The Khan and Thomas Model in Continuous Time

Final Project for Computational Economics

Victoria Gregory

June 3, 2016

1 Introduction

This paper describes my final project for John Stachurski's Computational Economics course at New York University. Following Achdou, Han, Lasry, Lions, and Moll (2015) and Ahn, Kaplan, Moll, and Winberry (2016), I demonstrate a new solution method for solving heterogeneous agent models in continuous time with aggregate shocks.

The solution method involves numerically solving a system of partial differential equations to compute the steady state objects of the model. To solve for the dynamics around the steady state, a locally accurate perturbation method is used. The method has a number of computational advantages relative to discrete-time methods. We will see that each iteration simply comes down to inverting a very sparse matrix. There is also a link between the value function and the stationary distribution that makes computing the stationary distribution very easy once one has solved for the value function. Additionally, the perturbation method does not rely on approximate aggregation, unlike the method of Krussel and Smith (1998).

Although the methods are sufficiently general, here I work with the Khan and Thomas (2008) model. This is a well-known model of firm investment which was designed to generate the "lumpy" investment patterns seen in the data. It features heterogeneous firms who face a fixed cost of capital adjustment.

I adapted the existing code[1] for Julia. In this document, I describe the model and solution method and briefly present some results. The accompanying Jupyter notebook describes the code in detail.[2]

2 Model

The description will be kept brief, but it is based on Khan and Thomas (2008) and follows the same notation as Winberry (2016).

[1] Available on Ben Moll's website: `http://www.princeton.edu/~moll/PHACTproject.htm`

[2] Available on my GitHub: `https://github.com/vgregory757/KhanThomas`

2.1 Firms

There is a continuum of heterogeneous firms, indexed by i, who produce, invest in capital, and hire labor. Their production function is:

$$Z(t)\varepsilon_i k_i^\theta n_i^\nu$$

where $Z(t)$ is an aggregate TFP shock common to all firms, ε_i is an idiosyncratic productivity shock, k_i is capital, n_i is labor, θ is the elasticity of output with respect to capital, and ν is the elasticity of output with respect to labor. The idiosyncratic shock follows a Poisson process taking on two values, ε_L and ε_H, with intensities λ_L and λ_H, respectively. The aggregate shock follows an Ornstein-Uhlenbeck process (the continuous time analogue of an AR(1) process):

$$d\log Z(t) = -(1-\rho_z)\log Z(t)dt + \sigma_z dW(t)$$

where $W(t)$ is a standard Brownian motion. This would correspond to an AR(1) process with autocorrelation ρ_z and standard deviation of shocks σ_z.

After production takes place, the firm will decide whether it wants to invest in capital for next period. Investment will result in the following drift for capital, $\dot{k} = i - \delta k$, where δ is the depreciation rate. If the firm invests ($i \neq 0$), it must pay an adjustment cost of $\chi_0 + \frac{\chi_1}{2}\left(\frac{i}{k}\right)^2 k$. This particular form of adjustment cost contains both a fixed and a variable term and is needed to generate the lumpy pattern of adjustment seen in the data.

2.2 Households

The representative household has preferences given by

$$\mathbb{E}\int_{t=0}^{\infty} e^{-\rho t}\frac{1}{1-\sigma}\left(C(t) - \chi\frac{N(t)^{1+\varphi}}{1+\varphi}\right)^{1-\sigma}$$

where $C(t)$ is consumption, $N(t)$ is labor supply, ρ is the continuous time discount factor, σ is the coefficient of relative risk aversion, χ determines the disutility of labor, and φ is the Frisch elasticity of labor supply.

2.3 Equilibrium

A *recursive competitive equilibrium* is a set of functions $v_i(k;t)$, value function for firm with productivity level i, capital stock k at time t, $g_i(k;t)$, distribution of firms with productivity i over capital at time t, $w(t)$, wage at time t, $\Lambda(t)$, stochastic discount factor at time t, and $Z(t)$, aggregate shock at time t, such that:

1. **Firm optimization** (Hamilton-Jacobi-Bellman equation)

$$\rho v_i(k;t) = \max_{i,n} \Lambda(t)\underbrace{\left[Z(t)\varepsilon_i k^\theta n^\nu - w(t)n - \left(i + \chi_0 \mathbb{1}\{i \neq 0\} + \frac{\chi_1}{2}\left(\frac{i}{k}\right)^2 k\right)\right]}_{:=\pi_i(i;t)}$$
$$+ \partial_k v_i(k;t)\underbrace{(i-\delta k)}_{=\dot{k}} + \lambda_i(v_j(k;t) - v_i(k;t)) + \frac{1}{dt}\mathbb{E}_t[dv_i(k;t)]$$

This is the recursive formulation of the firm's problem. Its individual states are $\lambda_i(t)$ and $k_j(t)$. The aggregate state is summarized by the index t. The last two terms compute expectations with respect to the idiosyncratic and aggregate states, respectively. Also note that by the envelope condition, $\partial_k v_i(k;t) = \partial_i \pi_i(i;t) = 1 + \frac{\chi_1 i}{k}$. This will be useful throughout the solution method.

2. **Goods market clearing**

   The household's first-order conditions lead to a stochastic discount factor, $\Lambda(t)$, which also governs how firms discount future streams of profits, as was seen in the HJB equation above.

$$\Lambda_t = \left(C(t) - \chi \frac{N(t)^{1+\varphi}}{1+\varphi} \right)^{-\sigma}$$

$$C(t) = \sum_i \int \left(Z(t)\varepsilon_i k^\theta n_i(k;t)^\nu - \left(i_i(k;t) + \chi_0 \mathbb{1}\{i_i(k;t) \neq 0\} + \frac{\chi_1}{2} \left(\frac{i_i(k;t)}{k} \right)^2 k \right) \right) g_i(k;t)$$

3. **Labor market clearing**

$$N(t) = \sum_i \int n_i(k;t) g_i(k;t) = \left(\frac{w(t)}{\chi} \right)^{\frac{1}{\varphi}}$$

   This also follows from the household's first-order conditions.

4. **Evolution of distribution** (Kolmogorov-Forward equation)

$$\partial_t g_i(k;t) = -\partial_k \left[(i_t(k;t) - \delta k) g_i(k;t) \right] - \lambda_i g_i(k;t) + \lambda_j(k;t) g_j(k;t)$$

   This governs the evolution of the distribution of firms over their individual states.

5. **Evolution of aggregate shock**

$$d \log Z(t) = -(1-\rho_z) \log Z(t) dt + \sigma_z dW(t)$$

3 Solution Method

Solving for the equilibrium with aggregate shocks requires a two-step solution method, as developed by Ahn, Kaplan, Moll, and Winberry (2016). The first step is to solve for the stationary equilibrium of the model without aggregate shocks. Achdou, Han, Lasry, Lions, and Moll (2015) outline finite difference methods to implement this step. The next step is to linearize the equilibrium conditions around the steady state and solve the resulting linear stochastic partial differential equation. This requires taking a Taylor expansion of the equilibrium conditions and then using Chris Sims' `gensys` to solve for the equilibrium. Both steps are outlined in more detail below.

3.1 Solving for the Steady State

The steady state of the Khan and Thomas model will consist of value functions defined over the individual states, a stationary distribution of firms over individual states, and a constant wage that satisfies the market clearing conditions.[3]

To solve for the value functions and distributions, a finite-difference method is used which approximates these objects over a grid of capital stocks. Denote this grid as $\mathbf{k} = (k_1, k_2, \dots, k_N)$, which is equally spaced with step size Δ_k. Also denote $v_{ij} = v_i(k_j)$ and $g_{ij} = g_i(k_j)$. Also define the vectors $\mathbf{v} = (v_{11}, v_{12}, \dots, v_{1N}, v_{21}, v_{22}, \dots, v_{2N})$ and $\mathbf{g} = (g_{11}, g_{12}, \dots, g_{1N}, g_{21}, g_{22}, \dots, g_{2N})$.

To implement the method, start with an initial guess for the value function, v_{ij}^0, and update the guess iteravely following:

$$\frac{v_{ij}^{n+1} - v_{ij}^n}{\Delta} + \rho v_{ij}^{n+1} = \pi(i_{ij}^n) + (v_{ij}^{n+1})'(i_{ij}^n - \delta k_j) + \lambda_i(v_{-ij}^{n+1} - v_{ij}^{n+1}) \tag{1}$$

where Δ is a step-size parameter, which can be arbitrarily large, and $(v_{ij}^{n+1})'$ is an approximation to the derivative, $\partial_k v_{ij}(t)$. This approximation can be one of the following (suppressing the n superscripts for now):

$$\partial_k v_{ij} \approx \frac{v_{ij} - v_{ij-1}}{\Delta_k} := v'_{ijB} \tag{2}$$

$$\partial_k v_{ij} \approx \frac{v_{ij+1} - v_{ij}}{\Delta_k} := v'_{ijF} \tag{3}$$

(2) is known as the backward difference and (3) is known as the forward difference. The presence of two types of approximations raises two questions, the first being when to use which approximation, and the second being what to use at the endpoints.

It turns out that the best way to address the first question is to use an *upwind scheme*. The idea is to use the forward difference when the drift of the state variable, capital, is positive, and a backward difference when the drift is negative. Denote the drift of capital by $s_{ij}^n = i_{ij}^n - \delta k_j$. To implement the upwind scheme, first compute the drift according to both the forward and backward difference, s_{ijF} and s_{ijB}. To do this, observe that optimal investment, conditional on the firm deciding to invest, can be found by maximizing the following expression:

$$\max_i \left(v'_{ij} i - \left[i + \chi_0 + \frac{\chi_1}{2} \left(\frac{i}{k} \right)^2 k \right] \right)$$

This is saying that the payoff to investing is the marginal value of an extra unit of capital times the amount of new capital minus the adjustment costs and the cost of investment itself. This yields the solution $i = \frac{k}{\chi_1}(v'_{ij} - 1)$. The firm then invests this amount if the profit function π is positive; if not, it chooses not to invest. This investment policy can be computed using both the forward and backward differences.

Once we have the drift computed using the forward and backward differences, we can use the following to approximate v'_{ij}:

$$v'_{ij} = v'_{ijF} \mathbb{1}\{s_{ijF} > 0\} + v'_{ijB} \mathbb{1}\{s_{ijB} < 0\} + \bar{v}'_{ij} \mathbb{1}\{s_{ijF} \leq 0 \leq s_{ijB}\} \tag{4}$$

[3] For the setup here,, we can simply ignore the SDF since it is a multiplicative constant on the profit function. We can also ignore the aggregate shock term on the production function, since the shock process in levels is stationary around 1.

At grid points where $s_{ijF} \leq 0 \leq s_{ijB}$, we set the drift equal to zero, meaning that $i_{ij}^n = \delta k_j$. We therefore set the derivative of the value function equal to the derivative of the profit function at this point, which is $\bar{v}'_{ij} = 1 + \delta\chi_1$ according to the envelope condition.

At the endpoints, we need to impose state constraints to ensure that capital does not go beyond the grid points. Here, we also set the drift equal to zero, as is done above.

For the investment policy, i_{ij}^n, we can just recompute it based on the upwind scheme's approximation of the derivative. Now we have everything we need to iterate on (1). But since it is actually a linear system in $2N$ unknowns, we can re-write it in matrix notation so that it can be solved very quickly on a computer. We can collect terms and define the variables:

$$
\begin{aligned}
x_{ij} &= -\frac{\min\{s_{ijB}, 0\}}{\Delta_k} \\
y_{ij} &= -\frac{\max\{s_{ijF}, 0\}}{\Delta_k} + \frac{\min\{s_{ijB}, 0\}}{\Delta_k} - \lambda_i \\
z_{ij} &= \frac{\max\{s_{ijF}, 0\}}{\Delta_k}
\end{aligned}
$$

so that (1) is now:

$$
\frac{v_{ij}^{n+1} - v_{ij}^n}{\Delta} + \rho v_{ij}^{n+1} = \pi(i_{ij}^n) + v_{i,j-1}^{n+1} x_{ij} + v_{ij}^{n+1} y_{ij} + v_{ij+1}^{n+1} z_{ij} + v_{-ij}^{n+1} \lambda_i
$$

In matrix notation, this is:

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

where

$$
\mathbf{A}^n = \begin{bmatrix}
y_{11} & z_{11} & 0 & \dots & 0 & \lambda_1 & 0 & 0 & 0 & 0 \\
x_{12} & y_{12} & z_{12} & 0 & \dots & 0 & \lambda_1 & 0 & 0 & 0 \\
0 & x_{13} & y_{13} & z_{13} & 0 & \dots & 0 & \lambda_1 & 0 & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\
0 & \ddots & \ddots & x_{1N} & y_{1N} & 0 & 0 & 0 & 0 & \lambda_1 \\
\lambda_2 & 0 & 0 & 0 & 0 & y_{21} & z_{21} & 0 & 0 & 0 \\
0 & \lambda_2 & 0 & 0 & 0 & x_{22} & y_{22} & z_{22} & 0 & 0 \\
0 & 0 & \lambda_2 & 0 & 0 & 0 & x_{23} & y_{23} & z_{23} & 0 \\
0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \dots & \dots & 0 & \lambda_2 & 0 & \dots & 0 & x_{2N} & y_{2N}
\end{bmatrix}, \pi^n = \begin{bmatrix}
\pi(i_{11}^n) \\ \vdots \\ \vdots \\ \pi(i_{1N}^n) \\ \pi(i_{21}^n) \\ \vdots \\ \vdots \\ \pi(i_{2N}^n)
\end{bmatrix}
$$

This system can be written even more compactly as:

$$
\mathbf{B}^n v^{n+1} = b^n \tag{5}
$$

where $\mathbf{B}^n = \left(\frac{1}{\Delta} + \rho\right)\mathbf{I} - \mathbf{A}^n$ and $b^n = \pi^n + \frac{1}{\Delta} v^n$.

To summarize, the algorithm for finding the value function is:

1. Approximate the derivative of the value function from (2), (3), and (4).

2. Compute i^n from $i = \frac{k}{\chi_1}(v'_{ij} - 1)$, using the derivative obtained from step 1. If the value of investing is positive, set this equal to i, otherwise set this equal to zero.

3. Update v^{n+1} using (5). This is computationally very efficient because of how sparse $\mathbf{B}^n$ is.

4. If v^{n+1} is sufficiently close to v^n, stop. If not, go back to step 1.

Aside from being vary fast and efficient, the advantage of this method is that once we have the value function, the Kolmogorov-Forward equation for the stationary distribution g_{ij} can be solved in just one step. For the steady state version of the model, this equation is:

$$0 = -\partial_k \left[(i_t(k) - \delta k) g_i(k)\right] - \lambda_i g_i(k) + \lambda_j g_j(k)$$

We first discretize this equation as:

$$0 = -[s_{ij} g_{ij}]' - \lambda_i g_{ij} + \lambda_{-i} g_{-ij}$$

Again, we need to decide where to use a forward and where to use a backward difference, and it turns out that the most convenient approximation is:

$$0 = -\frac{\max\{s^n_{ijF}, 0\} g_{ij} - g_{ij-1} \max\{s^n_{ij-1F}, 0\}}{\Delta_k} - \frac{g_{ij+1} \min\{s_{ij+1B}, 0\} - g_{ij} \min\{s_{ijB}, 0\}}{\Delta_k} - g_{ij}\lambda_i + \lambda_{-i} g_{-ij}$$

Re-arranging and writing in matrix form, we get:

$$\mathbf{A}^T g = 0 \tag{6}$$

where $\mathbf{A}^T$ is the transpose of the matrix from the HJB equation. So once the HJB is solved, we can find the stationary distribution simply by solving (6).

To complete the steady state solution, we just need to find the market clearing wage. This is done using a bisection method, as follows:

1. Guess a value for the wage, $w^{(n)}$.

2. Compute the firm's individual labor demand using:

$$n_{ij} = \left(\frac{\varepsilon_i k_j^\theta \nu}{w^{(n)}}\right)^{\frac{1}{1-\nu}}$$

   which comes from the firm's first-order condition for labor. Define the vector of labor policies as $\mathbf{n} = (n_{11}, n_{12}, \ldots, n_{1N}, n_{21}, n_{22}, \ldots, n_{2N})$.

3. Compute value functions and stationary distributions as outlined above.

4. The disutility of labor parameter, χ, is chosen so that in steady state, aggregate labor is $\frac{1}{3}$.[4] Therefore, we can compute excess labor supply as:

$$S(w^{(n)}) = \frac{1}{3} - \Delta_k \left(\mathbf{n}' \mathbf{g}^{(n)}\right)$$

   If $S(w^{(n)}) > 0$, increase the guess for w and go back to step 1. If $S(w^{(n)}) < 0$, decrease the guess for w and go back to step 1. Otherwise, stop.

[4] This is possible because of the form of the labor market clearing condition.

3.2 Solving for the Aggregate Dynamics

From now on, we bring back the t index to represent the aggregate state. Now, all of the equilibrium objects we need to solve for will additionally depend on t. To describe this part of the solution method, it is helpful to introduce a few more objects. Let $\mathbf{v}(t)$ be the time-varying version of $\mathbf{v}$ and define $\mathbf{g}(t)$ and $\boldsymbol{\pi}(t)$ similarly. Let $\mathbf{A}(t)$ be the time-varying version of the matrix $\mathbf{A}$ associated with $\mathbf{v}(t)$. Also denote the steady state analogues of these objects as $\mathbf{v}^*$ and $\mathbf{g}^*$ and the steady state stochastic discount factor as Λ^*. We can re-write the equilibrium conditions of the model in matrix form as follows:

1. **Hamilton-Jacobi-Bellman equation**

$$\rho \mathbf{v}(t) = \pi(t) + \mathbf{A}(t)\mathbf{v}(t) + \frac{1}{dt}\mathbb{E}_t[d\mathbf{v}(t)]$$

2. **Kolmogorov-Forward equation**

$$\partial_t \mathbf{g}(t) = \mathbf{A}(t)^T \mathbf{g}(t)$$

3. **Goods market clearing**[5]

$$\Lambda(t) - \left(C(t) - \chi \frac{N(t)^{1+\varphi}}{1+\varphi}\right)^{-\sigma} = 0$$

4. **Evolution of aggregate shock**

$$d \log Z(t) = -(1-\rho_z)\log Z(t)dt + \sigma_z dW(t)$$

To solve for the equilibrium dynamics, we will use Chris Sims' `gensys` algorithm, so it is useful to write the system in his general form. Define the vector of endogenous variables as:[6]

$$\mathbf{y}(t) = (\mathbf{v}(t), \mathbf{g}(t), \Lambda(t), \log Z(t))'$$

and similarly:

$$\dot{\mathbf{y}}(t) = (\dot{\mathbf{v}}(t), \dot{\mathbf{g}}(t), \dot{\Lambda}(t), \dot{\log} Z(t))'$$

Next, define the vector of innovations as $\mathbf{z}(t) = \frac{dW(t)}{dt}$. Then, define the vector of expectational errors for $\mathbf{v}(t)$ as $\boldsymbol{\eta}_{\mathbf{v}}(t) = -\frac{1}{dt}\mathbb{E}_t[d\mathbf{v}(t)]$. `gensys` will impose the restriction that $\mathbb{E}_t[\boldsymbol{\eta}_{\mathbf{v}}(t)] = 0$.

[5] We don't need labor market clearing because of Walras' law.

[6] The reason $\Lambda(t)$ is included as an endogenous variable instead of the wage is the following. The wage can always be computed just using parameters, pre-determined variables like capital, and the exogenous shocks. The equation turns out to be:

$$w = \left[(Z\nu)^{\varphi}\chi^{1-\nu}I^{(1-\nu)\varphi}\right]^{\frac{1}{\varphi+1-\nu}}$$

where $I = \left[\left(\boldsymbol{\varepsilon}\mathbf{k}^{\theta}\right)^{\frac{1}{1-\nu}}\right]'\mathbf{g}$. This comes from combining both the firm's and household's first-order conditions with respect to labor and doing a lot of algebra. The thing to notice here is that in the model with aggregate shocks, the wage does not depend directly on policy functions. This wasn't the case when solving for the steady state because χ was not yet determined. Once we have the wage, we can get labor demand from the labor market clearing condition, and we can use the policy functions to get aggregate consumption, and thus the SDF.

Lastly, define the residual function as:

$$\mathbf{F}(\mathbf{y}(t), \dot{\mathbf{y}}(t), \mathbf{z}(t), \boldsymbol{\eta}(t)) = \begin{bmatrix} \boldsymbol{\pi}(t) + \mathbf{A}(t)\mathbf{v}(t) + \dot{\mathbf{v}}(t) - \boldsymbol{\eta}_{\mathbf{v}}(t) - \rho\mathbf{v}(t) \\ \mathbf{A}(t)^T\mathbf{g}(t) - \dot{\mathbf{g}}(t) \\ \Lambda(t) - \left(C(t) - \chi\frac{N(t)^{1+\varphi}}{1+\varphi}\right)^{-\sigma} \\ -(1-\rho_z)\log Z(t)dt + \sigma_z\mathbf{z}(t) - \dot{\log} Z(t) \end{bmatrix} \tag{7}$$

In this form of the model, a steady state is a vector $\mathbf{y}^*$ such that $F(\mathbf{y}^*, \mathbf{0}, \mathbf{0}, \mathbf{0}) = \mathbf{0}$. We have already found this when we solved for the steady state: $\mathbf{y}^* = (\mathbf{v}^*, \mathbf{g}^*, \Lambda^*, 0)$.

The first step in solving for the aggregate dynamics is to take a first-order Taylor expansion of (7) around the steady state. This gives:

$$\mathbf{F}_1^*(\mathbf{y}(t) - \mathbf{y}^*) + \mathbf{F}_2^*\dot{\mathbf{y}}(t) + \mathbf{F}_3^*\mathbf{z}(t) + \mathbf{F}_4^*\boldsymbol{\eta}(t) \approx \mathbf{0} \tag{8}$$

Where $\mathbf{F}_1^*, \mathbf{F}_2^*, \mathbf{F}_3^*$, and $\mathbf{F}_4^*$ are the partial derivative matrices of $\mathbf{F}$ evaluated at the steady state. Computationally, the Jacobian is found by using an automatic differentiation routine.[7] For large systems like this one, automatic differentiation tends to be more efficient and more accurate than using a finite-difference algorithm.

The next step is to solve (8) with `gensys`. I will not discuss the algorithm in detail here because it is well-known and code is widely available. In short, the method posits a solution of the form:

$$\dot{\mathbf{y}}(t) \approx \mathbf{G}_1\mathbf{y}(t) + \boldsymbol{\Theta}\mathbf{z}(t) \tag{9}$$

and $\mathbf{G}_1$ and $\boldsymbol{\Theta}$ are returned by the algorithm. We now have an equation governing the evolution of the endogenous variables in terms of some known coefficient matrices, today's endogenous variables, and today's aggregate shock. This makes simulation of the model straightforward. Choose a size of the time step, Δ_t and approximate the time derivative by $\dot{\mathbf{y}}(t) \approx \frac{\mathbf{y}(t+\Delta_t)-\mathbf{y}(t)}{\Delta_t}$ and $\mathbf{z}(t)$ by $\mathbf{z}(t) = \frac{\sqrt{\Delta_t}\varepsilon(t)}{\Delta_t}$ where $\varepsilon(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Plugging this into (9) we get:

$$\mathbf{y}(t + \Delta_t) \approx (\Delta_t\mathbf{G}_t + \mathbf{I})\mathbf{y}(t) + \boldsymbol{\Theta}\sqrt{\Delta_t}\varepsilon(t) \tag{10}$$

so to simulate, all one needs to do is draw a sequence of shocks and iterate on (10).

4 Results

Now I report results regarding the policy functions, steady state distributions, and impulse responses to a TFP shock. The latter makes use of the solution to the model with aggregate dynamics, whereas the former two pertain to the steady state solution of the model.

A quick note on calibration: the parameter values come from Khan and Thomas (2008). They were chosen to match the firm investment distribution from Cooper and Haltiwanger (2006). Table 1 provides a summary of the economic and computational parameters.

[7] I used the package ForwardDiff.jl. See: `https://github.com/JuliaDiff/ForwardDiff.jl`

Economic	risk-aversion coefficient	σ	2
	discount rate	ρ	0.01
	Frisch elasticity	φ	0.5
	disutility of labor	χ	2.21
	returns to capital	θ	0.21
	returns to labor	ν	0.64
	depreciation rate	δ	0.025
	fixed adjustment cost	χ_0	0.001
	variable adjustment cost	χ_1	2
	autocorrelation of TFP shocks	ρ_z	0.95
	standard deviation of TFP shocks	σ_z	0.007
	sizes of idiosyncratic shocks	$[\varepsilon_L, \varepsilon_H]$	$[0.9, 1.1]$
	intensities of idiosyncratic shocks	$[\lambda_L, \lambda_H]$	$[0.25, 0.25]$
Computational	capital grid size	N	100
	tolerance level for value functions and wage		1e-6
	step size for implicit method	Δ	10000
	minimum capital value	k_1	3.17
	maximum capital value	k_N	4.76
	capital grid step size	Δ_k	0.016
	time step size	Δ_t	0.1
	number of simulation periods		200

Table 1: Calibration

4.1 Steady State Decision Rules and Distributions

The left panel of Figure 1 shows the investment policy functions for firms with both the high and low productivity levels. Firms with high productivity invest more than do low productivity firms because they receive higher profits, and thus are more willing to face the investment costs. Low productivity firms do not invest at all above a certain level of capital. These firms already have a high enough stock of capital such that they do not find it optimal to incur the fixed cost of investment.

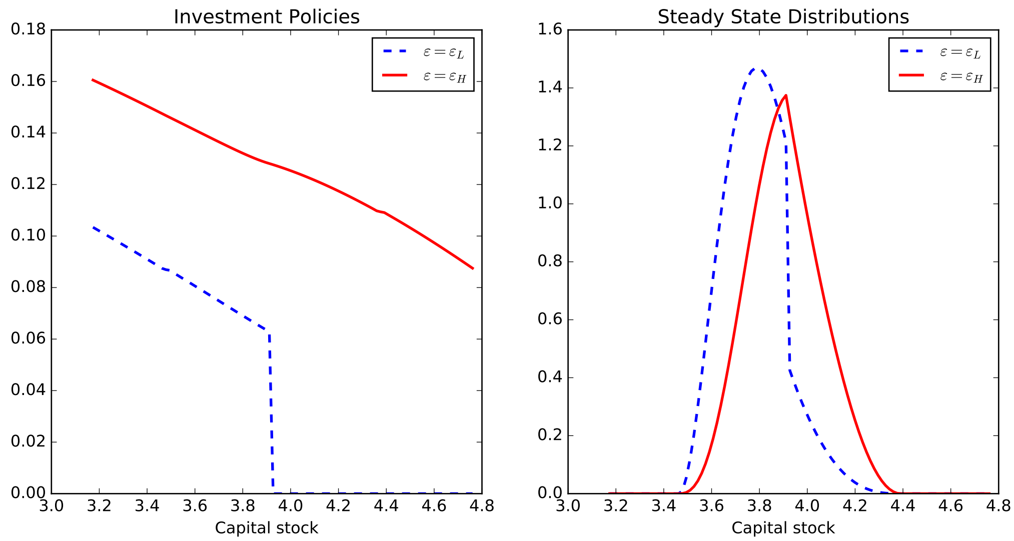

Figure 1: Policy functions and distributions

We can see some of the consequences of these decision rules come through in the shapes of the steady state distributions, in the right panel of Figure 1. Unsurprisingly, low productivity firms in general have smaller capital stocks than do high productivity firms. At the point at which low productivity firms no longer invest, there is a kink in the distribution. The only firms above that point are ones who received a low productivity shock; conditional on still being low productivity, they will immediately move leftwards in the distribution because they do not invest and are subject to capital depreciation. We see a kink at this point for the high productivity firms as well because of the low productivity, non-investing firms who get a high productivity shock.

4.2 Impulse Responses

Now we move onto the implications of the model with the aggregate TFP shocks. Figure 2 plots the impulse responses of the aggregate variables to a positive TFP shock. A higher level of TFP raises output and profits for the firm, encouraging investment. It also results in higher labor demand and increased wages. As the economy goes back to steady state, we see a discontinuous drop in investment. This happens when low productivity firms with high enough capital levels stop investing. We see no such drop in output, which leads to a discontinuous increase in consumption at this point. Investment is quickest to return to steady state, in contrast to capital which is the slowest-moving.

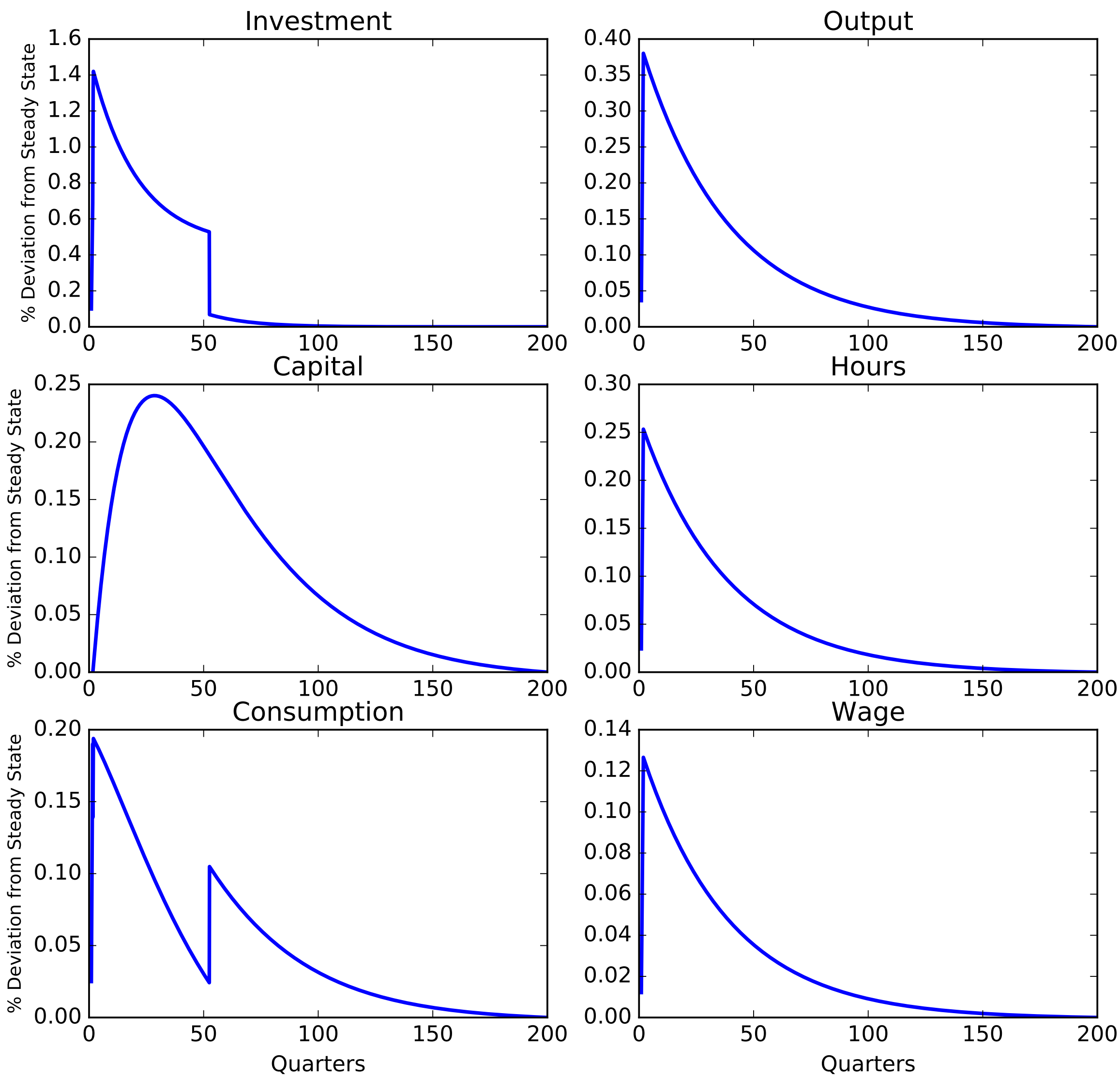

Figure 2: Impulse response functions

References

[1] Yves Achdou, Jiequn Han, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll. Heterogeneous agent models in continuous time. *Working Paper*, 2015.

[2] SeHyoun Ahn, Greg Kaplan, Benjamin Moll, and Thomas Winberry. There's no more excuses! a toolbox for solving heterogeneous agent models with aggregate shocks. *Working Paper*, 2016.

[3] Russell W Cooper and John C Haltiwanger. On the nature of capital adjustment costs. *The Review of Economic Studies*, 73(3):611–633, 2006.

[4] Aubhik Khan and Julia K Thomas. Idiosyncratic shocks and the role of nonconvexities in plant and aggregate investment dynamics. *Econometrica*, 76(2):395–436, 2008.

[5] Per Krusell and Anthony A Smith, Jr. Income and wealth heterogeneity in the macroeconomy. *Journal of Political Economy*, 106(5):867–896, 1998.

[6] Christopher A Sims. Solving linear rational expectations models. *Computational economics*, 20(1):1–20, 2002.

[7] Thomas Winberry. A toolbox for solving and estimating heterogeneous agent macro models. *Working Paper*, 2015.