Numerical Appendix for:

Micro and Macro Uncertainty

Andreas Schaab

December 27, 2020

Job Market Paper

Please find the latest version here.

Overview. This Numerical Appendix provides details on the structure, implementation and performance of the global solution method I develop to solve the two-asset HANK model presented in the paper. Section 1 summarizes the model equations, in part for convenience but also to recast some equations in the form used for numerical implementation. Section 2 discusses the structure of the algorithm. Section 3 provides additional implementation details, and discusses the performance and fit of the solution method. Section 4 develops a numerical method to decompose the transmission mechanism of exogenous shocks, generalizing the method of Kaplan et al. (2018) to state-space representations of heterogeneous-agent models with aggregate risk.

Contents

1 Model Summary

The main aggregate driving process of the model is the discount rate shock ρ_t. Indeed, in the main text I denote the aggregate state of the economy as (ρ_t, g_t) for simplicity. However, various experiments studied in the main text and appendix require an enlargement of the state space. For example, the fundamental risk shock I use in Section 5.1 to study the transmission mechanism of macro uncertainty requires adding $\sigma_{\rho,t}$ as an additional exogenous driving process to the state space. The fiscal policy experiments discussed in the appendix also require additional state variables. To study fiscal multipliers, I solve the model with G_t added as an aggregate state variable. To study policy uncertainty, I add both G_t and $\sigma_{G,t}$ as state variables. More broadly, every "shock experiment" we wish to study requires that we introduce the shock as an additional aggregate state variable and re-solve the model.

In short, it will be convenient in the context of this numerical appendix to summarize all aggregate driving processes using a flexible vector notatation $Z_t \in \mathbb{R}^d$. The aggregate state of the economy is then given by (Z_t, g_t). As discussed at length in the main text and appendix, I approximate the infinite-dimensional cross-sectional distribution using an approximation of the form $\hat{g}_t(\boldsymbol{x}) = F(\alpha_t)(\boldsymbol{x}) \approx g_t(\boldsymbol{x})$, where $\boldsymbol{x} = (a, k, z)$ is the vector of idiosyncratic household state variables and $\alpha_t \in \mathbb{R}^N$. Consequently, the aggregate state space of the *approximate economy*, whose numerical solution is the subject of this appendix, is given by $\hat{\Gamma}_t = (Z_t, \alpha_t)$. This approximate aggregate state vector has dimension $d + N$. Since this numerical appendix makes hardly any reference to the true mathematical model, I will drop the hat notation with the understanding that our discussion references the approximate economy unless explicitly stated otherwise.

Finally, Proposition 3 of the main text tells us that $d\alpha_t$ follows a time-homogeneous diffusion process with drift $\mu_\alpha(\Gamma_t)$ and volatility $\sigma_\alpha(\Gamma_t)$. Combining this with the assumption that the exogenous driving processes Z_t also follow diffusion processes, we can write the law of motion of the aggregate state as

$$d\Gamma_t = \mu_\Gamma(\Gamma_t) + \sigma_\Gamma(\Gamma_t) dB_t, \tag{1}$$

where B_t now refers to the vector of all aggregate risk factors that enter Z_t.

1.1 Household problem

The recursive representation of the household problem is developed in Appendix A.3 of the main text. In the present context, it will be convenient to collapse or "stack" the

two earnings types, so that $V_z\mu_z \equiv \lambda(z^j)[V(z^{-j}) - V(z^j)]$.[1] The resulting household Hamilton-Jacobi-Bellman (HJB) equation can then be written as

$$(\rho + \zeta)V = u(c, H) + V_k m + V_a s + V_z \mu_z + V_Z \mu_Z + \frac{1}{2}\sigma_Z^T V_{ZZ}\sigma_Z + V_\alpha \mu_\alpha \tag{2}$$

where

$$s = (r + \zeta)a + \left(r^k + \frac{\Pi^Q}{K}\right)k + e - q\iota - \psi(\iota, k) - c \tag{3}$$

$$m = (\zeta - \delta)k + \iota \tag{4}$$

are used as shorthand notation for the drift of the household's liquid and illiquid asset positions.[2] I suppress notationally the dependence on state variables with the understanding that both the value function, $V(\cdot)$, and the policy functions, $c(\cdot)$ and $\iota(\cdot)$ as well as the shorthand $s(\cdot)$ and $m(\cdot)$, are functions of the household states (a, k, z, Z, α). Similarly, all macroeconomic aggregates, such as the interest rate $r(\cdot)$ or the laws of motion $\mu_Z(\cdot)$ and $\mu_\alpha(\cdot)$, are understood as functions of the economy's aggregate state (Z, α). As before, V_x denotes the partial derivative of V with respect to x. Lastly, the policy functions $c(\cdot)$ and $\iota(\cdot)$ are implicitly defined in terms of the value function by the optimality conditions

$$u_c = V_a \tag{5}$$

and

$$V_k = \left(q + \psi_\iota\right)V_a. \tag{6}$$

State constrained boundary conditions. A complete definition of the partial differential equation (2) requires a set of boundary conditions. For simplicity, I impose reflecting boundaries in the aggregate dimensions (Z, α). Achdou et al. (2015) discuss how *state constrained boundary conditions* must be used in the idiosyncratic dimensions (a, k) to properly reflect the borrowing and short-sale constraints on liquid and illiquid asset positions. These boundary conditions in my model are identical to those used and discussed in Kaplan et al. (2018) since I use the same household portfolio structure. For details, please refer to the Achdou et al. (2015) Online Appendix as well as Kaplan et al. (2018).[3]

[1] Computationally, we simply stack the individual columns V^j associated with the discrete earnings types j.

[2] Recall that, for my main model, $\sigma_\alpha = 0$ since the Kolmogorov forward equation that governs the evolution of the cross-sectional distribution is not stochastic. Therefore, the usual second-derivative term with respect to α drops out.

[3] The appropriate treatment of state constrained boundary conditions is orthogonal to the implementation of adaptive sparse grid methods to solve the household problem and, for that matter, to the rest of my algorithm. In this context, my method therefore requires nothing that is not already discussed in these papers.

1.2 Firm problem

When working with a sequence-space representation of the model, Phillips curves oftentimes take the form of readily implementable differential or difference equations (e.g. Kaplan et al. (2018) or Auclert et al. (2020)). In my setting, deriving an implementable equation for inflation is more complicated and requires setting up and solving the value function associated with the firm problem. Since firms solve a dynamic problem in this model, they are exposed to and internalize aggregate risk, much like households.

The goal is to derive an implementable function $\pi(\Gamma)$ that associates a level of CPI inflation π that is consistent with firm optimization with every aggregate state Γ. My strategy will be as follows. First, I set up the recursive problem of an individual firm. The only state variable of the firm is its individual price, which I denote by p. It is important that, at this stage, firms understand their individual prices p may be different from the aggregate consumer price P. Taking all macroeconomic aggregates as given, including the aggregate price P, each small firm's value function then implies an inflation policy function. Finally, with these policy functions in hand, we aggregate by assuming symmetry to arrive at an aggregate Phillips curve $\pi(\Gamma)$.

In Section 2.2 of the main text, I introduce the sequence problem of the firm, which I restate here for convenience. Instead of referring to firms by using j indexation, I adopt the above-mentioned state space notation and associate a particular firm with its individual price, or its idiosyncratic state variable, p. The sequence problem is

$$\max_{\pi_t} \mathbb{E}_0 \int_0^\infty e^{-\int_0^t r_s^k ds} \left[\left(\frac{p_t}{P_t} - mc_t \right) \left(\frac{p_t}{P_t} \right)^{-\epsilon^f} Y_t \quad - \Lambda_t(\pi_t) \right] dt.$$

where firm-level inflation is $\pi_t = \dot{p}_t / p_t$ and

$$\Lambda_t(\pi_t) = \frac{\chi^f}{2} \pi_t^2 Y_t.$$

To solve the associated firm value function problem, it will turn out to be highly convenient to perform a state space transformation. Notice that, since all firms take the same P_t as given, we can also use a firm's *relative price* as its idiosyncratic state variable. Denoting $\tilde{p} = p/P$, the firm's recursive problem takes the form of a Hamilton-Jacobi-Bellman equation given by

$$r^k W^f(\tilde{p}, \Gamma) = \max_{\tilde{\pi}} \left\{ \tilde{p}^{1-\epsilon^f} Y - mc \tilde{p}^{-\epsilon^f} Y - \Lambda(\tilde{\pi}, \Gamma) + W_{\tilde{p}}^f \tilde{\pi} \tilde{p} + W_\Gamma^f \mu_\Gamma + \frac{1}{2} \sigma_\Gamma W_{\Gamma\Gamma}^f \sigma_\Gamma^T \right\}$$

where $\tilde{\pi} = \dot{\tilde{p}} / \tilde{p}$ denotes relative price inflation. I suppress the dependence of aggregates like Y and mc on Γ on the RHS. Furthermore, I use the superscript f to distinguish the firm value function W^f from the union value function W^u which I will derive below.

The only object left unspecified in this HJB is the adjustment cost term $\Lambda(\cdot)$, which we have to appropriately rewrite as a function of relative price inflation and the aggregate state. We have

$$\Lambda(\tilde{\pi}, \Gamma) = \frac{\chi^f}{2}\left(\frac{\tilde{\pi}}{1-\pi(\Gamma)} + \pi(\Gamma)\right)^2 Y(\Gamma),$$

where, importantly, $\pi(\Gamma) = \dot{P}/P$ now refers to aggregate CPI inflation. Suppressing the notational dependence of prices and aggregates on Γ, the firm HJB becomes

$$r^k W^f(\tilde{p}, \Gamma) = \max_{\tilde{\pi}} \left\{ \tilde{p}^{1-\epsilon^f} Y - mc\tilde{p}^{-\epsilon^f} Y - \frac{\chi^f}{2}\left(\frac{\tilde{\pi}}{1-\pi} + \pi\right)^2 Y + W^f_{\tilde{p}} \tilde{\pi}\tilde{p} \right. \tag{7}$$
$$\left. + W^f_{\Gamma}\mu_{\Gamma} + \frac{1}{2}\sigma_{\Gamma} W^f_{\Gamma\Gamma}\sigma^T_{\Gamma} \right\}.$$

The associated first-order condition for firm inflation is given by

$$\tilde{\pi} = \frac{(1-\pi)^2}{\chi^f Y} W^f_{\tilde{p}} \tilde{p} - \pi(1-\pi), \tag{8}$$

and is therefore a policy function taking as inputs the firm's idiosyncratic and aggregate state variables, $\tilde{\pi} = \tilde{\pi}(\tilde{p}, \Gamma)$.

Lastly, to characterize aggregate CPI inflation, we assume that all firms are symmetric and aggregate. In the context of relative price inflation, symmetry requires that

$$\tilde{\pi}(\tilde{p}, \Gamma) \mid_{\tilde{p}=1} = 0.$$

This condition maintains that, in any aggregate state Γ, a firm does not seek *relative* price inflation when its current price is symmetric to that of all other firms, $\tilde{p} = 1$. Plugging this into equation (8) and inverting, we arrive at the desired mapping

$$\pi(\Gamma) = \frac{W^f_{\tilde{p}}(1, \Gamma)}{\chi^f Y(\Gamma) + W^f_{\tilde{p}}(1, \Gamma)}, \quad \text{for all } \Gamma. \tag{9}$$

From here, our implementation strategy is very straightforward: Solve the partial differential equation (7) on the associated firm grid $(\tilde{p}, \Gamma)$, compute the partial derivative $W^f_{\tilde{p}}$, evaluate it at the point $\tilde{p} = 1$, and plug back into equation (9).

1.3 Union problem

My implementation strategy for the union problem largely mirrors that for the firm problem. For convenience, I restate union k's sequence problem

$$\max_{\pi^w_{k,t}} \mathbb{E}_0 \int_0^{\infty} e^{-\int_0^t (\rho_s + \zeta) ds} \left[\int u(c_t, h_t) g_t d\boldsymbol{x} - \frac{\chi^w}{2}\left(\pi^w_{k,t}\right)^2 L_t \right] dt.$$

As in the previous subsection, I now write the union's recursive problem using its *relative wage*, $\tilde{w} = W_k/W$, as the only idiosyncratic state variable. The recursive problem is then given by

$$(\rho+\zeta)W^u(\tilde{w},\Gamma) = \max_{\tilde{\pi}^w}\left\{\int u\Big(c(\boldsymbol{x},\Gamma;\tilde{w}),h(\boldsymbol{x},\Gamma;\tilde{w})\Big)gd\boldsymbol{x} - \frac{\chi^w}{2}\left(\pi_k^w\right)^2 L \right. \tag{10}$$
$$\left. + W^u_{\tilde{w}}\tilde{w}\tilde{\pi}^w\right\} + W^u_\Gamma\mu_\Gamma + \frac{1}{2}\sigma_\Gamma^T W^u_{\Gamma\Gamma}\sigma_\Gamma.$$

As before, the only step left is to express π_k^w in terms of relative wage inflation. We have

$$\pi_k^w = \frac{\tilde{\pi}^w + \pi^w - (\pi^w)^2}{1-\pi^w} = \frac{\tilde{\pi}^w}{1-\pi^w} + \pi^w.$$

Thus, the union's first-order condition can be written as

$$\tilde{\pi}^w = \frac{(1-\pi^w)^2}{\chi^w L}W^u_{\tilde{w}}\tilde{w} - \pi^w(1-\pi^w), \tag{11}$$

which characterizes the policy function for relative wage inflation, $\tilde{\pi}^w$, in terms of the state variables $(\tilde{w},\Gamma)$.

As in the context of the firm problem, a symmetric equilibrium requires that

$$\tilde{\pi}^w(\tilde{w},\Gamma)\mid_{\tilde{w}=1} = 0,$$

so that firms find it individually optimal, at the symmetric equilibrium point $\tilde{w}=1$, not to seek *relative* wage inflation. Plugging this into equation (11), evaluating at $\tilde{w}=1$, and rearranging yields

$$\pi^w(\Gamma) = \frac{W^u_{\tilde{w}}(1,\Gamma)}{\chi^w L(\Gamma) + W^u_{\tilde{w}}(1,\Gamma)}. \tag{12}$$

To implement this equation for nominal wage inflation, I solve the union HJB (10) on a grid over $(\tilde{w},\Gamma)$, take the partial derivative $W^u_{\tilde{w}}$ and evaluate it at $\tilde{w}=1$, and finally plug back into equation (12).

TBD. Discuss how I implement $c(\boldsymbol{x},\Gamma;\tilde{w})$ and $h(\boldsymbol{x},\Gamma;\tilde{w})$ (as functions of $\tilde{w}$) by projecting and appropriately aggregating the respective policy functions from the household grid over $(\boldsymbol{x},\Gamma)$ onto the union grid over $(\tilde{w},\Gamma)$.

1.4 Macro block

The "macro block" of the model is the set of equations that associates with each aggregate state Γ a set of endogenous macroeconomic objects. The set of relevant macroeconomic aggregates is given, in no particular order, by the functions

$$\{r, r^k, q, \Pi^Q, \tau^{\text{lump}}, H, w, Y, C, I, L, \pi, \pi^w, A, K, U, \Psi\}(\Gamma).$$

The **macro block** is then defined as the set of equations

$$
\begin{aligned}
Y &= K^{1-\beta}L^{\beta} \\
\Pi^{Q} &= qI - I - \Phi\left(\frac{I}{K}\right)K \\
q &= 1 + \Phi'\left(\frac{I}{K}\right) \\
\tau^{\text{lump}} &= \Pi + \tau^{\text{lab}}wL - \tau^{\text{UI}}U - G - rB^{G} \\
\Pi &= (1 - mc)Y \\
mc &= \frac{1}{\beta^{\beta}(1-\beta)^{1-\beta}}\left(r^{k}\right)^{1-\beta}\left(w\right)^{\beta} \\
r &= i - \pi \\
r^{k} &= \frac{1-\beta}{\beta}w\frac{L}{K} \\
i &= \max\left\{r^{*} + \bar{\pi} + \lambda_{\pi}\pi + \lambda_{Y}\log\left(\frac{Y}{Y^{*}}\right), 0\right\} \\
A &= B^{G} \\
Y &= C + I + \Phi + \Psi + G,
\end{aligned}
$$

together with equations (9) and (12) from the firm and union problems for $\pi(\Gamma)$ and $\pi^{w}(\Gamma)$, respectively. r^{*} and Y^{*} denote steady state values of the real interest rate and output, respectively. The equations of the macro block are combined with an **aggregation block** that characterizes the aggregate household sector by aggregating from the micro level,

$$
\begin{aligned}
A &= \int aF(\alpha)(\boldsymbol{x})d\boldsymbol{x} \\
K &= \int kF(\alpha)(\boldsymbol{x})d\boldsymbol{x} \\
C &= \int c(\boldsymbol{x})F(\alpha)(\boldsymbol{x})d\boldsymbol{x} \\
L &= \int zh(\boldsymbol{x})F(\alpha)(\boldsymbol{x})d\boldsymbol{x}
\end{aligned}
$$

$$I = \int \iota(\boldsymbol{x}) F(\alpha)(\boldsymbol{x}) d\boldsymbol{x}$$

$$\Psi = \int \psi(\iota, k) F(\alpha)(\boldsymbol{x}) d\boldsymbol{x}$$

$$U = 1 - \int z F(\alpha)(\boldsymbol{x}) d\boldsymbol{x},$$

where U is the aggregate unemployment rate. Since unions impose symmetric labor hours, the aggregate labor supply equation simplifies to $L = (1 - U)H$. One of the market clearing conditions can be dropped using Walras' law. Lastly, the exogenous aggregate discount rate process follows

$$d\rho = \theta_\rho(\bar{\rho} - \rho)dt + \sigma_\rho dB,$$

where the Brownian motion B is the only aggregate risk factor in the baseline model.

1.5 Deterministic steady state

Formally, the deterministic steady state, or stationary equilibrium in the language of Achdou et al. (2015), is defined as the equilibrium of my economy in the limit as $\sigma_\rho \to 0$ so that aggregate risk vanishes.

I compute the deterministic steady state of the model at the very beginning of the algorithm, and use it as a starting guess. In particular, I initialize the distribution representation $F^0(\alpha)(\boldsymbol{x})$ using the cross-sectional distribution that obtains in the deterministic steady state. See Section 2.3 for details.

1.6 Recursive equilibrium and state space representation

A recursive competitive equilibrium of approximate economy n is defined as the sets of functions $\{V, g, c, \iota\}(a, k, z, \Gamma)$ and $\{r, r^k, q, \tau^{\text{lump}}, H, w, Y, K, H, C, I, L, \pi, \pi^w\}(\Gamma)$ such that: (i) All agents optimize given general equilibrium prices, (ii) markets clear in all aggregate states Γ, and (iii) agents act according to beliefs that are consistent with the law of motion of approximate economy n.

2 Algorithm Structure

This section develops an algorithm to solve for the recursive equilibrium described above in Section 1.6. This algorithm consists of up to 3 hierarchical levels or, more formally, up to 3 nested fixed points. This high-level structure is illustrated below.

Algorithm Structure

Level 3: Update the distribution representation

↳ **Level 2**: Solve for GE prices to clear markets

↳ **Level 1**: Solve the household problem

The broad strategy of this algorithm is therefore as follows: In the nth outer iteration, corresponding to **Level 3**, we choose an approximate distribution representation

$$\hat{g}_t^n(\boldsymbol{x}) = F^n(\alpha_t)(\boldsymbol{x}) \approx g_t(\boldsymbol{x}).$$

Echoing the language I adopt in the main text, the nth *approximate economy* is then defined as the model that takes $\hat{g}_t^n(\boldsymbol{x})$ as its cross-sectional distribution. In other words, the nth approximate economy is defined as a restricted version of the true model which assumes that agents behave according to the *belief* that the cross-sectional distribution evolves according to $\frac{d}{dt}F^n(\alpha_t)(\boldsymbol{x})$.[4] A fixed point may then be constructed that solves a sequence of approximate economies with increasingly accurate fit. If the algorithm achieves convergence (heuristically, in the sense $\hat{g}_t^n(\boldsymbol{x}) \to^n g_t(\boldsymbol{x})$), the beliefs of agents in the nth approximate economy will become consistent with the model's true law of motion for n sufficiently large. This outer fixed point corresponds to **Level 3** of my algorithm, and is discussed in Section 2.3.

In the nth outer iteration, **Level 1** and **Level 2** of the algorithm together constitute a solution of the model that takes $\hat{g}_t^n(\boldsymbol{x})$ as its cross-sectional distribution. In particular, **Level 1** solves for the value and policy functions associated with the household problem taking as given all GE objects like prices. I use a value function iteration (VFI) algorithm which I discuss in Section 2.1. In this context, I build on the seminal contribution of Achdou et al. (2015) by, among other things, solving the associated VFI on an arbitrarily irregular grid.

[4] In the seminal Krusell and Smith (1998) algorithm, a "behavioral" version of the model is solved in which agents believe that capital follows the exogenously specified law of motion. In the same spirit, my algorithm solves a sequence of approximate economies in which agents believe the cross-sectional distribution evolves consistently with $d\alpha_t$.

In the middle step of my algorithm, **Level 2**, I then solve for the general equilibrium prices that clear markets while aggregating household behavior from the micro level using the policy functions obtained from the VFI. Broadly speaking, two approaches could be considered for this step. Achdou et al. (2015) and Kaplan et al. (2018) use a fixed-point iteration algorithm to solve for the impulse response functions of GE objects in a sequence-space representation of their models. In this paper, I develop a flexible and robust non-linear equation solver that delivers substantial performance gains. My approach is therefore in the spirit of Auclert et al. (2020) and related papers that leverage Newton-type solvers. I develop the structure of this algorithm in Section 2.2 and discuss its performance in Section 3.4.

In the remainder of this section, I flesh out the details of my algorithm by starting from the inner (VFI) and working my way to the outer (distribution representation) fixed point.

Comparison to Krusell and Smith (1998). Two main features distinguish the structure of my algorithm from Krusell and Smith (1998), aside from the obvious difference that I do not use a moments-based distribution approximation. First, algorithms in the spirit of Krusell and Smith (1998) skip **Level 2**. These algorithms do not have a distribution object on the grid which they could use to evaluate market clearing conditions. As a result, these algorithms resort to updating both the distribution representation (specifically, its law of motion) and general equilibrium prices in the outer fixed point of the algorithm.

Second, the original Krusell and Smith (1998) algorithm takes a particular distribution representation (i.e. the aggregate capital stock) as given from the start. While the algorithm features an outer fixed point, it uses this outer loop to update the law of motion of the distribution approximation (i.e. the law of motion of capital) and general equilibrium prices. My algorithm, on the other hand, updates and improves on the distribution approximation itself.

There have been several algorithms since Krusell and Smith (1998) that follow the traditional moments-based approach but specify a "distribution selection function" or "proxy distribution" to construct a distribution object on the grid from a given set of moments. Important examples include Algan et al. (2008) and Reiter (2010). These algorithms, like mine, use an outer loop to successively update and improve on the distribution representation.

Grid construction. In Section 3.3, I discuss where and how in this algorithm I leverage sparse grid adaptation. For the purposes of Section 2, it suffices to note that **Levels 2** and **3** always take the grid as given. By contrast, whenever the outer fixed point in **Level 1** is non-trivial, in the sense that there is some sequential updating on the distribution

representation and $n > 1$, then the grid must necessarily be re-constructed in every outer iteration.

To see this, consider the example where each outer iteration n adds one additional basis function to the representation $F(\alpha_t)(\boldsymbol{x})$ and, consequently, increases the dimensionality of $\alpha_t \in \mathbb{R}^n$ by one. Since the α_t constitute part of the aggregate state space of the approximate economy, one dimension must correspondingly be added to the grid in each outer iteration. In sum, my algorithm will generally reconstruct the grid after each outer iteration, but then keep the grid fixed at the two lower levels.

2.1 Level 1: household problem

This subsection focuses on solving the Hamilton-Jacobi-Bellman (HJB) equation (2) associated with the household problem, where c, ι, s and m are given by equations (3) through (6). The value function iteration algorithm I implement to do so corresponds to **Level 1** of my algorithm.

In this innermost fixed point, I solve equation (2) taking as given three kinds of objects: (1) A possibly irregular grid G constructed over the household state space $(\boldsymbol{x}, Z, \alpha)$. The grid G is only updated on the outer **Level 3** of the algorithm. (2) All general equilibrium objects that are taken as given by the household. Specifically, I take as given the set of functions $r(\Gamma)$, ... These mappings are updated on **Level 2** of the algorithm. (3) Finally, I take as given the form of the distribution representation, $F(\cdot)$, which is again only updated on **Level 3**.

Equation (2) is a time-independent parabolic partial differential equation (PDE) in $3 + d + N$ dimensions. I solve it using a method that is closely related to and builds on the seminal continuous-time value function iteration algorithm developed by Achdou et al. (2015). Many key features of their algorithm directly carry over to my setting. In particular, I also use finite-difference methods to discretize the PDE, I construct a time-marching fixed point, I use a semi-implicit scheme for updating the value function at each time step, and I use the upwinding scheme discussed in Kaplan et al. (2018) for the policy functions. Together, these algorithmic choices imply a discretization of equation (2) which takes the form

$$\begin{aligned}\frac{V^{i+1} - V^i}{\Delta} + (\rho + \zeta)V^{i+1} = \; & u^i + \left(D_k^S V^{i+1}\right) m^i + \left(D_a^S V^{i+1}\right) s^i + \left(D_z^S V^{i+1}\right) \mu_z \\ & + \left(D_Z^S V^{i+1}\right) \mu_Z + \frac{1}{2}\sigma_Z^T \left(D_{ZZ}^S V^{i+1}\right) \sigma_Z + \left(D_\alpha^S V^{i+1}\right) \mu_\alpha^i\end{aligned} \tag{13}$$

where the i superscript denotes the iteration of the time-marching fixed point. Time-marching is a commonly used method to solve time-independent parabolic PDEs. See

for example LeVeque (2007). As explained in the Achdou et al. (2015) Online Appendix, this algorithm is semi-implicit because the update V^{i+1} is defined implicitly by equation (13). It is semi- rather than fully implicit because we use the value function of the current iteration V^i to compute the policy functions, which is why u^i, s^i, m^i and μ_α^i are denoted with i rather than $i+1$ superscripts.

Operator notation. Equation (13) uses "operator notation" to denote partial derivatives or, after discretization, finite difference operations. In other words, I use the flexible notation $D_x V = V_x$ to denote the partial derivative of V with respect to x. Whenever V denotes a function that takes as inputs continuous state variables, D_x denotes the functional operator of differentiation. Whenever V refers to the discretized representation of the value function on a grid, D_x denotes the associated finite difference operator. In particular, a discretized finite difference operator D_x is simply a matrix that pre-multiplies the column vector V when represented on a grid.

Using operator notation will be highly convenient for several reasons. Most importantly, solving equation (13) on a sparse or irregular grid will simply require substituting the usual finite difference operators D_x, which are discussed in detail in Achdou et al. (2015), with a *sparse finite difference operator* D_x^S, whose construction we discuss in detail in Schaab and Zhang (2020). The structure of the value function iteration algorithm otherwise remains largely unchanged.[5]

Upwinding. As in Achdou et al. (2015) and Kaplan et al. (2018), I use an upwinding scheme to compute m^i and s^i, which is not made explicit in equation (13) for expositional clarity. In fact, since the structure of the household's portfolio problem in my model is identical to Kaplan et al. (2018), I can also use the same upwinding scheme. Please refer to that paper for additional details.[6]

Key differences from Achdou et al. (2015). My implementation of the household HJB deviates from the algorithm developed in Achdou et al. (2015) in three key regards.

1. I implement equation (13) on a highly adapted sparse grid. For a detailed and self-contained discussion, please see Schaab and Zhang (2020). At a high level, the main practical difference when solving differential equations on irregular grids is that *sparse finite difference operators* must be used in order to achieve a *consistent*

[5] When computed on a dense grid, the two finite difference operators coincide, with $D_x = D_x^S$.

[6] The choice of upwinding scheme is largely orthogonal to issues arising from the use of adaptive sparse grid methods when implementing equation (13). In particular, the main departure from Kaplan et al. (2018) is that we again have to use the sparse finite difference operator D_x^S rather than D_x when computing the upwinding solutions for m^i and s^i.

discretization scheme for the underlying differential equation. Using the usual finite difference operators associated with dense grids, as in Achdou et al. (2015), would not lead to a consistent discretization in the sense that the time-marching fixed point in equation (13) would not converge to the true V. However, the construction and use of the appropriate operators is largely modular with respect to the rest of the algorithm. That is, the sparse finite difference operators D_x^S can be constructed independently and outside of the value function iteration algorithm.

2. I implement an "extended VFI" that internally solves for the consistent law of motion μ_α. In this sense, μ_α^i is updated in each iteration of the time-marching scheme. In particular, the fixed point in (13) implies that when convergence is reached in $V^{i+1} - V^i$, the associated μ_α^i then automatically corresponds to a consistent forecast of the law of motion of α.[7]

3. I solve a fully recursive equilibrium of my model, adopting a state-space rather than a sequence-space representation. Equation (13) implements the stationary household value function that fully internalizes time variation in the aggregate state of the economy (Z, α). In particular, the second row of terms in equation (13) captures the effect of an expected change in the economy's aggregate state on the household value function. Both Achdou et al. (2015) and Kaplan et al. (2018) solve the household problem using a sequence-space representation.

Linear system. The discretized time-marching scheme in (13) gives rise to a linear system of equations that can be solved to obtain the update V^{i+1}. This linear system takes the form

$$\boldsymbol{A}^i V^{i+1} = \boldsymbol{B}^i. \tag{14}$$

Letting J denote the number of grid points, $\boldsymbol{A}^i$ is a $J \times J$ matrix and

$$\boldsymbol{B}^i = \frac{1}{\Delta} V^i + u^i$$

a $J \times 1$ column vector.

The matrix $\boldsymbol{A}^i$ represents a discretized (composite) functional operator given by

$$\boldsymbol{A}^i = \left(\frac{1}{\Delta} + \zeta\right)\boldsymbol{I} + \rho - \left(m^i D_k^S + s^i D_a^S + \mu_z D_z^S + \mu_Z D_Z^S + \frac{1}{2}\sigma_Z^T D_{ZZ}^S \sigma_Z + \mu_\alpha^i D_\alpha^S\right), \tag{15}$$

where $\boldsymbol{I}$ is a $J \times J$ diagonal matrix of 1s. It is now evident why expressing (13) using operator notation is highly convenient: We can construct $\boldsymbol{A}^i$ by directly summing the

[7] Ahn et al. (2017) solve a similarly structured VFI.

vector-matrix products on the RHS. In particular, each sparse finite difference operator D_x^S is only constructed *once* ex ante.[8] The terms associated with exogenous driving processes ($\mu_z D_z^S$, $\mu_Z D_Z^S$, and $\frac{1}{2}\sigma_Z^T D_{ZZ}^S \sigma_Z$) similarly don't change over the course of the value function iteration and can be computed ex ante. In each iteration i, we are therefore left with computing the policy functions (u^i, m^i and s^i), updating the law of motion μ_α^i (discussed below), and finally computing and summing the vector-matrix products in (15).

In Schaab and Zhang (2020), we show how to program a readily computable function Ξ that takes as its only inputs the drift and volatility coefficients of each state variable (i.e. a set of column-vectors) and outputs the matrix $\boldsymbol{A}^i$. For my two-asset HANK model, this implies

$$\Xi : \{s, m, \mu_z, \mu_Z, \sigma_Z, \mu_\alpha\} \mapsto \boldsymbol{A}.$$

What is particularly nice is that the structure of Ξ is so general that it can be applied to every HJB equation we have thus far come across in economics. This is largely due to the fact that the operators D_x^S can be computed entirely independently from the details of the economic application.

Krylov subspace methods. Commonly used linear equations solvers quickly falter when confronted with the system (14) in practice for two reasons. First, the grids on which I solve V quickly grow large because, in practice, the household problem has 8 to 10 state variables when we use 3 to 5 dimensions of distribution representation.

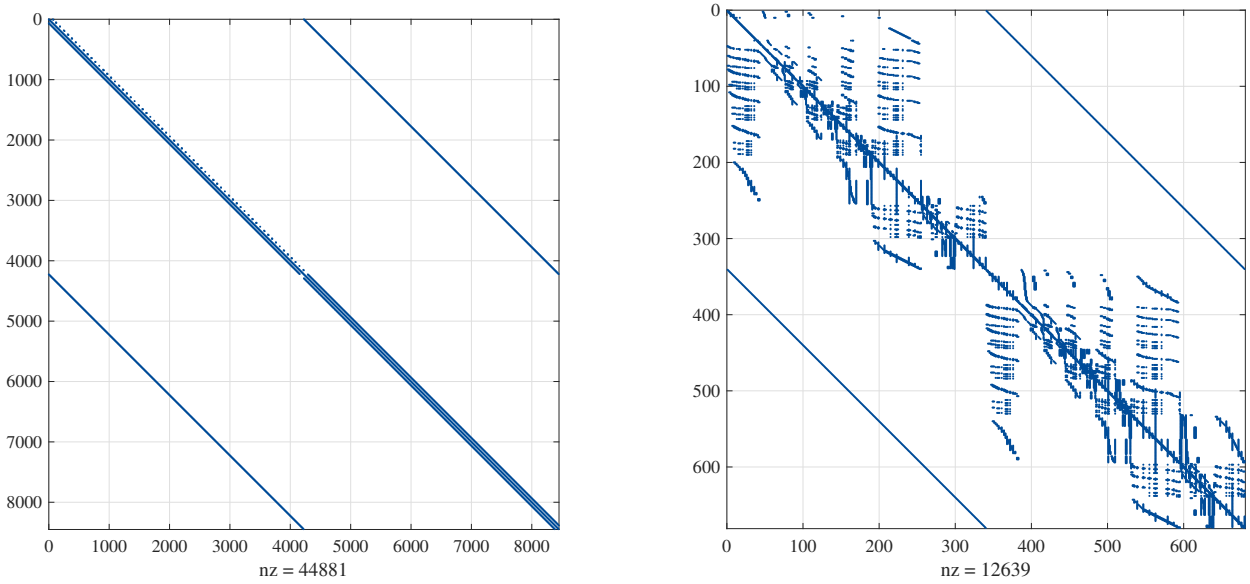
Second, the resulting matrix $\boldsymbol{A}^i$ is no longer as "nice" (i.e. highly sparse and symmetric) as in the context of dense grids. This is illustrated in Figure 1. I use Matlab's "spy( )" command to visualize the sparsity structure of $\boldsymbol{A}$. For illustration, I solve the household value function for the model's stationary equilibrium on two different grids. Panel (a) uses a fine, dense grid while panel (b) uses an adapted sparse grid. The adapted sparse grid features significantly fewer grid points (around 340 in total over the (a, k) dimensions) than the dense grid (around $4,200$ in total over the (a, k) dimensions).[9] However, this comes at the cost of a more complex sparsity structure in $\boldsymbol{A}$. Intuitively, this happens because sparse finite difference operators do not simply "look left and right" but rather draw information from many surrounding grid points. While this is one notable disadvantage of using adaptive sparse grid methods, the associated performance gains typically dominate.

In practice, therefore, I resort to using Krylov subspace methods. In particular, the

[8] This is in stark contrast to Achdou et al. (2015). They manually construct the vector-matrix products $\mu_x D_x$ from scratch in each iteration. While this is feasible (and sufficiently fast) for dense grids, leveraging the observation that the finite-difference operators D_x can be constructed independently is important in the context of irregular grids.

[9] The associated two-dimensional grid is illustrated in Figure 4.

Figure 1: Sparsity structure of functional operator $\boldsymbol{A}$



(a) Dense grid

(b) Adapted sparse grid

Notes. This figure uses Matlab's spy( ) command to plot the sparsity structure of the matrix $\boldsymbol{A}$. Each dot corresponds to a nonzero value. Panels (a) and (b) plot the $\boldsymbol{A}$ matrices associated with solutions of the (steady state) household value function for two different grids. Panel (a) uses a fine, dense grid. Panel (b) uses a highly adapted sparse grid. The solid diagonal lines in the upper right and lower left quadrants of these plots represent transitions across employment states. The points in the upper left and lower right quadrants represent transitions in portfolio positions.

asymmetry of $\boldsymbol{A}^i$ necessitates the use of the *generalized minimal residual* (GMRES) method.[10] For a superb introduction to iterative methods see Demmel (1997). The GMRES algorithm was originally developed in Saad and Schultz (1986).[11]

Solving for μ^i_α. Proposition 3 in the main text allows us to nest the computation of the consistent law of motion $d\alpha$ within the household VFI. I discuss there that the formulas for $\mu_\alpha(\Gamma)$ and $\sigma_\alpha(\Gamma)$ generically depend only on the household policy functions, as well as other objects that are taken as given and held fixed over the course of the household VFI.

[10] Ahn et al. (2017) also leverage Krylov subspace methods. Since they work on dense grids, much of the symmetry in $\boldsymbol{A}^i$ is preserved in their applications, which allows them to use a method that exploits this symmetry.

[11] When implementing this VFI in Matlab, GMRES typically starts outperforming the standard "mldivide" solver when the size of the grid J exceeds between 5,000 and 10,000 grid points.

Value function iteration i, therefore, starts with a guess V^i and computes the policy functions c^i, ι^i, s^i and m^i. At this stage, the only object left unspecified in equations (13) and (15) is μ_α^i (in my baseline model, $\sigma_\alpha = 0$). However, with both the policy functions for iteration i and the sparse finite difference operators in hand, μ_α^i can be readily computed using the formula from Proposition 3.

This approach shifts "complexity" from the outer fixed point (**Level 3**) to the inner fixed point (**Level 1**), which is desirable in many but not all applications. The traditional approach in the spirit of Krusell and Smith (1998) iterates on μ_α in the outermost fixed point by necessity because moments-based methods cannot compute μ_α as part of an extended household VFI. The traditional approach has the advantage of a more stable value function iteration step. This comes at the cost of requiring extra outer iterations in which the model is solved and simulated in order to find an update for μ_α.

In practice, leveraging the extended VFI structure to dispense with outer iterations is highly effective in simple models like the Krusell and Smith (1998) model or even a one-asset HANK model. In the context of more complex models like this paper's two-asset HANK model, solving for the laws of motion of *all* α_t inside the household VFI can lead to stability problems. Intuitively, there are more objects that change from one iteration to the next, which may slow and even thwart convergence in V^i. Concretely, this oftentimes comes at the cost of requiring a much smaller Δ and, consequently, many more iterations until V^i converges. Which side of this tradeoff dominates is not clear a priori and should, in practice, be weighed in the context of an algorithm's other bottlenecks.[12]

2.2 Level 2: general equilibrium prices and market clearing

I use a quasi-Newton algorithm to solve for the set of prices that clears markets in general equilibrium. This requires the construction of an objective function

$$\gamma : \text{price guesses} \mapsto \text{market clearing gap}$$

that associates an error metric with any potential price guess. It is most natural to derive this error metric from the "gap" in market clearing conditions implied by a particular price guess. After defining this function γ, we can simply feed it into a non-linear equation solver. I will provide additional details on this procedure in two steps. First, I discuss the construction of γ. Second, I sketch the quasi-Newton algorithm I use in practice.

[12] In practice, I implement my algorithm with a toggle that allows me to switch between these two approaches seamlessly. When solving the two-asset HANK model, the Δ required for VFI convergence when computing μ_α internally can be smaller by a factor of 100 or more.

2.2.1 Market clearing

Denote the number of discretized aggregate states or grid points by J^{agg}. To implement a Newton solver, the function γ must be of the form

$$\gamma : \mathbb{R}^{k \times J^{\text{agg}}} \to \mathbb{R}^{k \times J^{\text{agg}}}$$

where k denotes the number of prices or aggregates that we need to solve for across the J^{agg} aggregate states. Since Newton methods require the construction of the Jacobian matrix associated with γ, we naturally want k to be as small as possible to optimize performance.

My strategy will be to rearrange and reduce the macro block equations summarized in Section 1.4 so that I can directly compute all prices in terms of the smallest possible number of guesses. Concretely, I have to guess the four functions

$$\{Y, w, Q, \pi\}(\Gamma)$$

but can then compute all remaining GE objects using the equations from Section 1.4, including the relevant market clearing gap which becomes the output of $\gamma(\cdot)$.[13] For completeness, I will restate the relevant equations in the order I use them in practice. In the baseline model, I set $G(\Gamma) = 0$. Given $Y(\Gamma)$, $w(\Gamma)$, $q(\Gamma)$ and $\pi(\Gamma)$, as well as $K(\Gamma)$, $A(\Gamma)$ and $U(\Gamma)$ which can be computed directly by evaluating $F(\alpha)$ on the grid, we have

$$\begin{aligned}
L &= \left(Y K^{\beta-1}\right)^{\frac{1}{\beta}} \\
H &= \frac{L}{1-U} \\
r^k &= \frac{1-\beta}{\beta} w \frac{L}{K} \\
i &= \max\left\{r^* + \bar{\pi} + \lambda_\pi \pi + \lambda_Y \log\left(\frac{Y}{Y^*}\right), 0\right\} \\
r &= i - \pi \\
mc &= \frac{1}{\beta^\beta (1-\beta)^{1-\beta}} \left(r^k\right)^{1-\beta} \left(w\right)^{\beta} \\
\Pi &= (1 - mc) Y
\end{aligned}$$

[13] The present discussion applies to the current baseline model which takes the limit as wages become fully flexible, so that only goods prices are sticky. When both wages and prices are sticky, w becomes an additional state variable and the vector of price guesses must also include $\pi^w(\Gamma)$.

$$\Pi^Q = qI - I - \Phi\left(\frac{I}{K}\right)K$$

$$\tau^{\text{lump}} = \Pi + \tau^{\text{lab}} wL - \tau^{\text{UI}} U - rB^G.$$

Given our price guess, we have now computed all aggregate objects on the grid that are required to solve the household problem and obtain the policy functions $c(\cdot)$ and $\iota(\cdot)$, which lets us aggregate and compute $C(\Gamma)$, $I(\Gamma)$ and $\Psi(\Gamma)$. The first three out of four market clearing gaps can now be computed for each of the J^{agg} aggregate states Γ,

$$\chi_Y = Y - C - I - \Phi - \Psi$$

$$\chi_w = H^\eta - \frac{\epsilon^w - 1}{\epsilon^w}(1 - \tau^{\text{lab}})w \int z c(\boldsymbol{x})^{-\gamma} F(\alpha) d\boldsymbol{x}$$

$$\chi_q = q - 1 - \Phi'\left(\frac{1}{K}\int \iota(\boldsymbol{x}) F(\alpha)(\boldsymbol{x}) d\boldsymbol{x}\right).$$

Notice that, in the limit with flexible wages, χ_w corresponds to the optimality condition of the union which can set wages directly in the absence of adjustment costs. When wages are sticky, on the other hand, we compute the union problem and define an error metric for wage inflation much like the error metric for price inflation, χ_π, defined below.

The fourth and final market clearing gap comes from the firm problem. I solve the firm problem to obtain the policy function $\tilde{\pi}(\tilde{p}, \Gamma)$. Since we know that, to sustain a symmetric equilibrium, relative price inflation must be 0 when $\tilde{p} = 1$, we can use a projection to compute

$$\chi_\pi = \tilde{\pi}(1, \Gamma)$$

for all aggregate states Γ.

Putting everything together, our function becomes

$$\gamma : \{Y, w, q, \pi\}(\Gamma) \mapsto \{\chi_Y, \chi_w, \chi_q, \chi_\pi\}(\Gamma),$$

which we can now readily plug into any non-linear equation solver.

Importance of grid-based distribution representation. Unlike in the traditional Krusell and Smith (1998) algorithm, my approximation $F^n(\alpha)(\boldsymbol{x}) = g(\boldsymbol{x})$ implies that I can readily compute the cross-sectional distribution associated with approximate economy n on the grid. This is at the heart of everything discussed in Section 2.2. Without being able to use such a grid-based distribution object, I could not evaluate market clearing conditions or, consequently, solve for prices on the grid as discussed above.

2.2.2 A flexible quasi-Newton solver

In progress and coming soon.

2.3 Level 3: distribution representation

As discussed in the main text and appendix of the paper, my algorithm admits both parametric and non-parametric variants. The parametric algorithm chooses $F(\cdot)$ from a prespecified parametric family. In practice, however, I use a non-parametric algorithm to solve the two-asset HANK model. Therefore, I discuss the non-parametric approach first in this subsection. In the next subsection, I further elaborate on the parametric variant.

The method I use in practice focuses on approximations of the cross-sectional distribution that use basis function representations for which $F(\cdot)$ is affine in $\alpha_t \in \mathbb{R}^N$. That is,

$$F^N(\alpha_t)(\boldsymbol{x}) = C(\boldsymbol{x}) + \sum_{n=1}^{N} T^n(\boldsymbol{x})\alpha_t^n = C(\boldsymbol{x}) + T(\boldsymbol{x})\alpha_t, \tag{16}$$

where $T(\boldsymbol{x})$ and α_t stack the T^n and α^n, respectively. This form allows for a constant term $C(\boldsymbol{x})$ which I typically take to be the cross-sectional distribution associated with the deterministic steady state of the model. That is, $C(\boldsymbol{x}) = g^0(\boldsymbol{x}) = \lim_{\sigma_\rho \to 0} g(\boldsymbol{x}; \sigma_\rho)$.

The primary objective of my algorithm's outer fixed point (**Level 3**) is to find a set of basis functions $\{T^n\}$ for which the associated approximate economy is sufficiently close to the true model.[14] To formalize this approach, I proceed in three steps. First, I define and discuss an appropriate error metric that can guide the search for efficient $\{T^n\}$. Second, I show how to write an objective function that computes this error metric for any given distribution representation $F(\alpha)$. Third, I formally write down the estimation problem of choosing T^{n+1} to minimize our error in iteration n, and I discuss how to implement the associated fixed point in practice.

Forecast errors. In outer iteration n, **Level 1** and **Level 2** of my algorithm compute the solution of approximate economy n where the cross-sectional distribution is given by $F^n(\alpha)(\boldsymbol{x})$. We want to ask how far approximate economy n is from the true solution of the model. What is the source of potential discrepancy in this comparison based on the definition of recursive equilibrium in Section 1.6? The VFI on **Level 1** ensures that the optimality conditions for household behavior are respected. And solving the macro block

[14] As I discussed in Section 2.1, Proposition 3 of the main text can be used to compute the consistent law of motion $d\alpha$ as part of the household value function iteration step. However, this may not always be optimal because doing so takes its toll on the convergence properties of the VFI. Alternatively, we can compute the laws of motion for some but not all α inside the household VFI. The laws of motion of the remaining α are then updated in the outer fixed point, at the same time as $T(\boldsymbol{x})$ is updated. Indeed, this is the traditional approach of finding the consistent law of motion $d\alpha$: After solving and simulating the model for a given $F^n(\cdot)$, we can estimate the law of motion implied by the simulated cross-sectional distribution and update $d\alpha$ accordingly. For expositional clarity, I focus the present discussion around the case where all elements of $d\alpha$ are computed using the formulas of Proposition 3.

on **Level 2** implies that all markets clear in general equilibrium, respecting household behavior at the micro level. By construction, therefore, any residual discrepancy between approximate economy n and the true model must stem from remaining error in the distribution representation, $F^n(\alpha)(\boldsymbol{x}) \approx g(\boldsymbol{x})$.

To check how well $F^n(\alpha)(\boldsymbol{x})$ approximates $g(\boldsymbol{x})$, we would ideally like to simulate both approximate economy n and the true model for the same draw of exogenous shocks, and then directly compare the paths of the simulated cross-sectional distributions. Consider a simulation over time horizon $t \in [0, T]$, then we would ideally like to construct an error metric

$$\left|\left| F^n(\alpha_t)(\boldsymbol{x}) - g_t^{\text{true}}(\boldsymbol{x}) \right|\right|_{(t,\boldsymbol{x})\in\mathbb{R}\times\mathbb{R}^d}$$

and update the distribution representation F^{n+1} to account for residual error in this approximation.

While we cannot, of course, directly compute and simulate the true model, we can construct a fixed point much in this spirit, which, upon convergence, would imply that all agents in the economy behave according to consistent forecasts of the cross-sectional distribution. Consider again a solution of approximate economy n. For a given draw of the aggregate shock sequence, $\{\rho_t\}$, agents behave *as if* the cross-sectional distribution evolved according to $F^n(\alpha_t)(\boldsymbol{x})$, where $d\alpha_t = \mu_\alpha(\Gamma_t)dt + \sigma_\alpha(\Gamma_t)dB_t$. We can therefore directly compute household forecasts, by simply time-integrating the paths of all aggregate state variables. In the baseline model where $\Gamma = (\rho, \alpha)$, this simply requires integrating

$$\alpha_t = \alpha_0 + \int_0^t \mu_\alpha(\rho_s, \alpha_s)ds + \int_0^t \sigma_\alpha(\rho_s, \alpha_s)dBs$$

since we take $\{\rho_t\}$ as given exogenously. This yields the time series $\{\alpha_t\}_{[0,T]}$ and, therefore, the associated distribution forecast $\{F^n(\alpha_t)(\boldsymbol{x})\}_{[0,T]}$. And with time series for all aggregate state variables in hand, we can readily compute the time series of all other macroeconomic aggregates using the functions we computed on **Level 2** to clear markets. For example, $\{Y_t^{\text{lom}}\}_{[0,T]} = \{Y(\rho_t^{\text{lom}}, \alpha_t^{\text{lom}})\}_{[0,T]}$. I denote the time series obtained from such a simulation with the supercript "lom" for (l)aw (o)f (m)otion. In particular, with time step dt the implied distribution forecast evolves according to

$$\begin{aligned} g_{t+dt}^{n,\text{lom}} =& g_t^{\text{lom}} + \left(F_\alpha^n(\alpha_t^{\text{lom}})\mu_\alpha(\rho_t, \alpha_t^{\text{lom}}) + \frac{1}{2}\sigma_\alpha(\rho_t, \alpha_t^{\text{lom}})^T F_{\alpha\alpha}^n(\alpha_t^{\text{lom}})\sigma_\alpha(\rho_t, \alpha_t^{\text{lom}}) \right) dt \\ &+ F_\alpha^n(\alpha_t^{\text{lom}})\sigma_\alpha(\rho_t, \alpha_t^{\text{lom}})\sqrt{dt}\,\epsilon_t = F^n(\alpha_{t+dt}^{\text{lom}}), \end{aligned} \tag{17}$$

where $\{\epsilon_t\}$ is the shock sequence draw.

Alternatively, we can ask how the cross-sectional distribution would evolve between time steps t and $t + dt$ if, instead, we directly simulated household behavior at time t and

then computed g_{t+dt} by aggregating from the micro level. This idea is commonly used to compare the fit of heterogeneous-agent models in the literature. See for example Den Haan (2010). In particular, this approach is akin to a direct Monte Carlo simulation of household behavior at the micro level.

Following this approach, we can construct an alternative simulation which I denote using a "sim" superscript as follows: First, we initialize the economy at the same point and use the same draw of exogenous shocks. Second, at time t, we simulate household behavior at the micro level and update the cross-sectional distribution accordingly. Importantly, household behavior is still computed *as if* households believed the future evolution of the economy to follow the law of motion $d\alpha_t$. This is what makes this approach feasible. Third, while beliefs about the *future* remain as in the "lom" simulation, the macro block must be recomputed to find the prices that clear all markets *today*, using the new cross-sectional distribution $g_t^{n,\text{sim}}$ to evaluate market clearing conditions. Ensuring market clearing in this way is, of course, necessary to obtain a consistent solution.

Consider a fixed point that converges in the sense that, for n sufficiently large, the gap

$$\left|\left|g_t^{n,\text{lom}}(\boldsymbol{x}) - g_t^{n,\text{sim}}(\boldsymbol{x})\right|\right|_{(t,\boldsymbol{x})\in\mathbb{R}\times\mathbb{R}^d}$$

becomes sufficiently small. What would be the significance of convergence in this sense? In this case, approximate economy n is solved with agents behaving according to beliefs that, ex post, coincide sufficiently accurately with the actually realized evolution of the economy. We now have a solution of the model where (i) agents behave optimally given beliefs, (ii) all markets clear, and (iii) beliefs are sufficiently consistent with the true law of motion of the economy, thus satisfying our definition of a recursive equilibrium.

Computing forecast errors. How is the residual forecast error $|g_t^{n,\text{lom}}(\boldsymbol{x}) - g_t^{n,\text{sim}}(\boldsymbol{x})|$ computed in practice? $g_t^{n,\text{lom}}$ can be simulated straightforwardly and very cheaply by using equation (17).

In discrete time, Young (2010) discusses how to correctly simulate the cross-sectional distribution $g_t^{n,\text{sim}}$ via aggregation from simulated behavior at the micro level. A key observation in my context is that time-integration of the Kolmogorov forward equation is the proper continuous-time analog to the Young (2010) method.

In order to use the Kolmogorov forward equation to compute an update $g_{t+dt}^{n,\text{sim}}$, we simply need to compute household policy functions at time t and aggregate over those using the current cross-sectional distribution $g_t^{n,\text{sim}}$. Time step t of this simulation method is therefore structured as follows. At time t, the economy is in aggregate state $\Gamma_t^{n,\text{sim}}$.

(i) I start by projecting the household value function of approximate economy n onto the realized aggregate state. This yields a function $V(\boldsymbol{x}, \Gamma_t^{n,\text{sim}}) : \mathbb{R}^3 \to \mathbb{R}$ which encodes

household beliefs about the evolution of the economy's aggregate state starting from $\Gamma_t^{n,\text{sim}}$.

(ii) Next, I solve the non-linear macro block, following the procedure of Section 2.2 but using the simulated cross-sectional distribution $g_t^{n,\text{sim}}$. The policy functions of households, firms and unions must be recomputed to be consistent with these new prices. In particular, this is done holding beliefs fixed, which in the context of the household problem corresponds to holding $V(\boldsymbol{x}, \Gamma_t^{n,\text{sim}})$ fixed.

(iii) Finally, I use household policy functions, recomputed at the new prices, and plug these into the Kolmogorov forward equation to update the cross-sectional distribution and obtain g_{t+dt}^{sim}.

Step (ii) of this procedure is highly time-consuming because a non-linear equation solver must be employed to clear markets in every time step of every simulation.

The cost of simulation. In practice, the above-described simulation step can easily account for the lion's share of overall computation time, largely because the non-linear macro block must be solved at every time step, of which there are thousands per simulation. Several commonly used tricks, such as efficiently carrying over price guesses from one simulation to the next, can result in substantial performance gains.

The most effective method to speed up the simulation step, from my experience, is to simulate macroeconomic aggregates and the cross-sectional distribution on two different time grids.

Integrating the Kolmogorov forward equation to obtain an update for the cross-sectional distribution, g_{t+dt}^{sim}, typically requires a very fine time grid. To illustrate this point, consider households with liquid asset position $\hat{a}$, and assume that these households' optimal savings decision is $s(\hat{a}, \cdot)$. Households therefore transition out of liquid asset position $\hat{a}$ at rate $s(\hat{a}, \cdot)g_t(\hat{a}, \cdot)$. I abstract from offsetting forces for the purpose of illustration. If the discretized time step dt is too large, then the cumulative flow away from $\hat{a}$ over the course of the time step, given by $s(\hat{a}, \cdot)g_t(\hat{a}, \cdot)dt$, may be larger than the mass of households initially at $\hat{a}$. This happens because the flow rate $s(\hat{a}, \cdot)g_t(\hat{a}, \cdot)$ is held fixed over the course of a discretized time step. In continuous time, on the other hand, this flow rate is continuously adjusted so that the mass of households at any point in the state space can never become negative. In practice, the algorithm consistently breaks as soon as g_{t+dt}^{sim} becomes negative at any point in the state space. Therefore, to prevent this, the discretized time step dt must be sufficiently small.

I develop an adaptive algorithm that coarsens or extends the time grid used for integrating the Kolmogorov forward equation relative to the time grid used for simulating

macroeconomic aggregates whenever g^{sim}_{t+dt} becomes negative. In practice, I run the simulation of macroeconomic aggregates on a grid with 2,000 time steps, representing 1,000 quarters. This is the time grid on which the non-linear macro block is solved. Reducing the number of time steps in this grid can therefore lead to great performance gains.

On the other hand, the grid used for integrating the Kolmogorov forward equation typically has up to 20,000 or more time steps. This grid is adaptively adjusted over the course of the algorithm. Updating $g^{n,\text{sim}}$ is near costless. Consequently, refining this grid has hardly any impact on the runtime of the simulation step.[15]

Updating F^{n+1} to minimize forecast errors. Having first solved approximate economy n and then simulated it to obtain a time series of forecast errors associated with $g_t^{n,\text{lom}}(\boldsymbol{x}) - g_t^{n,\text{sim}}(\boldsymbol{x})$, updating F^{n+1} remains as the last step of outer iteration n. Concretely, this means finding a basis function $T^{n+1}(\boldsymbol{x})$ and using it in (16) to update F^{n+1}.

The key idea I employ here is that, for any candidate $\tilde{T}^{n+1}(\boldsymbol{x})$, we can cheaply recompute the associated $\tilde{F}^{n+1}(\alpha_t^{\text{lom}}) = \tilde{g}_t^{n+1,\text{lom}}$ leveraging Proposition 3 and the integration formula (17).

Formally, I set up the estimation problem

$$\min_{T^{n+1}(\boldsymbol{x}),\, \alpha_t} \left|\left| g_t^{n,\text{sim}}(\boldsymbol{x}) - g^0(\boldsymbol{x}) - \sum_{k=1}^{n+1} T^k(\boldsymbol{x})\alpha_t^k \right|\right|_{\mathbb{L}^2(t\times\boldsymbol{x})} \tag{18}$$

which takes as given $g^0(\boldsymbol{x})$ and $T^k(\boldsymbol{x})$ for $k \leq n$. Intuitively, the goal of this estimation problem is to find the new basis function $T^{n+1}(\boldsymbol{x})$ such that, when we properly recompute all α_t appropriately, we minimize the residual forecast error relative to $g_t^{n,\text{sim}}$. Lemma 11 of Appendix B.6 in the main text solves a variant of this estimation problem. In particular, we get

$$\alpha_t = \Big(T(\boldsymbol{x})'T(\boldsymbol{x})\Big)^{-1} T(\boldsymbol{x})'\Big[g_t^{n,\text{sim}}(\boldsymbol{x}) - g^0(\boldsymbol{x})\Big]. \tag{19}$$

And this solution is, of course, consistent with Proposition 3, which can be seen quickly by taking the derivative $\frac{d}{dt}$ in equation (19). Of course, approximate economy $n+1$ is not actually computed while searching for the efficient T^{n+1}. Instead, $g_t^{n,\text{sim}}$ is used directly here as a proxy, even though we would have to recompute and use $g_t^{n+1,\text{lom}}$ to exactly reflect what household believes would be under F^{n+1}. Subject to this caveat, the estimation problem (18) is therefore set up to find the basis function $T^{n+1}(\boldsymbol{x})$ that minimizes the remaining forecast error if household forecasts were recomputed under the resulting, more flexible representation $F^{n+1}(\alpha)(\boldsymbol{x})$.

[15] To generate my results, I typically run the entire simulation step one last time on a much finer time grid. (For example, to produce the out-of-sample forecast errors in Figure 3.)

2.3.1 Alternatives to the affine structure

The affine structure (16) is very flexible. However, we also have the prior intuition, in the spirit of Krusell and Smith (1998), that keeping track of the aggregate capital stock (as a moment of the distribution) directly might be a highly efficient use of one dimension of distribution representation.

In the context of a representation $g_t(\boldsymbol{x}) \approx F(\alpha_t)(\boldsymbol{x})$, it is not at all straightforward to let one dimension of $F(\alpha_t)$ correspond to a particular moment of the distribution. To circumvent this issue, I develop a variant of my algorithm that first *normalizes* $g_t(\boldsymbol{x})$ by the aggregate capital stock, then uses capital as the first dimension of the distribution representation, and finally constructs $F(\alpha_t)(\boldsymbol{x})$, using the remaining dimensions, to approximate the *normalized* distribution instead.

Formally, let $\omega_t(a, \tilde{k}, z)$ denote the cross-sectional distribution over liquid asset positions a, the *capital share* $\tilde{k}$ owned by the household, and employment state z. The resulting capital share distribution $\omega_t(a, \tilde{k}, z)$ is a direct transformation of $g_t(a, k, z)$ given by

$$\omega_t\left(a, \frac{k}{K_t}, z\right) = g_t(a, k, z),$$

where $\tilde{k} = k/K_t$ is the capital share. In this variant of the algorithm, I add K_t explicitly as an aggregate state variable, and then use the α_t to approximate ω_t.

2.4 Choosing $F(\cdot)$ from a parametric family

While the non-parametric algorithm discussed so far is particularly efficient to approximate $g_t(\boldsymbol{x})$ when $\boldsymbol{x}$ is higher-dimensional, a natural alternative approach is to pick $F(\cdot)$ directly from a parametric family. This is the approach I take when solving simpler models like a one-asset HANK model or the Krusell and Smith (1998) model, which I have solved with a $F(\alpha)$ representation consisting of over 20 Chebyshev polynomials.

In particular, when $F(\cdot)$ is chosen ex ante and not updated, the outer fixed point of my algorithm (**Level 3**) collapses to a single iteration. More surprisingly yet, no simulation step is required at any point in the resulting parametric algorithm.

For illustration, consider the concrete example where $x = \boldsymbol{x}$ is one-dimensional and we decide ex ante to represent the cross-sectional distribution using 10 Chebyshev polynomials. That is, we set $F^1(\alpha_t)(x) = g^0(x) + \sum_{n=1}^{10} T^n(x)\alpha_t^n$, where T^n is the nth Chebyshev polynomial. Recall that Proposition 3 allows us to compute the internally consistent law of motion for α_t as part of the household VFI. Therefore, all we need to do is run **Level 1** and **Level 2** of the algorithm *once*. This yields a solution of the approximate economy that corresponds to the desired Chebyshev polynomial approximation. In particular, all

markets clear, agents behave optimally under the belief that the cross-sectional distribution evolves according to $d\alpha_t$, and $d\alpha_t$ is, in fact, the law of motion implied by the approximate economy's solution. No simulation step is required.

For models like Krusell and Smith (1998), where no general equilibrium prices must be found to clear markets, not even **Level 2** of the algorithm is necessary. Indeed, a parametric variant of my algorithm solves the Krusell and Smith (1998) model in a *single* value function iteration step!

3 Algorithm Performance and Implementation Details

In this section, I discuss the performance of the algorithm. I also provide additional details on how to implement the algorithm of Section 3 in practice.

3.1 Algorithm convergence

In practice, I use the non-parametric algorithm discussed in Section 2.3 to globally solve the two-asset HANK model of my paper. For the main calibration, I work with distribution representations with dimensionality between 3 and 5. In other words, in practice I use

$$F^N(\alpha_t)(\boldsymbol{x}) = g^0(\boldsymbol{x}) + \sum_{n=1}^{N} T^n(\boldsymbol{x})\alpha_t^n$$

for N between 3 and 5. I discuss this basis function representation, as well as the construction of $\{T^n\}$, in detail in Section 2.3. This implies that the household problem has 8 to 10 state variables, while the firm and union problems both have 6 to 8 state variables.[16]

The first important question I want to address is whether this algorithm converges as we increase N.[17] Intuitively, convergence in this context means that household beliefs about the evolution of the cross-sectional distribution in approximate economy N become increasingly close to the actual evolution of the distribution when the economy is solved and simulated under these beliefs.

One approach to evaluate the convergence properties of my algorithm is by looking at households' forecast errors across the sequence of approximate economies. I will further discuss forecast errors in Section 3.2. An alternative and more illustrative approach is to directly compare the model solutions across approximate economies. I take this

[16] These numbers correspond to the model used for the experiments of Section 5 in the main text. In particular, the vector of state variables for the household problem is $(a, k, z, \rho, \sigma_\rho, \alpha)$, where σ_ρ is added for the transmission mechanism analysis in Section 5.1.

[17] I am still working on analytical convergence results for certain classes of basis functions.

approach in Figure 2, which compares solutions for output, $Y^N(\rho, \alpha)$, across a sequence of approximate economies. If the distance between Y^N and Y^{N+1} becomes increasingly small as N becomes large, then, heuristically, we can say that the algorithm converges. For purposes of illustration, Figure 2 plots output for all possible realizations of the discount rate shock, ρ, while holding all other aggregate state variables associated with the distribution representation, α, fixed at their mean, $\bar{\alpha}$. In other words, Figure 2 can also be interpreted as plotting the on-impact response of output to a shock ρ if we initialize the cross-sectional distribution at the risky steady state.

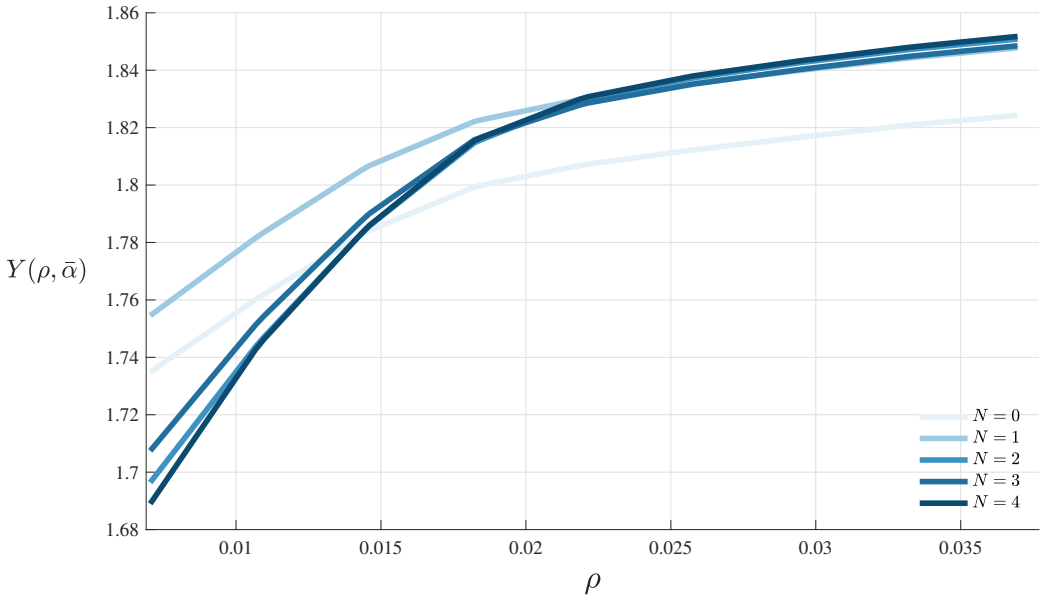
Figure 2 plots solutions for output across approximate economies with N, the maximum dimensionality of the distribution representation, ranging between 0 and 4. The $N = 0$ case, for example, corresponds to a solution of the model under the restriction that agents believe the cross-sectional distribution to be constant. As I discuss in detail in Section 2.3, $N = 1$ is a special case. In the spirit of Krusell and Smith (1998), we have a strong prior that the aggregate capital stock should already encapsulate much of the information households would otherwise glean from the entire cross-sectional distribution. I therefore follow Algan et al. (2008) to make the Krusell and Smith (1998) idea of directly keeping track of the capital stock operational in my framework. See Section 2.3.1 for details. The $N = 1$ case should therefore be interpreted as the analog of Krusell and Smith (1998) in my framework. The remaining cases, $N > 1$, precisely follow the non-parametric algorithm discussed in Section 2.3.[18]

Figure 2 suggests three noteworthy observations:

1. Approximate aggregation, in the sense of Krusell and Smith (1998), yields an accurate description of aggregate behavior in the model *during normal times*. For $N \geq 1$, the solutions for output lie on top of one another for $\rho > 0.025$. In this part of the state space, the economy is sufficiently far from the ZLB. Indeed, this finding suggests that the Krusell and Smith (1998) algorithm would deliver a reasonably accurate global solution of my two-asset HANK model in the absence of an occasionally binding macro constraint. It is only for ρ sufficiently low that my algorithm yields results that differ from the Krusell and Smith (1998) solution.

[18] In the current version of the paper, I work with distribution representations $F(\alpha_t)(\boldsymbol{x})$ that effectively hold unemployment fixed. Households have no reason to directly forecast the aggregate unemployment rate because it does not directly affect them. Households care about employment transition rates, which are a function of economic activity in this model rather than the aggregate level of unemployment. Conditional on households correctly forecasting the set of relevant prices, therefore, this restriction would be entirely innocuous. Indeed, I show below in Section 3.2 that my algorithm already implies accurate forecasts about economic activity, which suggests that relaxing this restriction would have no substantial effect on household behavior and, therefore, the rest of the model. That said, I am currently working on a solution of the model that relaxes this assumption.

Figure 2: Convergence of a sequence of approximate economies



Notes. The figure displays a state space representation of output, $Y^N(\rho, \bar{\alpha})$, plotted against all possible realizations of the discount rate shock, ρ, for a sequence of approximate economies indexed by N. I hold fixed (or project) all other dimensions of the aggregate state space at their mean, $\bar{\alpha}$. The figure illustrates visually that the sequence of approximate economies that I solve as part of my non-parametric algorithm converges. The $N = 0$ case corresponds to the approximate economy where households believe the cross-sectional distribution to be time-invariant. The $N = 1$ case corresponds, roughly speaking, to a solution of my model under the Algan et al. (2008) algorithm with the aggregate capital stock used as the only moment. The cases with $N > 1$ enlarge the distribution representation by using non-parametrically estimated basis functions. The figure highlights that approximate aggregate holds roughly during "normal times": For $\rho > 0.025$ the economy is sufficiently far away from the ZLB that macroeconomic behavior becomes linear. The $N = 1$ solution achieves a good fit, which can be seen from the coincidence of all lines with $N \geq 1$. Approximate aggregation fails badly during "crisis times", however: As ρ falls, the $N = 1$ solution attains an increasingly bad fit.

2. A Krusell and Smith (1998) solution of my model, the $N = 1$ case, implies substantially dampened macro uncertainty during crises. Intuitively, the endogenous volatility of output corresponds to the slope of the lines plotted in Figure 2, or $\sigma_{Y,t} \sim \frac{d}{d\rho} Y(\rho_t, \alpha_t)$. My algorithm converges to a solution of the model in which the sensitivity of economic activity to discount rate shocks (i.e. the slope of $Y(\rho, \cdot)$) is substantially larger close to and in the ZLB crisis region. This suggests that a Krusell and Smith (1998) solution of the model would considerably under-predict

the magnitude of endogenous uncertainty spikes during crises.

3. Figure 2 also suggests that the non-parametric algorithm converges - at least heuristically. The distance between Y^N and Y^{N+1} clearly decreases as N becomes larger. In fact, around $N = 3$, the only visually noticable discrepancy occurs for the lowest levels of ρ. This makes sense intuitively: We expect the largest degree of non-linearity in the model's aggregate behavior when the economy is in a deep recession. And this is exactly the region of the state space where approximate aggregation fails most obviously. These considerations suggest that, while the overall fit and convergence of my algorithm at $N = 4$ may already be acceptable, we would require an even finer distribution representation to increase the accuracy of forecasts in the crisis region. Section 3.2 will reach a similar conclusion by looking directly at household forecast errors in normal and crisis times.

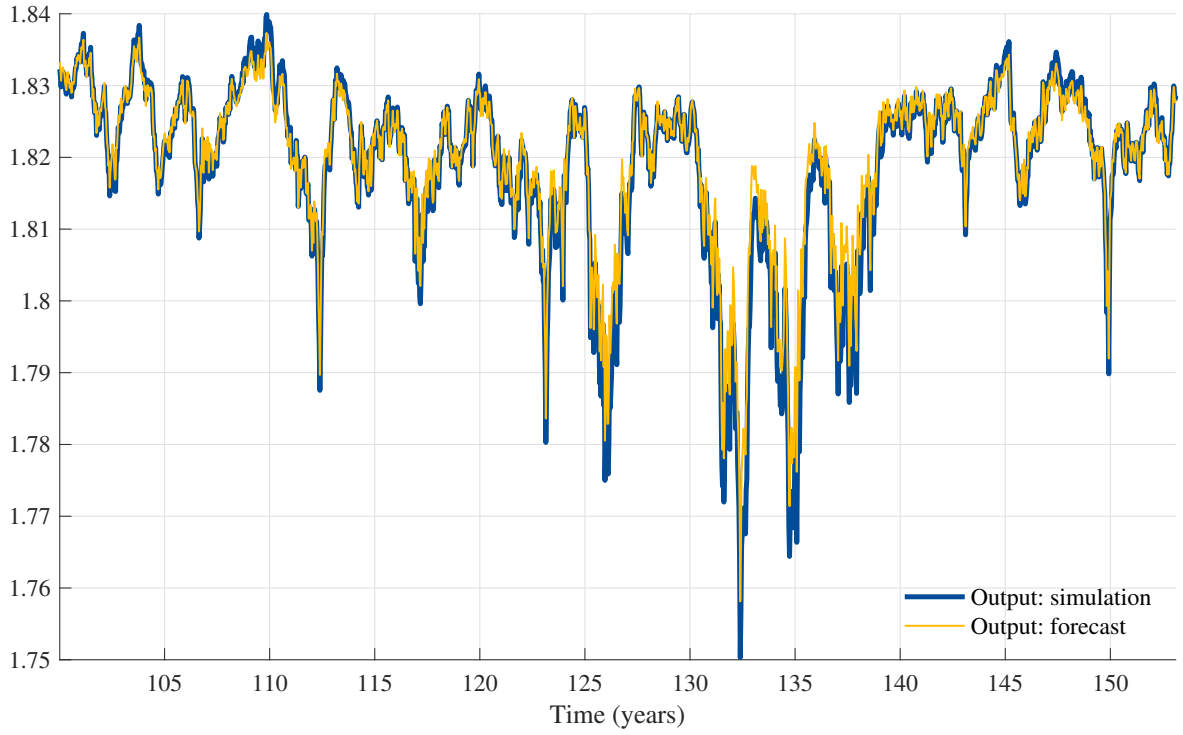
3.2 Algorithm fit

The most commonly used quantitative metric to assess the fit of solution methods for heterogeneous-agent macro models is the Den Haan (2010) metric. My implementation of this error metric compares two different simulations of the model for 8,000 time steps. The first simulation computes the evolution of the aggregate state directly from the law of motion implied by the $F^N(\alpha_t)(\boldsymbol{x})$ distribution representation. This simulation is equivalent to *household forecasts* because the household problem is solved with beliefs precisely consistent with the same law of motion. I denote this simulated time series for output by Y_t^{lom} and for the cross-sectional distribution by $g_t^{\text{lom}} = F^N(\alpha_t^{\text{lom}})$.

The second simulation initializes the economy in the risky steady state. Instead of using prices that are consistent with household forecasts, this simulation solves the non-linear macro block of the model at every time step to find the prices that clear all markets. These prices are then plugged into the household problem to compute policy functions. The cross-sectional distribution is then updated by directly using and aggregating the behavior of households at the micro level under the market-clearing set of prices. In this sense, this simulation is akin to a Monte Carlo simulation of household behavior at the micro level.[19] The time series of the cross-sectional distribution obtained from this simulation, g_t^{sim}, represents in some sense the correct behavior of the model because we ensure that markets clear in every time step. I denote the simulated time series for output by Y_t^{sim}.

[19] Indeed, the simulation algorithm I develop is the continuous-time analog of the Young (2010) method.

Figure 3: Den Haan (2010) forecast accuracy metric



Notes. This figure plots the time series for output from two distinct simulations of the model that use the same draw of exogenous discount rate shocks. Both simulations initialize the economy in the risky steady state and simulate over a horizon of 250 years with 8000 discretized time steps. In the first simulation (**blue**), the non-linear macro block is solved at each time step to find the prices that clear markets. Household behavior is then computed using these prices, and the cross-sectional distribution is updated by aggregating household behavior from the micro level. In the second simulation (**yellow**), the evolution of both prices and the cross-sectional distribution is evaluted directly from the law of motion implied by $F^N(\alpha_t)$. This second simulation corresponds precisely to how households would forecast the economy to evolve in response to the exogenously drawn shock series. The discrepancy between the **blue** and **yellow** lines therefore directly represents household forecast errors.

The Den Haan (2010) metric is then formally defined as

$$\epsilon_Y^{\text{DH}} = 100 \times \max_t \left| \log Y_t^{\text{sim}} - \log Y_t^{\text{lom}} \right|.$$

I illustrate the time series of household forecast error in Figure 3 by directly plotting Y_t^{sim} and Y_t^{lom} for the same underlying draw of shocks. The Den Haan (2010) metric would correspond to the maximum realization of the log difference between the blue line, Y_t^{sim}, and yellow line, Y_t^{lom}.

Importantly, this exercise is done *out of sample*. That is, I first run the non-parametric algorithm to find a distribution representation $F^N(\alpha_t)$ while holding fixed a given draw of exogenous shocks from one outer iteration to the next. With a solution of the model in hand, I then use a different draw of exogenous shocks for the simulations in Figure 3. Similarly, household forecasts are made at the beginning of time, $t = 0$. That is, households never condition their forecasts on any object associated with the simulation of g_t^{sim}.

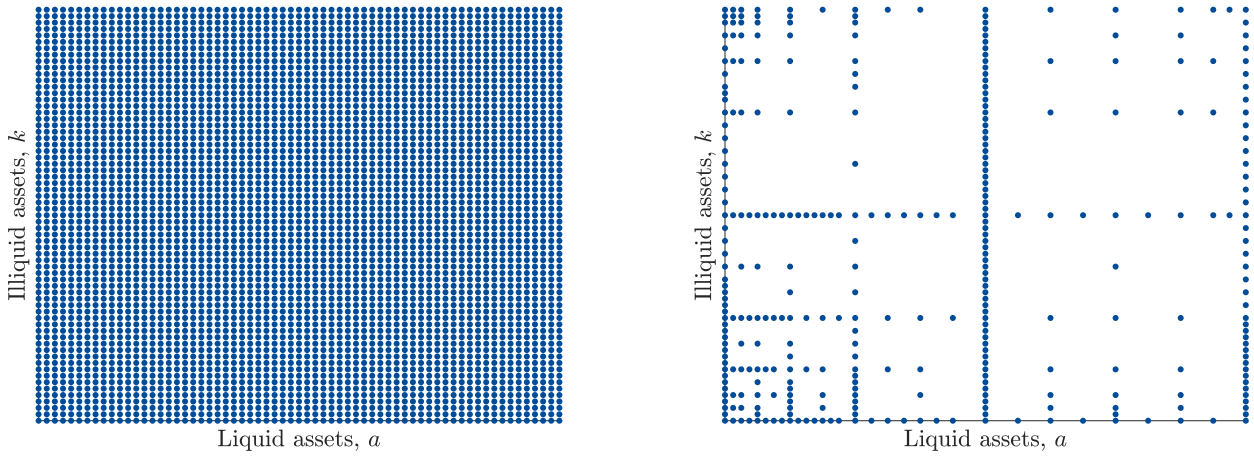
Figure 3 clearly illustrates that my solution method achieves a very good fit during normal times but still implies meaningful forecast errors during deep recessions. Concretely, when output is close to its median realization of 1.82, the blue and yellow lines typically lie almost exactly on top of one another. Forecast errors even remain small during moderate economic downturns. For example, consider the sample years 112 and 150. It is only during the deepest and most protracted recessions in this sample that forecast errors grow larger. This is, of course, consistent with my discussion of Figure 2: The behavior of macroeconomic aggregates in my model is roughly linear during normal times when the economy is sufficiently far from the ZLB. In this region of the state space, lower-dimensional distribution representations already achieve a good fit because approximate aggregation roughly holds. Closer to the crisis region, however, my model exhibits substantially non-linear dynamics. This requires a higher-dimensional distribution representation. While my method with N between 3 and 5 already achieves an improved fit relative to lower-dimensional representations (also see Figure 2), it remains an ongoing effort to further scale up the dimensionality of my solution and attain an even better forecast accuracy in the crisis region.

3.3 Gains from sparse grid adaptation

As part of my solution method, three value function iteration algorithms – corresponding to the household, firm and union problems – have to be implemented thousands of times. A complete "production run" that solves all versions of my model that I use for analysis and computes all my numerical results typically features hundreds of outer iterations (**Level 3**). And within each outer iteration n, solving for general equilibrium prices (**Level 2**) requires successively implementing the household, firm and union problems many times.[20]

[20]For example, model implementations that can be run on a personal workstation (in my case a 2019 MacBook Pro) will typically feature up to 100 aggregate states. This number can be considerably larger when the code is run on a cluster. Constructing the Jacobian matrix for the quasi-Newton algorithm once requires solving the household, firm and union VFIs as many times as there are aggregate states. My quasi-Newton method reconstructs this Jacobian matrix only a fraction of the times that a standard Newton method would require. Nonetheless, even a few dozen reconstructions of this matrix over the course of the entire production

Figure 4: Gains from grid adaptation



(a) Dense grid

(b) Adapted sparse grid

Notes. Panels (a) and (b) plot a dense grid and a highly adapted sparse grid over the two dimensions (a, k). The adapted grid takes as its baseline the dense grid in panel (a) and removes grid points according to the most stringent adaptation threshold value I use in practice. For illustration, no points are added that are not already on the dense grid.

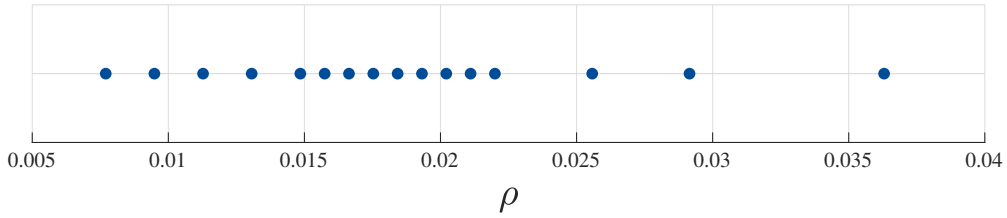
I leverage an adaptive sparse grid (ASG) toolbox for solving high-dimensional partial differential equations which is developed in a separate paper (Schaab and Zhang, 2020). Every value function iteration algorithm that is computed as part of my solution method is implemented on a highly adapted sparse grid which features only a fraction of the grid points that the associated dense grid would require. In this subsection, I provide some implementation details in the context of my application. For additional details, please refer to Schaab and Zhang (2020).

Idiosyncratc and aggregate states. When implementing a value function iteration algorithm with grid adaptation, it is typically helpful to distinguish between idiosyncratic and aggregate states. In a mathematical sense, an agent's value function is likely to feature significantly more concavity in an idiosyncratic dimension. Consequently, a much finer grid structure is required in these dimensions.

For several technical reasons that follow from this distinction, the idiosyncratic and aggregate dimensions of the grid are typically adapted separately. The current version of my algorithm simply adapts the idiosyncratic grid at the very beginning of the algorithm

run imply that all three VFIs must be solved thousands if not tens of thousands of times.

Figure 5: Clustering aggregate states around cusp of ZLB



Notes. The figure plots the unique values of ρ represented in a typical grid over the aggregate states (ρ, α). Grid points are densely clustered around the cusp of the ZLB.

based on the household value function in the deterministic steady state. This takes place even before outer iteration $n = 1$ is initialized. Figure 4 displays a pair of grids over the two idiosyncratic dimensions (a, k). In particular, the grid displayed in panel (b) takes the dense grid of panel (a) as its starting point and then removes points based on residual concavity in the household value function. For illustration, panel (b) uses the largest threshold value for grid point removal that I actually use in practice to solve the two-asset HANK model. In other words, I use adapted sparse grids over idiosyncratic states that are similar to but *no sparser* than the grid displayed in panel (b).

To construct the full grid, I use a tensor product that, roughly speaking, replicates the fully adapted grid over the idiosyncratic states for each aggregate state. In practice, I use no more than $J^{\text{agg}} = 100$ efficiently placed aggregate states when running the code interactively on a personal workstation. This number can be larger when running the code on a cluster. For example, Figure 5 plots the unique ρ values that are represented in a typical grid. Grid points are clustered around the cusp of the ZLB. In practice, I am currently not able to implement grids that are finer in the ρ dimension, with maximal adaptation level $l = 5$, than the grid in Figure 5.

3.4 Being stingy with Jacobians

My algorithm never reconstructs a Jacobian matrix unless absolutely necessary. As discussed in Section 2.2, I use a flexible and robust quasi-Newton method that can recycle previously computed Jacobians as a starting guess. In practice, this delivers substantial performance gains relative to a standard Newton algorithm. I recycle previously computed Jacobian matrices not only within the quasi-Newton step to solve for GE prices, but also across outer iterations.

Similarly, across outer iterations, every object computed in iteration n is used as a

starting guess for iteration $n+1$. This includes general equilibrium prices, value functions, simulated time series, Jacobians, and so on.

4 Transmission Decompositions with Aggregate Risk

Kaplan et al. (2018) have developed a seminal method for decomposing the transmission mechanism of exogenous shocks in the context of heterogeneous-agent models solved without aggregate risk and using a sequence-space representation. Their method has since been employed by numerous papers.

As a distinct methodological contribution, my paper develops an analog to their method in the context of models with aggregate risk that are solved globally using a state-space representation.

4.1 Sequence-space representation

I start by showing that the method of Kaplan et al. (2018) can be applied *literally* to global solutions of models with aggregate risk but then loses its practicality in the process.

To illustrate, consider the baseline model where the household problem state space is given by (a, k, z, ρ, α). As before, the household problem gives rise to the policy function $c(a, k, z, \rho, \alpha)$. This policy function is expressed using the state-space representation associated with the recursive equilibrium of the model.

The sequence-space representation is simply $\tilde{c}_t(a, k, z)$. Importantly, the aggregate state variables are simply collapsed into the t subscript. That is,

$$\tilde{c}_t(a, k, z) = c(a, k, z, \rho_t, \alpha_t).$$

The household problem only depends on t (or, for that matter, the aggregate state variables) through the general equilibrium prices that households are exposed to, such as the interest rate $r(\cdot)$ or the wage rate $w(\cdot)$. I use X to denote the set of all macroeconomic aggregates that households are exposed to.

Consider household consumption at time 0 in response to a (possibly trivial) exogenous perturbation $\{d\rho_t\}_{t\geq 0}$. Without aggregate uncertainty, in the spirit of Kaplan et al. (2018), we can relate the dependence of $c(\cdot)$ on prices and aggregate states via

$$\tilde{c}_0 = \tilde{c}\Big(\boldsymbol{x}_0, \{X(\rho_t, \alpha_t)\}_{t\geq 0}\Big) = \hat{c}\Big(\boldsymbol{x}_0, \{\rho_t, \alpha_t\}_{t\geq 0}\Big).$$

A household's consumption response at time $t = 0$ to the perturbation $\{d\rho_t\}_{t\geq 0}$ is a function of the household's initial idiosyncratic state, $\boldsymbol{x}_0$, and the future paths of all prices, $X_t = X(\rho_t, \alpha_t)$. Again, and importantly, this is the case without aggregate risk where ρ_t is

not stochastic beyond this initial perturbation. Finally, the second equality maintains that household consumption can be equivalently represented using the paths of state variables ρ_t and α_t. In conclusion, if we take partial derivatives here as in Kaplan et al. (2018), we have a straightforward and direct mapping between $d\rho_t$ and the set of prices dX_t.

Now consider the substantially more complicated case with aggregate risk. To do so formally, we require additional notation and structure. Let Ω denote the probability space induced by the Brownian motion B_t that is the model's sole aggregate risk factor, with $\omega \in \Omega$. The Brownian motion B_t induces a *law* over all possible sample paths $\{X_t(\omega)\}_{t\geq 0}$. When we now change or "shock" the model's initial condition, ρ_0, then the probability spaced Ω induced by B remains unchanged. However, the stochastic processes X_t or, put differently, the mapping

$$\omega \mapsto \{X_t(\omega)\}_{t\geq 0}$$

changes. And it is *this effect* on the probability distribution of future prices which should be captured by the transmission decomposition.

With aggregate risk, we now have

$$\tilde{c}_0 = \tilde{c}\Big(\boldsymbol{x}_0, \{X_t(\omega)\}_{t\geq 0,\omega\in\Omega}\Big) = \hat{c}\Big(\boldsymbol{x}_0, \{\rho_t(\omega), \alpha_t(\omega)\}_{t\geq 0,\omega\in\Omega}\Big).$$

In the sequence-space framework, household consumption at time $t = 0$ again depends on the aggregate state variables indirectly via prices. However, to consistently formulate this sequence-space representation, the household's initial consumption response now depends on *all possible sample paths* for prices, i.e. $\{X_t(\omega)\}_{t\geq 0,\omega\in\Omega}$.[21]

This representation generalizes the sequence-space framework implicit in the Kaplan et al. (2018) method. In their case, everything simplifies substantially, of course, because, with no aggregate risk, the probability space Ω is degenerate, and exactly one of the potential sample paths is realized *almost surely*. Therefore, policy functions in their setting only take as an input the single sample path of prices that is realized almost surely in the absence of aggregate risk.

Consider, finally, a variant of the partial derivative exercise used to decompose the initial response dc_0 when there is no aggregate risk. To simplify notation, assume that there is only a single general equilibrium price to which households are exposed, so that a perturbation $dX_t(\omega)$ can be interpreted more easily as "the change in price". We have

$$\frac{dc_0}{d\rho_0} = \int_0^\infty \int_\Omega \frac{\partial c_0}{\partial X_t(\omega)} \frac{dX_t(\omega)}{d\rho_0} d\omega dt,$$

where I now abuse notation by dropping the hat and tilde notation. The key takeaway here is that, since a sequence-space representation of $c_0(\cdot)$ depends on all possible sample

[21] In particular, $c(\cdot)$ is now a functional over the function space on prices $X_t(\omega)$ induced by B.

paths of prices, computing the response dc_0 similarly requires accounting for the partial effects through all possible sample paths for price $X_t(\omega)$.

Put slightly differently, and using language that is potentially more familiar, $\Omega = \{\omega\}$ is the set of all possible histories. Without aggregate risk, a single history is realized almost surely. Therefore, the integral $\int_\Omega$ becomes degenerate (in the sense of a Dirac function on the single history that is realized almost surely). This simplification is precisely what makes sequence-space representations feasible and highly effective in the absence of aggregate risk. With aggregate risk, however, the exercise in Kaplan et al. (2018), if taken literally, would compute the partial transmission of $d\rho_0$ through every possible future history of the single price $X_t(\omega)$. This makes the sequence-space representation less useful when there is aggregate risk.

4.2 State-space representation

Consider again the perturbation $d\rho_0$. In the state-space representation of the model, $d\rho_0$ is simply a particular move across the aggregate state space. I now denote by $X(\rho, \alpha)$ the relevant set of general equilibrium prices across a space (or grid) over (ρ, α). And a solution of the model yields the policy function $c(\boldsymbol{x}, \rho, \alpha)$.

With an eye towards implementation, let $X = \{X^i\}_i$, so that we can refer to a particular price X^i. Our goal is still, of course, to decompose the total behavioral response $dc_0/d\rho_0$ into a set of partial channels working through the prices X^i. Taking inspiration from the sequence-space representation, we can of course still write

$$\frac{dc_0}{d\rho_0} = \int_0^\infty \int_\Omega \sum_i \frac{\partial c_0}{\partial X_t^i(\omega)} \left[\frac{\partial X_t^i(\omega)}{\partial \rho_t(\omega)} \frac{d\rho_t(\omega)}{d\rho_0} + \frac{\partial X_t^i(\omega)}{\partial \alpha_t(\omega)} \frac{d\alpha_t(\omega)}{d\rho_0} \right] d\omega dt.$$

Next, we map back to the state-space representation, where we simply have

$$\frac{\partial X_t^i(\omega)}{\partial \rho_t(\omega)} = X_\rho^i(\rho_t(\omega), \alpha_t(\omega)).$$

That is, the partial response $\rho_t(\omega)$ has on $X_t^i(\omega)$ is simply the partial derivative of $X^i(\rho, \alpha)$ with respect to ρ but *evaluated at* $(\rho_t(\omega), \alpha_t(\omega))$.

The key idea behind my method is that there is a very simple way to compute the partial effect on c_0 through all possible sample paths $\{X_t^i(\omega)\}$ for price i. This idea exploits the fact that, under the state-space representation, X is simply a function of (ρ, α) and we can "hold fixed" its response to $d\rho_0$ by projecting $X(\rho, \alpha)$ onto $X(\bar{\rho}, \alpha)$. My approach still delivers a comprehensive decomposition of $d\rho_0$'s transmission through each price, adding together the effects of contemporaneous and future price changes (through the probability measure).

We can compute this partial effect in three steps. I will illustrate how to isolate the partial effect of $d\rho_0$ through the price X^i. First, solve the model and obtain the prices $X^i(\rho, \alpha)$. Second, recompute the household value function taking all prices X^{-i} as given, but imposing the projection

$$X^i(\rho, \alpha) = X^i(\bar{\rho}, \alpha),$$

where $\bar{\rho}$ is the value of ρ in the risky steady state. This is the key step. Intuitively, we solve again for household behavior *as if* changes in ρ had the usual effect on all prices X^{-i} but no effect on X^i. Crucially, this corresponds to the *partial* derivatives above because, by holding other prices X^{-i} fixed at the true solution, we do not capture other general equilibrium effects, which might pollute the partial decomposition.

Finally, we run two simulations of the model both initialized in the aggregate state $(\bar{\rho} + d\rho_0, \bar{\alpha})$. In the first case, we simulate household behavior at the micro level using the value function from the unrestricted model solution and aggregate to find dC_0, the on-impact response of total consumption. In the second case, we use the modified household value function to compute consumption behavior at the micro level such that households act as though price X^i did not exhibit a direct response to the underlying shock. Aggregating yields dC_0^i. Lastly, we can directly compare

$$dC_0 - dC_0^i,$$

which represents the partial contribution to dC_0 from households responding only and directly to the price change dX^i, in the sense $\{dX_t^i(\omega)\}_{t\geq 0, \omega \in \Omega}$, induced by the exogenous perturbation $d\rho_0$.

Notice that we can use this method for any aggregate variable that enters the household problem. In particular, we can let X^0 denote the aggregate shock ρ itself, in which case $dC_0 - dC_0^0$ captures, in the language of Kaplan et al. (2018), the *direct effect* of the shock $d\rho$.

References

Achdou, Y., J. Han, J.-M. Lasry, P.-L. Lions, and B. Moll (2015). Heterogeneous agent models in continuous time.

Ahn, S., G. Kaplan, B. Moll, T. Winberry, and C. Wolf (2017). Micro Heterogeneity and Aggregate Consumption Dynamics. *NBER Macroeconomics Annual, forthcoming*.

Algan, Y., O. Allais, and W. J. Den Haan (2008). Solving heterogeneous-agent models with parameterized cross-sectional distributions. *Journal of Economic Dynamics and Control 32*(3), 875–908.

Auclert, A., M. Rognlie, and L. Straub (2020). Micro jumps, macro humps: Monetary policy and business cycles in an estimated HANK model. Technical report, National Bureau of Economic Research.

Demmel, J. W. (1997). *Applied numerical linear algebra*. SIAM.

Kaplan, G., B. Moll, and G. L. Violante (2018). Monetary policy according to HANK. *American Economic Review 108*(3), 697–743.

Krusell, P. and A. A. J. Smith (1998). Income and Wealth Heterogeneity in the Macroeconomy. *Journal of Political Economy 106*(5), 867–896.

LeVeque, R. J. (2007). *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM.

Reiter, M. (2010). Solving the incomplete markets model with aggregate uncertainty by backward induction. *Journal of Economic Dynamics and Control 34*(1), 28–35.

Saad, Y. and M. H. Schultz (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing 7*(3), 856–869.

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*(1), 36–41.