SUPPLEMENT TO "FINANCIAL FRICTIONS AND THE WEALTH DISTRIBUTION"

(*Econometrica*, Vol. 91, No. 3, May 2023, 869–901)

JESÚS FERNÁNDEZ-VILLAVERDE
Department of Economics, University of Pennsylvania, NBER, and CEPR

SAMUEL HURTADO
Banco de España

GALO NUÑO
Banco de España

THIS APPENDIX compiles further details about the equilibrium conditions, our solution method, and additional results not reported in the main text.

APPENDIX A: EQUILIBRIUM CONDITIONS

An equilibrium in this economy is composed of a set of prices $\{w_t, rc_t, r_t, r_t^k\}_{t\geq 0}$, quantities $\{K_t, N_t, B_t, \widehat{C}_t, c_{it}\}_{t\geq 0}$, and a density $\{g_{it}(\cdot)\}_{t\geq 0}$ for $i \in \{1, 2\}$ such that:

1. Given w_t, r_t, and g_t, the solution of household m's problem (10) is $c_{mt} = c_i(a_t, G_t, N_t)$.
2. Given r_t^k, r_t, and N_t, the solution of the expert's problem (6) is $\widehat{C}_t$, K_t, and B_t.
3. Given K_t, the firm maximizes its profits and input prices are given by w_t and rc_t and the rate of return on capital by r_t^k.
4. Given w_t, r_t, and c_{it}, g_{it} is the solution of the KF equation (15).
5. Given r_t, g_{it}, and B_t, the debt market (11) clears and $N_t = K_t - B_t$.

We stack all the equilibrium conditions of the model (except the optimality condition for households) in two blocks. The first block includes all the variables that depend directly on N_t, B_t, and dZ_t:

$$w_t = (1-\alpha)K_t^{\alpha}, \tag{28}$$

$$rc_t = \alpha K_t^{\alpha-1}, \tag{29}$$

$$r_t = \alpha K_t^{\alpha-1} - \delta - \sigma^2 \frac{K_t}{N_t}, \tag{30}$$

$$dr_t^k = (rc_t - \delta)\,dt + \sigma\,dZ_t, \tag{31}$$

$$dN_t = \left(\alpha K_t^{\alpha-1} - \delta - \widehat{\rho} - \sigma^2\left(1 - \frac{K_t}{N_t}\right)\frac{K_t}{N_t}\right)N_t\,dt + \sigma K_t\,dZ_t. \tag{32}$$

Jesús Fernández-Villaverde: jesusfv@econ.upenn.edu
Samuel Hurtado: samuel.hurtado@bde.es
Galo Nuño: galo.nuno@bde.es

 https://doi.org/10.3982/ECTA18180

The second block includes the equations determining the aggregate consumption of the households, dB_t, dK_t, and $\frac{\partial g_{it}}{\partial t}$:

$$C_t = \sum_{i=1}^{2} \int c_i(a, G_t, N_t)\, dG_t(a, i), \tag{33}$$

$$dB_t = \left((1-\alpha) K_t^{\alpha} + \left(\alpha K_t^{\alpha-1} - \delta - \sigma^2 \frac{K_t}{N_t} \right) B_t - C_t \right) dt, \tag{34}$$

$$dK_t = dN_t + dB_t, \tag{35}$$

$$\frac{\partial g_{it}(a)}{\partial t} = -\frac{\partial}{\partial a}\big(s_i(a, G_t, N_t) g_{it}(a)\big) - \lambda_i g_{it}(a) + \lambda_j g_{jt}(a), \quad i \neq j = 1, 2. \tag{36}$$

The second block shows (i) how the density $\{g_{it}(\cdot)\}_{t \geq 0}$ for $i \in \{1, 2\}$ matters to determine C_t, (ii) that C_t pins down dB_t, and (iii) that once we have dB_t, we can calculate dK_t. Hence, computing the equilibrium of this economy is equivalent to finding C_t. Once C_t is known, all other aggregate variables follow directly.

APPENDIX B: NUMERICAL ALGORITHM

We describe the numerical algorithm used to solve for the equilibrium value function, $v_i(a, B, N)$, the density $g_i(a, B, N)$, and the aggregate debt B and equity N. The algorithm proceeds in three steps. We describe each step in turn.

Step 1: Solution to the Hamilton–Jacobi–Bellman Equation

The HJB equation is solved using an *upwind finite difference* scheme. It approximates the value function $v_i(a, B, N)$, $i = 1, 2$ on a finite grid with steps Δa, ΔB, ΔN: $a \in \{a_1, \ldots, a_J\}$, $B \in \{B_1, \ldots, B_L\}$, $N \in \{N_1, \ldots, N_M\}$, where:

$$\begin{aligned} a_j &= a_{j-1} + \Delta a = a_1 + (j-1)\Delta a, \quad 2 \leq j \leq J, \\ B_l &= B_{l-1} + \Delta B = B_1 + (l-1)\Delta L, \quad 2 \leq l \leq L, \\ N_m &= N_{m-1} + \Delta N = N_1 + (m-1)\Delta N, \quad 2 \leq m \leq M. \end{aligned}$$

The lower bound in the wealth space is $a_1 = 0$, such that $\Delta a = a_J/(J-1)$. We use the notation $v_{i,j,l,m} \equiv v_i(a_j, B_l, N_m)$, and similarly for the policy function $c_{i,j,l,m}$. The derivatives are evaluated according to

$$\begin{aligned} \frac{\partial v_i(a_j, B_l, N_m)}{\partial a} &\approx \partial_f v_{i,j,l,m} \equiv \frac{v_{i,j+1,l,m} - v_{i,j,l,m}}{\Delta a}, \\ \frac{\partial v_i(a_j, B_l, N_m)}{\partial a} &\approx \partial_b v_{i,j,l,m} \equiv \frac{v_{i,j,l,m} - v_{i,j-1,l,m}}{\Delta a}, \\ \frac{\partial v_i(a_j, B_l, N_m)}{\partial B} &\approx \partial_B v_{i,j,l,m} \equiv \frac{v_{i,j,l+1,m} - v_{i,j,l,m}}{\Delta B}, \\ \frac{\partial v_i(a_j, B_l, N_m)}{\partial Z} &\approx \partial_N v_{i,j,l,m} \equiv \frac{v_{i,j,l,m+1} - v_{i,j,l,m}}{\Delta N}, \end{aligned}$$

$$\frac{\partial^2 v_i(a_j, B_l, N_m)}{\partial N^2} \approx \partial^2_{NN} v_{i,j,l,m} \equiv \frac{v_{i,j,l,m+1} + v_{i,j,l,m-1} - 2v_{i,j,l,m}}{(\Delta N)^2}.$$

At each point of the grid, the first derivative with respect to a can be approximated with a forward (f) or a backward (b) approximation. In an upwind scheme, the choice of forward or backward derivative depends on the sign of the *drift function* for the state variable, given by

$$s_{i,j,l,m} \equiv w_{l,m} z_i + r_{l,m} a_j - c_{i,j,l,m}, \tag{37}$$

where

$$c_{i,j,l,m} = \left[\frac{\partial v_{i,j,l,m}}{\partial a}\right]^{-1/\gamma}, \tag{38}$$

$$w_{l,m} = (1-\alpha) Z (B_l + N_m)^{\alpha}, \tag{39}$$

$$r_{l,m} = \alpha Z (B_l + N_m)^{\alpha-1} - \delta - \sigma^2 \frac{(B_l + N_m)}{N_m}. \tag{40}$$

Let superscript n denote the iteration counter. The HJB equation is approximated by

$$\begin{aligned}
&\frac{v^{n+1}_{i,j,l,m} - v^n_{i,j,l,m}}{\Delta} + \rho v^{n+1}_{i,j,l,m} \\
&\quad = \frac{\left(c^n_{i,j,l,m}\right)^{1-\gamma} - 1}{1-\gamma} + \partial_f v^{n+1}_{i,j,l,m} s^n_{i,j,l,m,f} \mathbf{1}_{s^n_{i,j,n,m,f} > 0} \\
&\qquad + \partial_B v^{n+1}_{i,j,l,m} s^n_{i,j,l,m,b} \mathbf{1}_{s^n_{i,j,l,m,b} < 0} \\
&\qquad + \lambda_i \left(v^{n+1}_{-i,j,l,m} - v^{n+1}_{i,j,l,m}\right) + h_{l,m} \partial_B v_{i,j,l,m} + \mu^N_{l,m} \partial_N v_{i,j,l,m} \\
&\qquad + \frac{\left[\sigma^N_{l,m}\right]^2}{2} \partial^2_{NN} v_{i,j,l,m}
\end{aligned}$$

for $i = 1, 2$, $j = 1, \ldots, J$, $l = 1, \ldots, L$, $m = 1, \ldots, M$, where $\mathbf{1}(\cdot)$ is the indicator function and

$$\begin{aligned}
h_{l,m} &\equiv h(B_l, N_m), \\
\mu^N_{l,m} &\equiv \mu^N(B_l, N_m) = \alpha Z (B_l + N_m)^{\alpha} - \delta (B_l + N_m) - r_{l,m} B_l - \widehat{\rho} N_m, \\
\sigma^N_{l,m} &\equiv \sigma^N(B_l, N_m) = \sigma (B_l + N_m), \\
s^n_{i,j,l,m,f} &= w_{l,m} z_i + r_{l,m} a_j - \left[\frac{1}{\partial^n_f v_{i,j,l,m}}\right]^{1/\gamma}, \\
s^n_{i,j,l,m,b} &= w_{l,m} z_i + r_{l,m} a_j - \left[\frac{1}{\partial^n_b v_{i,j,l,m}}\right]^{1/\gamma}.
\end{aligned}$$

Thus, when the drift is positive ($s^n_{i,j,l,m,f} > 0$), we employ a forward approximation of the derivative, $\partial^n_f v_{i,j,l,m}$; when it is negative ($s^n_{i,j,l,m,b} < 0$), we employ a backward approximation, $\partial^n_b v_{i,j,l,m}$. The term $\frac{v^{n+1}_{i,j,l,m} - v^n_{i,j,l,m}}{\Delta} \to 0$ as $v^{n+1}_{i,j,l,m} \to v^n_{i,j,l,m}$.

Moving all terms involving v^{n+1} to the left-hand side and the rest to the right-hand side, we obtain

$$\begin{aligned}
&\frac{v^{n+1}_{i,j,l,m} - v^{n}_{i,j,l,m}}{\Delta} + \rho v^{n+1}_{i,j,l,m} \\
&\quad = \frac{\left(c^{n}_{i,j,n,m}\right)^{1-\gamma} - 1}{1-\gamma} + v^{n+1}_{i,j-1,l,m}\alpha^{n}_{i,j,l,m} + v^{n+1}_{i,j,l,m}\beta^{n}_{i,j,l,m} + v^{n+1}_{i,j+1,l,m}\xi^{n}_{i,j,l,m} \\
&\qquad + \lambda_i v^{n+1}_{-i,j,l,m} + v^{n+1}_{i,j,l+1,m}\frac{h_{l,m}}{\Delta B} + v^{n+1}_{i,j,l,m+1}\varkappa_{l,m} + v^{n+1}_{i,j,l,m-1}\varrho_{l,m},
\end{aligned} \tag{41}$$

where

$$\begin{aligned}
\alpha^{n}_{i,j} &\equiv -\frac{s^{n}_{i,j,B}\mathbf{1}_{s^{n}_{i,j,B}<0}}{\Delta a}, \\
\beta^{n}_{i,j,l,m} &\equiv -\frac{s^{n}_{i,j,l,m,f}\mathbf{1}_{s^{n}_{i,j,n,mF}>0}}{\Delta a} + \frac{s^{n}_{i,j,l,m,b}\mathbf{1}_{s^{n}_{i,j,l,m,b}<0}}{\Delta a} - \lambda_i - \frac{h_{l,m}}{\Delta B} - \frac{\mu^{N}_{l,m}}{\Delta N} - \frac{\left(\sigma^{N}_{l,m}\right)^2}{(\Delta N)^2}, \\
\xi^{n}_{i,j} &\equiv \frac{s^{n}_{i,j,F}\mathbf{1}_{s^{n}_{i,j,F}>0}}{\Delta a}, \\
\varkappa_{l,m} &\equiv \frac{\mu^{N}_{l,m}}{\Delta N} + \frac{\left(\sigma^{N}_{l,m}\right)^2}{2(\Delta N)^2} \\
&= \frac{\left[\alpha Z(B_l + N_m)^{\alpha} - \delta(B_l + N_m) - r_{l,m}B_l - \widehat{\rho}N_m\right]}{\Delta N} + \frac{\sigma^2(B_l + N_m)^2}{2(\Delta N)^2}, \\
\varrho_{l,m} &\equiv \frac{\left(\sigma^{N}_{l,m}\right)^2}{2(\Delta N)^2} = \frac{\sigma^2(B_l + N_m)^2}{2(\Delta N)^2},
\end{aligned}$$

for $i = 1, 2$, $j = 1, \ldots, J$, $l = 1, \ldots, L$, $m = 1, \ldots, M$. We consider boundary state constraints in a ($s^{n}_{i,1,B} = s^{n}_{i,J,F} = 0$). The boundary conditions in B and N are reflections.

In equation (41), the optimal consumption is set to

$$c^{n}_{i,j,n,m} = \left(\partial v^{n}_{i,j,l,m}\right)^{-1/\gamma}, \tag{42}$$

where $\partial v^{n}_{i,j,l,m} = \partial_f v^{n}_{i,j,l,m}\mathbf{1}_{s^{n}_{i,j,n,mF}>0} + \partial_b v^{n}_{i,j,l,m}\mathbf{1}_{s^{n}_{i,j,l,m,b}<0} + \partial \bar{v}^{n}_{i,j,l,m}\mathbf{1}_{s^{n}_{i,j,n,mF}\leq 0}\mathbf{1}_{s^{n}_{i,j,l,m,b}\geq 0}$.

In the above expression, $\partial \bar{v}^{n}_{i,j,l,m} = (\bar{c}^{n}_{i,j,n,m})^{-\gamma}$, where $\bar{c}^{n}_{i,j,n,m}$ is the consumption level such that the drift is zero: $\bar{c}^{n}_{i,j} = w_{l,m}z_i + r_{l,m}a_j$.

We define

$$\mathbf{A}^n_{l,m} = \begin{bmatrix} \beta^n_{1,1,l,m} & \xi^n_{1,1,l,m} & 0 & 0 & \cdots & 0 & \lambda_1 & 0 & \cdots & 0 \\ \alpha^n_{1,2,l,m} & \beta^n_{1,2,l,m} & \xi^n_{1,2,l,m} & 0 & \cdots & 0 & 0 & \lambda_1 & \ddots & 0 \\ 0 & \alpha^n_{1,3,l,m} & \beta^n_{1,3,l,m} & \xi^n_{1,3,l,m} & \cdots & 0 & 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha^n_{1,J-1,l,m} & \beta^n_{1,J-1,l,m} & \xi^n_{1,J-1,l,m} & 0 & \cdots & \lambda_1 & 0 \\ 0 & 0 & \cdots & 0 & \alpha^n_{1,J,l,m} & \beta^n_{1,J,l,m} & 0 & 0 & \cdots & \lambda_1 \\ \lambda_2 & 0 & \cdots & 0 & 0 & 0 & \beta^n_{2,1,l,m} & \xi^n_{2,1,l,m} & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & 0 & \lambda_2 & 0 & \cdots & \alpha^n_{2,J,l,m} & \beta^n_{2,J,l,m} \end{bmatrix},$$

$$\mathbf{v}^{n+1}_{l,m} = \begin{bmatrix} \mathbf{v}^{n+1}_{1,1,l,m} \\ \mathbf{v}^{n+1}_{1,2,l,m} \\ \vdots \\ \mathbf{v}^{n+1}_{1,J,l,m} \\ \mathbf{v}^{n+1}_{2,1,l,m} \\ \vdots \\ \mathbf{v}^{n+1}_{2,J,l,m} \end{bmatrix},$$

and

$$\mathbf{A}^n_m = \begin{bmatrix} \mathbf{A}^n_{1,m} & \frac{h_{1,m}}{\Delta B}\mathbf{I}_{2J} & \mathbf{0}_{2J} & \cdots & \mathbf{0}_{2J} & \mathbf{0}_{2J} \\ \mathbf{0}_{2J} & \mathbf{A}^n_{2,m} & \frac{h_{2,m}}{\Delta B}\mathbf{I}_{2J} & \cdots & \mathbf{0}_{2J} & \mathbf{0}_{2J} \\ \mathbf{0}_{2J} & \mathbf{0}_{2J} & \mathbf{A}^n_{3,m} & \cdots & \mathbf{0}_{2J} & \mathbf{0}_{2J} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & \mathbf{0}_{2J} & \mathbf{A}^n_{L-1,m} & \frac{h_{L-1,m}}{\Delta B}\mathbf{I}_{2J} \\ \mathbf{0}_{2J} & \mathbf{0}_{2J} & \cdots & \mathbf{0}_{2J} & \mathbf{0}_{2J} & \left(\mathbf{A}^n_{L,m} + \frac{h_{L,m}}{\Delta B}\mathbf{I}_{2J}\right) \end{bmatrix}, \qquad \mathbf{v}^{n+1}_m = \begin{bmatrix} \mathbf{v}^{n+1}_{1,m} \\ \mathbf{v}^{n+1}_{2,m} \\ \vdots \\ \mathbf{v}^{n+1}_{L,m} \end{bmatrix},$$

where $\mathbf{I}_n$ and $\mathbf{0}_n$ are the identity matrix and the zero matrix of dimension $n \times n$, respectively. We can also define

$$\mathbf{A}^n = \begin{bmatrix} \left(\mathbf{A}^n_1 + \mathbf{P}_1\right) & \mathbf{X}_1 & \mathbf{0}_{2J\times L} & \cdots & \mathbf{0}_{2J\times L} & \mathbf{0}_{2J\times L} \\ \mathbf{P}_2 & \mathbf{A}^n_2 & \mathbf{X}_2 & \cdots & \mathbf{0}_{2J\times L} & \mathbf{0}_{2J\times L} \\ \mathbf{0}_{2J\times L} & \mathbf{P}_3 & \mathbf{A}^n_3 & \cdots & \mathbf{0}_{2J\times L} & \mathbf{0}_{2J\times L} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & \mathbf{P}_{M-1} & \mathbf{A}^n_{M-1} & \mathbf{X}_{M-1} \\ \mathbf{0}_{2J\times L} & \mathbf{0}_{2J\times L} & \cdots & \mathbf{0}_{2J\times L} & \mathbf{P}_M & \left(\mathbf{A}^n_M + \mathbf{X}_M\right) \end{bmatrix},$$

$$\mathbf{v}^{n+1} = \begin{bmatrix} \mathbf{v}^{n+1}_1 \\ \mathbf{v}^{n+1}_2 \\ \vdots \\ \mathbf{v}^{n+1}_M \end{bmatrix},$$

$$\mathbf{X}_m = \begin{bmatrix} \varkappa_{1,m}\mathbf{I}_{2J} & \mathbf{0}_{2J} & \cdots & \mathbf{0}_{2J} & \mathbf{0}_{2J} \\ \mathbf{0}_{2J} & \varkappa_{2,m}\mathbf{I}_{2J} & \cdots & \mathbf{0}_{2J} & \mathbf{0}_{2J} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & \mathbf{0}_{2J} & \varkappa_{L-1,m}\mathbf{I}_{2J} & \mathbf{0}_{2J} \\ \mathbf{0}_{2J} & \mathbf{0}_{2J} & \mathbf{0}_{2J} & \mathbf{0}_{2J} & \varkappa_{L,m}\mathbf{I}_{2J} \end{bmatrix},$$

$$\mathbf{P}_m = \begin{bmatrix} \varrho_{1,m}\mathbf{I}_{2J} & \mathbf{0}_{2J} & \cdots & \mathbf{0}_{2J} & \mathbf{0}_{2J} \\ \mathbf{0}_{2J} & \varrho_{2,m}\mathbf{I}_{2J} & \cdots & \mathbf{0}_{2J} & \mathbf{0}_{2J} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ & & \mathbf{0}_{2J} & \varrho_{L-1,m}\mathbf{I}_{2J} & \mathbf{0}_{2J} \\ \mathbf{0}_{2J} & \mathbf{0}_{2J} & \mathbf{0}_{2J} & \mathbf{0}_{2J} & \varrho_{L,m}\mathbf{I}_{2J} \end{bmatrix},$$

$$\mathbf{u}^n = \begin{bmatrix} \frac{\left(c^n_{1,1,1,1}\right)^{1-\gamma}-1}{1-\gamma} \\ \frac{\left(c^n_{1,2,1,1}\right)^{1-\gamma}-1}{1-\gamma} \\ \vdots \\ \vdots \\ \frac{\left(c^n_{2,J,L,M}\right)^{1-\gamma}-1}{1-\gamma} \end{bmatrix}.$$

Then, equation (41) is a system of $2 \times J \times L \times M$ linear equations that can be written in matrix notation as

$$\frac{1}{\Delta}\left(\mathbf{v}^{n+1}-\mathbf{v}^n\right)+\rho\mathbf{v}^{n+1}=\mathbf{u}^n+\mathbf{A}^n\mathbf{v}^{n+1}.$$

The system in turn can be written as

$$\mathbf{B}^n\mathbf{v}^{n+1}=\mathbf{d}^n, \tag{43}$$

where $\mathbf{B}^n=(\frac{1}{\Delta}+\rho)\mathbf{I}-\mathbf{A}^n$ and $\mathbf{d}^n=\mathbf{u}^n+\frac{1}{\Delta}\mathbf{v}^n$.

The algorithm to solve the HJB equation begins with an initial guess $v^0_{i,j,l,m}$. Set $n=0$. Then:

1. Compute $c^n_{i,j,l,m}$, $i=1,2$ using (42).
2. Find $v^{n+1}_{i,j,l,m}$ solving the linear system of equations (43).
3. If $v^{n+1}_{i,j,l,m}$ is close enough to $v^n_{i,j,l,m}$, stop. If not, set $n:=n+1$ and proceed to step 1.

Most programming languages, such as `Julia` or `Matlab`, include efficient routines to handle sparse matrices such as $\mathbf{A}^n$.

Step 2: Solution to the KF Equation

The income-wealth distribution conditional on the current realization of aggregate debt $B=B_l$ and equity $N=N_m$ can be characterized by the KF equation:

$$\frac{\partial g}{\partial t}=-\frac{\partial}{\partial a}\left[s_i(a,B,N)g_{i,t}(a)\right]-\lambda_i g_{i,t}(a)+\lambda_{-i}g_{-i,t}(a), \quad i=1,2, \tag{44}$$

$$1 = \sum_{i=1}^{2} \int g_{it}(a)\, da. \tag{45}$$

If we define a time step Δt, we also solve this equation using a finite difference scheme. We use the notation $g_{i,j} \equiv g_i(a_j)$. The system can now be expressed as

$$\frac{g_{i,j,t+1} - g_{i,j}}{\Delta t} = -\frac{g_{i,j,t} s_{i,j,l,m,b} \mathbf{1}_{s_{i,j,l,m,b}>0} - g_{i,j-1,t} s_{i,j-1,l,m,f} \mathbf{1}_{s_{i,j-1,l,m,f}>0}}{\Delta a} - \frac{g_{i,j+1,t} s_{i,j+1,l,m,b} \mathbf{1}_{s_{i,j+1,l,m,b}<0} - g_{i,j,t} s_{i,j,l,m,b} \mathbf{1}_{s_{i,j,l,m,b}<0}}{\Delta a} - \lambda_i g_{i,j,t} + \lambda_{-i} g_{-i,j,t}.$$

In this case, let us define $\mathbf{g}_t = [g_{1,1,t}, g_{1,2,t} \ldots g_{1,J,t}, g_{2,1,t} \ldots g_{2,J,t}]'$ as the density conditional on the current state of B_l and N_m. We assume that $\mathbf{g}_0$ is the density in the DSS, and the update in the next time period is given by the KF equation:

$$\mathbf{g}_{t+1} = \left(\mathbf{I} - \Delta t \mathbf{A}_{l,m}^{\mathbf{T}}\right)^{-1} \mathbf{g}_t,$$

where $\mathbf{A}_{l,m}^{\mathbf{T}}$ is the transpose matrix of $\mathbf{A}_{l,m} = \lim_{n\to\infty} \mathbf{A}_{l,m}^{n}$, defined above.

Step 3: Update of the PLM Using a Neural Network

The vector $\boldsymbol{\theta}$ is recursively updated according to $\boldsymbol{\theta}_{m+1} = \boldsymbol{\theta}_m - \epsilon_m \nabla \mathcal{E}(\boldsymbol{\theta}; \boldsymbol{x}_j, \widehat{h}_j)$, where

$$\nabla \mathcal{E}(\boldsymbol{\theta}; \boldsymbol{x}_j, \widehat{h}_j) \equiv \left[\frac{\partial \mathcal{E}(\boldsymbol{\theta}; \boldsymbol{x}_j, \widehat{h}_j)}{\partial \theta_0^2}, \frac{\partial \mathcal{E}(\boldsymbol{\theta}; \boldsymbol{x}_j, \widehat{h}_j)}{\partial \theta_1^2}, \ldots, \frac{\partial \mathcal{E}(\boldsymbol{\theta}; \boldsymbol{x}_j, \widehat{h}_j)}{\partial \theta_{2,Q}^1} \right]^{\top}$$

is the gradient of the error function with respect to $\boldsymbol{\theta}$ evaluated at $(\mathbf{x_j}, \widehat{h}_j)$.

The step size $\epsilon_m > 0$ is selected in each iteration by line-search to minimize the error function in the direction of the gradient. The algorithm is run until $\|\boldsymbol{\theta}_{m+1} - \boldsymbol{\theta}_m\| < \varepsilon$, for a small $\varepsilon > 0$. The error gradient can be efficiently evaluated using backpropagation.

We fine-tune the training scheme of our neural network as follows. First, we employ a line-search algorithm because the training scheme needs to yield a consistently good approximation: a "not-good-enough" approximation in any of the dozens of iterations of the algorithm can make it break. Line-search is slower than a constant or adaptive learning rate, but it prevents bad steps in the minimization.

Second, the training scheme should not introduce big amounts of noise in each iteration. Otherwise, the noise can mask or prevent convergence (strict convergence criteria can only be met by chance, if at all). This is why we use a batch gradient descent (i.e., all training points are used in every gradient calculation and line-search step) instead of the more popular stochastic or minibatch gradient descents. The random choice of training points in each step, while common in much of the machine learning literature, introduces too much noise in our case.

Reducing stochastic elements in the solution method is also why we train the model with a grid approximation that clears out noise. We define a 101×101 grid over the (B, N) support, and assign each simulated point to one of the knots in that grid. Then, we run a linear regression, and use it to estimate the height of the PLM at that knot. This grid could later be used to solve the model using interpolation (e.g., with splines, or with natural neighbor interpolation) and linear extrapolation (and we do that as a robustness

check, finding similar results to those of the solution with the neural network). However, on a 2D surface. such extrapolation tends to generate ridges (because of the amplification of sample noise in far-away extrapolations), which could prevent convergence at the HJB step. Instead, we use those knots to train the neural network, which provides a good-enough fit in the visited area and a much smoother extrapolation to the nonvisited area of the (B, N) support.

In the first iteration of the algorithm, we do ten random initializations of the parameters of the neural network and ten subsequent training sessions. Later, we choose the best-performing trained network across those ten training sessions. From the second iteration onward, the neural network is initialized using weights that were found to be optimal in the previous iteration, and a single training session is carried out. This avoids re-introducing a stochastic element at each step, thus reducing unnecessary noise and saving running time. Using a small relaxation parameter in the PLM update step (it starts at 0.30 and exponentially decays toward 0.05, reaching 0.20 after five iterations and 0.10 after 16) makes the convergence of the full algorithm slower but smoother, with slow updates to the optimal neural network that help this nonrandom initialization work well.

Finally, in order to avoid scale problems, we normalize all the inputs of the neural network so that the range is determined to be $[-1, 1]$. To this end, we find the maximum and minimum values of the training sample and compute the mid-point between them. Then, we subtract the mid-point from every individual sample and divide the result by half of the max-min interval.

Complete Algorithm

We can now summarize the complete algorithm. We begin with a guess of the PLM $h^0(B, N)$. Set $n := 0$:

Step 1: *Household problem*. Given $h^n(B, N)$, solve the HJB equation to obtain an estimate of the value function $\mathbf{v}$ and of the matrix $\mathbf{A}$.

Step 2: *Distribution.* Given $\mathbf{A}$, simulate T periods of the economy using the KF equation and obtain the aggregate debt $\{B_{t+1} = \sum_{i=1}^{2}\sum_{j=1}^{J}\frac{g_{ijt+1}(B_t,N_t)}{\Delta t}\Delta a\}_{t=0}^{T}$ and equity $\{N_t\}_{t=0}^{T}$. Given $\varepsilon_t \overset{iid}{\sim} \mathcal{N}(0,1)$, the law of motion of equity is

$$N_t = N_{t-1} + \left[\alpha Z(B_t + N_t)^{\alpha} - \delta(B_t + N_t) - r_t B_t - \widehat{\rho} N_t\right]\Delta t + \sigma(B_t + N_t)\sqrt{\Delta t}\varepsilon_t.$$

Step 3: *PLM*. Update the PLM using a neural network: h^{n+1}. If $\|h^{n+1} - h^n\| < \varepsilon$, where ε is a small positive constant, then stop. If not, return to step 1.

The code is currently optimized for computation on four cores using `Matlab`. Solving the nonlinear version of the model takes approximately 5 hours on a 2021 high-end workstation (which also allows for multiple instances of the code to run at the same time, which accelerates the more time-consuming sensitivity analysis computation).

The breakdown of time is 15% for solving the HJB; 63% for the simulation in the KFE; 19% for training the neural network that approximates the PLM; and 3% for other tasks (mostly plotting and convergence diagnostics).

APPENDIX C: A GENERAL ALGORITHM FOR SOLVING HETEROGENEOUS AGENT MODELS

The algorithm discussed in the previous section is an instance of a much more general strategy for solving a large class of heterogeneous agent models, in both discrete and

continuous time. For example, Fernández-Villaverde, Marbet, Nuño, and Rachedi (2022) used the method in this paper to compute a discrete-time, heterogeneous agent New Keynesian (HANK) model with aggregate shocks and the zero lower bound (ZLB) on the nominal interest rate.[1]

In general, to solve these models, we need to deal with the distribution of agents G_t and the operator $H(\cdot)$ that characterizes how G_t evolves:

$$G_{t+1} = H(G_t, S_t)$$

in discrete time, or

$$\frac{\partial G_t}{\partial t} = H(G_t, S_t)$$

in continuous time, given the other aggregate states of the economy S_t.

The challenge, thus, is to track G_t and compute $H(G_t, S_t)$. Building on the ideas of Krusell and Smith (1998), we propose the following algorithm:

1. Summarizing the distribution: If we deal with N discrete types of agents, we keep track of the $N-1$ weights of each type of agent but one (as the weights need to sum up to 1). If we deal with continuous types, we extract a finite number of features from G_t. These can be moments, Q-quantiles, weights in a mixture of normals, or many other options.

2. Substituting the operator $H(\cdot)$ by a parameterized PLM $h(\cdot)$: We stack either the weights or features of the distribution in a vector μ_t and assume that μ_t follows a PLM:

$$\mu_{t+1} = h(\mu_t, S_t)$$

in discrete time, or

$$\frac{\partial \mu_t}{\partial t} = h(\mu_t, S_t)$$

in continuous time, instead of $H(G_t, S_t)$.

3. Parameterizing the PLM: We parameterize $h(\mu_t, S_t)$ as $h(\mu_t, S_t; \theta)$ using a neural network, where θ is the vector of network weights. The details of the architecture (i.e., hyperparameters such as number of hidden layers, activation function, number of nodes,...) will depend on the details of the model. These hyperparameters can be determined by standard techniques (e.g., cross-validation, drop-out).

4. Training the neural network: We determine the unknown weights θ to ensure that an economy where μ_t follows $h(\mu_t, S_t; \theta)$ replicates as well as possible the behavior of an economy where G_t follows $H(\cdot)$. To do so:

1. We guess an initial value θ^0 of the neural network weights. We set $n := 0$.
2. We construct a time series of μ_t simulating from the PLM and the relevant equilibrium conditions of the model.
3. We train the neural network weights on the simulated data and obtain θ^{n+1} by minimizing a quadratic error function between μ_{t+1} and μ_t (in discrete time) or $\frac{\mu_{t+1}-\mu_t}{\Delta t}$ and μ_t (in continuous time). Other suitable loss functions, for example, adding regularization terms, are possible.

[1]The computation in Fernández-Villaverde et al. (2022) also demonstrates that the details of the PLM in our model (e.g., continuous time, number of aggregate state variables, etc.) are not an essential part of the algorithm.

4. We iterate on steps 2–3 until $\|\theta^{n+1} - \theta^n\| < \epsilon$ given a norm $\|\cdot\|$ and tolerance level $\epsilon > 0$.

An example: Consider the basic Krusell–Smith model in discrete time, where we have two aggregate variables: an aggregate productivity shock Z_t and household distribution $G_t(a, z)$ on individual assets, a, and individual labor productivity, z, where

$$\int G_t(a, z)\, da = K_t.$$

We summarize $G_t(a, \cdot)$ with the log of its mean: $\mu_t = \log K_t$. Then, we parameterize a PLM $h(\mu_t, Z_t; \theta)$ using a neural network and train it as described above.

In fact, the original proposal of Krusell and Smith (1998) to find

$$\underbrace{\log K_{t+1}}_{\mu_{t+1}} = \underbrace{\theta_0(Z_t) + \theta_1(Z_t) \log K_t}_{h(\mu_t, s_t; \theta)},$$

where one determines $\{\theta_0(Z_t), \theta_1(Z_t)\}$ by running an OLS on repeated simulations until convergence, is a particular case of this approach. Linear regression is nothing but a sparse neural network with a (state-dependent) linear activation function. The OLS procedure quickly minimizes the quadratic error function for convenient functional forms such as log-linear.

Similarities and differences with Krusell and Smith (1998): The previous paragraphs show how close we are to the spirit of Krusell and Smith (1998): (i) we summarize the salient features (from an economic perspective) of the agents' distribution; (ii) we parameterize the evolution of these features; and (iii) we determine these parameters by repeated simulations. We simply argue that neural networks provide a constructive procedure to accomplish these tasks:

1. Where Krusell and Smith (1998) proposed using moments of G_t, we propose either sticking with moments or using other features of G_t as required.
2. Where Krusell and Smith (1998) proposed using a (state-dependent) log-linear function for the PLM, we propose using a more general neural network.
3. Where Krusell and Smith (1998) proposed using OLS, we propose using standard algorithms like stochastic gradient descent.

The third item is a small difference: it is just for numerical convenience; and since we use the first moment of the distribution, the first item above is moot in our paper. The key to our idea is, thus, the second item: using neural networks to approximate any PLM.

Further comments on neural networks: Krusell and Smith (1998) did not offer much guidance regarding feature and parameterization selection in general models. While in their model keeping track of the log of the mean with a (state-dependent) log-linear PLM works very well, how do we proceed when solving an arbitrary heterogeneous agent model?

The main text outlined several reasons why neural networks were an attractive possibility to approximate a PLM. Here, we repeat once again that neural networks can also tackle the "curse of dimensionality" when either μ_t or S_t is highly dimensional. Thus, neural networks can keep track of many features from G_t, such as many Q-quantiles. While this result is not relevant to our model, it has been put to good use by Ebrahimi Kahou, Fernández-Villaverde, Perla, and Sood (2021) by solving economic models with hundreds of state variables that would saturate alternatives, such as Chebyshev polynomials. Ebrahimi Kahou et al. (2021) also showed that a neural network would find a log-linear PLM if this is, indeed, the best approximation, even if we do not supply the neural network with this information ex ante.

Recall, nonetheless, that our general algorithm also shares some of the drawbacks of Krusell and Smith (1998). For example, we approximate a self-justified equilibrium given by the PLM, which can be far from the rational expectations equilibrium. While the flexibility of our richly parameterized PLM might help minimize this risk, we should keep this point in mind.

APPENDIX D: BUILDING THE LIKELIHOOD FUNCTION

Let us assume that the econometrician has access to $D+1$ observations of output, $Y_0^D = \{Y_0, Y_\Delta, Y_{2\Delta}, \ldots, Y_D\}$ at fixed time intervals $[0, \Delta, 2\Delta, \ldots, D\Delta]$. The derivations below would be similar for observables other than output. Since we have one aggregate shock in the model (to capital), we can only use one observable in our likelihood. Otherwise, we would suffer from stochastic singularity. If we wanted to have more observables, we would need to either enrich the model with more shocks or introduce measurement shocks in the observables. In those situations, we might need to resort to a sequential Monte Carlo approximation to the filtering problem described by the associated Kushner–Stratonovich equation of our dynamic system (see, in discrete time, Fernández-Villaverde and Rubio-Ramírez (2007)).

The likelihood function $\mathcal{L}_D(Y_0^D|\Psi)$ for our observations of output and given some parameter values $\Psi = \{\alpha, \delta, \sigma, \widehat{\rho}, \rho, \gamma, z_1, z_2, \lambda_1, \lambda_2\}$ has the form

$$\mathcal{L}_D\big(Y_0^D|\Psi\big) = \prod_{d=1}^{D} p_Y(Y_{d\Delta}|Y_{(d-1)\Delta}; \Psi),$$

where $p_Y(Y_{d\Delta}|Y_{(d-1)\Delta}; \Psi) = \int f_{d\Delta}(Y_{d\Delta}, B)\, dB$ is the conditional density function of $Y_{d\Delta}$ given $Y_{(d-1)\Delta}$, and the density function for output and debt, $f_{d\Delta}(Y_{d\Delta}, B)$, implied by the solution of the model. Our task is, then, to compute the sequences of conditional densities $p_Y(Y_{d\Delta}|Y_{(d-1)\Delta}; \Psi)$ at the fixed time intervals $[0, \Delta, 2\Delta, \ldots, D\Delta,]$.

To do so, we obtain the diffusion of $Y_t = (B_t + N_t)^\alpha$. Applying Itô's lemma, we get

$$\begin{aligned} dY_t &= \frac{\partial (B+N)^\alpha}{\partial B}\, dB_t + \frac{\partial (B+N)^\alpha}{\partial N}\, dN_t + \frac{1}{2}\frac{\partial^2 (B+N)^\alpha}{\partial N^2}\sigma^2 (B+N)^2\, dt \\ &= \mu^Y(B_t, Y_t)\, dt + \sigma_t^Y(Y_t)\, dZ_{t,} \end{aligned} \tag{46}$$

where

$$\begin{aligned} \mu^Y(B_t, Y_t) = \alpha Y_t^{\frac{\alpha-1}{\alpha}} \Bigg\{ & h\big(B_t, Y_t^{\frac{1}{\alpha}} - B_t\big) + \alpha Y_t + \left[\frac{(\alpha-1)\sigma^2}{2} - \delta\right] Y_t^{\frac{1}{\alpha}} \\ & - \left(\alpha Y_t^{\frac{\alpha-1}{\alpha}} - \delta - \sigma^2 \frac{Y_t^{\frac{1}{\alpha}}}{Y_t^{\frac{1}{\alpha}} - B_t}\right) B_t - \hat{\rho}\big(Y_t^{\frac{1}{\alpha}} - B_t\big) \Bigg\}, \end{aligned}$$

and $\sigma^Y(Y_t) = \alpha\sigma Y_t$.

With equation (46), the density $f_t^d(Y, B)$ follows the KF equation in the interval $[(d-1)\Delta, d\Delta]$:

$$\frac{\partial f_t}{\partial t} = -\frac{\partial}{\partial Y}\big[\mu^Y(Y, B) f_t(Y, B)\big] - \frac{\partial}{\partial B}\big[h\big(B, Y^{\frac{1}{\alpha}} - B\big) f_t^d(Y, B)\big]$$

$$+\frac{1}{2}\frac{\partial^2}{\partial Y^2}\left[\left(\sigma^Y(Y)\right)^2 f_t(Y,B)\right]. \tag{47}$$

At the beginning of the interval, we have $f_{(d-1)\Delta}(Y,B)=\delta(Y-Y_{(d-1)\Delta})f_{(d-2)\Delta}(B|Y_{(d-1)\Delta})$, where $f_{(d-1)\Delta}(B|Y_{(d-1)\Delta})$ is the probability of B conditional on $Y=Y_{(d-1)\Delta}$:

$$f_{(d-2)\Delta}(B|Y_{(d-1)\Delta})=\frac{f_{(d-2)\Delta}(Y_{(d-1)\Delta},B)}{f_{(d-2)\Delta}(Y_{(d-1)\Delta})}=\frac{f_{(d-2)\Delta}(Y_{(d-1)\Delta},B)}{\int f_{(d-2)\Delta}(Y_{(d-1)\Delta},B)\,dB},$$

if $d\geq 2$, $f_{-1}(B)=f(B)$ is the ergodic distribution of B, and $\delta(\cdot)$ is the Dirac delta function. Since the operator in the KF equation (47) is the adjoint of the infinitesimal generator employed in the HJB, we only need to transpose and invert a sparse matrix that has already been computed when we solved the HJB.

Lo (1988) provided technical assumptions that must be satisfied for this estimation to work. In our model, these conditions are met provided that $h(B,N)$ is twice continuously differentiable in B and N and three times continuously differentiable in Ψ, which is guaranteed if (i) $h(B,N)$ is approximated using a neural network with our softplus activation function and (ii) Ψ lies in the interior of a finite-dimensional closed and compact parameter space.[2]

APPENDIX E: BUILDING THE LIKELIHOOD FUNCTION WITH MICRO DATA

A promising avenue to improve the estimation in the main text is to add micro observations, which bring much additional information and help in integrating different levels of aggregation to assess the empirical validity of the model. More concretely, let $X_t\equiv[g_t(a,z);N_t]'$ be a vector of observations on the asset holdings of agents in this economy (households, $g_t(a,z)$, and the expert, N_t). Imagine, as before, that we have $D+1$ observations of X_t at fixed time intervals $[0,\Delta,2\Delta,\ldots,D\Delta,]: X_0^D=\{X_0,X_\Delta,X_{2\Delta},\ldots,X_D\}$.

At this moment, we need to assume—as is typically done in models with heterogeneous agents and aggregate shocks—that the *conditional no aggregate uncertainty* (CNAU) condition holds. This condition implies that if households are distributed on the interval $I=[0,1]$ according to the Lebesgue measure Φ, then $G_t(A\times Z)=\Phi(i\in I:(a_t^i,z_t^i)\in A\times Z)$, for any subsets $A\subset[0,\infty)$, $Z\subset\{z_1,z_2\}$. That is, the probability under the conditional distribution is the same as the probability according to the Lebesgue measure across I.

The likelihood that an individual agent $i\in I$ at time $t=d\Delta$ is at state $(a_{d\Delta}^i,z_{d\Delta}^i,B_{d\Delta},N_{d\Delta})$ is $f_{d\Delta}^d(a_{d\Delta}^i,z_{d\Delta}^i,B_{d\Delta},N_{d\Delta})$. The log-likelihood is then $\log[f_{d\Delta}^d(a_{d\Delta}^i,z_{d\Delta}^i,B_{d\Delta},N_{d\Delta})]$. Notice that this log-likelihood is a function of i.

The conditional aggregate log-likelihood across all agents is

$$\log p_X(X_{d\Delta}|X_{(d-1)\Delta};\Psi)=\int\log\left[f_{d\Delta}^d(a_{d\Delta}^i,z_{d\Delta}^i,B_{d\Delta},N_{d\Delta})\right]\Phi(di),$$

and, taking into account the CNAU condition, we get

$$\int\log\left[f_{d\Delta}^d(a_{d\Delta}^i,z_{d\Delta}^i,B_{d\Delta},N_{d\Delta})\right]\Phi(di)=\int\log\left[f_{d\Delta}^d(a,z,B_{d\Delta},N_{d\Delta})\right]G_{d\Delta}(da,dz)$$

[2]In our model, output is a flow variable, whereas, in the data, it is the cumulative production over a quarter. Thus, a precise definition of the observable would be $Y_{d\Delta}^{\text{agg}}=\int_{(d-1)\Delta}^{d\Delta}Y_s\,ds$. In our paper, this expression can be approximated with a high degree of accuracy by $Y_{d\Delta}^{\text{agg}}\approx Y_{d\Delta}\Delta$.

$$= \sum_{i=1}^{2} \int_0^{\infty} \log\bigl[f_{d\Delta}^d(a, z_i, B_{d\Delta}, N_{d\Delta})\bigr] g_{d\Delta}(a, z)\, da,$$

where, in the second line, we have applied the definition of the Radon–Nikodym derivative to get the differential in a.

The density $f_t^d(a, z, B, N)$ follows the KF equation:

$$\begin{aligned}\frac{\partial f_t^d}{\partial t} = &-\frac{\partial}{\partial a}\bigl(s_i(a_t, B_t, N_t) f_t^d(a, z_i, B, N)\bigr) - \lambda_i f_t^d(a, z_i, B, N) + \lambda_j f_t^d(a, z_j, B, N) \\ &- \frac{\partial}{\partial B}\bigl[h(B, N) f_t^d(a, z_i, B, N))\bigr] - \frac{\partial}{\partial N}\bigl[\mu_t^N(B, N) f_t^d(a, z_i, B, N)\bigr] \\ &+ \frac{1}{2}\frac{\partial^2}{\partial N^2}\bigl[\bigl(\sigma_t^N(B, N)\bigr)^2 f_t^d(B, N)\bigr], \quad (i \neq j = 1, 2),\end{aligned} \tag{48}$$

where $f_{(d-1)\Delta}^d = g_{(d-1)\Delta}(a, z)\delta(B - B_{(d-1)\Delta})\delta(N - N_{(d-1)\Delta})$, an easy-to-evaluate expression.

More concretely, we use the notation $f_{i,j,l,m}^d \equiv f_i^d(a_j, B_l, N_m)$ and define a time step $\Delta t = \frac{\Delta}{S}$, where $1 \ll S \in \mathbb{N}$ is a constant. If we solve the KF equation (48) using a finite difference scheme, we have, for $t = (d-1)\Delta$ and $s = 1, \ldots, S-1$,

$$\mathbf{f}_{t+s\Delta t}^d = \bigl(\mathbf{I} - \Delta t \mathbf{A}^{\mathbf{T}}\bigr)^{-1} \mathbf{f}_{t+(s-1)\Delta t}^d,$$

$$\mathbf{f}_t^d = \mathbf{g}_t \boldsymbol{\delta}_{N_{(d-1)\Delta}} \boldsymbol{\delta}_{B_{(d-1)\Delta}},$$

where $\boldsymbol{\delta}$ is the Kronecker delta and $\mathbf{f}_t^d = [f_{1,1,1,t}, f_{1,1,1,2,t}, \ldots, f_{2,J,L,M,t}]'$.

We approximate $p_X(X_{d\Delta}|X_{(d-1)\Delta}; \gamma) = \sum_{i=1}^{2}\sum_{j=1}^{J}\sum_{l=1}^{L} f_{i,j,l^d,m}^d g_{i,j}^d \Delta a \Delta B$, where $f_{i,j,l^d,m}^d$ is the density evaluated at the observed equity point $N_{d\Delta}$, $f_i^d(a_j, B_l, N = N_{d\Delta})$ and $g_{i,j}^d$ are the elements of the observed distribution $\mathbf{g}_{d\Delta}$.

APPENDIX F: MAXIMIZING THE LIKELIHOOD FUNCTION

We maximize the likelihood function by searching on a grid between 0.013 and 0.015 with a step 0.0002 (we played extensively with σ values to determine the region of high likelihood before starting the grid search).

We plot the resulting log-likelihood in Figure S1. The point estimate, 0.0142, is drawn as a vertical discontinuous red line, with a standard error of 0.00011342, computed by the local derivative of the function. The smoothness of the plot confirms that our algorithm has successfully converged, since changes in one parameter value do not lead us to different numerical solutions.

APPENDIX G: ALTERNATIVE SOLUTION METHODS

Figure S2 plots the PLM phase diagram corresponding to the solution of our model using the Krusell–Smith algorithm. This figure is equivalent to Figure 5 in the main text (where we use neural networks). It is easy to see that the multiplicity of SSS(s) disappears if we do not consider nonlinearities. Thus, agents in the model expect a unimodal ergodic distribution.

Next, we show that the approximation to the PLM computed with Chebyshev polynomials is not satisfactory. In Figure S3, we plot the PLM obtained with an algorithm similar to

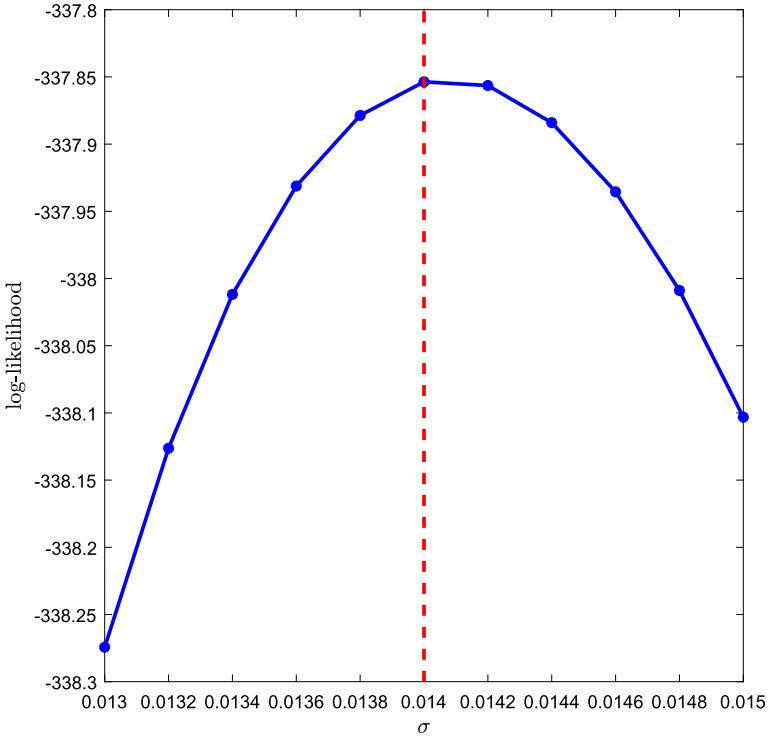


FIGURE S1.—Log-likelihood for different values of σ and point estimate.

ours, but where we substitute a linear combination of Chebyshev polynomials for the neural network and we select the coefficients of that linear combination to fit the simulated data as well as possible.

While, at first inspection, the PLM in panel (a) seems sensible, a closer examination of the scale of the y-axis reveals implausible movements in $h(B, N)$. These variations are seen better in panel (b), where we zoom in on $h(B, N)$ with a smaller range of debt and equity. The PLM is well approximated in the ergodic distribution (shaded area in the center) but, as soon as we move slightly outside that area, the oscillating features of polynomial approximations reassert themselves. Using this PLM begets unstable simulations and unreliable results. Similar problems appear in solutions construed with splines or similar series approximations: extrapolation requires a well-behaved basis and neural networks do an excellent job at such a task.

APPENDIX H: CONVERGENCE TO THE SSS(S)

How do we know that the two SSS(s) described above are stable? The state space $(g(\cdot), N)$ is infinite-dimensional and, hence, we cannot check convergence numerically for all possible initial states. Instead, we analyze convergence for points visited in the ergodic distribution.

Figure S4 considers an array of different initial income-wealth densities and equity levels $(g_0(\cdot), N_0)$, selected from the simulations used to compute the aggregate ergodic distribution and analyze the transitional dynamics when no aggregate shocks arrive (agents continue forming their expectations assuming $\sigma > 0$). We plot, in red, the paths converg-

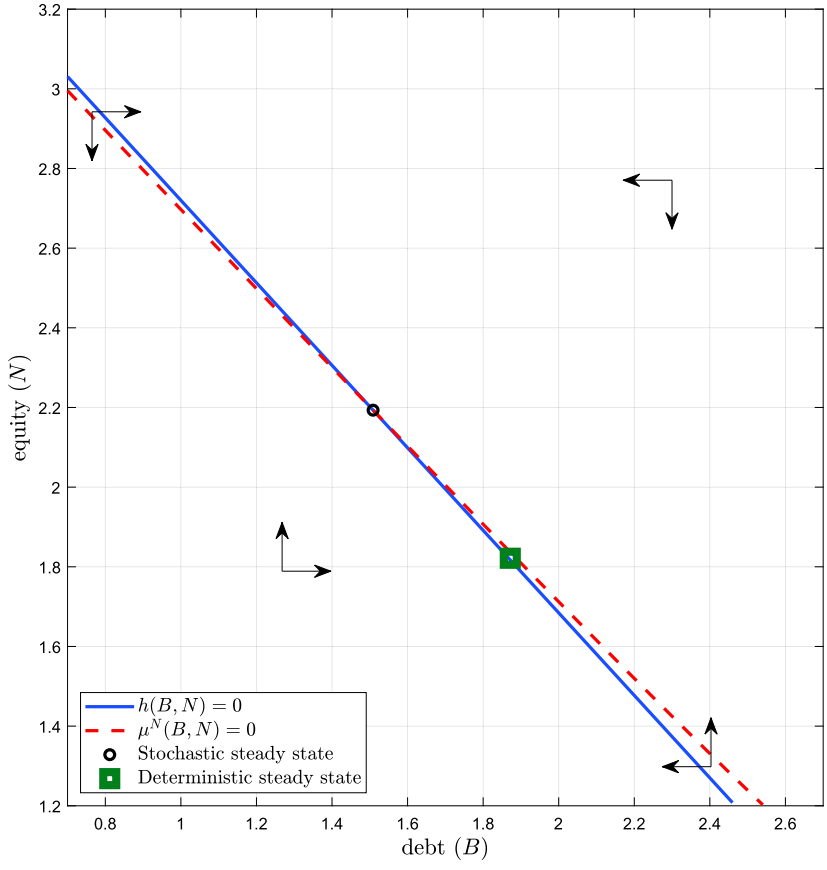


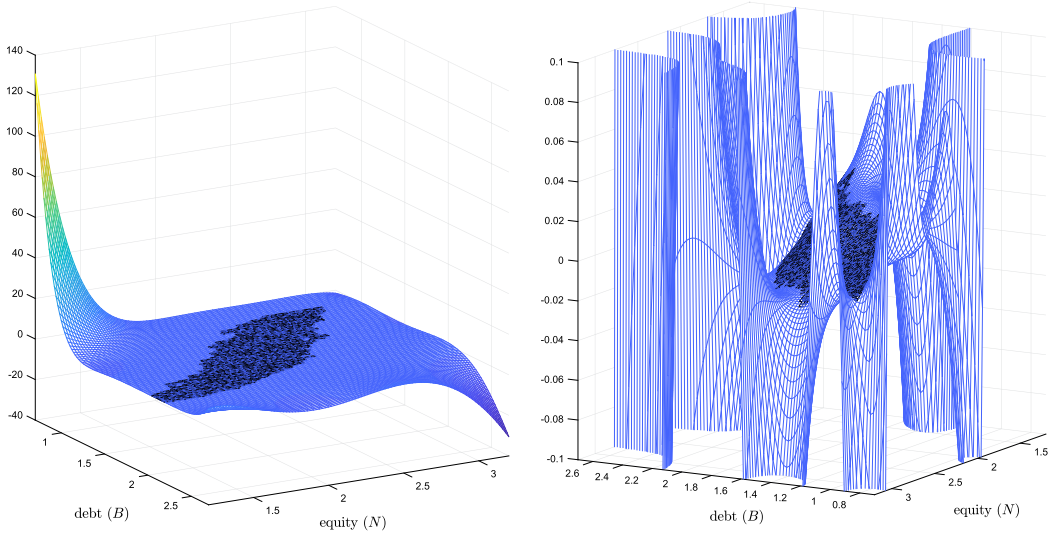
FIGURE S2.—Phase diagram, DSS, and SSS(s), linear PLM.

ing toward the LL-SSS and, in green, the paths converging toward the HL-SSS. In all cases, the economy converges to the SSS(s)—denoted by small circles—on the basin of attraction of the initial condition. However, the convergence path is plodding and it may take centuries.

Notwithstanding, we cannot rule out that, for other initial conditions, the model would not converge to an SSS. This limitation is related to the self-justified nature of the solution. The PLM is computed based on the income-wealth distributions visited along the ergodic distribution. One could potentially find a distribution that would lead to alternative dynamics.[3]

Interestingly, equity and debt often overshoot their SSS values. For example, when the economy starts with low levels of equity and high debt, the expert issues even more debt for a while. Only as the expert accumulates wealth through excess returns undisturbed by shocks (we are in the deterministic convergence path) does equity grow and debt fall.

[3]Similarly, one could also find other equilibria beyond the one we compute (although, despite our efforts, we failed to do so). Recall that the multiplicity of SSS(s) is different from a possible multiplicity of equilibria: in our model, we are in one basin of attraction or another depending on the sequence of shocks the economy has experienced, but the equilibrium we compute is unique given the initial condition and sequence of shocks.



(a) PLM (whole) (b) PLM (zoom)

FIGURE S3.—PLM with Chebyshev polynomials.

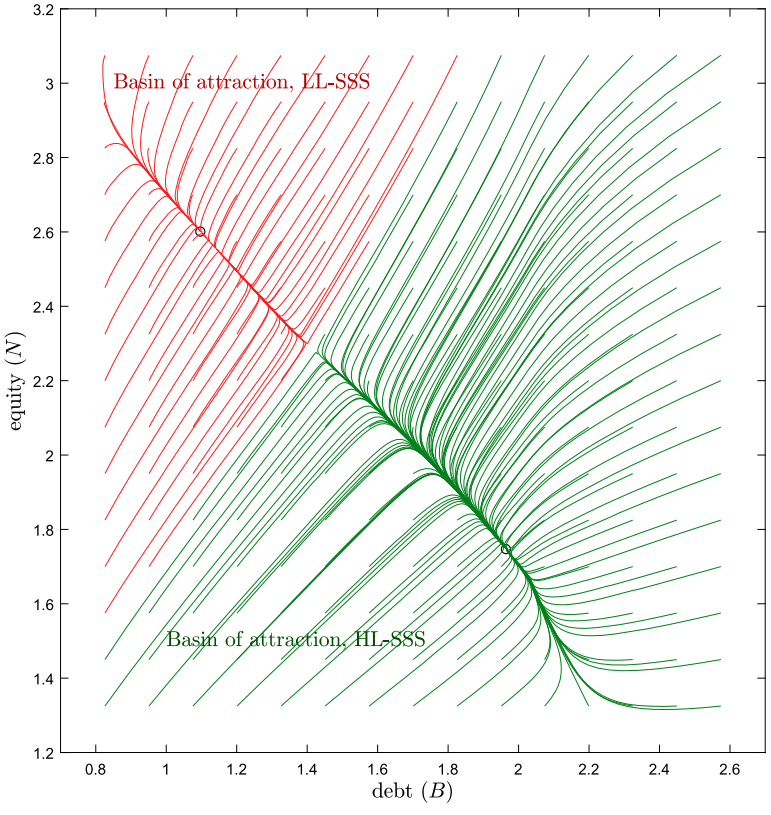


FIGURE S4.—Convergence paths.

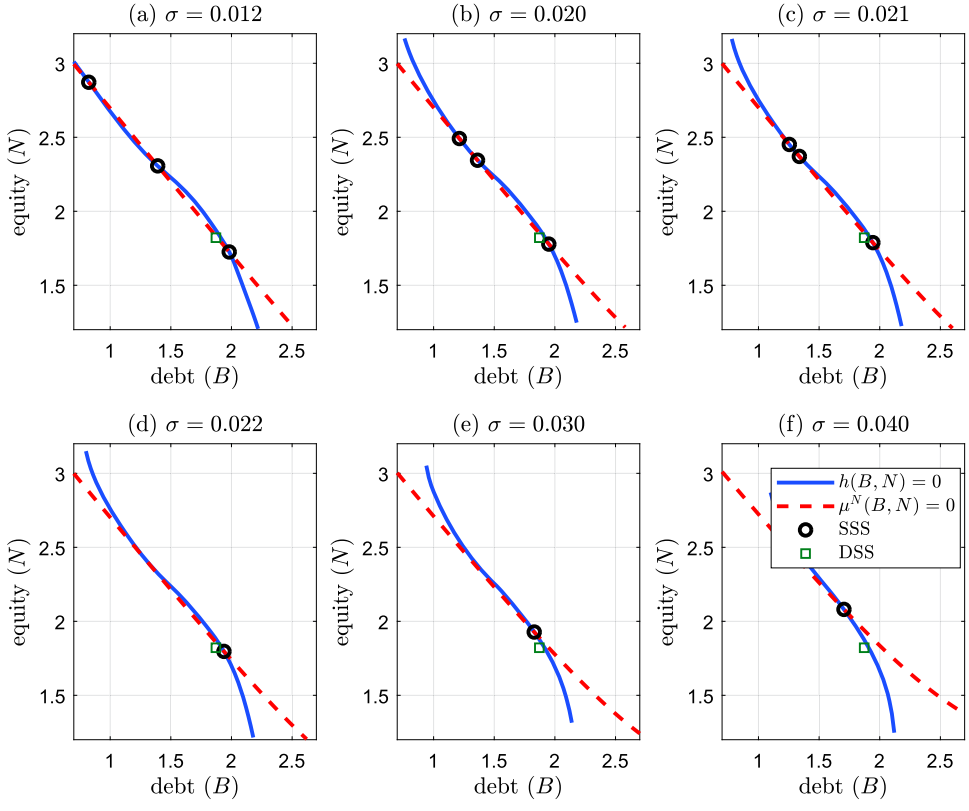


FIGURE S5.—Phase diagram as a function of σ.

APPENDIX I: COMPARATIVE STATICS

This appendix explores how the values of the SSS(s) change as we move some parameter values of the model (while keeping all the other parameter values constant). The exercises clarify the supply and demand of debt by the expert and households.

Figure S5 explores the role of σ. Each panel plots, for a different value of σ, the phase diagram of the economy following the same convention as in Figure 5. For low values of σ, the precautionary motive of households is mild and, thus, we can sustain the LL-SSS (in addition to the HL-SSS). As we increase σ, the precautionary motive becomes stronger and $h(B, N) = 0$ bends more, until the LL-SSS disappears. Even with high risk-free rates, households demand few bonds. Similarly, the HL-SSS moves to the left (i.e., less debt and more equity) as the expert is exposed to additional capital risk. This effect becomes sufficiently strong that the HL-SSS, instead of being to the right of the DSS (i.e., more debt and more equity than the DSS because of the higher excess return induced by precautionary savings), crosses to the left of the DSS. In our economy, the leverage in the HL-SSS is a negative function of σ, a roughly constant function in the unstable SSS, and an increasing function in the LL-SSS (until the additional SSS(s) disappear).

Figure S6 plots the phase diagram of the model for nine different values of z_1 (still keeping the ergodic mean of z equal to 1) from 0.67 to 0.92. Each panel follows the same convention as in Figure 5 and we only plot the segments of $h(B, N)$ and $\mu^N(B, N)$ visited in the ergodic distribution.

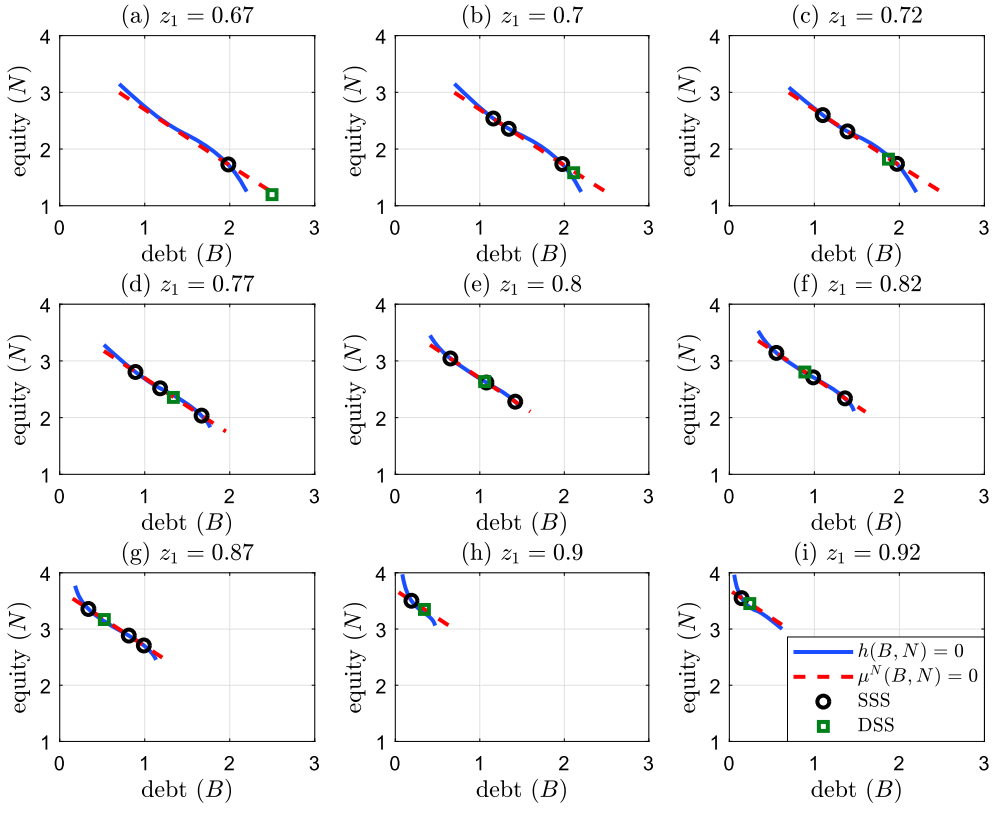


FIGURE S6.—Phase diagram as a function of z_1.

For low levels of z_1, the only SSS is the HL-SSS (see the left and center panels at the top row). Idiosyncratic risk is so high that households demand enough debt to sustain only one SSS, which, besides, has more debt than the DSS. As z_1 increases, we cross a threshold around 0.69 and we find three SSS(s), with the same interpretation as in Figure S6. As z_1 rises, the HL-SSS moves to the left of the DSS since households demand less debt to self-insure against idiosyncratic risk. By the time z_1 reaches 0.9, the precautionary demand for debt by households is now so weak that only the LL-SSS survives.

We can combine both previous exercises in Figure S7, where we plot the values of the SSS(s) as we simultaneously move aggregate and idiosyncratic risk. This figure complements Figure 11 in the main text.

Figure S8 shows the value of the SSS(s) as we vary $\widehat{\rho}$, the discount factor of the expert. As the expert becomes more impatient, the level of leverage in the HL-SSS and LL-SSS increases slightly, while the level of leverage at the DSS rises much more strongly. The reason is that as $\widehat{\rho}$ grows, households are relatively more patient and, therefore, more willing to accumulate bonds and increase leverage.

Finally, Figure S9 draws the ergodic distributions of equity and debt as z_1 varies. For low levels of z_1, most of the ergodic mass accumulates in the region of high debt and low equity. As z_1 increases, the ergodic mass spreads toward the upper left corner, first slowly, but gathering steam by the time we reach $z_1 = 0.85$. At this level, there is a bifurcation and the region around the LL-SSS becomes predominant and the higher leverage region eventually disappears. This change in the ergodic distribution is crucial for aggre-

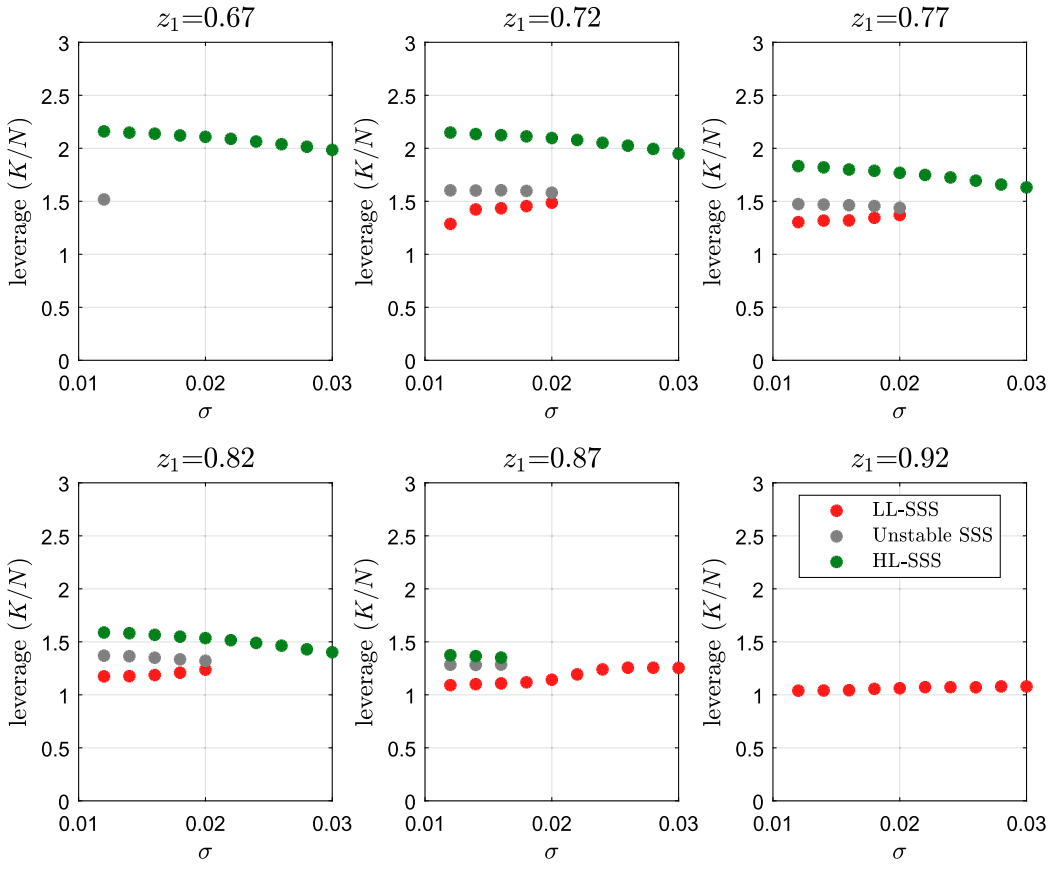


FIGURE S7.—SSS(s) as a function of σ and z_1.

gate fluctuations since, as we saw in Figure 6, the responses of the economy to a capital shock heavily depend on leverage.

APPENDIX J: VALUE FUNCTIONS

The second row of Figure S10 plots the value functions of the households as a function of assets for low- and high-labor productivity at the HL-SSS and the LL-SSS. For easy reference, in the first row of Figure S10, we reproduce the distributions of households.

The comparison of value functions shows that, for all levels of assets, households prefer to be at the LL-SSS than at the HL-SSS. This fact is not a surprise since, at the LL-SSS, the economy is less volatile and households have concave preferences only over consumption (not allowing, therefore, substitution with leisure when productivity is low). However, precautionary behavior also means that, at the HL-SSS, we will have more rich households.

The bottom row of Figure S10 shows how the value function changes after a two-standard-deviations negative capital shock: poorer households are worse off (they have lower wages), but wealthier households are better off, as their bonds pay a higher interest rate. The effect is more acute at the HL-SSS, as the persistence of wages and the risk-free interest rate are higher.

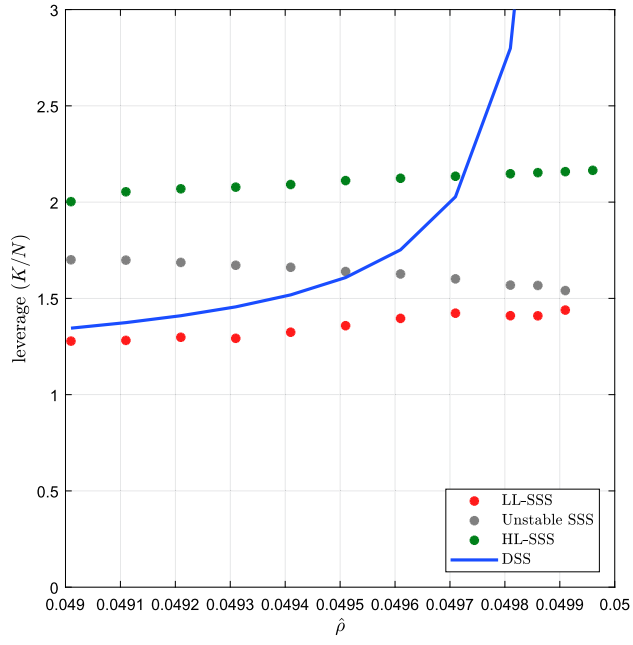


FIGURE S8.—SSS(s) as a function of $\widehat{\rho}$.

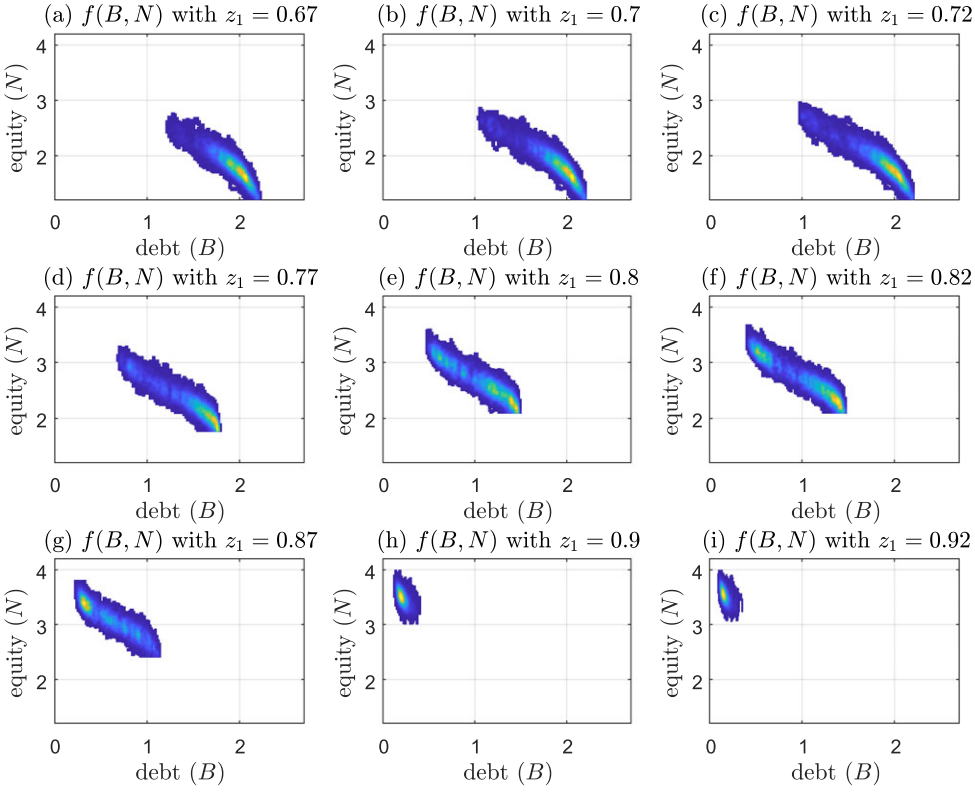


FIGURE S9.—$f(B,N)$ as a function of z_1. Lighter colors indicate higher probability.

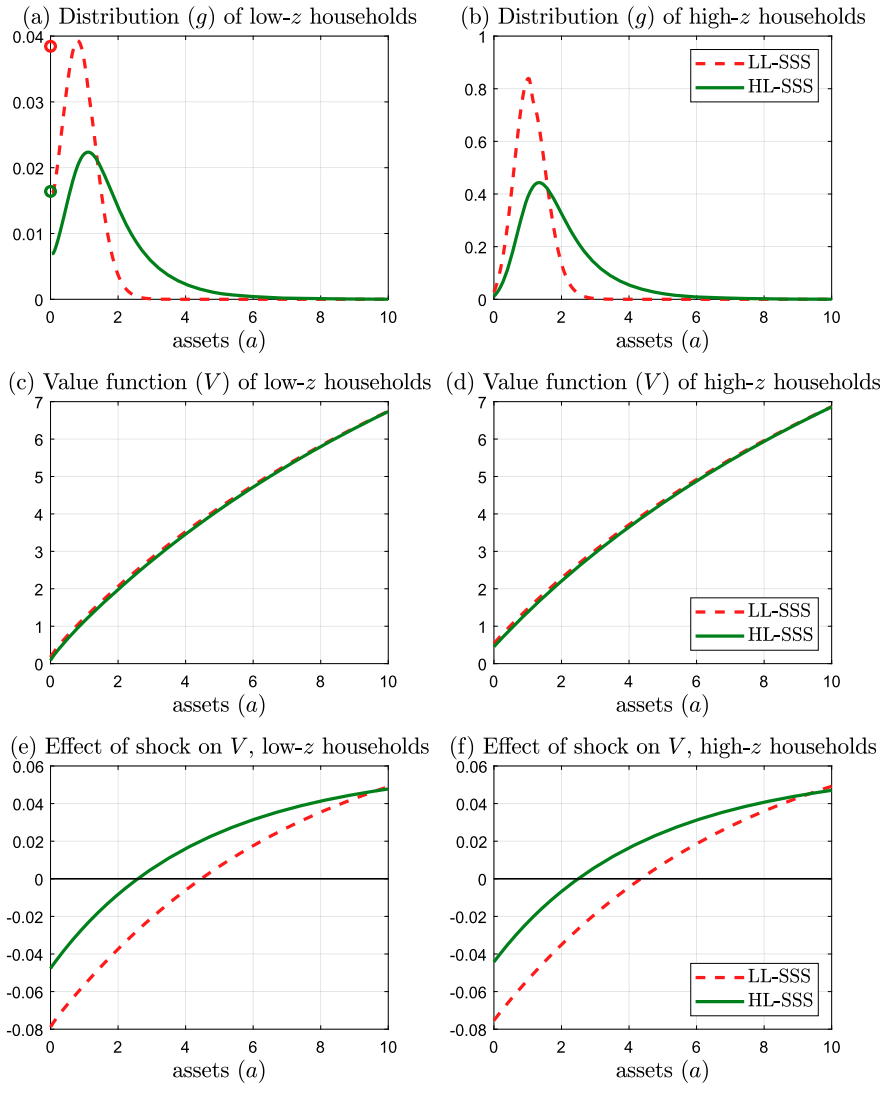


FIGURE S10.—Wealth distribution and value functions in the DSS and SSS.

APPENDIX K: GIRFS: HIGH VERSUS LOW HOUSEHOLD HETEROGENEITY

Figure S11 reports the GIRFs of the baseline, high household heterogeneity version of the model (continuous green line) to a two-standard-deviations negative capital shock when the economy is at the HL-SSS. These GIRFs are, by construction, identical to the GIRFs (also in continuous green lines) in Figure 6. Figure S11 also plots the GIRFs (in discontinuous blue line) at the unique SSS existing when $z_1 = 0.97$, the value in the top row in Figure 11.

Both sets of GIRFs vary considerably. The fall in output (panel a) is more persistent when we have more heterogeneity. We will show below how this is related to the dynamics of consumption and savings by wealthy households. The reduction in households' total consumption (panel b) is lower at impact when heterogeneity is high. In comparison, the

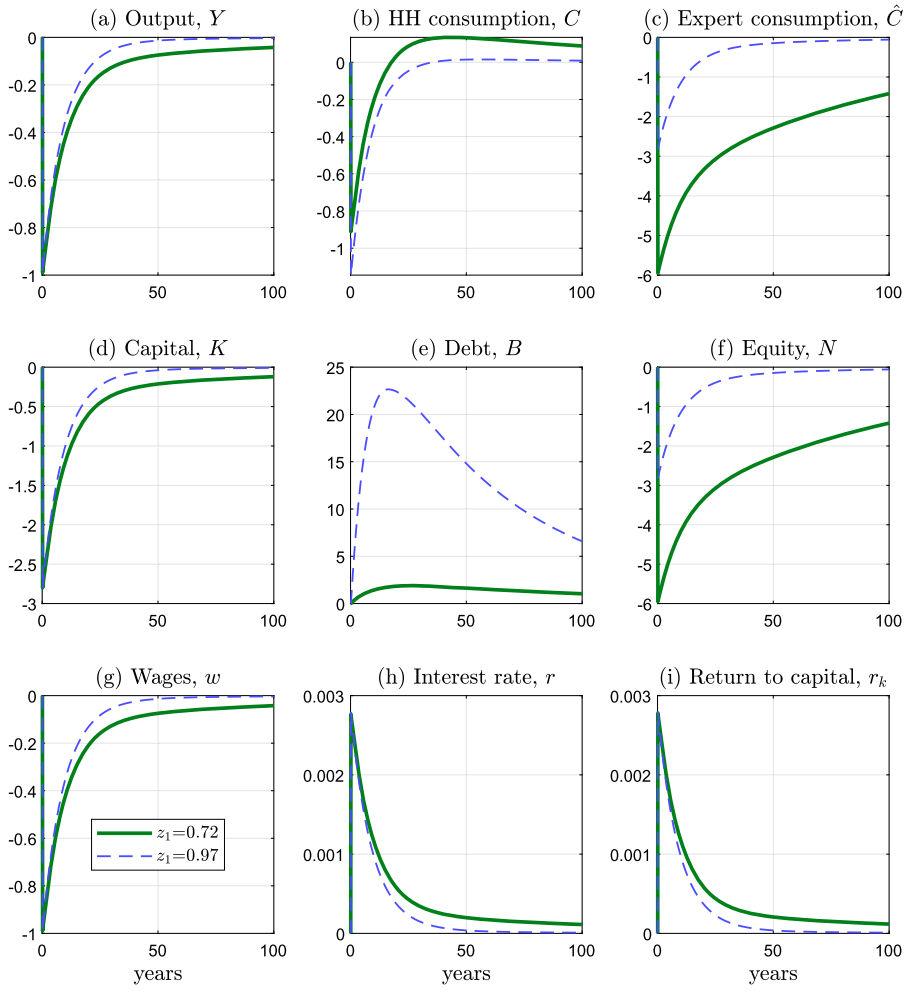


FIGURE S11.—GIRFs, different levels of heterogeneity.

consumption of the expert drops much more (panel c). This is because the expert starts with higher leverage; thus, his equity drops more (panel f) and he does not issue much additional debt for a long time (panel e).

Figure S11 also clarifies the connection between changes in microeconomic conditions with aggregate outcomes. Imagine an economy that, due to technological change or structural transformation, starts having a more turbulent labor market, with households rotating in and out of unemployment more often (or suffering, in an alternative interpretation of z, more changes to their wages while employed). The increased precautionary saving lowers the risk-free interest rate and increases, on average, leverage. Thus, the economy becomes more volatile even when the volatility of the aggregate shocks remains constant.

APPENDIX L: CONSUMPTION DECISIONS

Figure S12 documents the heterogeneity of consumption effects. The figure plots the consumption decision rules for high-productivity households ($z = z_2$) along the asset axis

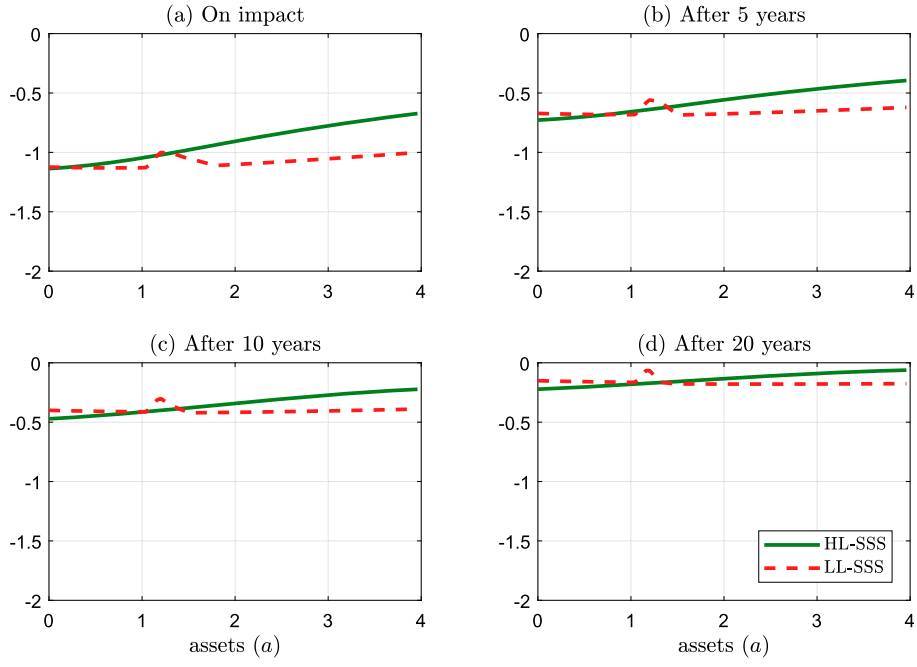


FIGURE S12.—Difference in consumption decision rules at different points in time after the shock.

(results for $z = z_1$ are qualitatively similar) when a two-standard-deviations negative capital shock hits the economy at the HL-SSS (continuous green line). To facilitate interpretation, we plot the *difference* in the consumption decision rules with respect to the case without the shock at different points in time: at impact (panel a); after 5 years (panel b); after 10 years (panel c); and after 20 years (panel d). At impact, all households reduce their consumption, but poorer households do so by a larger amount, reflecting their lower income. For instance, richer households (level of assets of 4) reduce their consumption by around one-third less than poor households (level of assets of 0). The difference in consumption reduction survives over time. The asymmetry in the consumption response is much smaller when we have a two-standard-deviations negative capital shock at the LL-SSS. Since the risk-free interest rate is less persistent in this case, the intertemporal substitution mechanism is weaker.

The asymmetric consumption responses have a direct impact on how the wealth distribution evolves. To illustrate this point, Figure S13 draws what we call the *distributional impulse response functions*, or DIRFs. A DIRF is the natural analog of a GIRF except that, instead of plotting the evolution of an aggregate variable such as output or wages, we plot the evolution of the wealth density $g_t(\cdot)$. More concretely, Figure S13 plots the difference between the density before and after the shock, $g_t(\cdot) - g_0(\cdot)$. Time, in years, is plotted on the y-axis, assets on the x-axis, and the DIRFs on the z-axis. A positive value of the DIRF at a given asset level and point in time should be read as the density is higher at that asset level and point in time than it would have been in the absence of a shock. A negative value has the opposite interpretation. In the left panel of Figure S13, we plot the DIRF to a two-standard-deviations negative capital shock when the economy is at the HL-SSS. In the right panel, we plot the DIRF to a two-standard-deviations negative capital shock when the economy is at the LL-SSS.

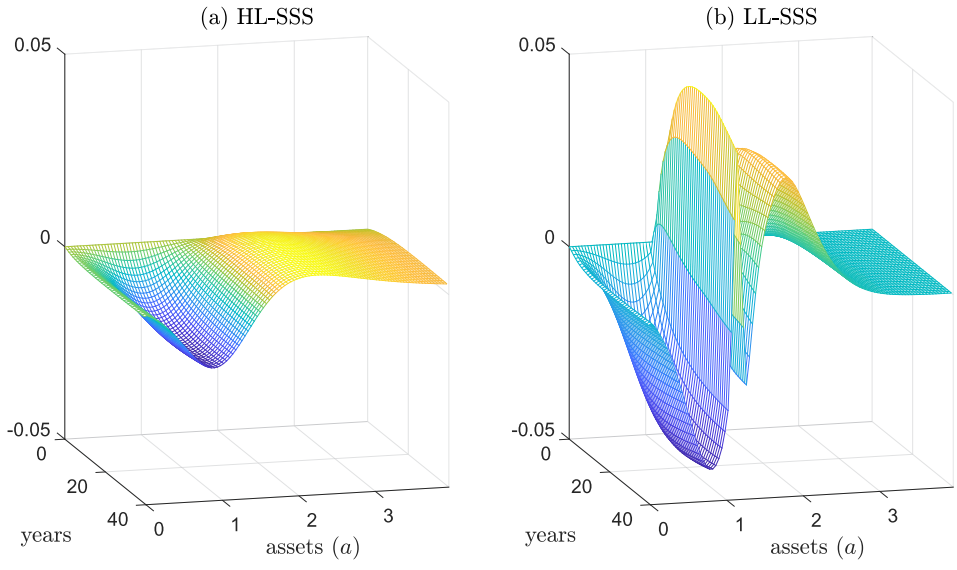


FIGURE S13.—DIRFs at the HL-SSS and LL-SSS.

In panel (a), we see how households with low assets must draw from their wealth to smooth consumption (even if consumption still drops) to compensate for lower income. This mechanism makes the DIRF negative in that region. In comparison, households with higher assets reduce their consumption to respond to a temporarily higher risk-free rate and accumulate wealth. Thus, the DIRF is positive in the region of high assets. These effects are more pronounced in panel (b). In the LL-SSS, poor households have too little debt to smooth consumption, and wealthy households accumulate much additional debt as the risk-free interest rate changes.

REFERENCES

EBRAHIMI KAHOU, MAHDI, JESÚS FERNÁNDEZ-VILLAVERDE, JESSE PERLA, AND ARNAV SOOD (2021): "Exploiting Symmetry in High-Dimensional Dynamic Programming," Working Paper 28981, National Bureau of Economic Research. [10]

FERNÁNDEZ-VILLAVERDE, JESÚS, AND JUAN F. RUBIO-RAMÍREZ (2007): "Estimating Macroeconomic Models: A Likelihood Approach," *Review of Economic Studies*, 74, 1059–1087. [11]

FERNÁNDEZ-VILLAVERDE, JESÚS, JÖEL MARBET, GALO NUÑO, AND OMAR RACHEDI (2022): "Inequality and the Zero Lower Bound,", University of Pennsylvania, Available at https://www.sas.upenn.edu/∼jesusfv/inequality_ZLB.pdf. [9]

KRUSELL, PER, AND ANTHONY A. SMITH (1998): "Income and Wealth Heterogeneity in the Macroeconomy," *Journal of Political Economy*, 106, 867–896. [9-11]

LO, ANDREW (1988): "Maximum Likelihood Estimation of Generalized Itô Processes With Discretely Sampled Data," *Econometric Theory*, 4, 231–247. [12]

Co-editor Dave Donaldson handled this manuscript.

Manuscript received 3 March, 2020; final version accepted 23 January, 2023; available online 23 February, 2023.