BLS WORKING PAPERS

U.S. Department of Labor
U.S. Bureau of Labor Statistics
Office of Prices and Living Conditions

Solving the Incomplete Markets Model With Aggregate Uncertainty: The Method of Mixtures

Erick Sager, U.S. Bureau of Labor Statistics

Working Paper 487
March 2016

Solving the Incomplete Markets Model With Aggregate Uncertainty: The Method of Mixtures

Erick Sager*

Bureau of Labor Statistics

March 6, 2016

Abstract

In this paper I propose a new method for computing equilibrium in economies featuring heterogeneous agents, incomplete markets and aggregate uncertainty. The new method approximates the endogenous joint distribution of wealth and income by replacing stochastic simulation procedures with *iteration* on distribution functions. By construction, the approximate distribution satisfies an *intratemporal consistency condition* that imposes stationarity on both the distribution and the law of motion for aggregate state variables. I show that the Method of Mixture Distributions is capable of obtaining a solution faster than existing computational algorithms while attaining a high level of accuracy. Lastly, I provide an extension of the algorithm for computing equilibrium in an economy with non-trivial market clearing, showing that the algorithm is suitable for computing models in which prices cannot be forecasted by a finite set of moments from the distribution.

Keywords: Idiosyncratic Risk, Business Cycles, Numerical Methods

JEL Classification Codes: C63, E21, E32

*I would like to thank V.V. Chari, Larry Jones, Aubhik Khan, Felix Kübler, Lilia Maliar, Serguei Maliar, Toshihiko Mukoyama, Christopher Phelan, Michael Reiter, Víctor Ríos-Rull, Kjetil Storesletten, David Wiczer and Eric Young for their comments and insights. I would also like to thank participants of the Midwest Macroeconomics Conferences at WUSTL and the University of Missouri, Society for Computational Economics and Finance Conference in Taipei and the World Congress of the Econometric Society in Montreal. The views expressed herein are those of the author and not necessarily those of the Bureau of Labor Statistics or the U.S. Department of Labor. All errors are my own.

1 Introduction

This paper develops a new method for computing equilibrium in models with heterogeneous agents, incomplete markets and aggregate uncertainty. I illustrate this new method's mechanics within the context of Krusell and Smith's (1998) production economy. I compare this paper's computational method to other prominent methods and find that it is faster and approximately as accurate.

In the presence of aggregate uncertainty, heterogeneous agent macroeconomic models with incomplete markets are notoriously difficult to compute. These economies are not generically Pareto efficient[1] and therefore solving for equilibria requires explicitly computing prices. Furthermore, in models with a continuum of heterogeneous agents, such as Krusell and Smith's (1998) model, model inhabitants must infer future prices from the entire distribution of individual actions. Because this distribution is an infinite dimensional object, including it as a state variable is computationally intractable.

The method developed in this paper overcomes these computational hurdles by incorporating three main features. First, it follows previous work in reducing the size of the state space by replacing the distribution with a finite set of distributional moments. Second, it follows Reiter (2010) in non-parametrically representing the distribution as a function of a finite set of its moments. And third, this paper extends Young's (2010) procedure for non-stochastically updating distributions by developing a new method for constructing an ergodic set of distributions by iteration instead of simulation.

In effect, this third feature provides a new technique for constructing a distribution from its moments and it is the crux of this paper. Conceptually, I construct the set of distributions that would arise endogenously if model inhabitants could compute future prices from a finite set of distributional moments. In such a counterfactual environment, distributions and the law of motion for moments must be mutually consistent: future distributions can be computed from the law of motion for their moments, and likewise future moments can be computed from the contemporaneous distribution. It is precisely this consistency that I exploit to iteratively construct the *ergodic set of distributions*.

The numerical implementation of this method constructs distributions by iterating on a fixed point operator. At each step of the procedure, this operator enforces the mutual

[1] Dávila, Hong, Krusell, and Ríos-Rull (2012) characterize the source of inefficiency in this class of models, both theoretically and quantitatively.

consistency between distributions and the law of motion for moments. At a stationary point, the set of distributions that characterize wealth and income before individuals engage in market activity will equal the set of distributions that attain after markets clear. Consistency is enforced when the operator constructs a new iteration of distributions as mixtures of distributions from the previous iteration. Hence this computational method is called the "Method of Mixture Distributions."

The Method of Mixture Distributions has an important advantage over Krusell and Smith's (1998) original computational algorithm: the Method of Mixture Distributions replaces the simulation of distributions with a faster iterative procedure that constructs an ergodic set of distributions on which the law of motion for moments can be computed directly. Put differently, the Method of Mixture Distributions computes the ergodic set of distributions at grid points on the aggregate state space and directly computes the relationship between aggregate moments today and tomorrow from the policy function and distributions. More generally, constructing mixture distributions can provide an alternative to simulation in a large class of computational methods. I provide one such example in the final section of this paper, in which I apply the Method of Mixture distributions to an endowment economy.

In a series of numerical tests, I compare the Method of Mixture Distributions to alternative methods featured in the *Journal of Economic Dynamic's* symposium (see Den Haan, Judd, and Juillard (2010)). Relative to those methods, I find that the Method of Mixture Distributions is over 20 times faster and is just as accurate.

Finally, in terms of programming complexity, this method is fairly simple to code. This paper's algorithm is very similar to Aiyagari's (1994), except (i) it generalizes the computation of distributions, and (ii) computes a multi-dimensional fixed point rather than a one-dimensional fixed point. While these two differences are important for tackling the problem at hand, they do not add much in terms of programming time.

2 Method of Mixture Distributions

In this section I will discuss the key features of my computation algorithm, define the computational problem, and lastly provide details on the numerical implementation of the algorithm. Whereas the first part of this section provides a heuristic characterization,

it is in the final part of this section that I more rigorously define the key features of the computational method.

2.1 Key Features

Near Rationality I assume that agents do not observe the entire joint distribution over wealth and income. Instead, I assume that agents observe a finite set of moments from the distribution and make decisions using this restricted information set. Therefore, in order to forecast future prices, agents must forecast the future values of distribution moments. This assumption of *near rationality* has become a standard device in computational algorithms for reducing the dimensionality of the state space.[2]

Approximate Distributions In order to check whether market clearing conditions are satisfied, I must compute aggregate outcome that are implied by individual policy functions. Put in the context of Krusell and Smith's (1998) economic environment (to be explicated in section the next section), I must check whether individual behavior is consistent with the aggregate law of motion for capital, and I must compute the level of aggregate capital that is implied by the policy function for savings. However, in order to compute aggregate capital, I must know the joint distribution over wealth and income. Following Reiter (2010), I approximate the true joint distribution using a mapping from the aggregate state space to the space of distributions.[3]

Intratemporal Consistency A novel feature in this computational algorithm is the choice of mapping from the aggregate state space to distributions. Conceptually, I choose the mapping that would endogenously arise under the counterfactual assumption that the true economy in fact only required a set of moments to perfectly characterize the distribution. In such a counterfactual economy, the law of motion for the distribution is uniquely determined from the law of motion for the aggregate state variables.

More concretely, I consider the set of mappings that have the following *moment consistency condition*: if I choose a set of moments from the aggregate state space, then the

[2]See Den Haan (1996), Krusell and Smith (1998) and Castañeda, Díaz-Giménez, and Ríos-Rull (1998) for early examples. There are too many subsequent uses of this assumption to provide a comprehensive list here.

[3]Reiter (2010) calls this mapping a "distribution selection function." He approximates the distribution by defining a metric for the distance between distributions and finding the closest distribution (under the metric) to the steady state distribution without aggregate uncertainty. The solution is chosen from the set of distributions that have the desired moments.

mapping generates a distribution with the selected set of moments. In this way, the mapping allows us to condition the family of constructed distributions on a finite set of moments.

However, an additional condition is needed to ensure that the endogenous law of motion for distributions generates subsequent distributions that also satisfy the moment consistency condition. I call this additional condition the *intratemporal consistency condition*: if the distribution at the beginning-of-period satisfies the moment consistency condition then so will the new distribution at the end-of-period, that is generated by agents' savings decisions.

Put together, the two conditions ensure that the distribution and the law of motion for the aggregate state are uniquely determined by a finite set of moments. Furthermore, when the conditions hold, the law of motion for the distribution's moments will induce an end-of-period distribution that also satisfies the moment consistency condition.[4]

2.2 Approximate Equilibrium

Environment To build notation, we will define the approximate equilibrium that will be computed. For the sake of parsimony, I will only consider recursive competitive equilibrium and leave issues of existence and uniqueness aside.[5] For a richer description of the economic environment, see Den Haan, Judd, and Juillard (2010).

The economy consists of a continuum of individuals and a representative firm. Time is discrete and infinite.

Individual agents have preferences over consumption described by $u(c) = c^{1-\sigma}/(1-\sigma)$, for $\sigma > 1$. In the recursive problem, let the individual agent's state vector be (a, ε, k, z), which consists of individual savings $a \in \mathcal{A}$, individual employment status $\varepsilon \in \mathcal{E}$, ag-

[4]While Reiter (2010) uses the moment consistency condition on distributions, he only uses it to define the set of distributions in his minimization routine (see footnote 3). He then assumes that the set of distributions is ergodic and computes the law of motion that is induced by policy functions and the approximate set of distributions. Instead, my method simultaneously computes distributions and the laws of motion for moments, without such a minimization procedure. Instead my method iterates on the law of motion for moments and constructs a set of distributions that satisfy the two consistency conditions at each step of the iteration.

[5]Miao (2006) proves the existence of a sequential equilibrium of an environment that generalizes the Krusell and Smith (1998) economy. However, Miao (2006) cannot prove that his sequential formulation yields Krusell and Smith's (1998) wealth-recursive representation.

gregate capital $k \in \mathcal{K}$ and the aggregate productivity shock $z \in \mathcal{Z}$. The space of assets is given by $\mathcal{A} = [\underline{a}, \infty)$ where the lower bound is due to a constraint on how much an individual agent may borrow in any period. Employment status is a binary variable, indicating that an agent is either employed ($\varepsilon = 1$) or unemployed ($\varepsilon = 0$). Likewise, the aggregate productivity shock only takes on two values, $\mathcal{Z} = \{\underline{z}, \bar{z}\}$ with $\underline{z} < \bar{z}$. Lastly, the space of aggregate capital is $\mathcal{K} = [0, \infty)$.

Implicit in this definition of the state vector is the assumption that the first moment of the distribution, k, is sufficient to characterize the distribution. This assumption is made to simplify explication of the method, but could easily be relaxed by adding additional moments to the state vector.

The representative firm operates a constant returns to scale production technology, taking aggregate capital and labor as inputs at given factor prices. Let the production technology take the Cobb-Douglas form $zk^{\alpha}l(z)^{1-\alpha}$, for $\alpha \in (0,1)$. Aggregate labor $l(z)$ is a given function of z, while gross rents, $R(k,z)$, and the wage rate, $w(k,z)$, are given functions of (k,z). Implicit in the rental rate is the depreciation of aggregate capital, denoted $\delta \in (0,1)$.

Stochastic state variables evolve according to a four-state Markov chain, in which the distribution of idiosyncratic shocks changes with the aggregate shock, and the aggregate shock evolves independently. The conditional probability that the aggregate shock z' is realized next period given this period's shock is z is denoted $\pi_z(z'|z)$. The joint conditional probability that an agent realizes individual shock ε' and aggregate shock z' next period given today's individual and aggregate shocks are (ε, z) is denoted $\pi(\varepsilon', z'|\varepsilon, z)$.

Equilibrium Now define a *near rational recursive competitive equilibrium* as a value function $v : \mathcal{A} \times \mathcal{E} \times \mathcal{K} \times \mathcal{Z} \to \mathbb{R}$, a policy function $g : \mathcal{A} \times \mathcal{E} \times \mathcal{K} \times \mathcal{Z} \to \mathcal{A}$, a law of motion for aggregate capital $G : \mathcal{K} \times \mathcal{Z} \to \mathcal{K}$, indexed distributions[6] $F : \mathcal{B} \times \mathcal{K} \times \mathcal{Z} \to [0,1]$ and prices $w : \mathcal{K} \times \mathcal{Z} \to \mathbb{R}_+$ and $R : \mathcal{K} \times \mathcal{Z} \to \mathbb{R}_+$ such that

(i) Given (w, R, G), v solves the Bellman equation:

$$v(a, \varepsilon, k, z) = \max_{c, a' \in \Gamma(a, \varepsilon, k, z)} u(c) + \beta \sum_{\varepsilon'} \sum_{z'} \pi(\varepsilon', z'|\varepsilon, z) v(a', \varepsilon', k', z')$$

$$\Gamma(a, \varepsilon, k, z) = \left\{ c \geq 0, a' \geq \underline{a} \;\middle|\; c + a' \leq w(k,z)\varepsilon + R(k,z)a \right\}$$

[6] As is standard, $\mathcal{B}$ denotes the Borel σ-algebra on $\mathcal{A} \times \mathcal{E}$.

$$k' = G(k, z)$$

with associated policy function for savings: g.

(ii) Given (w, R), firms choose (k, l) optimally by setting marginal products to factor prices:

$$w(k, z) = (1 - \alpha)\frac{zk^{\alpha}l(z)^{1-\alpha}}{l(z)}$$

$$R(k, z) = \alpha\frac{zk^{\alpha}l(z)^{1-\alpha}}{k} + (1 - \delta)$$

(iii) Asset and labor markets clear:

$$k = \sum_{\varepsilon} \int_{\mathcal{A}} a \; F(da, \varepsilon; k, z)$$

$$l(z) = \sum_{\varepsilon} \int_{\mathcal{A}} \varepsilon \; F(da, \varepsilon; k, z)$$

(iv) and; (g, G, F) satisfy the *intratemporal consistency condition*:

$$F(\hat{a}, \varepsilon; \hat{k}, z) = \int_{\mathcal{K}} \mathbb{1}\left[G(k, z) = \hat{k}\right] \left(\int_{\mathcal{A}} \mathbb{1}\left[g(a, \varepsilon, k, z) = \hat{a}\right] F(da, \varepsilon; k, z) \right) dk$$

Discussion A brief discussion of the moment and intratemporal consistency conditions is warranted. Notice that the moment consistency condition is built into the definition of asset market clearing. That is, given a level of capital demanded by the representative firm (k), the asset market clearing condition imposes a condition on the distribution. That condition is the moment consistency condition: the first moment of the indexed distribution $F(a, \varepsilon; k, z)$ must equal its index moment, k.

Next, the intratemporal consistency condition simply states that the distribution is generated by the agents' optimal savings decisions and the aggregate law of motion for aggregate capital. The condition defines a fixed point operator on the distribution, which ensures that the beginning-of-period distribution can be recovered from the end-of-period distribution.[7]

[7] To see this, the distribution on the right hand side of the intratemporal consistency condition is the distribution that agents observe at the beginning of the period. After agents save resources for next period, in a way that is consistent with the law of motion for aggregate capital, there is a new distribution over

Put together, the moment and intratemporal consistency conditions ensure that both the beginning-of-period and end-of-period distributions can be indexed by a finite set of their moments. Therefore, if the intratemporal consistency condition is satisfied, then the moment consistency condition must hold on the end-of-period distribution. The first moment of the end-of-period distribution is therefore equal to the level of capital implied by the law of motion for capital:

$$G(k,z) = \sum_{\varepsilon} \int_{\mathcal{A}} g(a,\varepsilon,k,z) F(da,\varepsilon;k,z)$$

which, by the moment consistency condition, means that next period's distribution over (a,ε) is given by $F(a,\varepsilon;G(k,z),z')$ for any z'.

2.3 Numerical Implementation

Heuristic Algorithm The algorithm is organized into three main steps at each iteration.

- **Step 0:** ***(Iteration*** 0***)***
  Guess an initial aggregate law of motion for capital, $G_0(k,z)$.
- **Step 1:** ***(Iteration*** n***)***
  Given $G_n(k,z)$, compute the optimal policy function, $g(a,\varepsilon,k,z|G_n)$.
- **Step 2:** ***(Iteration*** n***)***
  Given $G_n(k,z)$ and $g(a,\varepsilon,k,z|G_n)$, compute the intratemporally consistent distributions, $F(a,\varepsilon;k,z|G_n)$.
- **Step 3:** ***(Iteration*** n***)***
  Check whether the convergence criterion is satisfied. If so, stop. Otherwise, update the guess on the aggregate law of motion to obtain $G_{n+1}(k,z)$ and return to **Step 1** ***(Iteration*** $n+1$***)***.

In the remainder of the section, I will define the convergence criterion and solver choice for the aggregate law of motion for capital in the outer loop (Step 3), discuss the solution method for the policy function (Step 1)and then finally provide details about computing

savings. This end-of-period distribution is given by the left hand side of the intratemporal consistency condition.

the intratemporally consistent distributions (Step 2). I spend the most time providing details on distributions, as this is the novel component of the computational method.

Preliminaries I select gridpoints in both the a and k directions. The grid over savings contains n_a discrete nodes and the grid over aggregate capital contains n_k discrete nodes. The grids over savings and aggregate capital are denoted, respectively, by:

$$\mathbb{G}_a \triangleq \{a_1, a_2, \ldots, a_{n_a}\} \subseteq [\underline{a}, \bar{a}] \subset \mathcal{A}$$

$$\mathbb{G}_k \triangleq \{k_1, k_2, \ldots, k_{n_k}\} \subseteq [\underline{k}, \bar{k}] \subset \mathcal{K}$$

Furthermore, define the grids over the stochastic state variables as $\mathbb{G}_\varepsilon \triangleq \{0, 1\}$ and $\mathbb{G}_z \triangleq \{\underline{z}, \bar{z}\}$ for the idiosyncratic and aggregate shocks, respectively.

Computing the Law of Motion The objective of the numerical algorithm is to find a function $G : \mathcal{K} \times \mathcal{Z} \to \mathcal{K}$, approximated by a continuous piecewise linear function on discrete collocation points $\mathbb{G}_k \times \mathbb{G}_z$, that satisfies the moment and intratemporal consistency conditions (up to the required precision). Accordingly, at each iteration, check whether the aggregate law of motion satisfies the convergence criterion:

$$\sup_{k,z} |e_n(k, z)| < \epsilon \quad \text{for} \quad \epsilon > 0$$

where for all $(k, x) \in \mathbb{G}_k \times \mathbb{G}_z$ the residual function at iteration n is given by:

$$e_n(k, z) \triangleq G_n(k, z) - \sum_{\varepsilon} \sum_{a} g(a, \varepsilon, k, z | G_n) f(a, \varepsilon; k, z | G_n)$$

and $f(a, \varepsilon; k, z | G_n)$ is the density function (to be further discussed below) associated with distribution $F(a, \varepsilon; k, z | G_n)$ and a given law of motion $G_n(k, z)$. If the convergence criterion is not satisfied, update the guess for the law of motion for capital.[8]

Computing Policy Functions The inner loop consists of finding the policy function, $g(a, \varepsilon, k, z | G_n)$, that solves the individual agent's maximization problem given aggregate

[8] Following Gaspar and Judd (1997), I update the guess on the aggregate law of motion for capital using Fixed Point Iteration:

$$\begin{aligned} G_{n+1}(k, z) &= G_n(k, z) - \Delta \cdot e_n(k, z | G_n) \\ &= (1 - \Delta) G_n(k, z) + \Delta \cdot \sum_{\varepsilon} \sum_{a} g(a, \varepsilon, k, z | G_n) f(a, \varepsilon; k, z | G_n) \end{aligned}$$

where the relaxation parameter $\Delta > 0$ governs the speed and stability of convergence.

law of motion $G_n(k,z)$. There are many algorithms that can quickly and accurately compute a solution. I find that methods that operate on the Euler equation, such as variants of Carroll's (2006) endogenous gridpoint method, or Maliar and Maliar's (2013) envelope condition method are both fast and accurate in this application.[9]

Computing Distributions Given the policy function, the next step is to compute the intratemporally consistent distribution over $(a,\varepsilon) \in \mathbb{G}_a \times \mathbb{G}_\varepsilon$, for each point in the aggregate state grid $(k,z) \in \mathbb{G}_k \times \mathbb{G}_z$. Following Reiter (2010), it will be useful to work with non-parametric density functions, specified as histograms with mass on gridpoints $\mathbb{G}_a \times \mathbb{G}_\varepsilon$.

Similar to economies without aggregate uncertainty,[10] I compute densities as the fixed point of an operator on probability measures. However, in the context of the current economic environment with aggregate uncertainty, I compute a *set of densities* that are generated by the policy function and the law of motion for aggregate capital. Each density in the set is associated with a point on the aggregate state grid $(\mathbb{G}_k \times \mathbb{G}_z)$. Furthermore, the intratemporal consistency condition defines the fixed point operator in this environment. Therefore, any set of densities that satisfy the moment and intratemporal consistency conditions will be ergodic, in the sense that the operator maps a set of densities into itself. This is the set of densities that endogenously arise under the equilibrium evolution of the economy.

In order to numerically implement the fixed point operator that defines an intratemporally consistent distribution (this is Step 2 of the overall algorithm, stated at the beginning of section 2.3), I will proceed according to the following heuristic outline:

- **Step 2 :** ***(Iteration*** 0***)***
  Given G_n and $g(a,\varepsilon,k,z|G_n)$, guess an initial beginning-of-period density, denoted $f_0(a,\varepsilon;k,z|G_n)$.

- **Step 2a:** ***(Iteration*** i***)***
  Given the beginning-of-period density $f_i(a,\varepsilon;k,z|G_n)$, compute the end-of-period den-

[9] Generally, Carroll's (2006) endogenous gridpoint method is preferable for computing policy functions in Aiyagari-Bewley-Huggett economies because the programmer can avoid all root-finding procedures. In economic environments where the "inversion step" of the endogenous gridpoint method cannot be performed without a root-finding procedure, other computational methods may be preferable. See Maliar and Maliar (2013) for speed and accuracy benchmarks for fixed point iteration when computing the neoclassical growth model.

[10] See Huggett (1993) and Ríos-Rull (1997) for an extensive overview of these methods.

sity:

$$f_i'(\hat{a}, \varepsilon; k, z|G_n) \triangleq \sum_{a \in \mathbb{G}_a} \omega_a\big(\hat{a}, g(a, \varepsilon, k, z|G_n)\big) f_i(a, \varepsilon; k, z|G_n)$$

where $\hat{a} \in \mathbb{G}_a$ and the function $\omega_a : \mathbb{G}_a \times \mathcal{A} \to [0, 1]$ will be defined below.

- **Step 2b:** ***(Iteration i)***
  Given the end-of-period density $f_i'(a, \varepsilon; k, z|G_n)$, compute the first moment of the end-of-period density:

$$\Gamma_i(k, z) \triangleq \sum_{\varepsilon \in \mathbb{G}_\varepsilon} \sum_{a \in \mathbb{G}_a} a f_i'(a, \varepsilon; k, z|G_n)$$

- **Step 2c:** ***(Iteration i)***
  Given the end-of-period density $f_i'(a, \varepsilon; k, z|G_n)$, compute the beginning-of-period density for iteration $i + 1$:

$$f_{i+1}(a, \varepsilon; \hat{k}, z|G_n) = \sum_{k \in \mathbb{G}_k} \omega_k\big(\hat{k}, k, |\Gamma_i(\cdot, z)\big) f_i'(a, \varepsilon; k, z|G_n)$$

  where $\hat{k} \in \mathbb{G}_k$ and the function $\omega_k : \mathbb{G}_k \times \mathbb{G}_k \to [0, 1]$ will be defined below.

- **Step 2d:** ***(Iteration i)***
  Verify whether the density satisfies the convergence criterion, for some $\eta > 0$ small:

$$\sup_{(a,\varepsilon,k,z)} \Big| f_{i+1}(a, \varepsilon; k, z|G_n) - f_i(a, \varepsilon; k, z|G_n) \Big| < \eta$$

  If so, stop and go to **Step 3** ***(iteration n)***. If not, go to **Step 2a** ***(iteration $i + 1$)***.

Next I must define and characterize the functions ω_a and ω_k. These functions govern how a set of distributions is mapped into itself and, therefore, are crucial for constructing the ergodic set of densities.

First I characterize the function $\omega_a(a, a')$. This function governs the mapping from the set of beginning-of-period densities into the set of end-of-period densities.[11] To compute the end-of-period density, I find the level of assets today (a) for which an agent will save $\hat{a}$, and then allocate that beginning-of-period mass of agents with savings a to the end-of-period density associated with savings $\hat{a}$. However, the value of a for which

[11] The function ω_a is nearly equivalent to the transition function in standard Aiyagari-Bewley-Huggett economies (see Ríos-Rull (1997) for a detailed characterization). The main difference between ω_a and the transition function is that ω_a takes stochastic state variables (idiosyncratic and aggregate shocks) as constants, hence the nomenclature, "*intratemporal* consistency condition."

$g(a, \varepsilon, k, z|G_n) = \hat{a}$ is not necessarily on the asset grid, $\mathbb{G}_a$. In order to ensure the density is defined on gridpoints, I follow Young (2010) by introducing a lottery that probabilistically assigns mass from the beginning-of-period to end-of-period densities.[12]

Let $a' \triangleq g(a, \varepsilon, k, z|G_n)$ for some (a, ε, k, z) on the grid. For all $a_j \in \mathbb{G}_a$ with $j = 2, \ldots, n_a$, the lotteries are given by the following function:

$$\omega_a(a_j, a') \triangleq \begin{Bmatrix} \frac{a_{j+1} - a'}{a_{j+1} - a_j} & \text{if} \quad a' \in [a_j, a_{j+1}] \\ \frac{a' - a_{j-1}}{a_j - a_{j-1}} & \text{if} \quad a' \in [a_{j-1}, a_j] \\ 0 & \text{elsewhere} \end{Bmatrix}$$

In words, when the value of the policy function $g(a, \varepsilon, k, z|G_n)$ falls in between asset grid points $[a_j, a_{j+1}]$, assign complementary fractions of the mass in the beginning-of-period density to each of those nodes. That is, assign a fraction $(a_{j+1} - a')/(a_{j+1} - a_j)$ of $f_i(a, \varepsilon; k, z|G_n)$ to node $(a_j, \varepsilon; k, z)$ and a fraction $(a' - a_j)/(a_{j+1} - a_j)$ to node $(a_{j+1}, \varepsilon; k, z)$.

Next I characterize the function $\omega_k(\hat{k}, k|\Gamma(\cdot, z))$. This function governs the mapping from the set of end-of-period densities back into the set of beginning-of-period densities. To compute the beginning-of-period density, I first find the level of aggregate capital today (k) for which the law of motion for aggregate capital implies that tomorrow's aggregate capital is $\hat{k}$. However, the value of k for which $\Gamma_i(k, z) = \hat{k}$ is not necessarily on the aggregate capital grid, $\mathbb{G}_k$. This is problematic because, as before, I wish to construct density functions that are defined on gridpoints. Because densities are indexed by the *aggregate state*, we cannot use simple lotteries to redistribute mass in such a way that the constructed density has $\hat{k}$ as a first moment. To address this issue, I develop the *method of mixture distributions* as follows.

I construct a conditional distribution over end-of-period densities, $f_i'(a, \varepsilon; k, z)$. The conditional distribution probabilistically allocates *distributions* over (a, ε) to aggregate states (k, z),[13] thereby constructing the beginning-of-period density as a mixture distribution. By construction, the conditional distribution over densities will generate a mixture dis-

[12] Because this economy consists of a continuum of agents, an argument invoking the law of large numbers ensures that these lotteries do not introduce any cross-sectional sampling variation.

[13] Similar to footnote number 12, since there is a continuum of model agents, an argument invoking the law of large numbers ensures that mixture distributions do not introduce any time-series sampling variation.

tribution with the desired first moment, $\hat{k} \in \mathbb{G}_k$.

The conditional distribution function is given by $\omega_k : \mathbb{G}_k \times \mathbb{G}_k \to [0,1]$. For $j = 2, \ldots, n_k$, let $k_j \in \mathbb{G}_k$ and let $z \in \mathbb{G}_z$. The conditional distribution function is given by:

$$\omega_k\big(\hat{k}, k|\Gamma(\cdot,z)\big) \triangleq \left\{ \begin{array}{ll} \dfrac{\hat{k} - \Gamma(k_j,z)}{\Gamma(k_{j+1},z) - \Gamma(k_j,z)} & \text{if} \quad \hat{k} \in [\Gamma(k_j,z), \Gamma(k_{j+1},z)] \text{ and } k = k_{j+1} \\ \dfrac{\Gamma(k_{j+1},z) - \hat{k}}{\Gamma(k_{j+1},z) - \Gamma(k_j,z)} & \text{if} \quad \hat{k} \in [\Gamma(k_j,z), \Gamma(k_{j+1},z)] \text{ and } k = k_j \\ 0 & \text{elsewhere} \end{array} \right\}$$

In words, first find the adjacent end-of-period distributions for which $\hat{k}$ falls in between their first moments. That is, find $\{k_j, k_{j+1}\}$ for which $\hat{k} \in [\Gamma(k_j,z), \Gamma(k_{j+1},z)]$. Then assign a fraction $(\Gamma_{j+1} - \hat{k})/(\Gamma_{j+1} - \Gamma_j)$ of the end-of-period density $f_i'(a,\varepsilon;k_j,z)$ and a complementary fraction $(\hat{k} - \Gamma_j)/(\Gamma_{j+1} - \Gamma_j)$ of the end-of-period density $f_i'(a,\varepsilon;k_{j+1},z)$ to the beginning-of-period density $f_{i+1}(a,\varepsilon;\hat{k},z)$. A generic mixture distribution takes the form:

$$f_{i+1}(a,\varepsilon;\hat{k},z) = \left(1 - \frac{\hat{k} - \Gamma_j}{\Gamma_{j+1} - \Gamma_j}\right) f_i'(a,\varepsilon;k_j,z) + \left(\frac{\hat{k} - \Gamma_j}{\Gamma_{j+1} - \Gamma_j}\right) f_i'(a,\varepsilon;k_{j+1},z)$$

where $\Gamma_j \triangleq \Gamma_i(k_j,z)$ and $\Gamma_{j+1} \triangleq \Gamma_i(k_{j+1},z)$.

I will end this discussion by verifying that the mixture distribution in fact has $\hat{k}$ as a first moment.

Proposition 1: *The first moment of the mixture distribution*

$$f_{i+1}(a,\varepsilon;\hat{k},z) = \sum_{k \in \mathbb{G}_k} \omega_k(\hat{k}, k|\Gamma(\cdot,z)) f_i'(a,\varepsilon;k,z)$$

is $\hat{k}$ when the conditional distribution is $\omega_k(\hat{k}, k|\Gamma(\cdot,z))$, which is given above.

Proof: *We can verify the proposition directly. Take $\hat{k} \in [\Gamma(k_j,z), \Gamma(k_{j+1},z)]$. Define $\omega_k \triangleq (\hat{k} - \Gamma_j)/(\Gamma_{j+1} - \Gamma_j)$ for $\Gamma_j \triangleq \Gamma(k_j,z)$.*

$$\hat{k} \stackrel{?}{=} \sum_{\varepsilon \in \mathbb{G}_\varepsilon} \sum_{a \in \mathbb{G}_a} a f_{i+1}(a,\varepsilon;\hat{k},z)$$

$$\hat{k} \stackrel{?}{=} \sum_{\varepsilon \in \mathbf{G}_\varepsilon} \sum_{a \in \mathbf{G}_a} a \left[(1-\omega_k) f_i'(a, \varepsilon; k_j, z) + \omega_k f_i'(a, \varepsilon; k_{j+1}, z) \right]$$

$$\hat{k} \stackrel{?}{=} (1-\omega_k)\Gamma(k_j, z) + \omega_k \Gamma(k_{j+1}, z)$$

$$\hat{k} \stackrel{?}{=} \frac{\Gamma(k_{j+1}, z) - \hat{k}}{\Gamma(k_{j+1}, z) - \Gamma(k_j, z)} \cdot \Gamma(k_j, z) + \frac{\hat{k} - \Gamma(k_j, z)}{\Gamma(k_{j+1}, z) - \Gamma(k_j, z)} \cdot \Gamma(k_{j+1}, z)$$

$$\hat{k} \stackrel{\checkmark}{=} \hat{k}$$

Therefore the conditional distribution ω_k over densities $f_i'(a, \varepsilon; k, z)$ induces a beginning-of-period distribution with the desired first moment.■

Initialization The algorithm requires initial guesses on both the distribution and the aggregate law of motion for capital. A good initial guess on the aggregate law of capital should satisfy the following properties: (i) $G(k, z)$ is strictly increasing in both k and z, (ii) $G(k, z)$ is (weakly) concave in k, (iii) there exist $\bar{k}$ and $\underline{k}$ such that $G(\bar{k}, \bar{z}) = \bar{k}$, $G(\underline{k}, \underline{z}) = \underline{k}$ and $\bar{k} > \underline{k}$.

An initial guess on distributions is obtained by a variant of Reiter's (2010) method. In effect, I compute the steady state distribution (without aggregate uncertainty), then I choose a desired first moment from the grid, and finally I normalize the domain of the distribution so that the transformed distribution has the desired first moment.

3 Numerical Tests

In this section I compare the Method of Mixture Distributions to alternative methods. Comparisons focus on the speed and accuracy of each method. The section ends with a discussion of the differences between each computational method.

3.1 Computational Details

I use the same parameter values as Den Haan et al. (2010) in their symposium on computing the incomplete market model with aggregate uncertainty. These parameter values are standard in the literature.

In the following numerical tests, I will compete different computational methods against each other. In order to facilitate a direct comparison across methods, I will use the same interpolation method, the same discretization of state variables (e.g. grids) and the same convergence criteria. In particular, only linear interpolation is used. Furthermore, I use the same grids for each method. The asset grid is divided into $n_a = 250$ non-uniformly spaced nodes on the interval $[\underline{a}, 200]$, with the following polynomial rule[14] governing node placement:

$$a_j = \underline{a} + \left(\frac{j-1}{n_a - 1}\right)^2 (200 - \underline{a}) \qquad \forall\, j = 1, 2, \ldots, n_a$$

The aggregate capital grid is divided into $n_k = 12$ uniformly spaced nodes on the interval $[33, 45]$. All programs were run on a 3.0 GHz Intel Core i7-3450M processor. No programs were parallelized despite multi-core processing capability. All code was written in Matlab R2014a.

3.2 Results: Computational Time

In order compare computational speed across methods, I will consider two experiments. The reported times are the average across 50 runs of an algorithm.

Experiment 1 In the first experiment, taken from Den Haan et al.'s (2010) JEDC symposium, I compute a solution when the coefficient of relative risk aversion equals $\sigma = 1.1$. Then I use this solution as a starting guess for solving the model with a CRRA coefficient equal to $\sigma = 1$. In Table 1, I have reproduced Table 2 in Den Haan (2010), which reports the results of this experiment across each method in the JEDC symposium. Since Den Haan and Rendahl's (2010) method of "Explicit Aggregation" was the fastest, I have normalized its speed to unity and provided the relative speeds for the other methods. We observe that Reiter's (2010) method, on which this paper's method builds, is the second fastest. Young (2010) performs an updated version of Krusell and Smith's (1998) algorithm, which is 44 times slower than Den Haan and Rendahl's (2010) Explicit Aggregation algorithm.

Now I run numerical experiment 1 on my method, Den Haan and Rendahl's (2010) method and Young's (2010) version of the Krusell and Smith (1998) algorithm, denoted

[14] See Maliar et al. (2010) for more details about this node placement method.

Table 1: Computational Time for JEDC Symposium

Method	Time, normalized*
Den Haan and Rendahl (2010)	1
Reiter (2010)	6.7 times slower than Explicit Aggregation
Young (2010)	44 times slower than Explicit Aggregation
Maliar, Maliar, and Valli (2010)	46 times slower than Explicit Aggregation
Algan, Allais, and Den Haan (2010)	391 times slower than Explicit Aggregation

*Times normalized to Den Haan and Rendahl (2010) method.

respectively as "MIX," "XPA" and "KSY." To facilitate comparison, I coded each method using the same grids, interpolation routines, parameter values and convergence criteria. In principle, all remaining differences in computation time is due to pure differences in methods. The research reported in Table 1 was not coded in such a comparable manner, and therefore elapsed time comparisons are confounded by simple differences in programming conventions.

The results, which are reported in Table 2, show that the Method of Mixture Distributions is 20.1 times faster than Den Haan and Rendahl's (2010) Explicit Aggregation method and 118.3 times faster than Krusell and Smith's (1998) method.

Table 2: Computational Time

	Experiment 1			Experiment 2		
Method	**Inner**	**Outer**	**Total**	**Inner**	**Outer**	**Total**
KSY	42.1 sec.	135.4 sec.	177.4 sec.	87.1 sec.	210.9 sec.	298.0 sec.
XPA	27.4 sec.	2.8 sec.	30.2 sec.	149.5 sec.	13.1 sec.	163.6 sec.
MIX	1.0 sec.	0.5 sec.	1.5 sec.	2.2 sec.	1.3 sec.	4.6 sec.
KSY/MIX	42.1	270.8	118.3	36.4	175.9	62.9
XPA/MIX	27.4	5.6	20.1	51.8	9.3	27.4

Experiment 2 For the second experiment, I time the computational speed of the Method of Mixtures, the Explicit Aggregation method and the Young's (2010) algorithm under an alternate specification for initial guesses. In this experiment, initial guesses are not computed in advance of timing the methods. Instead, the length of time required to initialize guesses is also timed. This experiment contrasts with the first, in which initial guesses on policy functions and laws of motion for capital (and the distribution, for the Intratemporally Consistent Distributions) were computed before the methods were

timed.

This experiment disproportionately slows the Method of Mixtures, which computes the steady state distribution in order to provide an initial guess on the intratemporally consistent distribution. On the other hand, I assume that the Explicit Aggregation algorithm and the Krusell-Smith algorithm are given a standard guess for the initial law of motion: $G(k,z) = (1-\delta)k$. It turns out that this guess is a relatively close approximation to the solution.

Table 2 shows that the Method of Mixtures is 27.4 times faster than the Explicit Aggregation and 62.9 times faster than the Krusell-Smith algorithm. These faster computational speeds exist despite the advantages given to the Explicit Aggregation and Krusell-Smith algorithms in terms of initial guesses.

3.3 Results: Accuracy

Following the JEDC symposium (Den Haan, Judd, and Juillard (2010)), I perform three accuracy tests on the Krusell-Smith method, Explicit Aggregation method and Method of Mixtures. These tests measure the percent error in Euler equations and the aggregate law of motion for capital. The details are described in the JEDC symposium. The results are reported in Table 3, in which I present the mean (L_1), maximum (L_∞) and 99-th percentile $(\mathcal{P}_{99})$ of the simulated errors under each test.

Table 3: Accuracy Tests*

	Static Euler Equation Errors			Dynamic Euler Equation Errors			Aggregate Law of Motion Errors		
Statistic	KSY	XPA	MIX	KSY	XPA	MIX	KSY	XPA	MIX
L_1	0.0022	0.0234	0.0028	0.0106	0.0254	0.0112	0.0506	0.4811	0.1089
$\mathcal{P}_{99}$	0.0076	0.1710	0.0091	0.0315	0.1860	0.0336	0.1387	0.6815	0.3084
L_∞	0.0297	0.2424	0.0268	0.1966	0.3222	0.1875	0.1643	0.7254	0.3791

*Numerical values are statistics from the distribution of percent errors generated from a 10,000 period simulation. L_1, L_∞ and $\mathcal{P}_x$ are, respectively, the average and maximum errors in percentages and the x-th percentile of the distribution of percent errors.

Table 3 contains the accuracy results. We observe that the Method of Mixtures and the Krusell-Smith method yield nearly identical results across both static and dynamic

Euler equation errors, while the Explicit Aggregation method's errors are an order of magnitude higher. The Krusell-Smith method has the highest accuracy for the aggregate law of motion, while the Method of Mixtures generates roughly 2 times higher errors, and the Explicit Aggregation methods generates roughly twice as large errors as the Method of Mixtures.

3.4 Discussion

Simulation and Regression The classic algorithm, originally proposed by Krusell and Smith (1998) and updated by Young (2010), computes the ergodic set of distributions over a long sample path for idiosyncratic and aggregate shocks. Using the realized distributions, the algorithm computes the law of motion for aggregate capital by computing the time-series for aggregate capital from the time-series of distributions and estimating the log-linear relationship between aggregate capital today and tomorrow from the simulated data.

The Method of Mixture Distributions, on the other hand, replaces simulation with an iterative procedure. The method computes the ergodic set of distributions at grid points on the aggregate state space and directly computes the relationship between aggregate capital today and tomorrow from the policy function and distributions. Therefore the law of motion for aggregate capital can be any arbitrary function that maps an aggregate state vector today into aggregate capital tomorrow.

The iterative method proves faster than simulation-based techniques (such as Young's (2010) reformulation of Krusell and Smith (1998) or Reiter's (2010) reference distributions). Simulation in Krusell and Smith's (1998) computational method is time intensive and significantly slows down convergence.

Simulation and Reference Distributions Reiter's (2010) method and the Method of Mixture Distributions differ critically in how they update distributions. Reiter (2010) solves for the policy function and aggregate law of motion implied by a particular mapping from distributional moments to distributions. Upon obtaining a solution, his method simulates distributions implied by the policy function and uses a weighted average of these simulated distributions to provide a new mapping between distributional moments and distributions.

The Method of Mixture Distributions, on the other hand, does not use simulation at any step. As established, simulation is time consuming relative to iteration. And while both methods update distributions as weighted averages of prior distributions, the Method of Mixture Distributions imposes an additional condition on how exactly this is performed. [15]

Explicit Aggregation Den Haan and Rendahl's (2010) Explicit Aggregation method does not require computing distributions. In principle, this feature of their method should make Explicit Aggregation faster than the Method of Mixture Distributions. However, the version of Explicit Aggregation advocated by Den Haan and Rendahl (2010) requires an aggregate state space containing aggregate shocks and the total capital held by *both* unemployed and employed agents. The Method of Mixture Distributions only requires aggregate shocks and aggregate capital and therefore has a smaller aggregate state vector. It is this feature that makes the Method of Mixture Distributions the faster algorithm.[16]

4 Non-Trivial Market Clearing

In Krusell and Smith's (1998) production economy, market clearing prices can be recovered from the representative firm's optimality conditions. As a result, a solution to the model can be written in terms of a law of motion for the aggregate state variables. Instead, now consider an endowment economy,[17] in which agents trade a non-contingent bond that is in zero net supply. The market clearing price ensures that for every agent who wishes to borrow resources, there exists an agent who is willing to lend those resources. Consequently, the market clearing price must be computed using the distribution over debt position and cannot be simply recovered from a representative firm's optimality conditions.

In this section, I demonstrate how to use the Method of Mixture Distributions to compute

[15] Algan, Allais, and Den Haan (2010) also provide a mapping between distributional moments and distributions. They parameterize the distribution and, at each iteration of the algorithm, perform a root-finding procedure to update the distribution. However, the root-finder slows the overall algorithm. Furthermore, Algan et al. (2010) use Reiter's (2010) simulation method to recover "reference moments" to increase the accuracy of the parameterized distribution.

[16] Note well that (although I have not yet tested this statement) the Explicit Aggregation method may be the faster method when the state space is restricted to only aggregate shocks and aggregate capital. The next draft will explore this possibility.

[17] See Huggett (1993) or Den Haan (1997).

the set of market clearing prices in an endowment economy with aggregate uncertainty. It turns out that the method easily extends to the computation of an endowment economy. The main difference is that in the endowment economy, we know the first moment of the distribution: market clearing requires that it is zero. Therefore, we will construct mixture distributions that enforce market clearing prices at each iteration. Furthermore, as was the case for the Krusell and Smith (1998) model, the Method of Mixture Distributions can be viewed as an extension of Young's (2010) method that replaces simulation with iteration.

In the remainder of this section I will describe the algorithm for endowment economies and then report the results of speed and accuracy comparisons. The description will follow the same steps as the previous sections.

4.1 Pure Exchange Economy

Environment Agents in the endowment economy trade a non-contingent bond, $b \in \mathcal{B}$, at a price $q \in \mathcal{Q}$. Agents solve the following dynamic program:

$$v(b,\varepsilon,q,z) = \max_{(c,b')\in\Gamma(b,\varepsilon,q,z)} u(c) + \beta \sum_{\varepsilon'}\sum_{z'} \pi(\varepsilon',z'|\varepsilon,z) v(b',\varepsilon',q',z')$$

$$\Gamma(b,\varepsilon,q,z) = \left\{ c \geq 0, b' \geq \underline{b} \;\middle|\; c + qb' \leq z\varepsilon + b \right\}$$

$$q' = Q(z')$$

The associated policy function for bond holdings is denoted $g(b,\varepsilon,q,z)$ and is indexed by the aggregate endowment shock and the contemporaneous bond price. $Q : \mathcal{Z} \to \mathcal{Z}$ is the equilibrium pricing function, which agents use to forecast future bond prices.

The market clearing condition requires:

$$\sum_{\varepsilon} \int_{\mathcal{B}} b f(b,\varepsilon;z) db = 0$$

$$\sum_{\varepsilon} \int_{\mathcal{B}} g(b,\varepsilon;Q(z),z) f(b,\varepsilon;z) db = 0$$

where $f(b,\varepsilon;z)$ is the density over bond holdings and idiosyncratic income shock which is indexed by the aggregate endowment shock. The first market clearing condition re-

quires that bonds are in zero net supply at the beginning-of-period, before trade occurs. This is an equilibrium restriction on the density function. The second condition ensures that prices $q = Q(z)$ clear bond markets after trade occurs at the end-of-period, given policy functions.

4.2 Numerical Implementation

Preliminaries We will place the endogenous state variables on a bond grid $b \in \mathbb{G}_b \subseteq \mathcal{B}$ with n_b nodes and price grid $q \in \mathbb{G}_q \subseteq \mathcal{Q}$ with n_q nodes. As before, the exogenous state variables are placed on grids given by $\mathbb{G}_z$ and $\mathbb{G}_\varepsilon$.

Computing Equilibrium Prices To compute the bond prices, first guess that $Q_0(z)$ is the initial equilibrium price function. At each iteration (n), compute a new price function $Q'_n(z)$ from a given guess $Q_n(z)$. To update the guess, we again follow Gaspar and Judd (1997) in using Fixed Point Iteration:

$$Q_{n+1}(z) = Q_n(z) + \Delta \cdot \Big(Q'_n(z) - Q_n(z) \Big)$$

Iterate until $Q'_n(z)$ equals $Q_n(z)$ up to the required precision.

Computing Policy Functions Given a price function $Q_n(z)$, we will compute the policy function $g(b, \varepsilon, q, z|Q_n)$ for each bond price on the grid $q \in \mathbb{G}_q$. Agents use $Q_n(z')$ to forecast future prices and take $q \in \mathbb{G}_q$ as today's bond price. As before, I use the Endogenous Gridpoint Method.

Computing Densities I compute a density function $f(b, \varepsilon; z)$ that satisfies an intratemporal consistency condition of the following form:

$$f(\hat{b}, \varepsilon; z) = \int_{\mathcal{Q}} \mathbb{1}\left[\Gamma(q, z) = 0\right] \left(\int_{\mathcal{B}} \mathbb{1}\left[g(b, \varepsilon, q, z|Q) = \hat{b} \right] f(b, \varepsilon; z) db \right) dq$$

$$\Gamma(q, z) = \sum_{\varepsilon} \int_{\mathcal{B}} g(b, \varepsilon, q, z|Q) f(b, \varepsilon; z) db$$

This condition requires that the beginning-of-period distribution is consistent with the end-of-period distribution, in the same way that was discussed above. However, intratemporal consistency changes in one distinct way: the market clearing condition requires that the first moment of the bond-holding distribution is zero.

In order to numerically implement the fixed point operator that defines an intratemporally consistent density, I will proceed according to the following heuristic outline:

- **Step 0 :** ***(Iteration*** 0***)***
  Given Q_n and $g(b, \varepsilon, q, z|Q_n)$, guess an initial beginning-of-period density, denoted $f_0(b, \varepsilon; z|Q_n)$.

- **Step 1:** ***(Iteration*** i***)***
  Given the beginning-of-period density $f_i(b, \varepsilon; z|Q_n)$, compute the end-of-period density:
  $$f_i'(\hat{b}, \varepsilon; q, z|Q_n) \triangleq \sum_{b \in \mathbb{G}_b} \omega_b\big(\hat{b}, g(b, \varepsilon, q, z|Q_n)\big) f_i(b, \varepsilon; z|Q_n)$$
  where $\hat{b} \in \mathbb{G}_b$ and the function $\omega_b : \mathbb{G}_b \times \mathcal{B} \to [0, 1]$ will be defined below.

- **Step 2:** ***(Iteration*** i***)***
  Given the end-of-period density $f_i'(b, \varepsilon; q, z|Q_n)$, compute the first moment of the end-of-period density:
  $$\Gamma_i(q, z) \triangleq \sum_{\varepsilon \in \mathbb{G}_\varepsilon} \sum_{b \in \mathbb{G}_b} b f_i'(b, \varepsilon; q, z|Q_n)$$

- **Step 3:** ***(Iteration*** i***)***
  Given the end-of-period density $f_i'(b, \varepsilon; q, z|Q_n)$, compute the beginning-of-period density for iteration $i + 1$:
  $$f_{i+1}(b, \varepsilon; z|Q_n) = \sum_{q \in \mathbb{G}_q} \omega_q\big(q|\Gamma_i(\cdot, z)\big) f_i'(b, \varepsilon; q, z|Q_n)$$
  where the function $\omega_q : \mathbb{G}_q \to [0, 1]$ will be defined below.

- **Step 4:** ***(Iteration*** i***)***
  Verify whether the density satisfies the convergence criterion, for some $\eta > 0$ small:
  $$\sup_{(b, \varepsilon, z)} \Big| f_{i+1}(b, \varepsilon; z|Q_n) - f_i(b, \varepsilon; z|Q_n) \Big| < \eta$$
  If so, go to **Step 5:** ***(Iteration*** i***)*** and recover the price function.
  If not, go to **Step 1** ***(iteration*** $i + 1$***)***.

- **Step 5:** ***(Iteration*** n***)***
  Given $f(b, \varepsilon; z|Q_n)$ has converged, use the corresponding $\Gamma(q, z)$ to find market clear-

ing prices for each $z \in \mathbb{G}_z$. That is find $Q_n'(z)$ such that

$$\Gamma\left(Q_n'(z), z\right) = 0 \quad \forall\, z \in \mathbb{G}_z$$

Continue to updating the price function (as detailed above).

Next I must define and characterize the functions ω_b and ω_q. These functions govern how a set of distributions is mapped into itself and, therefore, are crucial for constructing the ergodic set of densities. The function ω_b is defined equivalently to ω_a in Section 2.3. The function ω_q is similar to ω_k (defined in Section 2.3), but has some important differences that will now be discussed.

The function ω_q governs the mapping from the set of end-of-period densities back into the set of beginning-of-period densities. To compute the beginning-of-period density, I first find today's bond price (q) for which the bond market clears (e.g. the distribution has a first moment equal to zero). However, the value of q for which $\Gamma_i(q, z) = 0$ is not necessarily on the bond price grid, $\mathbb{G}_q$. To find the density function that corresponds to a the market clearing bond price, I develop the *Method of Mixture Distributions* as follows.

I construct a conditional distribution over end-of-period densities, $f_i'(b, \varepsilon; q, z)$. The conditional distribution probabilistically allocates *distributions* over (b, ε) to aggregate states (z) and contemporaneous bond prices (q), thereby constructing the beginning-of-period density as a mixture distribution. By construction, the conditional distribution over densities will generate a mixture distribution with the desired first moment of zero.

The conditional distribution function is given by $\omega_q : \mathbb{G}_q \to [0, 1]$. For $j = 2, \ldots, n_q$, let $q_j \in \mathbb{G}_q$ and let $z \in \mathbb{G}_z$. The conditional distribution function is given by:

$$\omega_q(q|\Gamma(\cdot, z)) \triangleq \left\{ \begin{array}{ll} \dfrac{-\Gamma(q_j, z)}{\Gamma(q_{j+1}, z) - \Gamma(q_j, z)} & \text{if} \quad \Gamma(q_j, z) \leq 0 \leq \Gamma(q_{j+1}, z) \text{ and } q = q_{j+1} \\ \dfrac{\Gamma(q_{j+1}, z)}{\Gamma(q_{j+1}, z) - \Gamma(q_j, z)} & \text{if} \quad \Gamma(q_j, z) \leq 0 \leq \Gamma(q_{j+1}, z) \text{ and } q = q_j \\ 0 & \text{elsewhere} \end{array} \right\}$$

In words, first find the adjacent end-of-period distributions that have first moments around zero (the market clearing first moment). That is, find $\{q_j, q_{j+1}\}$ for which $0 \in [\Gamma(q_j, z), \Gamma(q_{j+1}, z)]$. Then assign a fraction $(\Gamma_{j+1})/(\Gamma_{j+1} - \Gamma_j)$ of the end-of-period den-

sity $f_i'(b, \varepsilon; q_j, z)$ and a complementary fraction $(-\Gamma_j)/(\Gamma_{j+1} - \Gamma_j)$ of the end-of-period density $f_i'(b, \varepsilon; q_{j+1}, z)$ to the beginning-of-period density $f_{i+1}(b, \varepsilon; z)$. A generic mixture distribution takes the form:

$$f_{i+1}(b, \varepsilon; z) = \left(1 - \frac{-\Gamma_j}{\Gamma_{j+1} - \Gamma_j}\right) f_i'(b, \varepsilon; q_j, z) + \left(\frac{-\Gamma_j}{\Gamma_{j+1} - \Gamma_j}\right) f_i'(b, \varepsilon; q_{j+1}, z)$$

where $\Gamma_j \triangleq \Gamma_i(q_j, z)$ and $\Gamma_{j+1} \triangleq \Gamma_i(q_{j+1}, z)$.

4.3 Numerical Tests

Computational Details I use the same parameters in Den Haan's (1997) baseline specification. The asset grid is divided into $n_b = 250$ non-uniformly spaced nodes on the interval $[\underline{b}, 10]$ according to the polynomial rule:

$$a_j = \underline{b} + \left(\frac{j-1}{n_b - 1}\right)^2 (10 - \underline{b}) \qquad \forall\, j = 1, 2, \ldots, n_b$$

where the borrowing limit is set to $\underline{b} = -1.0$. The asset price grid is divided into $n_q = 12$ uniformly spaced nodes on the interval $[0.99, 1.10]$.

Computational Time I repeat versions of the two experiments in section 3.2. In the first experiment, I compute the solution when the coefficient of relative risk aversion equals $\sigma = 3.0$. Then I use this solution as a starting guess for solving the model with a coefficient of $\sigma = 1.5$. In the second experiment, I time the computational speed of each method under standard guesses. For each method I guess the initial policy function is $b'(b, \varepsilon; q, z) = (1 - \beta)b$. For the Method of Mixtures, I must initialize distributions by computing the steady state distribution and find the implied price function. The initial price function for the other two methods is set to $Q(z) = z$.

Results are reported in Table 4. The table shows that the Krusell-Smith algorithm is 91 times and 33 times slower than the Method of Mixtures in experiments 1 and 2, respectively. With respect to the Explicit Aggregation method, computation under experiment 1 did not converge under a wide set of starting values. For experiment 2, while the Explicit Aggregation method converged quickly, we will see that the solution is inaccurate. From this set of observations, I conclude that this paper's particular implementation of the Explicit Aggregation method is not well equipped to compute the equilibria of Pure

Exchange economies. Further work is needed to determine whether another implementation may be more robust.

Table 4: Computational Time

Method	**Experiment 1 Total**	**Experiment 2 Total**
KSY	721.4 sec.	791.4 sec.
XPA	∞ sec.	154.6 sec.
MIX	7.9 sec.	24.3 sec.
KSY/MIX	91.3	32.6
XPA/MIX	∞	6.4

Accuracy Accuracy results for each of the three tests in section 3.3 are reported in Table 5. We observe that the Method of Mixtures and the Krusell-Smith method yield nearly identical accuracy results across each test, while the Explicit Aggregation method has higher errors across each test. Note that the high errors are a result of the simple specification for the bond prices. In principle, if prices were conditioned on additional aggregate state variables then agents' forecasts of future prices would be more accurate. The results in Table 5 are meant to be suggestive of the relative accuracies across methods under a common set of assumptions on primitives, not the absolute accuracy achievable.

Table 5: Accuracy Tests*

	Static Euler Equation Errors			Dynamic Euler Equation Errors			Aggregate Law of Motion Errors		
Statistic	KSY	XPA	MIX	KSY	XPA	MIX	KSY	XPA	MIX
L_1	2.3526	3.3576	2.3247	2.4461	3.3258	2.4433	0.3813	4.1632	0.3829
L_∞	4.8535	5.4582	4.8287	18.3377	21.9826	18.2313	1.2102	7.1987	1.0526

*Numerical values are statistics from the distribution of percent errors generated from a 10,000 period simulation. L_1 and L_∞ are, respectively, the average and maximum errors in percentages.

4.4 Discussion

The extension of the Method of Mixture Distributions for the endowment economy is very similar to the method as it is applied to a production economy. The main difference is that the market clearing condition for an endowment economy requires that bonds

be held in zero net supply. Therefore, the Method of Mixture Distributions constructs distributions that satisfy the market clearing condition.

Furthermore, the Method of Mixture Distributions can be thought of as an extension of Young's (2010) method. Young's (2010) method essentially guesses a market clearing set of prices as a function of the aggregate state, computes policy functions and then simulates distributions to recover the relationship between prices and aggregate states. In terms of computational intensity, simulation in an endowment economy requires finding a market clearing bond price at each period of time along the sample path. Given that simulation requires a long sample path, a large number of root-finding computations must be implemented.

The Method of Mixture Distributions replaces simulation in Young's (2010) method with iteration. Iteration on the intratemporally consistent density operator constructs densities as mixtures over conditional distributions (ω_q) in a way that induces market clearing. Therefore iteration avoids implementing root-finding computations, except upon convergence when a root-finder recovers the market clearing prices directly. Therefore, the speed gains from iteration can be large. Furthermore, this method allows for an arbitrary pricing function whereas simulation specifies a particular function and estimates its parameters.

5 Conclusion

This paper provided a new method for computing equilibrium in a large class of economies featuring heterogeneous agents, incomplete markets and aggregate uncertainty. Following the *Journal of Economic Dynamics and Control's* January 2010 symposium, I illustrated this new method's mechanics within the context of Krusell and Smith's (1998) production economy. I additionally provided an extension of the method to endowment economies, in which market clearing prices must be solved explicitly. I compared the Method of Mixture Distributions to other methods in the JEDC symposium and found that it is faster and approximately as accurate.

The Method of Mixture Distributions can be regarded as an alternative to stochastic simulation, such as Young's (2010) method, that avoids *both* cross-sectional and time-series sampling error. The Method of Mixture Distributions approximates the ergodic

set of distribution functions through an iterative procedure, instead of simulating a long sample path of distributions. As a result, the Method of Mixture Distributions is faster and does not require parametric specifications for either the law of motion for aggregate state variables (or for bond price functions in an endowment economy).

Lastly, the Method of Mixture Distributions only requires a parsimonious aggregate state vector. Therefore, relative to methods that do not utilize any sort of simulation, such as Den Haan and Rendahl's (2010) method of Explicit Aggregation, the Method of Mixture Distributions is capable of computing equilibrium faster – it does not require as many functional evaluations over the aggregate state space.

References

Aiyagari, S. R. (1994): "Uninsured Idiosyncratic Risk and Aggregate Saving," *The Quarterly Journal of Economics*, 109, 659–84.

Algan, Y., O. Allais, and W. J. Den Haan (2010): "Solving the Incomplete Markets Model with Aggregate Uncertainty Using Parameterized Cross-Sectional Distributions," *Journal of Economic Dynamics and Control*, 34, 59–68.

Anderson, C. R. and C. J. Elion (2004): "Accelerated Solutions of Nonlinear Equations Using Runge-Kutta Methods," *UCLA Computational and Applied Mathematics Report # 04-26*.

Carroll, C. D. (2006): "The Method of Endogenous Gridpoints for Solving Dynamic Stochastic Optimization Problems," *Economics Letters*, 91, 312–320.

Castañeda, A., J. Díaz-Giménez, and J.-V. Ríos-Rull (1998): "Exploring the Income Distribution Business Cycle Dynamics," *Journal of Monetary Economics*, 42, 93–130.

Chang, Y. and S.-B. Kim (2007): "Heterogeneity and Aggregation: Implications for Labor-Market Fluctuations," *American Economic Review*, 97, 1939–1956.

Dávila, J., J. H. Hong, P. Krusell, and J.-V. Ríos-Rull (2012): "Constrained Efficiency in the Neoclassical Growth Model With Uninsurable Idiosyncratic Shocks," *Econometrica*, 80, 2431–2467.

Den Haan, W. J. (1996): "Heterogeneity, Aggregate Uncertainty, and the Short-Term Interest Rate," *Journal of Business & Economic Statistics*, 14, 399–411.

——— (1997): "Solving Dynamic Models With Aggregate Shocks And Heterogeneous Agents," *Macroeconomic Dynamics*, 1, 355–386.

——— (2010): "Comparison of Solutions to the Incomplete Markets Model with Aggregate Uncertainty," *Journal of Economic Dynamics and Control*, 34, 4–27.

Den Haan, W. J., K. L. Judd, and M. Juillard (2010): "Computational Suite of Models with Heterogeneous Agents: Incomplete Markets and Aggregate Uncertainty," *Journal of Economic Dynamics and Control*, 34, 1–3.

Den Haan, W. J. and P. Rendahl (2010): "Solving the Incomplete Markets Model with Aggregate Uncertainty Using Explicit Aggregation," *Journal of Economic Dynamics and Control*, 34, 69–78.

Gaspar, J. and K. Judd (1997): "Solving Large-Scale Rational-Expectations Models," *Macroeconomic Dynamics*, 1, 45–75.

Huggett, M. (1993): "The Risk-Free Rate in Heterogeneous-Agent Incomplete-Insurance Economies," *Journal of Economic Dynamics and Control*, 17, 953–969.

Khan, A. and J. K. Thomas (2008): "Idiosyncratic Shocks and the Role of Nonconvexities in Plant and Aggregate Investment Dynamics," *Econometrica*, 76, 395–436.

Krusell, P. and A. A. Smith (1997): "Income And Wealth Heterogeneity, Portfolio Choice, And Equilibrium Asset Returns," *Macroeconomic Dynamics*, 1, 387–422.

——— (1998): "Income and Wealth Heterogeneity in the Macroeconomy," *Journal of Political Economy*, 106, 867–896.

Maliar, L. and S. Maliar (2013): "Envelope Condition Method Versus Endogenous Grid Method for Solving Dynamic Programming Problems," *Economics Letters*, 120, 262–266.

Maliar, L., S. Maliar, and F. Valli (2010): "Solving the Incomplete Markets Model with Aggregate Uncertainty Using the Krusell-Smith Algorithm," *Journal of Economic Dynamics and Control*, 34, 42–49.

Miao, J. (2006): "Competitive Equilibria of Economies with a Continuum of Consumers and Aggregate Shocks," *Journal of Economic Theory*, 128, 274–298.

Reiter, M. (2010): "Solving the Incomplete Markets Model with Aggregate Uncertainty by Backward Induction," *Journal of Economic Dynamics and Control*, 34, 28–35.

Ríos-Rull, J.-V. (1997): "Computation of equilibria in heterogeneous agent models," *Federal Reserve Bank of Minneapolis Staff Report # 231*.

Young, E. R. (2010): "Solving the Incomplete Markets Model with Aggregate Uncertainty Using the Krusell-Smith Algorithm and Non-Stochastic Simulations," *Journal of Economic Dynamics and Control*, 34, 36–41.

A Appendix: Pseudo Algorithms

(0) Obtain initial distribution

- Compute the steady state density function over $\mathcal{G}_a \times \mathcal{G}_\varepsilon$, denoted f_{ss}
- Transform f_{ss} to f_0 s.t. for each $S \equiv (k, z) \in \mathcal{G}_k \times \mathcal{G}_z$:

$$\sum_{\varepsilon} \int_{\mathcal{A}} a f_0(a, \varepsilon; k, z) = k \tag{MCC}$$

- Transformation 1 (Reiter): Rescale $\mathcal{G}_a$ so that MCC holds and then interpolate f_0 back onto grid.
- Or Transformation 2 (Sager): Compute steady state for k_L and k_H corresponding to different β and use:

$$f(a, \varepsilon; k, z) = (1 - \omega(k))\lambda_{ss}(k_L) + \omega(k)\lambda_{ss}(k_H) \quad \forall k \in \mathcal{G}_k$$

  where the MCC is guaranteed by setting weights as:

$$\omega(k) = \frac{k_H - k}{k_H - k_L}$$

(1) Compute policy function given f_i

- Guess $g = g_1(a, \varepsilon; k, z)$
- Compute

$$G(k, z|f_i) = \sum_{\varepsilon} \int_A g(a, \varepsilon; k, z) f_i(a, \varepsilon; k, z) da$$

- Compute $g_i(a, \varepsilon; k, z)$ given $G(k, z|f_i)$ via EGM

$$c(a', \varepsilon; k, z) = u_c^{-1}\left(\beta \sum_{\varepsilon', z'} \frac{\pi(\varepsilon', z'|\varepsilon, z)}{\pi(z'|z)} u_c\left(y(s'|f_i) - g(s'|f_i)\right) R(G(k, z|f_i), z')\right)$$

$$\hat{a}(a', \varepsilon; k, z) = R(k, z)^{-1} \left(c(a', \varepsilon; k, z) + a' - w(k, z)\varepsilon \right)$$

$$g(a', \varepsilon; k, z) = \Phi \left(\hat{a}(a', \varepsilon; k, z), a' | a' \in \mathcal{G}_a \right)$$

- If $g_{i+1} = g_i$, stop.

(2) Iterate once on distribution $f_i \to f_{i+1}$

- Compute

$$\hat{f}_i(a', \varepsilon; k', z) = \int_{\mathcal{K}} \int_{\mathcal{A}} 1[g_{i+1}(a, \varepsilon; k, z | f_i) = a', G(k, z | f_i) = k'] f_i(a, \varepsilon; k, z) da dk$$

  such that

$$G(k, z | f_i) = \sum_{\varepsilon} \int_{\mathcal{A}} g_{i+1}(a, \varepsilon; k, z) f_i(a, \varepsilon; k, z) da$$

- Update

$$f_{i+1}(a, \varepsilon; k, z) = (1 - \delta) f_i(a, \varepsilon; k, z) + \delta \hat{f}_i(a, \varepsilon; k, z)$$

- Return to (1)

A.1 Pure Exchange Economy

Here I outline the pseudo-algorithm for computing equilibria of the pure exchange economy studied in Den Haan (1997), as was featured in the main text.

(0) Obtain initial distribution

- Compute f_{ss}
- Transform f_{ss} to f_0 s.t. for each $z \in \mathcal{G}_z$:

$$\sum_{\varepsilon} \int_{\mathcal{B}} b f_0(b, \varepsilon; z) = 0 \qquad \text{(MCC)}$$

- Transformation 1 (Reiter): Rescale $\mathcal{G}_b$ so that MCC holds and then interpolate f_0 back onto grid.
- Or Transformation 2 (Sager): Compute steady state for z_L and z_H to initialize $f_0(b, \varepsilon; z)$.
  - Set $g_0(b, \varepsilon; q, z) = g_{ss}(b, \varepsilon; Q_{ss}(z), z) \cdot \left(1 - \frac{q}{Q_{ss}(z)}\right)^{\kappa}$, $\kappa > 0$

(1) Compute policy function given $f_i(b, \varepsilon; z)$ and initial guess $g_i(b, \varepsilon; q, z)$

- Compute
$$Q_i(z) = \left\{ q \quad \text{s.t.} \quad \sum_{\varepsilon} \int_{\mathcal{B}} g_i(b, \varepsilon; q, z) f_i(b, \varepsilon; z) db = 0 \right\}$$
- Compute $g_{i+1}(b, \varepsilon; q, z)$ given $Q_i(z)$ and $s' = (b', \varepsilon'; Q(z'), z')$ via EGM
$$c_i(b', \varepsilon; q, z) = u_c^{-1} \left(q^{-1} \beta \sum_{\varepsilon', z'} \frac{\pi(\varepsilon', z'|\varepsilon, z)}{\pi(z'|z)} u_c \left(y(s') - Q_i(z') g_i(s') \right) \right)$$
$$\hat{b}_i(b', \varepsilon; q, z) = c_i(b', \varepsilon; q, z) + qb' - z\varepsilon$$
$$g_{i+1}(b', \varepsilon; q, z) = \Phi \left(\hat{b}_i(b', \varepsilon; q, z), b' | b' \in \mathcal{G}_b \right)$$
- If $g_{i+1} = g_i$, stop.

(2) Iterate once on distribution $f_i \to f_{i+1}$

- Compute
$$\hat{f}_i(b', \varepsilon; z) = \int_{\mathcal{Q}} \int_{\mathcal{B}} 1[g_{i+1}(b, \varepsilon; q, z | f_i) = b', G_i(q, z) = 0] f_i(b, \varepsilon; z) db dq$$
such that
$$G_i(q, z) = \sum_{\varepsilon} \int_{\mathcal{B}} g_{i+1}(b, \varepsilon; q, z) f_i(b, \varepsilon; z) db$$
- Update
$$f_{i+1}(b, \varepsilon; z) = (1 - \delta) f_i(b, \varepsilon; z) + \delta \hat{f}_i(b, \varepsilon; z)$$
- Return to (1)

A.2 Production Economy with Two Moments in State Space

Consider the production economy in section 2.2. Let ν be the variance of the wealth distribution and expand the aggregate state space to include both the first and second moments of the distribution: $S \equiv (k, \nu, z)$.

Let $s = (a, \varepsilon; k, \nu, z)$. Compute four steady states corresponding to $\{k_i, \nu_i\}_{i \in \{L,H\}}$. Transform into initial guess $f_0(a, \varepsilon; k, \nu, z)$ using bilinear interpolation on distributions.

Forecasts:

$$G_i^k(k,\nu,z) = \sum_{\varepsilon} \int_{\mathcal{A}} g_i(a,\varepsilon;k,\nu,z) f_i(a,\varepsilon;k,\nu,z) da$$

$$G_i^\nu(k,\nu,z) = \sum_{\varepsilon} \int_{\mathcal{A}} g_i(a,\varepsilon;k,\nu,z)^2 f_i(a,\varepsilon;k,\nu,z) da$$

Otherwise, inner loop stays the same: $g_i \mapsto g_{i+1}$

Outer loop becomes: $f_i \mapsto f_{i+1}$

$$\hat{f}_i(a,\varepsilon;k,\nu,z) = \int_{\mathcal{K}\times\mathcal{V}\times\mathcal{A}} \Gamma_i\left(\hat{a},\hat{k},\hat{v}|a,k,\nu\right) f_i(\hat{a},\varepsilon;\hat{k},\hat{v},z) d\hat{a} d\hat{v} d\hat{k}$$

$$\Gamma_i\left(\hat{a},\hat{k},\hat{v}|a,k,\nu\right) = \mathbb{1}\left[g_{i+1}(\hat{a},\varepsilon;\hat{k},\hat{v},z) = a, \hat{G}_i^k(\hat{k},\hat{v},z) = k, \hat{G}_i^\nu(\hat{k},\hat{v},z) = \nu\right]$$

$$\hat{G}_i^k(k,\nu,z) = \sum_{\varepsilon} \int_{\mathcal{A}} g_{i+1}(a,\varepsilon;k,\nu,z) f_i(a,\varepsilon;k,\nu,z) da$$

$$\hat{G}_i^\nu(k,\nu,z) = \sum_{\varepsilon} \int_{\mathcal{A}} g_{i+1}(a,\varepsilon;k,\nu,z)^2 f_i(a,\varepsilon;k,\nu,z) da$$

$$f_{i+1}(s) = (1-\delta) f_i(s) + \delta \hat{f}_i(s)$$

where the numerical implementation of Γ contains a two dimensional root finder over (k,ν):

$$\Psi(\hat{k},\hat{v}|k,\nu,z) = \begin{bmatrix} \hat{G}_i^k(\hat{k},\hat{v},z) - k \\ \hat{G}_i^\nu(\hat{k},\hat{v},z) - \nu \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

This is the most computationally expensive step of the algorithm.

Instead of a root finder, split into two interpolation routines:

$$\hat{K}(\hat{v}) \equiv G^{k-1}(k;\hat{v})$$

$$H(\hat{v}) \equiv G^v(\hat{K}(\hat{v}),\hat{v})$$

$$\hat{v}^* = H^{-1}(\nu)$$

$$\hat{k}^* = \hat{K}(\hat{v}^*)$$

Relative to KS98 with two moments, I avoid the simulation step. However, I must solve

a two-dimensional fixed point $\Psi(\hat{k}, \hat{v}) = 0$ for each element of the aggregate state space $S = (k, \nu, z)$ at each iteration. As the number of included moments increases, it is possible that simulation will be a faster method. This is because simulation automatically generates a time series $\{k_t, \nu_t\}_{t=0}^{T}$ and N-linear regressions will be faster to compute than an N-dimensional fixed point for N sufficiently large.

$$k_{t+1} = \beta_k + \beta_{kk} k_t + \beta_{k\nu} \nu_t + \epsilon_k$$

$$\nu_{t+1} = \beta_\nu + \beta_{\nu k} k_t + \beta_{\nu\nu} \nu_t + \epsilon_\nu$$

A.3 Economy with Equity and Bonds

Following Krusell and Smith (1997), consider an economy in which agents have access to two types of assets: one-period equity and one-period debt.

Let $s = (a, b, \varepsilon; k, q, z)$. Compute four steady states and transform into initial guess $f_0(a, b, \varepsilon; k, z)$ (elaborate).

Forecasts:

$$G_i^k(k, q, z) = \sum_{\varepsilon} \int_{\mathcal{A}\times\mathcal{B}} g_i^a(a, b, \varepsilon; k, q, z) f_i(a, b, \varepsilon; k, z) da db$$

$$G_i^q(k, q, z) = \sum_{\varepsilon} \int_{\mathcal{A}\times\mathcal{B}} g_i^b(a, b, \varepsilon; k, q, z) f_i(a, b, \varepsilon; k, z) da db$$

$$Q_i(k, z) = \left\{ q \quad \text{s.t.} \quad G_i^q(k, q, z) = 0 \right\}$$

Otherwise, inner loop stays the same: $g_i \mapsto g_{i+1}$

- Define:

$$s' = \left(g_i^a(s), g_i^b(s), \varepsilon'; G_i^k(k, q, z), Q_i(k, z), z' \right)$$

$$c_i'(s') = y(s') - g_i^a(s') - Q_i(z') g_{i+1}^b(s')$$

$$y(s) = w(S)\varepsilon + R(S)a + b$$

- Compute policy functions with EEb, EEa, BC (respectively):

$$c_{i+1}(a, b, \varepsilon; k, q, z) = u_c^{-1} \left(\beta \sum_{\varepsilon', z'} \frac{\pi(\varepsilon', z'|\varepsilon, z)}{\pi(z'|z)} u_c \left(c_i'(s') \right) R(S') \right)$$

$$g^a_{i+1}(a,b,\varepsilon;k,q,z) = u_c^{-1}\left(q^{-1}\beta\sum_{\varepsilon',z'}\frac{\pi(\varepsilon',z'|\varepsilon,z)}{\pi(z'|z)}u_c\left(c'_i(s')\right)\right)\cdot\frac{g^a_i(s)}{c_{i+1}(s)}$$

$$g^b_{i+1}(a,b,\varepsilon;k,q,z) = q^{-1}\Big(y(s) - c_{i+1}(s) - g^a_i(s)\Big)$$

Outer loop becomes: $f_i \mapsto f_{i+1}$

- Update the distribution:

$$\hat{f}_i(a,b,\varepsilon;k,z) = \int_{\mathcal{K}\times\mathcal{Q}\times\mathcal{A}\times\mathcal{B}}\Gamma_i\left(\hat{a},\hat{b},\hat{k},\hat{q}|a,b,k\right)f_i(\hat{a},\hat{b},\varepsilon;\hat{k},z)d\hat{a}d\hat{b}d\hat{k}d\hat{q}$$

$$\Gamma_i\left(\hat{a},\hat{b},\hat{k},\hat{q}|a,b,k\right) = \mathbb{1}\left[\Psi^s_i(\hat{a},\hat{b},\varepsilon;\hat{k},\hat{v},z|a,b) = 0, \Psi^S_i(\hat{k},\hat{q},z|k,0) = 0\right]$$

$$f_{i+1}(s) = (1-\delta)f_i(s) + \delta\hat{f}_i(s)$$

- The numerical implementation of Γ contains a two two-dimensional root finders. The first gives the aggregate state today that delivers the aggregate state tomorrow:

$$\Psi^S_i(\hat{k},\hat{q},z|k,0) \equiv \begin{bmatrix}\hat{G}^k_i(\hat{k},\hat{q},z) - k\\ \hat{G}^q_i(\hat{k},\hat{q},z) - 0\end{bmatrix}$$

  Given the aggregate state, the second fixed point gives the individual state today that delivers the individual state tomorrow:

$$\Psi^s_i(\hat{a},\hat{b},\varepsilon;\hat{k},\hat{q},z|a,b) \equiv \begin{bmatrix}g^a_i(\hat{a},\hat{b},\varepsilon;\hat{k},\hat{q},z) - a\\ g^b_i(\hat{a},\hat{b},\varepsilon;\hat{k},\hat{q},z) - b\end{bmatrix}$$

  where

$$\hat{G}^k_i(k,q,z) = \sum_\varepsilon\int_{\mathcal{A}\times\mathcal{B}}g^a_{i+1}(a,b,\varepsilon;k,q,z)f_i(a,b,\varepsilon;k,z)dadb$$

$$\hat{G}^q_i(k,q,z) = \sum_\varepsilon\int_{\mathcal{A}\times\mathcal{B}}g^b_{i+1}(a,b,\varepsilon;k,q,z)f_i(a,b,\varepsilon;k,z)dadb$$

  This is the most computationally expensive step of the algorithm.

Alternatively to a two-dimensional root finder for the aggregate fixed point:

$$\hat{K}(\hat{q}) \equiv G^{k-1}(k;\hat{q},z)$$

$$H(\hat{q}) \equiv G^q(\hat{K}(\hat{q}), \hat{q}, z)$$

$$\hat{q}^* = H^{-1}(0)$$

$$\hat{k}^* = \hat{K}(\hat{q}^*)$$

A.4 Heterogeneous Firm Economy

Following Khan and Thomas (2008), consider an economy with heterogeneous firms that are subject to frictions that distort investment decisions.

Let $s = (a, \varepsilon; k, p, z)$. Compute steady states and transform into initial guess $f_0(a, \varepsilon; k, z)$.

Forecasts:

$$G_i^k(k, p, z) = \sum_{\varepsilon} \int_{\mathcal{A}} g_i(a, \varepsilon; k, p, z) f_i(a, \varepsilon; k, z) da$$

$$G_i^{rc}(k, p, z) = \sum_{\varepsilon} \int_{\mathcal{A}} \left[y(a, \varepsilon; k, p, z) + (1 - \delta)k - c_i(a, \varepsilon; k, p, z) \right] f_i(a, \varepsilon; k, z) da$$

$$P_i(k, z) = \left\{ p \quad \text{s.t.} \quad G_i^k(k, p, z) = G_i^{rc}(k, p, z) \right\}$$

Otherwise, inner loop stays the same: $g_i \mapsto g_{i+1}$

- Compute policy functions of Bellman equation:

$$\pi(s) = \max_n \{ z\varepsilon F(k, n) - w(k, z)n + (1 - \delta)k \}$$

$$V(s) = p\pi(s) - pw(k, z) \cdot \int_0^{\bar{\xi}} \xi G(d\xi) + G(\bar{\xi}) W(\varepsilon, k, p, z) + (1 - G(\bar{\xi})) W^c(\varepsilon, k, p, z)$$

$$W(s) = \max_{a'} \; -\gamma p a' + \beta \sum_{\varepsilon', z'} \pi(\varepsilon', z' | \varepsilon, z) V(a', \varepsilon'; G_i^k(k, p, z), P_i(G_i(k, p, z), z'), z')$$

$$W^c(s) = \max_{a' \in [\underline{\kappa} a, \bar{\kappa} a]} -\gamma p a' + \beta \sum_{\varepsilon', z'} \pi(\varepsilon', z' | \varepsilon, z) V(a', \varepsilon'; G_i^k(k, p, z), P_i(G_i(k, p, z), z'), z')$$

Outer loop becomes: $f_i \mapsto f_{i+1}$

- Update the distribution:

$$\hat{f}_i(a, \varepsilon; k, z) = \int_{\mathcal{K} \times \mathcal{P} \times \mathcal{A}} \Gamma_i \left(\hat{a}, \hat{k}, \hat{p} | a, k \right) f_i(\hat{a}, \varepsilon; \hat{k}, z) d\hat{a} d\hat{k} d\hat{p}$$

$$\Gamma_i\left(\hat{a},\hat{k},\hat{q}|a,k\right) = \mathbb{1}\left[\Psi_i^s(\hat{a},\varepsilon;\hat{k},\hat{p},z|a) = 0, \Psi_i^S(\hat{k},\hat{p},z|k,0) = 0\right]$$

$$f_{i+1}(s) = (1-\delta)f_i(s) + \delta\hat{f}_i(s)$$

- The numerical implementation of Γ contains root finders. The first gives the aggregate state today that delivers the aggregate state tomorrow:

$$\Psi_i^S(\hat{k},\hat{p},z|k,0) \equiv \begin{bmatrix} \hat{G}_i^k(\hat{k},\hat{p},z) - k \\ \hat{G}_i^{rc}(\hat{k},\hat{p},z) - \hat{G}_i^k(\hat{k},\hat{p},z) \end{bmatrix}$$

  Given the aggregate state, the second fixed point gives the individual state today that delivers the individual state tomorrow:

$$\Psi_i^s(\hat{a},\varepsilon;\hat{k},\hat{p},z|a) \equiv g_i(\hat{a},\varepsilon;\hat{k},\hat{q},z) - a$$

  where

$$\hat{G}_i^k(k,p,z) = \sum_{\varepsilon}\int_{\mathcal{A}} g_{i+1}(a,\varepsilon;k,p,z)f_i(a,\varepsilon;k,z)da$$

$$\hat{G}_i^{rc}(k,p,z) = \sum_{\varepsilon}\int_{\mathcal{A}} \left[y(a,\varepsilon;k,p,z) + (1-\delta)k - c_{i+1}(a,\varepsilon;k,p,z)\right] f_i(a,\varepsilon;k,z)da$$

  This is the most computationally expensive step of the algorithm.

A.5 Indivisible Labor and Incomplete Markets

Following Chang and Kim (2007), consider a production economy similar to Krusell and Smith (1998) in which agents receive idiosyncratic income shocks and choose whether or not to be employed each period.

Let $s = (a,\varepsilon;k,z)$ be the state vector. If an agent is employed they work a fixed number of hours, denoted $\bar{h}$, and otherwise an agent works zero hours but receives an unemployment benefit. Let the value of being employed be denoted by $v^e(a,\varepsilon;k,z)$ and the value of being unemployed be denoted by $v^u(a,\varepsilon;k,z)$. The agent's labor decision is denoted $h(a,\varepsilon;k,z) \in \{0,\bar{h}\}$ and is chosen according to:

$$v(a,\varepsilon;k,z) = \max_{h\in\{0,\bar{h}\}}\left\{\left(1-h/\bar{h}\right)v^u(a,\varepsilon;k,z) + \left(h/\bar{h}\right)v^e(a,\varepsilon;k,z)\right\}$$

Guess $\{v_0^u(s), v_0^e(s), g_0(s), f_0(s)\}$. Compute steady states and transform into initial guess $f_0(a, \varepsilon; k, z)$. Use guesses on (v^u, v^e) to compute $v(s)$ and $h(s)$.

Forecast capital and compute future prices:

$$G_i(k,z) = \sum_{\varepsilon} \int_{\mathcal{A}} g_i(a,\varepsilon;k,z) f_i(a,\varepsilon;k,z) da$$

$$l_i(k,z) = \sum_{\varepsilon} \int_{\mathcal{A}} \varepsilon h_i(a,\varepsilon;k,z) f_i(a,\varepsilon;k,z) da$$

$$r_i'(k,z,z') = \alpha z' G_i(k,z)^{\alpha-1} l_i(G_i(k,z),z')^{1-\alpha} - \delta$$

$$w_i'(k,z,z') = (1-\alpha) z' G_i(k,z)^{\alpha} l_i(G_i(k,z),z')^{-\alpha}$$

Inner loop: $g_i \mapsto g_{i+1}$, $v_i \mapsto v_{i+1}$

- Compute policy and value functions for each type, where $(\hat{g}_i, \hat{c}_i)$ are recovered from the budget constraint and Euler equation:

$$\hat{v}_i(s) = \max_{h \in \{0,\bar{h}\}} \left\{ \left(1 - h/\bar{h}\right) v_i^u(s) + \left(h/\bar{h}\right) v_i^e(s) \right\}$$

$$\hat{v}_i^u(s) = u(\hat{c}_i(a,\varepsilon;k,z), 0) + \beta \hat{v}_i(a,\varepsilon;k,z)$$

$$\hat{v}_i^e(s) = u(\hat{c}_i(a,\varepsilon;k,z), \bar{h}) + \beta \hat{v}_i(a,\varepsilon;k,z)$$

- Dampen (v^u, v^e, g)
- Recover employment decision and savings decision:

$$h_{i+1}(s) = \arg\max_{h \in \{0,\bar{h}\}} \left\{ \left(1 - h/\bar{h}\right) v_{i+1}^u(s) + \left(h/\bar{h}\right) v_{i+1}^e(s) \right\}$$

$$\hat{g}_i(s) = \left(1 - h_{i+1}(s)/\bar{h}\right) \hat{g}_i^u(s) + \left(h_{i+1}(s)/\bar{h}\right) \hat{g}_i^e(s)$$

Outer loop becomes: $f_i \mapsto f_{i+1}$, $l_i \mapsto l_{i+1}$

- Update the distribution:

$$\hat{f}_i(a,\varepsilon;k,z) = \int_{\mathcal{K}\times\mathcal{A}} \Gamma_i\left(\hat{a},\hat{k}|a,k\right) f_i(\hat{a},\varepsilon;\hat{k},z) d\hat{a} d\hat{k}$$

$$\Gamma_i\left(\hat{a},\hat{k}|a,k\right) = \mathbb{1}\left[g_i(\hat{a},\varepsilon;\hat{k},z) = a, G_i(\hat{k},z) = k\right]$$

$$f_{i+1}(s) = (1-\delta)f_i(s) + \delta\hat{f}_i(s)$$