Exploration of continuous time models using Julia
Computational Economics Term paper

James Graham

June 2, 2016

1 Introduction

This paper documents and demonstrates a solution method for a simple continuous time, heterogeneous agents model economy. In doing so, I make significant use of the lecture notes on Benjamin Moll's website Moll (2015*a*). Most of the code I use is modified and translated to Julia from the Matlab codes found at the HACT project website, Moll (2015*b*).

In this paper I work with the benchmark Huggett (1993) economy. This is a simple general equilibrium model in which agents solve a consumption-savings problem subject to idiosyncratic income fluctuations and a borrowing constraint.

I solve two versions of the model. First, I take a model in which income can only take on two states, and the ability to switch between the states occurs with a Poisson arrival rate. Second, I take a model in which there is a continuum of income states, and the evolution of income is governed by a mean-reverting diffusion process.

I solve the two versions of the model using Julia. Although the solution may use a large number of grid points, the model itself is fairly simple and so does not take long to solve. For this model, then, Julia does not exhibit large absolute speed gains over, say, Matlab. However, Julia is still faster, and coding the model in it would be useful if using models with larger state spaces.

One major drawback of using Julia is the difficulty I had in producing plots. For various reasons, several of the main plotting packages did not work on my computer, and I had to resort to using the Gadfly package. Gadfly is a fairly clean plotting tool, however it is lacking many features that we might want to work with (e.g. different line styles, Latex integration, 3D plots, legends, easy exporting of figures). I've done the best that I can with Gadfly, but admit that I am somewhat disappointed with the results relative to Matlab. For this reason, I have chosen to include Python's matplotlib plots in this PDF, while using Gadfly plots in the Jupyter notebook.

2 Model

I present here the benchmark Huggett (1993) model. It features the consumption savings problem of agents, who are subject to a borrowing constraint and idiosyncratic fluctuations in income. I present the model in continuous time, and use finite difference methods for solving the ODEs and PDEs that come from the model.

There are a continuum of households, indexed by $l \in [0,1]$. Each household solves the following continuous time problem

$$\begin{aligned}
\max_{c_t} \quad & E_0 \int_{t=0}^{\infty} e^{-\rho t} u(c(t)) dt \\
s.t. \quad & \dot{a} = z(t) + ra - c(t) \\
& a \geq \underline{a}
\end{aligned}$$

where ρ is the discount rate, $u(c) = c(t)^{1-\gamma}/1-\gamma$, assets follow a law of motion denoted $\dot{a}$, and income, z, follows some stochastic process. Note that both assets and income are state variables for the problem. Note that we will sometimes refer to the evolution of assets under a solution to the problem via the savings function, $s(a,z)$.

The solution to the household's problem is described by the Hamilton-Jacobi-Bellman equation. We can derive this heuristically by making use of a discrete time Bellman equation and Ito's Lemma. First, however, we must take a stand on the process that governs the evolution of income. I consider two such processes: a two-state Markov process driven by the Poisson arrival of switching shocks; and an Ornstein-Uhlenbeck (i.e. mean-reverting) diffusion process.

2.1 HJB: Two-state income process with Poisson switching shocks

Suppose the income process has two states $z \in \{z_1, z_2\}$, and can switch between these states with a Poisson intensity λ_1 and λ_2 (e.g. can switch from state 1 to state 2 with Poisson intensity λ_1). Let i denote the income state a household currently finds itself in. Then, we can write a discrete time approximation to the Bellman equation for the household problem. Consider a discrete unit of time, Δt, over which the household enjoys the flow rate of utility $u(c_i(a))$, and over which assets are adjusted by the amount Δa. Then we can write the Bellman as

$$v_i(a) = \max_c u(c)\Delta t + e^{-\rho \Delta t} \left[(1 - \lambda_i \Delta t) v_i(a + \Delta a) + \lambda_i \Delta t v_{-i}(a + \Delta a) \right]$$

Subtracting $e^{-\rho\Delta t} v_i$ from both sides, and dividing through by Δt, we get

$$\begin{aligned}
\frac{(1 - e^{-\rho\Delta t})}{\Delta t} v_i(a) &= \max_c u(c) + e^{-\rho\Delta t} \frac{1}{\Delta t} \left[(1 - \lambda_i \Delta t)[v_i(a + \Delta a) - v_i(a)] + \lambda_i \Delta t [v_{-i}(a + \Delta a) - v_i(a)] \right] \\
&= \max_c u(c) + e^{-\rho\Delta t} \left[(1 - \lambda_i \Delta t) \frac{[v_i(a + \Delta a) - v_i(a)]}{\Delta a} \frac{\Delta a}{\Delta t} + \lambda_i [v_{-i}(a + \Delta a) - v_i(a)] \right]
\end{aligned}$$

Now, taking the limit as $\Delta t \to 0$, we arrive at the HJB equation[1]

$$\begin{aligned}
\rho v_i(a) &= \max_c u(c) + v_i'(a)\dot{a} + \lambda_i [v_{-i}(a) - v_i(a)] && (1) \\
&= \max_c u(c) + v_i'(a)[z + ra - c] + \lambda_i [v_{-i}(a) - v_i(a)] && (2)
\end{aligned}$$

[1]On the left-hand side, we use L'Hôpital's rule to take the limit. One the right-hand side, we use the definition of a derivative to take the limit.

2.2 HJB: Continuous state income process with a diffusion process

Suppose the log of the income process follows an Ornstein-Uhlenbeck (i.e. mean-reverting) diffusion process:

$$d \log(z) = -\theta \log(z)dt + \sigma dW(t)$$

where θ is associated with the persistence of the process, and $dW(t)$ is a standard Brownian motion.[2] Since income shows up linearly in the budget constraint of the household, we need to know how z evolves. We can use Ito's Lemma to show that[3]

$$dz = \left(-\theta \log(z) + \frac{\sigma^2}{2}\right) zdt + \sigma\sqrt{z}dW(t)$$

For ease of notation, we will write this as $dz = \mu(z)dt + \sigma(z)dW(t)$

As above, we can write a discrete time approximation to the Bellman equation for the household problem. Again, Δt is a discrete unit of time, over which the household enjoys the flow rate of utility $u(c)$, assets are adjusted by the amount Δa, and income adjusts by the amount Δz. Now we can write the Bellman as

$$v(a, z) = \max_c \; u(c)\Delta t + e^{-\rho\Delta t}E\left[v(a + \Delta a, z + \Delta z)\right]$$

Subtracting $e^{-\rho\Delta t}v(a, z)$ from both sides, and dividing through by Δt, we get

$$\frac{(1 - e^{-\rho\Delta t})}{\Delta t}v(a, z) = \max_c \; u(c) + e^{-\rho\Delta t}\frac{1}{\Delta t}E\left[v(a + \Delta a, z + \Delta z) - v(a, z)\right] \quad (3)$$

Note that when taking the limit as $\Delta t \to 0$, the term $\frac{1}{\Delta t}E\left[v(a + \Delta a, z + \Delta z) - v(a, z)\right]$, is not equivalent to the derivative of v. Rather, we need to use Ito's Lemma, which in this case tells us that

$$\lim_{\Delta t \to 0} \quad \frac{1}{\Delta t}E\left[dv(a, z)\right] = \frac{\partial v(a, z)}{\partial a}da + \frac{\partial v(a, z)}{\partial z}dz + \frac{1}{2}\frac{\partial^2 v(a, z)}{\partial z^2}(dz)^2$$

And so taking the limit as $\Delta t \to 0$ through all of equation (3), we find the HJB equation

$$\rho v(a, z) = \max_c \; u(c) + \partial_a v(a, z)[z + ra - c] + \partial_z v(a, z)\mu(z) + \frac{1}{2}\partial_{zz} v(a, z)\sigma(z)^2 \quad (4)$$

2.3 HJB: boundary conditions

Notice that the HJBs are partial differential equations. To pin down solutions to them, we need some boundary conditions. One nice feature of the continuous time nature of the problem presented here is that the first order conditions always hold (which is in contrast to discrete time models, in which the Euler equation only holds with inequality when there

[2] A standard Brownian motion is: $dW(t) = \epsilon_t\sqrt{dt}$, where $\epsilon_t \sim \mathcal{N}(0, 1)$.

[3] For a variable x that follows a diffusion process, we can derive the evolution of some function $f(x)$ via Ito's Lemma: $df(x) = \frac{\partial f}{\partial x}dx + \frac{1}{2}\frac{\partial^2 f(x)}{\partial x^2}(dx)^2$. In our case, $f(\log(z)) = e^{\log(z)}$.

are borrowing constraints present). Thus, we find that $u'(c) = v_i'(a)$ and $u'(c) = v'(a,z)$, for each type of income process respectively. We impose that at the borrowing constraint, it must be that assets are not decreasing: $0 \leq \dot{a} = z + r\underline{a} - c$. This implies that $c \leq z + r\underline{a}$. Since the utility function is concave, we have the boundary condition that:

$$v'(\underline{a}, z) \geq u'(z + r\underline{a})$$

For the general diffusion process, we will want to discretize the income process for computation. In doing so, we will also need some boundary conditions for z. In particular, we assume that there are reflective barriers for z at its upper and lower bounds $\underline{z}$, $\overline{z}$. This generates the following boundary conditions:

$$\partial_z v(a, \underline{z}) = \partial_z v(a, \overline{z}) = 0$$

2.4 Kolmogorov Forward Equations

Because households experience idiosyncratic shocks to income, and their decision rule responds to these shocks, households are distributed across income and asset states. In order to aggregate households decision rules (e.g. to figure out the aggregate asset holdings), we need to keep track of the distribution over time. We do this by writing down a Kolmogorov Forward Equation (KFE).

2.4.1 Poisson income process

For the case of an income process that evolves according to Poisson shocks, we derive the KFE as follows. First, note that for income state i, the CDF over asset holdings at any point in time is: $G_i(a,t) = Pr(\tilde{a}_t \leq a, \tilde{z}_t = z_i)$. At any point in time, households save at the rate $s_i(\tilde{a}_t)$. Then, over any period of time Δ, assets will accumulate according to $\tilde{a}_{t+\Delta} = \tilde{a}_t + \Delta s_i(\tilde{a}_t)$. Now, since households with income i switch to income j with Poisson intensity λ_i (and vice versa for income j), we can see that the CDF over asset holdings at time $t + \Delta$ are given by:

$$\begin{aligned} G_i(a, t+\Delta) &= Pr(\tilde{a}_{t+\Delta} \leq a, \tilde{z}_{t+\Delta} = z_i) \\ &= (1 - \Delta\lambda_i) Pr(\tilde{a}_t \leq a - s_i(a), \tilde{z}_t = z_i) + \Delta\lambda_j Pr(\tilde{a}_t \leq a - s_j(a), \tilde{z}_t = z_j) \end{aligned}$$

where the first term following the second equality refers to the mass of households whose income remained in state i, and the second term refers to the mass of households whose income changed from state j to state i. Now, if we subtract $G_i(a,t)$ from each side, and divide by Δ, we have

$$\begin{aligned} \frac{G_i(a, t+\Delta) - G_i(a,t)}{\Delta} = &\frac{G_i(a - \Delta s_i(a), t) - G_i(a,t)}{\Delta s_i(a)} s_i(a) \\ &- \lambda_i G_i(a - \Delta s_i(a), t) + \lambda_j G_j(a - \Delta s_j(a), t) \end{aligned}$$

Taking the limit as $\Delta \to 0$ yields

$$\partial_t G_i(a,t) = - g_i(a,t) s_i(a) - \lambda_i G_i(a,t) + \lambda_j G_j(a,t)$$

where $g_i(a,t)$ is the PDF over assets for households with income in state i. Finally, take the derivative of everything with respect to a, to get the KFE:

$$\partial_t g_i(a,t) = \partial_a[g_i(a,t)s_i(a)] - \lambda_i g_i(a,t) + \lambda_j g_j(a,t) \tag{5}$$

2.4.2 Ornstein-Uhlenbeck process

The KFE is more difficult to derive for the case with a diffusion process.[4] For this reason, I report that the KFE in this case is:

$$\partial_t g(a,z,t) = \partial_a[g(a,z,t)s(a,z)] - \partial_z[g_i(a,z,t)\mu(z)] + \frac{1}{2}\partial_{zz}[g_i(a,z,t)\sigma(z)^2] \tag{6}$$

Note that like the HJB equation under the diffusion process for income, the KFE is affected by the drift and uncertainty introduced by the diffusion process.

3 Huggett economy equilibrium

A stationary equilibrium in the Huggett economy is a value function v, consumption policy c, interest rate r, a joint distribution function over assets and income g, and a transition function defined by the KFE(s) such that

- Given the interest rate r, the value function v is the solution to the HJB equation, and the consumption policy c maximizes the HJB equation.
- The distribution function is a probability distribution:

$$1 = \int_{\underline{z}}^{\overline{z}} \int_{\underline{a}}^{\overline{a}} g(a,z)dadz$$

- The interest rate clears the asset market so that aggregate assets, $S(r)$, are in zero net supply:

$$0 = S(r) \equiv \int_{\underline{z}}^{\overline{z}} \int_{\underline{a}}^{\overline{a}} ag(a,z)dadz$$

- The distribution function and the KFE(s) satisfy a stationary distribution when $\partial_t g = 0$, so that in the Poisson income case:

$$\begin{aligned} 0 =& \partial_a[g_1(a,t)s_1(a)] - \lambda_1 g_1(a,t) + \lambda_2 g_2(a,t) \\ 0 =& \partial_a[g_2(a,t)s_2(a)] - \lambda_2 g_2(a,t) + \lambda_1 g_1(a,t) \end{aligned}$$

  and in the Ornstein-Uhlenbeck income case:

$$0 = \partial_a[g(a,z,t)s(a,z)] - \partial_z[g_i(a,z,t)\mu(z)] + \frac{1}{2}\partial_{zz}[g_i(a,z,t)\sigma(z)^2]$$

[4] Refer to Moll (2015a) for details.

4 Computational solution to the model and equilibrium

The HJB and KFE equations sometimes have closed form solutions in the form of solutions to systems of PDEs. However, computational solutions are handy for our purposes.[5]

An advantageous solution method involves approximating derivatives using an "upwind scheme". The main idea is that whenever the drift of the state variable is positive, we use a forward difference approximation to the derivative of the value function, and whenever the drift of the state variable is negative, we use a backward difference approximation to the derivative of the value function.

Computation under Poisson income process

For simplicity, I will first consider the Poisson case. We define an evenly spaced set of grid points over assets with $i = 1, \cdots, I$. The two income states are denoted $j = 1, 2$. Let the evolution of the assets (the state variable) be denoted by $s_{i,j} = z_j + ra_i - c_{i,j}$. Let $v_{i,j} \equiv v(a_i, z_j)$. The derivative of the value function with respect to assets is denoted $v'_{i,j}$. Then optimal consumption is given by the envelope condition, $c_{i,j} = (u')(v'_{i,j})$.

Computationally, the derivative of the value function can be described by either the forward or backwards approximations:

$$
\begin{aligned}
v'_{i,j,F} &\approx \frac{v_{i+1,j} - v_{i,j}}{\Delta a} \\
v'_{i,j,B} &\approx \frac{v_{i-1,j} - v_{i,j}}{\Delta a}
\end{aligned} \tag{7}
$$

We compute consumption under each approximation, and then can define a "forward" and "backward" asset drift via:

$$
\begin{aligned}
s_{i,j,F} &= z_j + ra_i - (u')(v'_{i,j,F}) \\
s_{i,j,B} &= z_j + ra_i - (u')(v'_{i,j,B})
\end{aligned}
$$

Now, we define an "upwind scheme" derivative as

$$
v'_{i,j} = v'_{i,j,F}\mathbb{1}\{s_{i,j,F} > 0\} + v'_{i,j,B}\mathbb{1}\{s_{i,j,B} < 0\} + \bar{v}'_{i,j}\mathbb{1}\{s_{i,j,F} < 0 < s_{i,j,B}\} \tag{8}
$$

where $\bar{v}'_{i,j} = u'(z_j + ra_i)$, which corresponds to $s_{i,j} = 0$ which we assume at grid points where $s_{i,j,F} \leq 0 \leq s_{i,j,B}$.

Now, note that on the grid points, it is the case that

$$
\begin{aligned}
\rho v_{i,j} &= u(c_{i,j}) + v'_{i,j}[z_j + ra_i - c_{i,j}] + \lambda_j[v_{i,-j} - v_{i,j}], \quad j = 1, 2 \\
c_{i,j} &= (u')(v'_{i,j})
\end{aligned}
$$

We want to compute the value function via an iterative method. Let $n = 1, 2, \cdots$ be the iteration number. Then we can implicitly update the HJB equation via

$$
\frac{v_{i,j}^{n+1} - v_{i,j}^{n}}{\Delta} + \rho v_{i,j}^{n+1} = u(c_{i,j}^{n}) + (v_{i,j}^{n+1})'[z_j + ra_i - c_{i,j}^{n}] + \lambda_j[v_{i,-j}^{n+1} - v_{i,j}^{n+1}]
$$

[5]... especially when we economists don't know all that much about the solutions to PDEs ...

Substituting in our forward and backward approximations to the derivative, we can write

$$
\begin{aligned}
\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta}+\rho v_{i,j}^{n+1} &= u(c_{i,j}^{n})+(v_{i,j,F}^{n+1})'[z_j+ra_i-c_{i,j,F}^{n}]^{+}+(v_{i,j,B}^{n+1})'[z_j+ra_i-c_{i,j,B}^{n}]^{+}+\lambda_j[v_{i,-j}^{n+1}-v_{i,j}^{n+1}] \\
&= u(c_{i,j}^{n})+(v_{i,j,F}^{n+1})'[s_{i,j,F}^{n}]^{+}+(v_{i,j,B}^{n+1})'[s_{i,j,B}^{n}]^{+}+\lambda_j[v_{i,-j}^{n+1}-v_{i,j}^{n+1}] \\
&= u(c_{i,j}^{n})+\frac{v_{i+1,j}^{n+1}-v_{i,j}^{n+1}}{\Delta a}[s_{i,j,F}^{n}]^{+}+\frac{v_{i-1,j}^{n+1}-v_{i,j}^{n+1}}{\Delta a}[s_{i,j,B}^{n}]^{+}+\lambda_j[v_{i,-j}^{n+1}-v_{i,j}^{n+1}]
\end{aligned}
$$

where the superscripts $+,-$ for a variable x denote the positive and negative parts of x i.e. $x^{+}=\max\{x,0\}$, $x^{-}=\min\{x,0\}$. Note that we can now collect terms to express the HJB as

$$
\frac{v_{i,j}^{n+1}-v_{i,j}^{n}}{\Delta}+\rho v_{i,j}^{n+1}=u(c_{i,j}^{n})+v_{i-1,j}^{n+1}x_{i,j}^{n}+v_{i,j}^{n+1}y_{i,j}^{n}+v_{i+1,j}^{n+1}z_{i,j}^{n}
$$

where

$$
\begin{aligned}
x_{i,j}^{n} &= -\frac{(s_{i,j,B}^{n})^{-}}{\Delta a} \\
y_{i,j}^{n} &= -\frac{(s_{i,j,F}^{n})^{+}}{\Delta a}+\frac{(s_{i,j,B}^{n})^{-}}{\Delta a}-\lambda_j \\
z_{i,j}^{n} &= \frac{(s_{i,j,F}^{n})^{+}}{\Delta a}
\end{aligned}
$$

From the boundary conditions for the HJB, we know that the state equation must drift up (positive) when at the lower bound, and drift down (negative) when at the upper bound. This means that $(s_{1,j,B}^{n})^{-}=\min\{s_{1,j,B}^{n},0\}=0$ and $(s_{I,j,F}^{n})^{+}=\max\{s_{I,j,F}^{n},0\}=0$, and so $x_{1,j}=z_{I,j}=0,\ j=1,2$. Hence, $v_{0,j}^{n+1}$ and $v_{I+1,j}^{n+1}$ are never used.

Notice that the above is a linear system of $2\times I$ equations in $2\times I$ unknowns (the value functions). We can rewrite the system of equations above in matrix notation:

$$
\frac{1}{\Delta}(v^{n+1}-v^{n})+\rho v^{n+1}=u^{n}+\mathbf{A}^{n}v^{n+1} \tag{9}
$$

where the matrix $\mathbf{A}^{n}$ is constructed from the elements $x_{i,j}^{n}, y_{i,j}^{n}, z_{i,j}^{n}$ described above. We note that for even moderately sized grid spaces, most of the matrix $\mathbf{A}$ is filled with zeros. For this reason, it turns out to be efficient to use sparse matrix methods to find a solution.

Now, we need to compute the KFE equations. Solving the KFEs turns out to be very convenient once we have solved the HJB equation, as we will see. First, note that we can discretize the KFEs and the summing-to-one restriction as:

$$
\begin{aligned}
0 &= -[s_{i,j}g_{i,j}]'-\lambda_j g_{i,j}+\lambda_{-j}g_{i,-j} \quad j=1,2 \\
1 &= \sum_{i=1}^{I} g_{i,1}\Delta a+\sum_{i=1}^{I} g_{i,2}\Delta a
\end{aligned}
$$

We need to choose an approximation for the derivative $[s_{i,j}g_{i,j}]'$. Again using an "upwind scheme", we can write the KFEs as

$$0 = -\frac{(s_{i,j,F}^n)^+ g_{i,j} - (s_{i-1,j,F}^n)^+ g_{i-1,j}}{\Delta a} - \frac{(s_{i+1,j,B}^n)^- g_{i+1,j} - (s_{i,j,B}^n)^- g_{i,j}}{\Delta a} - \lambda_j g_{i,j} + \lambda_{-j} g_{i,-j}$$

Now collecting terms, we have

$$0 = g_{i-1,j} z_{i-1,j} + g_{i,j} y_{i,j} + g_{i+1,j} x_{i+1,j} + \lambda_{-j} g_{i,-j}$$

where

$$\begin{aligned} x_{i+1,j}^n &= -\frac{(s_{i+1,j,B}^n)^-}{\Delta a} \\ y_{i,j}^n &= -\frac{(s_{i,j,F}^n)^+}{\Delta a} + \frac{(s_{i,j,B}^n)^-}{\Delta a} - \lambda_j \\ z_{i-1,j}^n &= \frac{(s_{i-1,j,F}^n)^+}{\Delta a} \end{aligned}$$

As above, we can rewrite the system in matrix notation:

$$0 = (\mathbf{A}^n)^T g \tag{10}$$

where we want to use the A^n that corresponds to the solution of the HJB equation above (i.e. the A^n used in the final iteration). Note that this equation can be solved using eigenvalue methods, however, we need to be careful to impose the summing-to-one restriction: $1 = \sum_{i=1}^{I} g_{i,1}\Delta a + \sum_{i=1}^{I} g_{i,2}\Delta a$.

Computation under Ornstein-Uhlenbeck income process

The Ornstein-Uhlenbeck income case is very similar to the Poisson income case, although with somewhat more complicated notation. Here, we define evenly spaced grid points over assets, $i = 1, \cdots, I$, and income, $j = 1, \cdots, J$. As before let the evolution of assets be denoted by $s_{i,j} = z_j + ra_i - c_{i,j}$, let the value function be $v_{i,j} \equiv v(a_i, z_j)$, the derivate with respect to assets is denoted $\partial_a v_{i,j}$, and the derivative with respect to income is $\partial_z v_{i,j}$. Consumption is then given by the envelope condition $c_{i,j} = (u')(\partial_a v_{i,j})$.

The forward and backward approximations to the derivatives are

$$\begin{aligned} \partial_{a,F} v_{i,j} &\approx \frac{v_{i+1,j} - v_{i,j}}{\Delta a} \\ \partial_{a,B} v_{i,j} &\approx \frac{v_{i-1,j} - v_{i,j}}{\Delta a} \end{aligned} \tag{11}$$

We do not need to define an upwind scheme for the derivatives with respect to income, because the income process is a diffusion, it is strictly positive everywhere, and has well-defined boundary conditions. We can then use either a forward or backward approximation; here I choose to adopt a forward difference approximation:

$$\begin{aligned} \partial_z v_{i,j} &\approx \frac{v_{i,j+1} - v_{i,j}}{\Delta z} \\ \partial_{zz} v_{i,j} &\approx \frac{(v_{i,j+1} - v_{i,j}) - (v_{i,j} - v_{i,j-1})}{(\Delta z)^2} \end{aligned}$$

The forward and backward asset drifts are defined as in the previous section, as is the upwind scheme. On the grid points it is the case that

$$\begin{aligned}\rho v_{i,j} &= u(c_{i,j}) + \partial_a v_{i,j}[z_j + ra_i - c_{i,j}] + \mu_j \partial_z v_{i,j} + \frac{\sigma_j^2}{2}\partial_{zz} v_{i,j} \\ c_{i,j} &= (u')(\partial_a v_{i,j})\end{aligned}$$

To compute the value function via iteration, let $n = 1, 2, \cdots$ be the iteration number, and the we can implicitly update the HJB equation via the upwind approximation scheme:

$$\begin{aligned}\frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta} + \rho v_{i,j}^{n+1} &= u(c_{i,j}^n) + \partial_{a,F} v_{i,j}^{n+1}[z_j + ra_i - c_{i,j,F}^n]^+ + \partial_{a,B} v_{i,j,B}^{n+1}[z_j + ra_i - c_{i,j,B}^n]^- \\ &\quad + \mu_j \partial_z v_{i,j}^{n+1} + \frac{\sigma_j^2}{2}\partial_{zz} v_{i,j}^{n+1} \\ &= u(c_{i,j}^n) + \frac{v_{i+1,j}^{n+1} - v_{i,j}^{n+1}}{\Delta a}(s_{i,j,F})^+ + \frac{v_{i,j}^{n+1} - v_{i-1,j}^{n+1}}{\Delta a}(s_{i,j,B})^- \\ &\quad + \mu_j \frac{v_{i,j+1}^{n+1} - v_{i,j}^{n+1}}{\Delta z} + \frac{\sigma_j^2}{2}\frac{v_{i,j+1}^{n+1} - 2v_{i,j}^{n+1} + v_{i,j-1}^{n+1}}{(\Delta z)^2}\end{aligned}$$

Now collecting terms, we have

$$\frac{v_{i,j}^{n+1} - v_{i,j}^n}{\Delta} + \rho v_{i,j}^{n+1} = u(c_{i,j}^n) + v_{i-1,j}^{n+1} x_{i,j}^n + v_{i,j}^{n+1}(y_{i,j}^n + \nu_j) + v_{i+1,j}^{n+1} z_{i,j}^n + v_{i,j-1}^{n+1}\chi_j + v_{i,j+1}^{n+1}\zeta_j$$

where

$$\begin{aligned}x_{i,j}^n &= -\frac{(s_{i,j,B}^n)^-}{\Delta a} \\ y_{i,j}^n &= -\frac{(s_{i,j,F}^n)^+}{\Delta a} + \frac{(s_{i,j,B}^n)^-}{\Delta a} \\ z_{i,j}^n &= \frac{(s_{i,j,F}^n)^+}{\Delta a} \\ \chi_j &= \frac{\sigma_j^2}{2(\Delta z)^2} \\ \nu_j &= -\frac{\mu_j}{\Delta z} - \frac{\sigma_j^2}{(\Delta z)^2} \\ \zeta_j &= \frac{\mu_j}{\Delta z} + \frac{\sigma_j^2}{2(\Delta z)^2}\end{aligned}$$

From the boundary conditions for assets, we know that the state equation must drift up (positive) when at the lower bound, and drift down (negative) when at the upper bound. This means that $(s_{1,j,B}^n)^- = \min\{s_{1,j,B}^n, 0\} = 0$ and $(s_{I,j,F}^n)^+ = \max\{s_{I,j,F}^n, 0\} = 0$, and so $x_{1,j} = z_{I,j} = 0$ for all j. Hence, $v_{0,j}^{n+1}$ and $v_{I+1,j}^{n+1}$ are never used.

From the boundary conditions for income, we have that $\partial_z v_{i,1} = \frac{v_{i,1} - v_{i,0}}{\Delta z} = 0$ and so $v_{i,0} = v_{i,1}$, and $\partial_z v_{i,J} = \frac{v_{i,J+1} - v_{i,J}}{\Delta z} = 0$ and so $v_{i,J+1} = v_{i,J}$.

The above is a linear system of $I \times J$ equations. Let the $I \times J$ matrix $\mathbf{C}$ contain all of the elements in χ_j, ν_j, ζ_j, and let the $I \times J$ matrix $\mathbf{B^n}$ contain all of the elements in $x_{i,j}^n, y_{i,j}^n, z_{i,j}^n$. Let $\mathbf{A}^n = \mathbf{B}^n + \mathbf{C}$. Then, as in the previous section, we can write the system of equations in matrix notation as

$$\frac{1}{\Delta}(v^{n+1} - v^n) + \rho v^{n+1} = u^n + \mathbf{A}^n v^{n+1} \tag{12}$$

We now need to compute the KFE equation. We can discretize the KFE and the summing-to-one restriction as:

$$\begin{aligned} 0 &= -\partial_a[s_{i,j} g_{i,j}] - \partial_z[\mu_j g_{i,j}] + \frac{1}{2}\partial_{zz}[\sigma_j^2 g_{i,j}] \\ 1 &= \sum_{i=1}^{J}\sum_{i=1}^{I} g_{i,j}\Delta a \Delta z \end{aligned}$$

We approximate the derivative with respect to assets with an upwind scheme, and approximate the derivative with respect to income with a forward difference:

$$\begin{aligned} 0 = &-\frac{(s_{i,j,F}^n)^+ g_{i,j} - (s_{i-1,j,F}^n)^+ g_{i-1,j}}{\Delta a} - \frac{(s_{i+1,j,B}^n)^- g_{i+1,j} - (s_{i,j,B}^n)^- g_{i,j}}{\Delta a} \\ &- \frac{\mu_{j+1} g_{i,j+1} - \mu_j g_{i,j}}{\Delta z} + \frac{1}{2}\frac{(\sigma_{j+1}^2 g_{i,j+1} - \sigma_j^2 g_{i,j}) - (\sigma_j^2 g_{i,j} - \sigma_{j-1}^2 g_{i,j-1})}{(\Delta z)^2} \end{aligned}$$

Now collecting terms, we have

$$0 = g_{i-1,j} z_{i-1,j}^n + g_{i,j}(y_{i,j}^n + \nu_j) + g_{i+1,j} x_{i+1,j}^n + g_{i,j-1}\chi_{j-1} + g_{i,j+1}\zeta_{j+1}$$

where

$$\begin{aligned} x_{i+1,j}^n &= -\frac{(s_{i+1,j,B}^n)^-}{\Delta a} \\ y_{i,j}^n &= -\frac{(s_{i,j,F}^n)^+}{\Delta a} + \frac{(s_{i,j,B}^n)^-}{\Delta a} \\ z_{i-1,j}^n &= \frac{(s_{i-1,j,F}^n)^+}{\Delta a} \\ \chi_j &= \frac{\sigma_j^2}{2(\Delta z)^2} \\ \nu_j &= -\frac{\mu_j}{\Delta z} - \frac{\sigma_j^2}{(\Delta z)^2} \\ \zeta_j &= \frac{\mu_j}{\Delta z} + \frac{\sigma_j^2}{2(\Delta z)^2} \end{aligned}$$

Again, we can rewrite the system in matrix notation:

$$0 = (\mathbf{A}^n)^T g \tag{13}$$

where we want to use the A^n that corresponds to the solution of the HJB equation above (i.e. the A^n used in the final iteration). Note that this equation can be solved using eigenvalue methods, however, we need to be careful to impose the summing-to-one restriction: $1 = \sum_{i=1}^{J}\sum_{i=1}^{I} g_{i,j}\Delta a \Delta z$.

4.1 Algorithm

In order to solve the agent's problem and find an equilibrium, we solve the HJB and KFE equations described above, and use a bisection method over the interest rate. I use the following algorithm for the model with a Poisson income process:

1. Guess an interest rate r^0. For $t = 0, 1, 2, \cdots$ follow:
   (a) Guess $v^0_{i,j}$ for $i = 1, \cdots, I$, and $j = 1, 2$. For $n = 0, 1, 2, \cdots$ follow:
      i. Compute the derivate of the value function, $(v^n_{i,j})'$, using the upwind scheme in equations (7) and (8)
      ii. Compute consumption using $c^n_{i,j} = (u')^{-1}[(v^n_{i,j})']$
      iii. Compute the next value function iterate, $n+1$, from equation (9)
      iv. If v^{n+1} is close to v^n under some convergence criterion, stop. If not, set $n = n+1$ and return to step i.
   (b) Take the matrix $\mathbf{A}$ associated with the final step in the HJB equation computation. Solve for the stationary distribution, g, from the system of KFEs in (10).
   (c) Compute asset supply, $S(r^t) = \sum_{i=1}^{I} a_i g_{i,1} \Delta a + \sum_{i=1}^{I} a_i g_{i,2} \Delta a$. If $S(r^t)$ is close to zero under some convergence criterion, stop. If not: when $S(r^t) > 0$ decrease the interest rate guess, and when $S(r^t) < 0$ increase the interest rate guess. Set $t = t+1$, and return to step (a).

For the Ornstein-Uhlenbeck income process I use the following algorithm:

1. Guess an interest rate r^0. For $t = 0, 1, 2, \cdots$ follow:
   (a) Guess $v^0_{i,j}$ for $i = 1, \cdots, I$, and $j = 1, \cdots, J$. For $n = 0, 1, 2, \cdots$ follow:
      i. Compute the derivate of the value function with respect to assets, $\partial_a v^n_{i,j}$, using the upwind scheme in equations (11) and (8)
      ii. Compute consumption using $c^n_{i,j} = (u')^{-1}(\partial_a v^n_{i,j})$
      iii. Compute the next value function iterate, $n+1$, from equation (12)
      iv. If v^{n+1} is close to v^n under some convergence criterion, stop. If not, set $n = n+1$ and return to step i.
   (b) Take the matrix $\mathbf{A}$ associated with the final step in the HJB equation computation. Solve for the stationary distribution, g, from the system of KFE in (13).
   (c) Compute asset supply, $S(r^t) = \sum_{i=1}^{J} \sum_{i=1}^{I} a_i g_{i,j} \Delta a \Delta z$. If $S(r^t)$ is close to zero under some convergence criterion, stop. If not: when $S(r^t) > 0$ decrease the interest rate guess, and when $S(r^t) < 0$ increase the interest rate guess. Set $t = t+1$, and return to step (a).

5 Results

I modified and translated into Julia the Matlab codes found at the HACT project website Moll (2015*b*). Although the models I have presented here do not have a large state space, and so do not take long to solve, I found that the Julia code is up to several orders of magnitude faster than the Matlab code.

However, one major drawback of using Julia is the difficulty encountered in plotting. For various reasons, several of the main plotting packages did not work on my computer, so for the Jupyter notebook I had to resort to using the Gadfly package. Gadfly is a fairly clean plotting tool, however it is lacking many features that we might want to work with: e.g. different line styles, Latex integration, 3D plots, simple legends. Another problem is that I had a lot of difficulty getting Gadfly to produce plots that can be compiled in a latex file. As a work-around, I used the Julia package DataFrames to output the model data, and plotted the graphs using the Matplotlib package in Python.

Before discussing the results of the model, I briefly describe the parameterization I used for the model. Note that I have not chosen the parameters for realism, rather I have chosen them to produce clear, distinct results that allow us to discuss the model's mechanisms. For the Poisson income process version of the model, I use the following. The discount factor is $\rho = 0.05$, and the risk aversion parameter in the CRRA utility function is $\gamma = 2$. As noted in Moll (2015*a*), the implicit updating scheme for computation of the HJB equation allows for arbitrary values of the time step, Δ, so I use 1000 which produces stable results. I set the borrowing constraint to $\underline{a} = -0.15$, and use 1000 points for the asset grid space in the region $[\underline{a}, \overline{a}] = [-0.15, 5]$. Income in the low state is $z_1 = 0.1$, and in the high state is $z_2 = 0.2$. The Poisson switching intensities are $\lambda_1 = 1.2$ and $\lambda_2 = 1.5$.

For the diffusion process model, I also use $\rho = 0.05$ and $\gamma = 2$. Again, $\Delta = 1000$, and $[\underline{a}, \overline{a}] = [-0.15, 5]$. For the state space, I use 200 points for the asset grid space in the region $[\underline{a}, \overline{a}] = [-0.15, 5]$, and 200 points for the income grid space. To parameterize the Ornstein-Uhlenbeck process, first note that the stationary distribution for $\log(z)$ is $\mathcal{N}(0, \frac{\sigma^2}{\theta})$. I set $\frac{\sigma^2}{\theta} = 0.1$. Since z is log-normally distributed with zero mean, we find that the mean of z is given by $\exp(\frac{\sigma^2}{2\theta})$. Assuming that the autocorrelation of z is 0.5, then the parameter $\theta = -\log(0.5)$, where $(1 - \theta)$ is the equivalent of the discrete time AR(1) autocorrelation coefficient. This value is approximately 0.31. I opt for a low serial correlation in the income process, because this increases the likelihood that agents with high incomes suffer negative income shocks, which induces them to save more for precautionary reasons. I use an income grid space that is 0.75 and 1.25 times the mean of z, $\exp(\frac{\sigma^2}{2\theta})$, from above.

Computation under Poisson income process

Figures 1 and 2 plot the consumption and savings policy functions for the model under each of the income states. We can see that the consumption policy function exhibits curvature near the borrowing constraint, but is approximately linear further from the constraint. Agents in the high income state consume more close to and at the borrowing constraint, because of their higher incomes. The low income agent dissaves over the entire asset space, while the high income agent saves over most of the asset space. Low income agents dissave in order to smooth consumption over states, while high income agent's saving is due to precautionary

Figure 1: Consumption function under Poisson income process

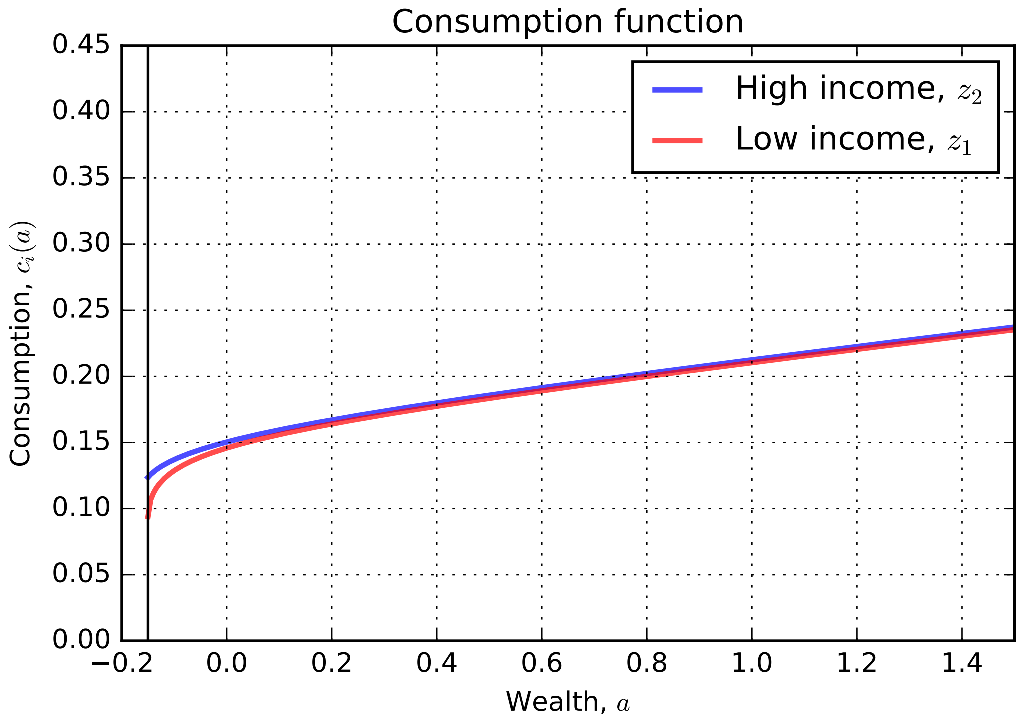
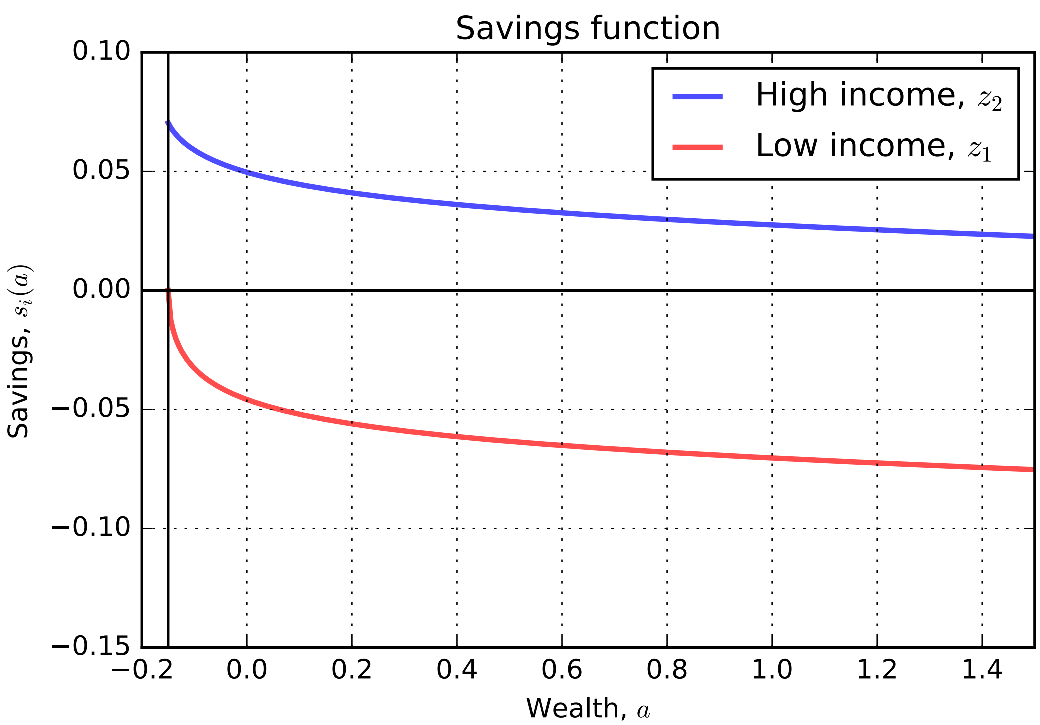


Figure 2: Savings function under Poisson income process



behaviour: an aversion to hitting the lower bound.

Figure 3: Stationary distribution under Poisson income process

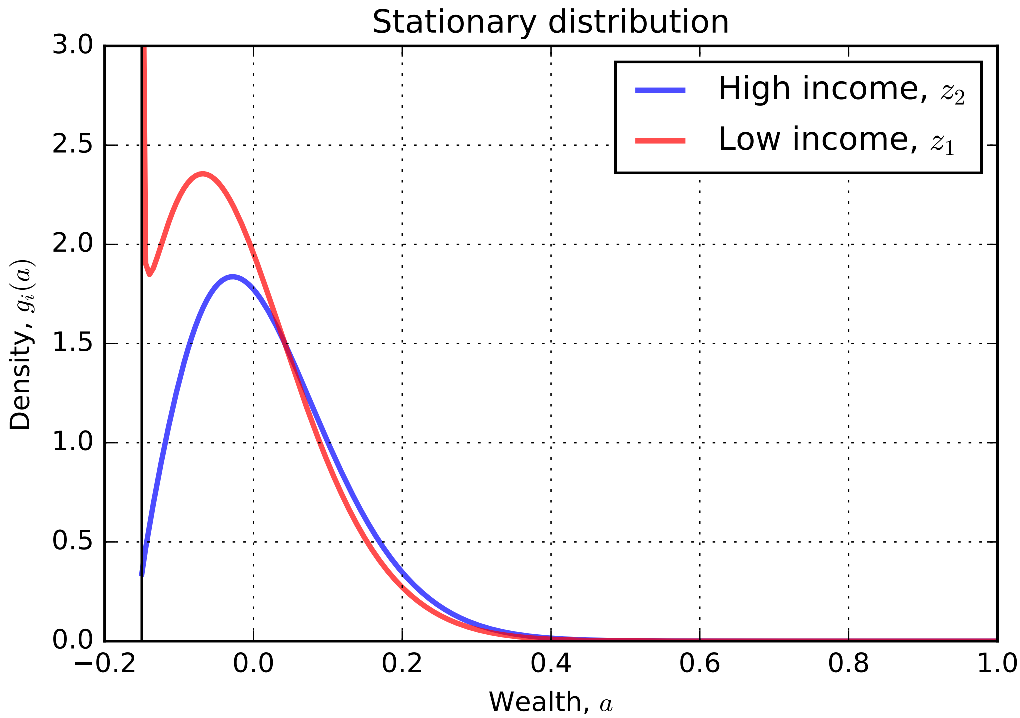


Figure 3 plots the stationary distribution of agents across assets and the two income states. We can see that the low income state generates a Dirac mass point at the borrowing constraint. The distribution under the high income state is shifted left relative to the low income state, reflecting the fact that those agents are saving to avoid the borrowing constraint. As a result, very few of these agents find themselves at the borrowing constraint. Note that there is more mass in the distribution under the low income state because the Poisson switching intensity is larger in the high income state (i.e. agents are more likely to switch from high income to low income than vice versa).

Computation under Ornstein-Uhlenbeck income process

Figures 4 and 5 plot the consumption and savings policy functions for the model across asset and income states. As in the previous example, we can see that the consumption policy function exhibits curvature near the borrowing constraint and is approximately linear further from the constraint. There also appears to be some curvature with respect to income for lower income states. The savings function shows that The lower income agents tend to dissave, while the higher income agents save. One interesting feature of the graph is the part of the surface towards the back left: on the borrowing constraint, the savings function is zero across the lower half of the income space, and rising thereafter. This suggests that agents who are borrowing constrained spend all of their income up until some threshold income, when it is worthwhile for them to save their way out of the constraint.

Figure 4: Consumption function under diffusion income process

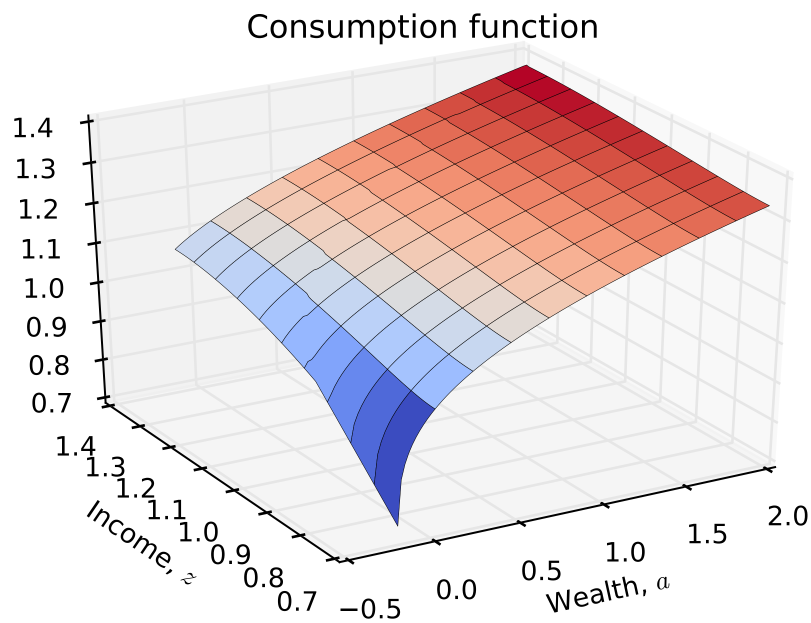


Figure 5: Savings function under diffusion income process

Figures 6 and 7 both plot the stationary distribution of agents across income and assets, each viewed from a different position. The first figure highlights that low income agents are more likely to be borrowing constrained and that, as before, there are mass points of agents on the borrowing constraint for low income levels. Interestingly, the mass point peaks at a higher income level than the lower bound of the state space. I think that this is because conditional on being at the borrowing constraint, income is log-normally distributed (in fact, the peak of the distribution among borrowing constrained agents is at the mean of the distribution over z). The second figure shows that high income agents tend to be away from the borrowing constraint, and hold positive wealth/assets due to the precautionary motive.

Figure 6: Stationary distribution under diffusion income process (view 1)

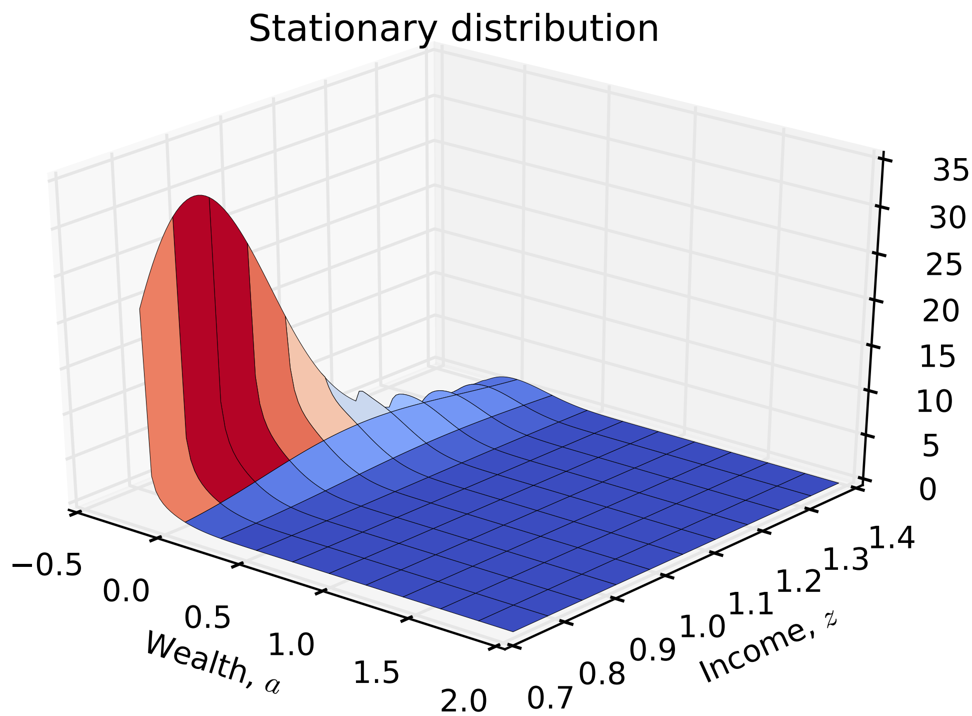
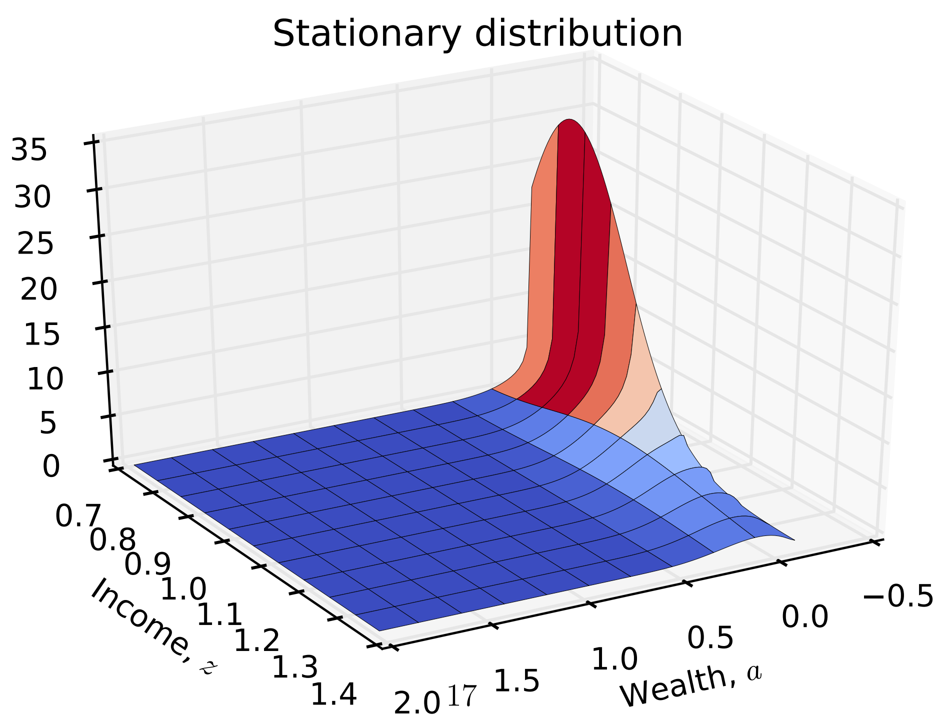


Figure 7: Stationary distribution under diffusion income process (view 2)



References

Huggett, Mark. 1993. "The risk-free rate in heterogeneous-agent incomplete-insurance economies." *Journal of economic Dynamics and Control*, 17(5): 953–969.

Moll, Benjamin. 2015*a*. "Continuous time models lecture notes." *http://www.princeton.edu/~moll/notes.htm*, Accessed: 2010-06-02.

Moll, Benjamin. 2015*b*. "HACT project codes." *http://www.princeton.edu/~moll/HACTproject.htm*, Accessed: 2010-06-02.