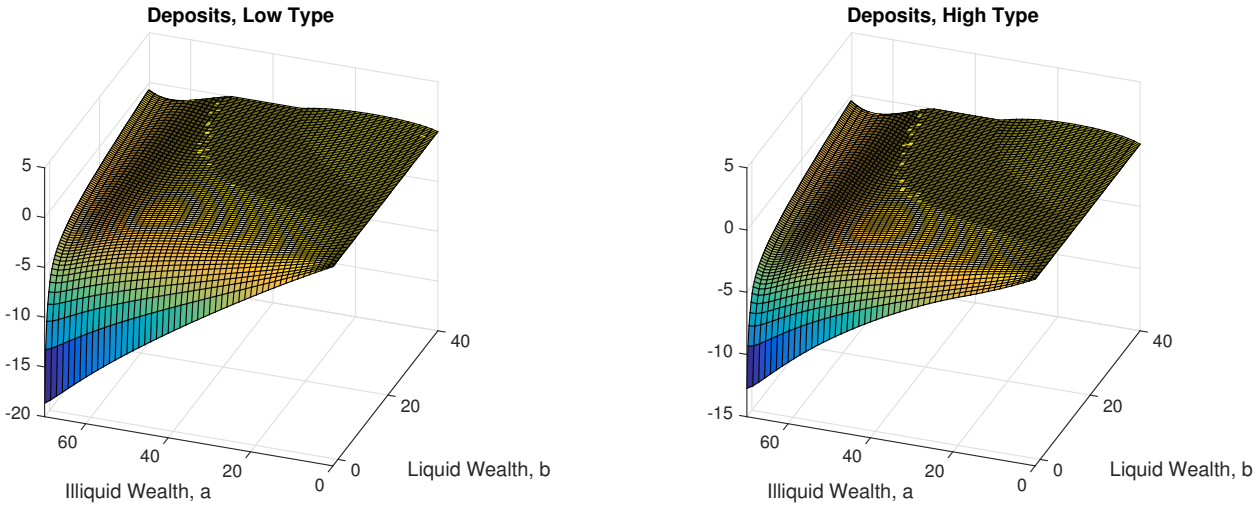

Figure 3: Deposits

5 Results

References

Achdou, Yves, Jiequn Han, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll. 2017. "Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach." Princeton University Working Papers.

Kaplan, Greg, Benjamin Moll, and Giovanni L. Violante. 2016. "Monetary Policy According to HANK." National Bureau of Economic Research Working Paper 21897.

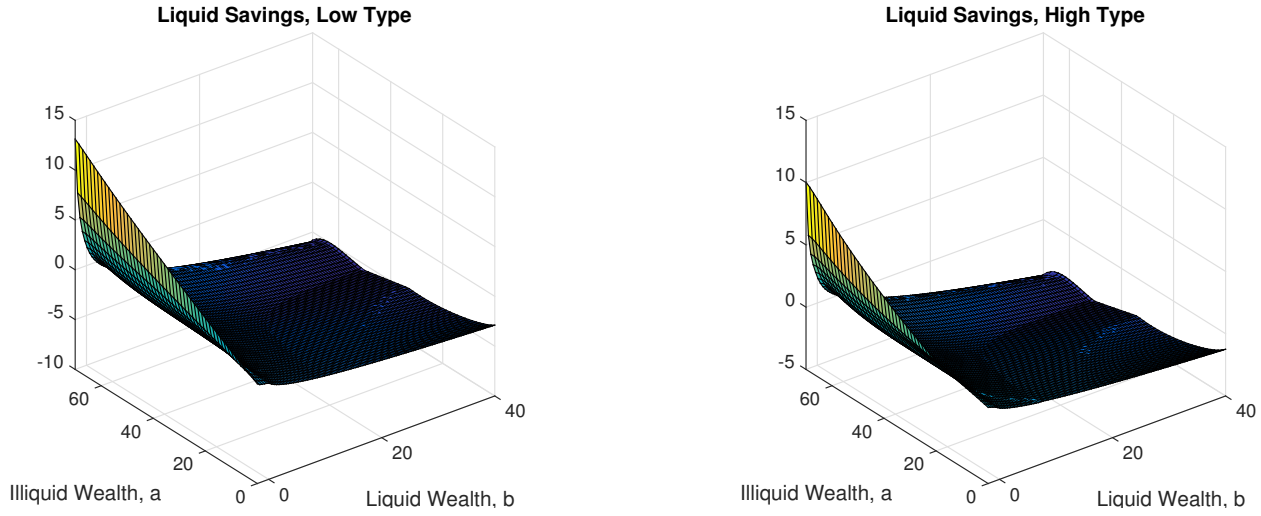

Figure 4: Liquid Savings

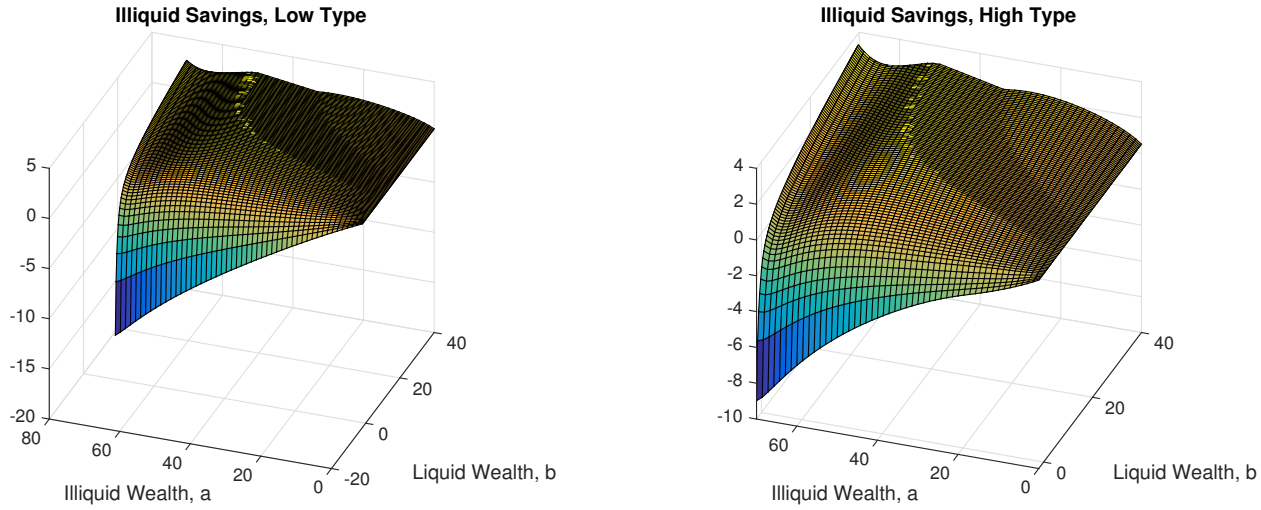

Figure 5: Illiquid Savings

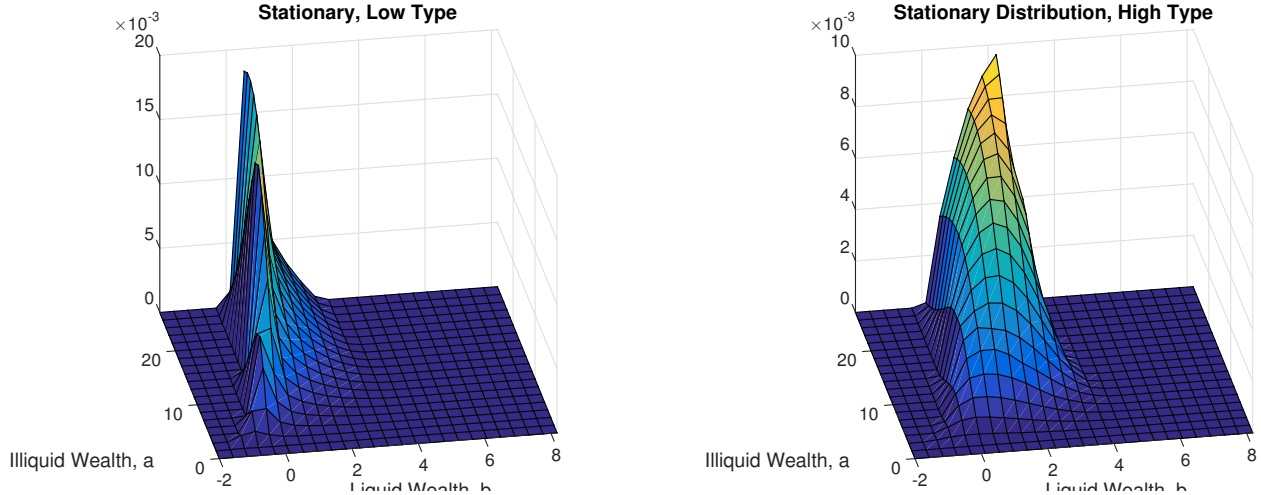

Figure 6: Stationary Distributions