

# Deep Learning for Economists II: Quantitative Economics

---

Jesús Fernández-Villaverde<sup>1</sup>

November 29, 2024

<sup>1</sup>University of Pennsylvania

## Functional equations

---

# Functional equations

- A large class of problems in economics search for a function  $d$  that solves a *functional equation*:

$$\mathcal{T}(d) = \mathbf{0}$$

- More formally:

1. Let  $J^1$  and  $J^2$  be two functional spaces and let  $\mathcal{T} : J^1 \rightarrow J^2$  be an operator between these two spaces.
2. Let  $\Omega \subseteq \mathbb{R}^I$ .
3. Then, we need to find a function  $d : \Omega \rightarrow \mathbb{R}^m$  such that  $\mathcal{T}(d) = \mathbf{0}$ .

- Points to remember:

1. Regular equations are particular examples of functional equations.
2.  $\mathbf{0}$  is the space zero, different in general from the zero in the reals.
3. This formalism deals both with market equilibrium and social planner problems.

## Example I: Decision rules

- Take the basic stochastic neoclassical growth model:

$$\max \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t u(c_t)$$

$$c_t + k_{t+1} = e^{z_t} k_t^\alpha + (1 - \delta) k_t, \forall t > 0$$

$$z_t = \rho z_{t-1} + \sigma \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, 1)$$

- The first-order condition:

$$u'(c_t) = \beta \mathbb{E}_t \{ u'(c_{t+1}) (1 + \alpha e^{z_{t+1}} k_{t+1}^{\alpha-1} - \delta) \}$$

## Example I: Decision rules

- There is a decision rule (a.k.a. policy function) that gives the optimal choice of consumption and capital tomorrow given the states today:

$$d = \begin{cases} d^1(k_t, z_t) = c_t \\ d^2(k_t, z_t) = k_{t+1} \end{cases}$$

- Then:

$$\begin{aligned} \mathcal{T} &= u'(d^1(k_t, z_t)) \\ -\beta \mathbb{E}_t \left\{ u' \left( d^1 \left( d^2(k_t, z_t), z_{t+1} \right) \right) \left( 1 + \alpha e^{z_{t+1}} \left( d^2(k_t, z_t) \right)^{\alpha-1} - \delta \right) \right\} &= 0 \end{aligned}$$

- If we find  $d$ , and a transversality condition is satisfied, we are done!

## Example II: Conditional expectations

- Let us go back to our Euler equation:

$$u'(c_t) - \beta \mathbb{E}_t \left\{ u'(c_{t+1}) (1 + \alpha e^{z_{t+1}} k_{t+1}^{\alpha-1} - \delta) \right\} = 0$$

- Define now:

$$d = \begin{cases} d^1(k_t, z_t) = c_t \\ d^2(k_t, z_t) = \mathbb{E}_t \left\{ u'(c_{t+1}) (1 + \alpha e^{z_{t+1}} k_{t+1}^{\alpha-1} - \delta) \right\} \end{cases}$$

- Why? Example: ZLB.

- Then:

$$\mathcal{T}(d) = u'(d^1(k_t, z_t)) - \beta d^2(k_t, z_t) = 0$$

## Example III: Value functions

- There is a recursive problem associated with the previous sequential problem:

$$V(k_t, z_t) = \max_{k_{t+1}} \{u(c_t) + \beta \mathbb{E}_t V(k_{t+1}, z_{t+1})\}$$

$$c_t = e^{z_t} k_t^\alpha + (1 - \delta) k_t - k_{t+1}, \forall t > 0$$

$$z_t = \rho z_{t-1} + \sigma \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, 1)$$

- Then:

$$d(k_t, z_t) = V(k_t, z_t)$$

and

$$\mathcal{T}(d) = d(k_t, z_t) - \max_{k_{t+1}} \{u(c_t) + \beta \mathbb{E}_t d(k_{t+1}, z_{t+1})\} = 0$$

## Deep learning to solve functional equations

- General idea: substitute  $d(\mathbf{x})$  by  $d^n(\mathbf{x}, \theta)$  where  $\theta$  is an  $n - \text{dim}$  vector of coefficients to be determined.
- We can “learn” this function with deep learning (notice: notation with just one layer):

$$d \cong d^n(\mathbf{x}; \theta) = \theta_0 + \sum_{m=1}^M \theta_m \phi(z_m)$$

where  $\phi(\cdot)$  is an arbitrary activation function and:

$$z_m = \sum_{n=0}^N \theta_{n,m} x_n$$

- We need to sample the desired function and minimize a loss function.

- Basic setup:
  1. Natural sampling: simulated paths of the economy or grid points.
  2. Natural loss function: the implied error in the equilibrium/optimality conditions.
  3. Natural minimization algorithm: Stochastic gradient descent or variations.
- However, you want to think about this more as a framework than a concrete set of instructions.
- Let us go over different applications. Enumeration, no exhaustive list.

## Examples of code

1. An LQ optimal control problem:

[https://github.com/Mekahou/Fun-Stuff/blob/main/codes/linear%20quadratic%20DP%20DNN/3.%20LQ\\_DP\\_DNN\\_Training\\_Main.ipynb](https://github.com/Mekahou/Fun-Stuff/blob/main/codes/linear%20quadratic%20DP%20DNN/3.%20LQ_DP_DNN_Training_Main.ipynb)

2. A Neoclassical growth model (discrete time):

[https://colab.research.google.com/drive/1jbSti3LkxASZg04Bkod0EBJFXdf6xu9\\_?usp=sharing](https://colab.research.google.com/drive/1jbSti3LkxASZg04Bkod0EBJFXdf6xu9_?usp=sharing)

3. A Neoclassical growth model (continuous time):

[https://colab.research.google.com/drive/1rFfqbJYn26\\_nm-CS21Fbw\\_QV7ccM8X1T?usp=sharing](https://colab.research.google.com/drive/1rFfqbJYn26_nm-CS21Fbw_QV7ccM8X1T?usp=sharing)

4. An OLG model:

<https://github.com/sischei/DeepEquilibriumNets>

5. A Krusell-Smith and a financial frictions HA code:

<https://github.com/jesusfv/financial-frictions>

## Models with a representative agent

---

# The stochastic neoclassical growth model

- A representative household:

$$\max \mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t \log(c)$$

$$c_t + k_{t+1} = w_t + (1 + r_t - \delta) k_t, \forall t > 0$$

$$\lim_{T \rightarrow \infty} \mathbb{E} \beta^T k_{T+1} c_T^{-1} k_0, z_0 = 0$$

- A representative firm:

$$y_t = (e^{z_t})^{1-\alpha} k_t^\alpha$$

$$z_t = \rho z_{t-1} + \sigma \nu_t, \quad \nu_t \sim \mathcal{N}(0, \sigma)$$

Solving the model with  $\rho < 1$  is actually harder!

- Input markets are competitive and aggregate resource constraint holds:  $y_t = c_t + k_{t+1} - (1 - \delta) k_t$ .

## The equilibrium Markov process

- With recursive notation, the states of the model are  $X \equiv \begin{bmatrix} k & z \end{bmatrix}^\top \in \mathbb{R}_+^2$ .
- Euler equation as a functional equation on  $k'(k, z)$ :

$$1 = \mathbb{E} \left[ \beta \frac{c(k, z)}{c(k'(k, z), z')} \left( 1 - \delta + \alpha(e^{z'})^{1-\alpha} k'(k, z)^{\alpha-1} \right) \right]$$

where  $c(k, z) \equiv (e^z)^{1-\alpha} k^\alpha + (1 - \delta)k - k'(k, z)$ .

- Thus, conditioning on a policy:

$$X' = \Phi(X, z; k') = \begin{bmatrix} k'(k, z) \\ \rho z + \sigma \nu \end{bmatrix}$$

## A deep learning approximation

- We approximate  $k'(k, z)$  with a deep neural network  $k'_\theta(k, z)$  that belongs to the architecture  $\mathcal{H}(\theta)$ .
- Euler equation error:

$$\mathcal{L}(k'_\theta, X) = \left[ 1 - \sum_{i=1}^M \omega_i \left[ \beta \frac{c(k, z)}{c(k'_\theta, z'(z, \nu_i))} (1 - \delta + \alpha(z'(z, \nu_i))^{1-\alpha} (k'_\theta)^{\alpha-1}) \right] \right]^2$$

where  $z'(z, \nu_i) = \rho \log z + \sigma \nu_i$ .

- Goal:

$$k'^* = \arg \min_{k'_\theta \in \mathcal{H}(\theta)} \mathbb{E}_{X \sim \mu^*(X_0, T_{\text{pop}}; k'^*)} [\mathcal{L}(k'_\theta, X)]$$

where  $\mu^*(X_0, T_{\text{pop}}; k'^*)$  population distribution.

## Solving the equilibrium loop

- If we knew  $\mu^*(\cdot)$ , a trivial algorithm to solve the problem above would be to sample  $\mathcal{D}$  points from  $\mu^*(\cdot)$  and solve the empirical risk:

$$\theta^* = \arg \min_{\theta \in \Theta} \frac{1}{|\mathcal{D}|} \sum_{X \in \mathcal{D}} \mathcal{L}(k'_\theta, X).$$

- The challenge, of course, is that we do not know  $\mu^*(\cdot)$  precisely because of the equilibrium feedback loop.
- This is the main difference between applications of deep learning in engineering/natural sciences and economics.

## The idea

---

- Sample from a misspecified population distribution and periodically regenerate those samples as the policy function is refined.
- That is, given a  $k'_i$ , we solve:

$$k'_{i+1} \equiv \arg \min_{k'_\theta \in \mathcal{H}(\theta)} \mathbb{E}_{X \sim \mu^*(x_0, T_{\text{pop}}; k'_i)} [\mathcal{L}(k'_\theta, X)],$$

until convergence.

- In our case, we start with a  $k'_1(\cdot)$  policy from the Solow model with constant savings rate  $s_{ss}$ ,  
$$k'_1(k, z) = s_{ss}z^{1-\alpha}k^\alpha + (1 - \delta)k.$$

## An iterative algorithm

1. Solve the empirical risk minimization problem given the current  $\mathcal{D}_i$ ,

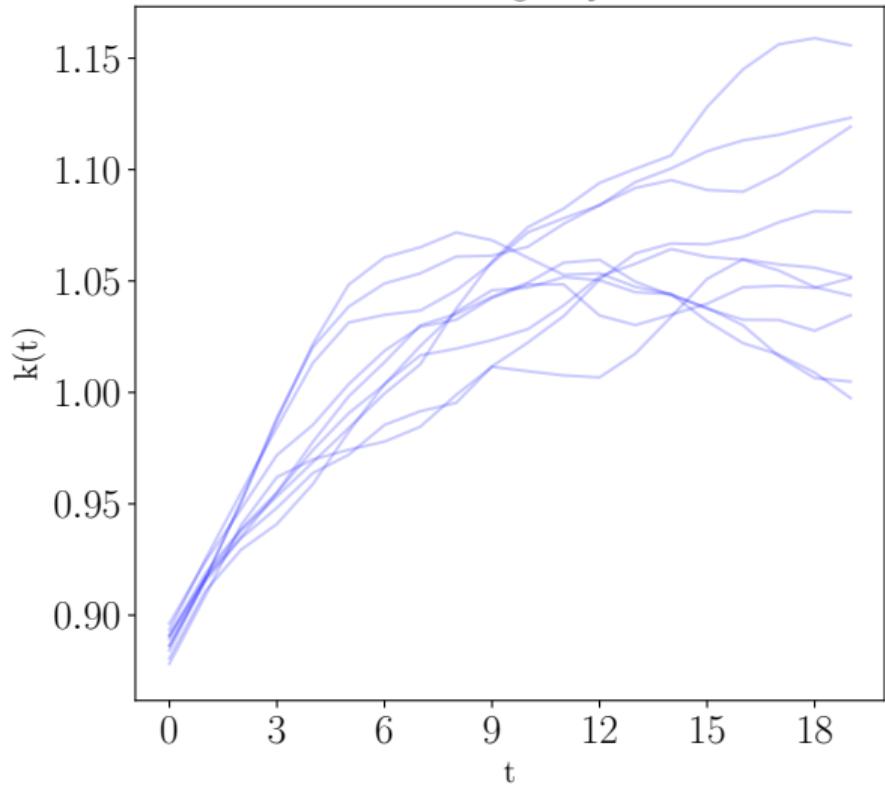
$$\theta_{i+1} = \arg \min_{\theta \in \Theta} \frac{1}{|\mathcal{D}_i|} \sum_{x \in \mathcal{D}_i} \mathcal{L}(k'_\theta, X).$$

2. Generate a new  $\mathcal{D}_{i+1}$  using samples from  $\mu^*(X_0, T_{\text{pop}}, k'_{\theta_{i+1}})$ .
3. Iterate until the empirical risk is below a threshold.

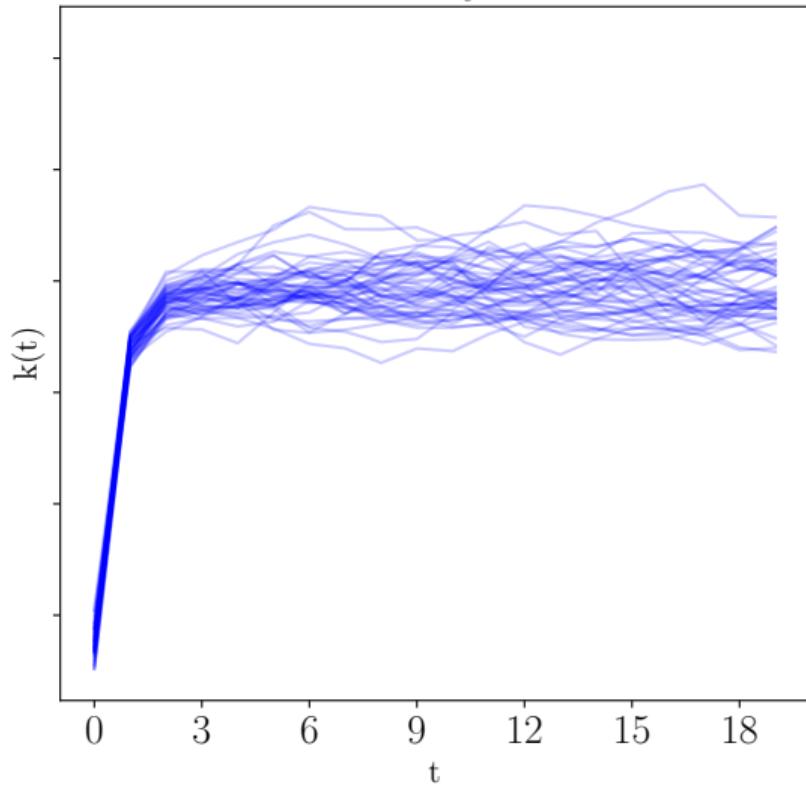
## Parameters, hyperparameters, and optimizer

- Parameter values:  $\beta = 0.9$ ,  $\alpha = 1/3$ ,  $\delta = 0.2$ ,  $\rho = 0.9$ , and  $\sigma = 0.025$ .
- $M = 7$  nodes for quadrature.
- A deep neural network with four layers, each with 128 nodes, a ReLU activation function  $\max\{0, x\}$ , and a softplus final layer,  $\log(1 + e^x)$ . This leads to approximately  $50K$  parameters in  $\theta$ .
- $T_{\text{pop}} = 20$  and generate trajectories  $(k_t, z_t)$  for  $t = 0, \dots, T_{\text{pop}}$ .
- $X_0 \sim \mathcal{N}(\begin{bmatrix} k_0 & z_0 \end{bmatrix}^\top, \Sigma)$  with  $z_0 = 1$ ,  $k_0 = 0.8 \times k_{ss}$ , and  $\Sigma \equiv \begin{bmatrix} 0.008^2 & 0 \\ 0 & 0.02^2 \end{bmatrix}$ .
- 100 iterations of the L-BFGS optimizer and regenerate the  $\mathcal{D}_i$  every five iterations.
- The algorithm, coded in Jax, runs in approximately 1.2 seconds using only the CPU of a laptop.

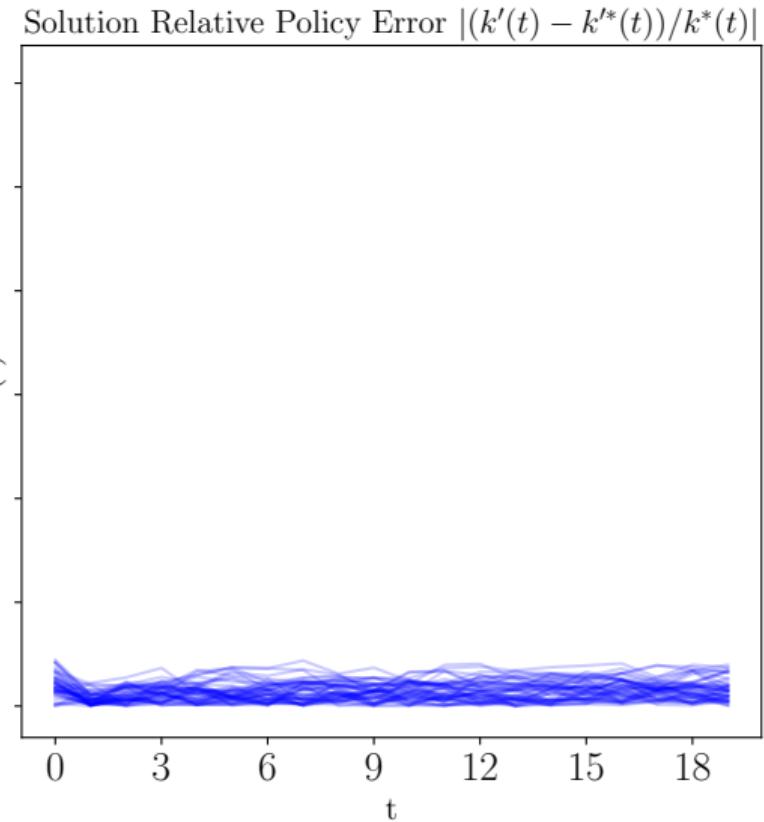
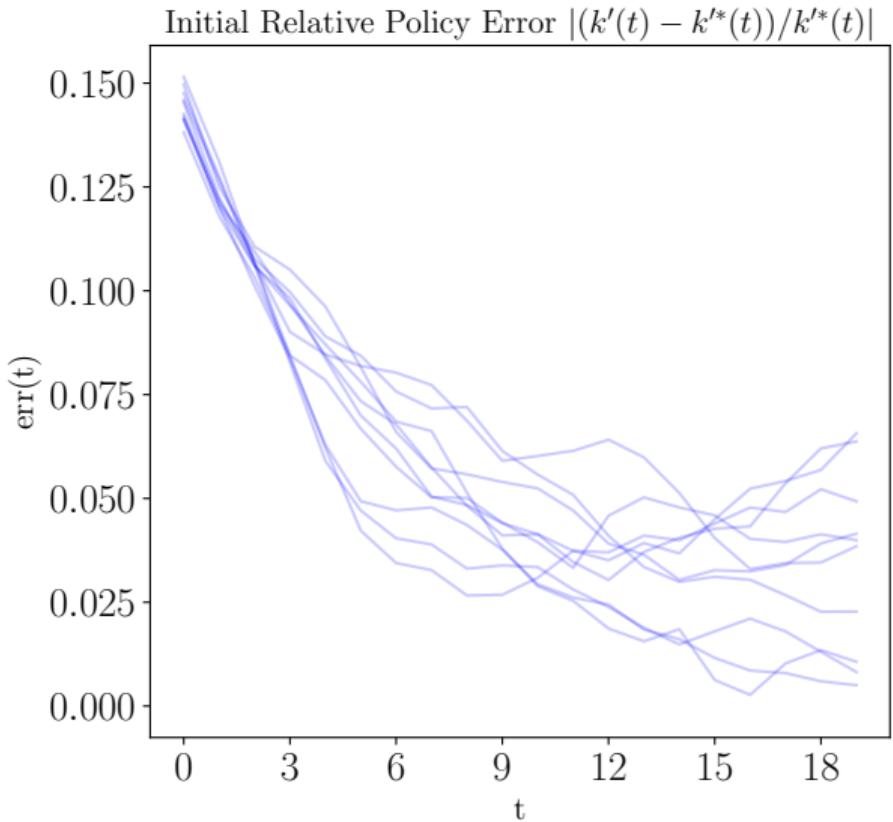
Initial Training Trajectories



Solution Trajectories



**Figure 1:** Initial vs. final trajectories of capital.



**Figure 2:** Initial vs. final policy errors

## **Models with heterogeneous agents**

---

## The challenge

---

- To compute and take to the data models with heterogeneous agents, we need to deal with:
  1. The distribution of agents  $G_t$ .
  2. The operator  $H(\cdot)$  that characterizes how  $G_t$  evolves:

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

or

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

given the other aggregate states of the economy  $S_t$ .

- How do we track  $G_t$  and compute  $H(G_t, S_t)$ ?

## A common approach

- If we are dealing with  $N$  discrete types, we keep track of  $N - 1$  weights.
- If we are dealing with continuous types, we extract a finite number of features from  $G_t$ :
  1. Moments.
  2. Q-quantiles.
  3. Weights in a mixture of normals...
- We stack either the weights or features of the distribution in a vector  $\mu_t$ .
- We assume  $\mu_t$  follows the operator  $h(\mu_t, S_t)$  instead of  $H(G_t, S_t)$ .
- We parametrize  $h(\mu_t, S_t)$  as  $h^j(\mu_t, S_t; \theta)$ .
- We determine the unknown coefficients  $\theta$  such that an economy where  $\mu_t$  follows  $h^j(\mu_t, S_t; \theta)$  replicates as well as possible the behavior an economy where  $G_t$  follows  $H(\cdot)$ .

## Example: Basic Krusell-Smith model

- Two aggregate variables: aggregate productivity shock and household distribution  $G_t(a, z)$  where:

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

- We summarize  $G_t(a, \cdot)$  with the log of its mean:  $\mu_t = \log K_t$  (extending to higher moments is simple, but tedious).
- We parametrize  $\underbrace{\log K_{t+1}}_{\mu_{t+1}} = \underbrace{\theta_0(s_t) + \theta_1(s_t) \log K_t}_{h^j(\mu_t, s_t; \theta)}$ .
- We determine  $\{\theta_0(s_t), \theta_1(s_t)\}$  by OLS run on a simulation.

## Problems

---

- No much guidance regarding feature and parameterization selection in general cases.
  - Yes, keeping track of the log of the mean and a linear functional form works well for the basic model. But what about an arbitrary model?
  - Method suffers from “curse of dimensionality”: difficult to implement with many state variables or high  $N$ /higher moments.
  - Lack of theoretical foundations (Does it converge? Under which metric?).

## How can deep learning help?

- Deep learning addresses challenges:
  1. How to extract features from an infinite-dimensional object efficiently.
  2. How to parametrize the non-linear operator mapping how distributions evolve.
  3. How to tackle the “curse of dimensionality.”
- Given time limitations, today I will discuss the last two points.
- In our notation of  $y = f(x)$ :
  1.  $y = \mu_{t+1}$ .
  2.  $x = (\mu_t, S_t)$ .

## **Example: Financial frictions and the wealth distribution**

---

## Motivation

- Recently, many papers have documented the nonlinear relations between financial variables and aggregate fluctuations.
- For example, **Jordà et al. (2016)** have gathered data from 17 advanced economies over 150 years to show how output growth, volatility, skewness, and tail events all seem to depend on the levels of leverage in an economy.
- Similarly, **Adrian et al. (2019a)** have found how, in the U.S., sharply negative output growth follows worsening financial conditions associated with leverage.
- Can a fully nonlinear DSGE model account for these observations?
- To answer this question, we postulate, compute, and estimate a continuous-time DSGE model with a financial sector, modeled as a representative financial expert, and households subject to uninsurable idiosyncratic labor productivity shocks.

## The main takeaway

- The interaction between the supply of bonds by the financial sector and the precautionary demand for bonds by households produces significant *endogenous aggregate risk*.
- This risk induces an endogenous regime-switching process for output, the risk-free rate, excess returns, debt, and leverage.
- Mechanism: *endogenous aggregate risk*.
- We generate:
  1. Multimodal distributions of aggregate variables ([Adrian et al., 2019b](#)).
  2. Time-varying levels of volatility and skewness for aggregate variables ([Fernández-Villaverde and Guerrón, 2020](#)).
  3. Supercycles of borrowing and deleveraging ([Reinhart and Rogoff, 2009](#)).

## HA vs. RA

---

- Our findings are in contrast with the properties of the representative household version of the model.
- While the consumption decision rule of the households is close to linear with respect to the household state variables, it is sharply nonlinear with respect to the aggregate state variables.
- This point is more general: agent heterogeneity might matter even if the decision rules of the agents are linear with respect to individual state variables.
- Thus, changes in the forces behind precautionary savings affect aggregate variables, and we can offer a novel and simultaneous account of:
  1. The recent heightened fragility of the advanced economies to adverse shocks.
  2. The rise in wealth inequality witnessed before the 2007-2008 financial crisis.
  3. The increase in debt and leverage experienced during the same period.
  4. The low risk-free interest rates of the last two decades.

## Methodological contribution

---

- New approach to (globally) compute and estimate with the likelihood approach HA models:
  1. Computation: we use tools from machine learning.
  2. Estimation: we use tools from inference with diffusions.
- Strong theoretical foundations and many practical advantages.
  1. Deal with a large class of arbitrary operators efficiently.
  2. Algorithm that is i) easy to code, ii) stable, iii) scalable, and iv) massively parallel.

# The firm

- Representative firm with technology:

$$Y_t = K_t^\alpha L_t^{1-\alpha}$$

- Competitive input markets:

$$w_t = (1 - \alpha) K_t^\alpha L_t^{-\alpha}$$

$$rc_t = \alpha K_t^{\alpha-1} L_t^{1-\alpha}$$

- Aggregate capital evolves:

$$\frac{dK_t}{K_t} = (\iota_t - \delta) dt + \sigma dZ_t$$

- Instantaneous return rate on capital  $dr_t^k$ :

$$dr_t^k = (rc_t - \delta) dt + \sigma dZ_t$$

## The expert

- Representative expert (financial intermediary) with preferences:

$$\mathbb{E}_0 \left[ \int_0^\infty e^{-\hat{\rho}t} \log(\hat{C}_t) dt \right]$$

- Expert rents capital  $K_t$  to firms and issues risk-free debt  $B_t$  at rate  $r_t$  to households.
- Financial friction: expert cannot issue state-contingent claims and must absorb all risk from capital.
- Expert's net wealth  $N_t = K_t - B_t$  evolves:

$$\begin{aligned} dN_t &= K_t dr_t^k - r_t B_t dt - \hat{C}_t dt \\ &= \left[ (r_t + \omega_t (rc_t - \delta - r_t)) N_t - \hat{C}_t \right] dt + \sigma \omega_t N_t dZ_t \end{aligned}$$

where  $\omega_t \equiv \frac{K_t}{N_t}$  is the leverage ratio.

- $K_t$  follows:

$$dK_t = dN_t + dB_t$$

- Continuum of infinitely-lived households with unit mass with preferences:

$$\mathbb{E}_0 \left[ \int_0^\infty e^{-\rho t} \frac{c_t^{1-\gamma} - 1}{1-\gamma} dt \right]$$

- Heterogeneous in wealth  $a_m$  and labor productivity  $z_m$  for  $m \in [0, 1]$ .
- $z_t$  units of labor valued at wage  $w_t$  evolves stochastically following a Markov chain:
  1.  $z_t \in \{z_1, z_2\}$ , with  $z_1 < z_2$  and ergodic mean 1.
  2. Jump intensity from state 1 to state 2:  $\lambda_1$  (reverse intensity is  $\lambda_2$ ).
- Distribution of households  $G_t(a, z)$ .

## Households II

- Households save  $a_t \geq 0$  in the riskless debt issued by experts with an interest rate  $r_t$ :

$$da_t = (w_t z_t + r_t a_t - c_t) dt = s(a_t, z_t, K_t, G_t) dt$$

- Optimal choice:  $c_t = c(a_t, z_t, K_t, G_t)$ .
- Total consumption by households:

$$C_t \equiv \int c(a_t, z_t, K_t, G_t) dG_t(a, z)$$

- $G_t(a, z)$  has a Radon-Nikodym derivative  $g_t(a, z)$  that follows the KF equation:

$$\frac{\partial g_{it}}{\partial t} = -\frac{\partial}{\partial a} (s(a_t, z_t, K_t, G_t) g_{it}(a)) - \lambda_i g_{it}(a) + \lambda_j g_{jt}(a), \quad i \neq j = 1, 2$$

where  $g_{it}(a) \equiv g_t(a, z_i)$ ,  $i = 1, 2$ .

## Market clearing

1. Total amount of labor rented by the firm is equal to labor supplied:

$$L_t = \int z dG_t = 1$$

Then, total payments to labor are given by  $w_t$ .

2. Total amount of debt of the expert equals the total households' savings:

$$B_t \equiv \int adG_t (da, dz)$$

with  $dB_t = (w_t + r_t B_t - C_t) dt$ .

3. By the resource constraint,  $dK_t = (Y_t - \delta K_t - C_t - \hat{C}_t) dt + \sigma K_t dZ_t$ , we get:

$$\iota_t = \frac{Y_t - C_t - \hat{C}_t}{K_t}$$

## Equilibrium

An equilibrium in this economy is composed by a set of prices  $\{w_t, rc_t, r_t, r_t^k\}_{t \geq 0}$ , quantities  $\{K_t, N_t, B_t, \hat{C}_t, c_{mt}\}_{t \geq 0}$ , and a density  $\{g_t(\cdot)\}_{t \geq 0}$  such that:

1. Given  $w_t$ ,  $r_t$ , and  $g_t$ , the solution of the household  $m$ 's problem is  $c_t = c(a_t, z_t, K_t, G_t)$ .
2. Given  $r_t^k$ ,  $r_t$ , and  $N_t$ , the solution of the expert's problem is  $\hat{C}_t$ ,  $K_t$ , and  $B_t$ .
3. Given  $K_t$ , firms maximize their profits and input prices are given by  $w_t$  and  $rc_t$ .
4. Given  $w_t$ ,  $r_t$ , and  $c_t$ ,  $g_t$  is the solution of the KF equation.
5. Given  $g_t$  and  $B_t$ , the debt market clears.

## Characterizing the equilibrium I

- First, we proceed with the expert's problem. Because of log-utility:

$$\hat{C}_t = \hat{\rho} N_t$$

$$\omega_t = \hat{\omega}_t = \frac{rc_t - \delta - r_t}{\sigma^2}$$

- We can use the equilibrium values of  $rc_t$ ,  $L_t$ , and  $\omega_t$  to get the wage:

$$w_t = (1 - \alpha) K_t^\alpha$$

the rental rate of capital:

$$rc_t = \alpha K_t^{\alpha-1}$$

and the risk-free interest rate:

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

## Characterizing the equilibrium II

- Expert's net wealth evolves as:

$$dN_t = \underbrace{\left( \alpha K_t^{\alpha-1} - \delta - \hat{\rho} - \sigma^2 \left( 1 - \frac{K_t}{N_t} \right) \frac{K_t}{N_t} \right) N_t dt}_{\mu_t^N(B_t, N_t)} + \underbrace{\sigma K_t}_{\sigma_t^N(B_t, N_t)} dZ_t$$

- And debt as:

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

- Nonlinear structure of law of motion for  $dN_t$  and  $dB_t$ .
- We need to find:

$$C_t \equiv \int c(a_t, z_t, K_t, G_t) g_t(a, z) dadz$$

$$\frac{\partial g_{it}}{\partial t} = -\frac{\partial}{\partial a} (s(a_t, z_t, K_t, G_t) g_{it}(a)) - \lambda_i g_{it}(a) + \lambda_j g_{jt}(a), \quad i \neq j = 1, 2$$

# The DSS

- No aggregate shocks ( $\sigma = 0$ ), but we still have idiosyncratic household shocks.
- Then:

$$r = r_t^k = rc_t - \delta = \alpha K_t^{\alpha-1} - \delta$$

and

$$dN_t = (\alpha K_t^{\alpha-1} - \delta - \hat{\rho}) N_t dt$$

- Since in a steady state the drift of the expert's wealth must be zero, we get:

$$K = \left( \frac{\hat{\rho} + \delta}{\alpha} \right)^{\frac{1}{\alpha-1}}$$

and:

$$r = \hat{\rho} < \rho$$

- The value of  $N$  is given by the dispersion of the idiosyncratic shocks (no analytic expression).

## How do we find aggregate consumption?

- As in Krusell and Smith (1998), households only track a finite set of  $n$  moments of  $g_t(a, z)$  to form their expectations.
- No exogenous state variable (shocks to capital encoded in  $K$ ). Instead, there are two endogenous states.
- For ease of exposition, we set  $n = 1$ . The solution can be trivially extended to the case with  $n > 1$ .
- More concretely, households consider a *perceived law of motion* (PLM) of aggregate debt:

$$dB_t = h(B_t, N_t) dt$$

where

$$h(B_t, N_t) = \frac{\mathbb{E}[dB_t | B_t, N_t]}{dt}$$

## A new HJB equation

- Given the PLM, the household's Hamilton-Jacobi-Bellman (HJB) equation becomes:

$$\begin{aligned}\rho V_i(a, B, N) = & \max_c \frac{c^{1-\gamma} - 1}{1-\gamma} + s \frac{\partial V_i}{\partial a} + \lambda_i [V_j(a, B, N) - V_i(a, B, N)] \\ & + h(B, N) \frac{\partial V_i}{\partial B} + \mu^N(B, N) \frac{\partial V_i}{\partial N} + \frac{[\sigma^N(B, N)]^2}{2} \frac{\partial^2 V_i}{\partial N^2}\end{aligned}$$

$i \neq j = 1, 2$ , and where

$$s = s(a, z, N + B, G)$$

- We solve the HJB with a first-order, implicit upwind scheme in a finite difference stencil.
- Sparse system. Why?
- Alternatives for solving the HJB? Meshfree, FEM, deep learning, ...

## An algorithm to find the PLM

- 1) Start with  $\mathbf{h}_0$ , an initial guess for  $\mathbf{h}$ .
- 2) Using current guess  $\mathbf{h}_n$ , solve for the household consumption,  $\mathbf{c}_m$ , in the HJB equation.
- 3) Construct a time series for  $B_t$  by simulating by  $J$  periods the cross-sectional distribution of households with a constant time step  $\Delta t$  (starting at DSS and with a burn-in).
- 4) Given  $B_t$ , find  $N_t$ ,  $K_t$ , and:

$$\hat{\mathbf{h}} = \left\{ \hat{h}_1, \hat{h}_2, \dots, \hat{h}_J \equiv \frac{B_{t_j + \Delta t} - B_{t_j}}{\Delta t}, \dots, \hat{h}_J \right\}$$

- 5 ) Define  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_J\}$ , where  $\mathbf{x}_j = \{x_j^1, x_j^2\} = \{B_{t_j}, N_{t_j}\}$ .
- 6) Use  $(\hat{\mathbf{h}}, \mathbf{X})$  and a universal nonlinear approximator to obtain  $\mathbf{h}_{n+1}$ , a new guess for  $\mathbf{h}$ .
- 7) Iterate steps 2)-6) until  $\mathbf{h}_{n+1}$  is sufficiently close to  $\mathbf{h}_n$ .

## A universal nonlinear approximator

- We approximate the PLM with a neural network (NN):

$$h(\mathbf{x}; \theta) = \theta_0^1 + \sum_{q=1}^Q \theta_q^1 \phi \left( \theta_{0,q}^2 + \sum_{i=1}^D \theta_{i,q}^2 x^i \right)$$

where  $Q = 16$ ,  $D = 2$ , and  $\phi(z) = \log(1 + e^z)$ .

- $\theta$  is selected as:

$$\theta^* = \arg \min_{\theta} \frac{1}{2} \sum_{j=1}^J \left\| h(\mathbf{x}_j; \theta) - \hat{h}_j \right\|^2$$

## Estimation with aggregate variables I

- $D + 1$  observations of  $Y_t$  at fixed time intervals  $[0, \Delta, 2\Delta, \dots, D\Delta]$ :

$$Y_0^D = \{Y_0, Y_\Delta, Y_{2\Delta}, \dots, Y_D\}.$$

- More general case: sequential Monte Carlo approximation to the Kushner-Stratonovich equation ([Fernández-Villaverde and Rubio Ramírez, 2007](#)).
- We are interested in estimating a vector of structural parameters  $\Psi$ .
- Likelihood:

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

where

$$p_Y(Y_{d\Delta} | Y_{(d-1)\Delta}; \Psi) = \int f_{d\Delta}(Y_{d\Delta}, B) dB.$$

given a density,  $f_{d\Delta}(Y_{d\Delta}, B)$ , implied by the solution of the model.

## Estimation with aggregate variables II

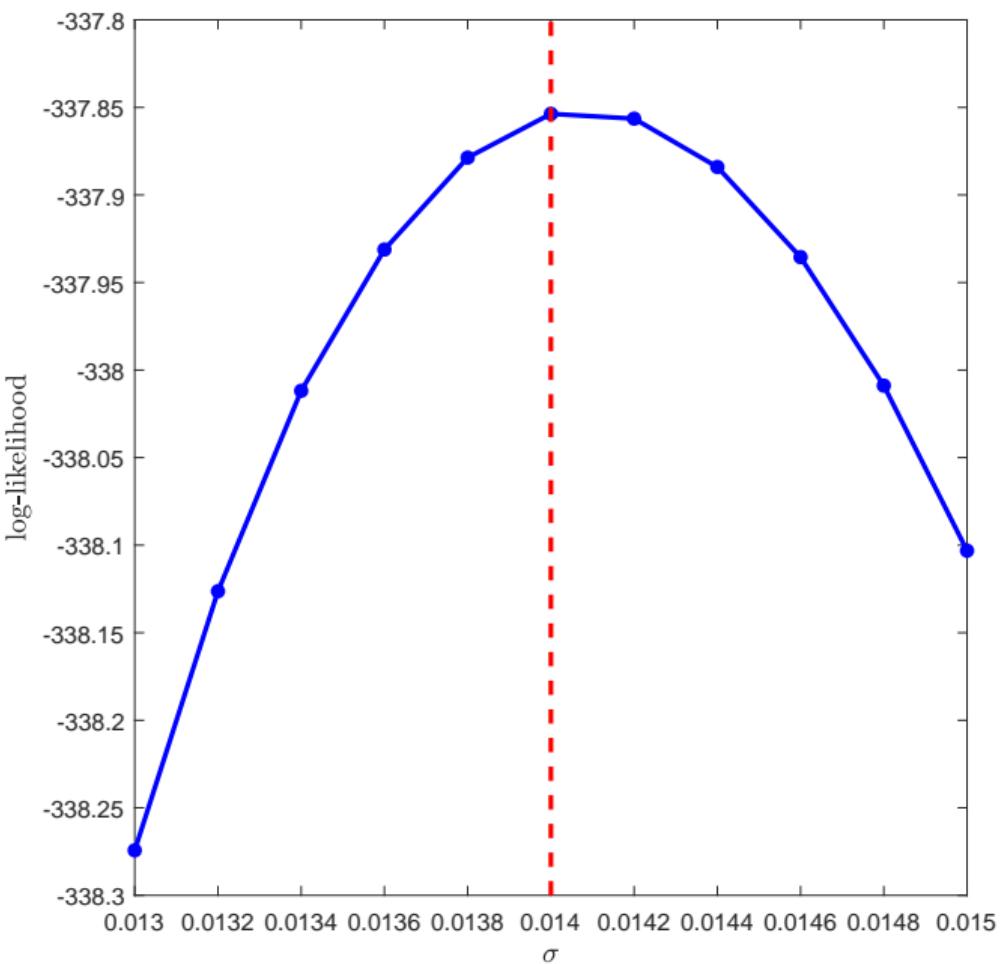
- After finding the diffusion for  $Y_t$ ,  $f_t^d(Y, B)$  follows the Kolmogorov forward (KF) equation in the interval  $[(d - 1)\Delta, d\Delta]$ :

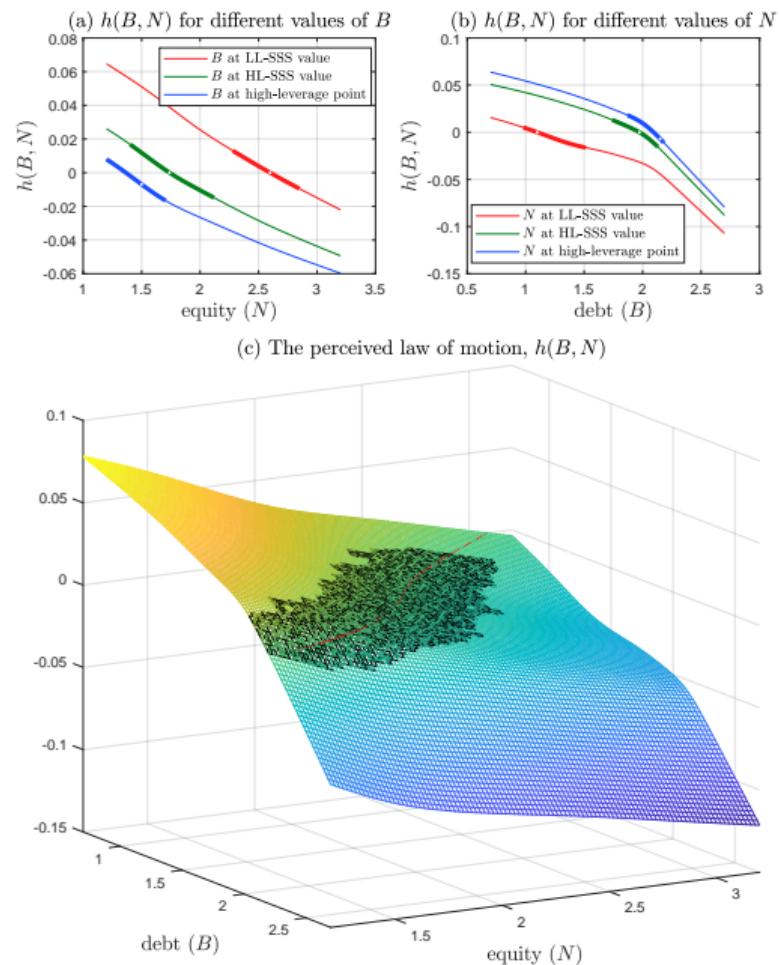
$$\begin{aligned}\frac{\partial f_t}{\partial t} &= -\frac{\partial}{\partial Y} [\mu^Y(Y, B)f_t(Y, B)] - \frac{\partial}{\partial B} [h(B, Y^{\frac{1}{\alpha}} - B)f_t^d(Y, B)] \\ &\quad + \frac{1}{2} \frac{\partial^2}{\partial Y^2} [(\sigma^Y(Y))^2 f_t(Y, B)]\end{aligned}$$

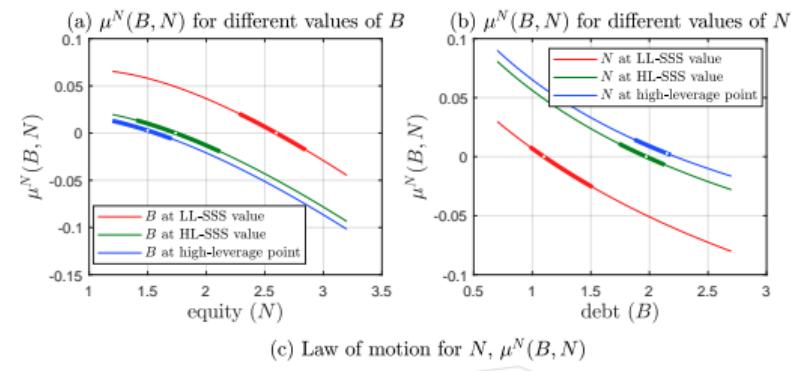
- The operator in the KF equation is the adjoint of the infinitesimal generator of the HJB.
- Thus, the solution of the KF equation amounts to transposing and inverting a sparse matrix that has already been computed.
- Our approach provides a highly efficient way of evaluating the likelihood once the model is solved.
- Conveniently, retraining the neural network is easy for new parameter values.

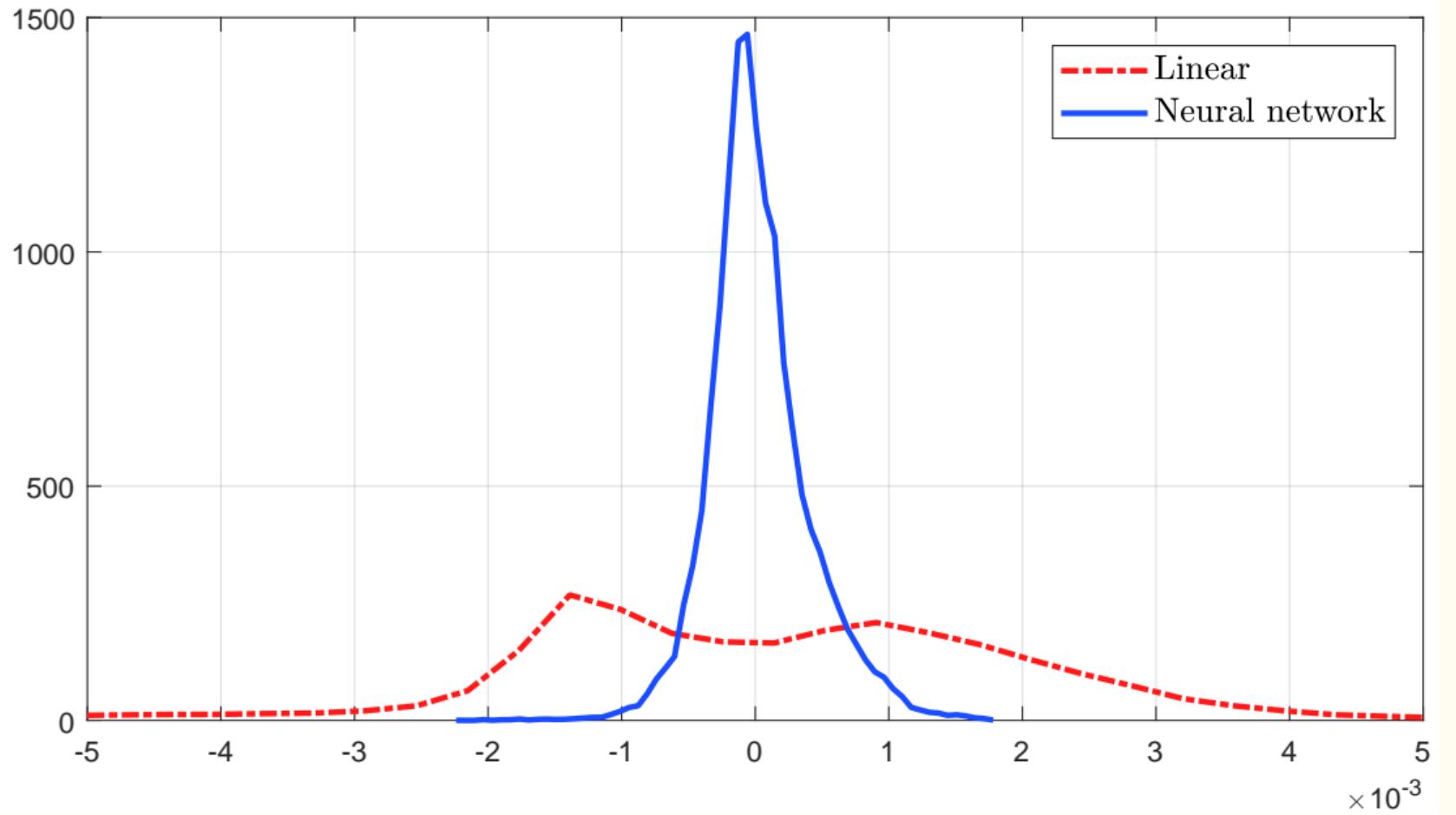
## Parametrization

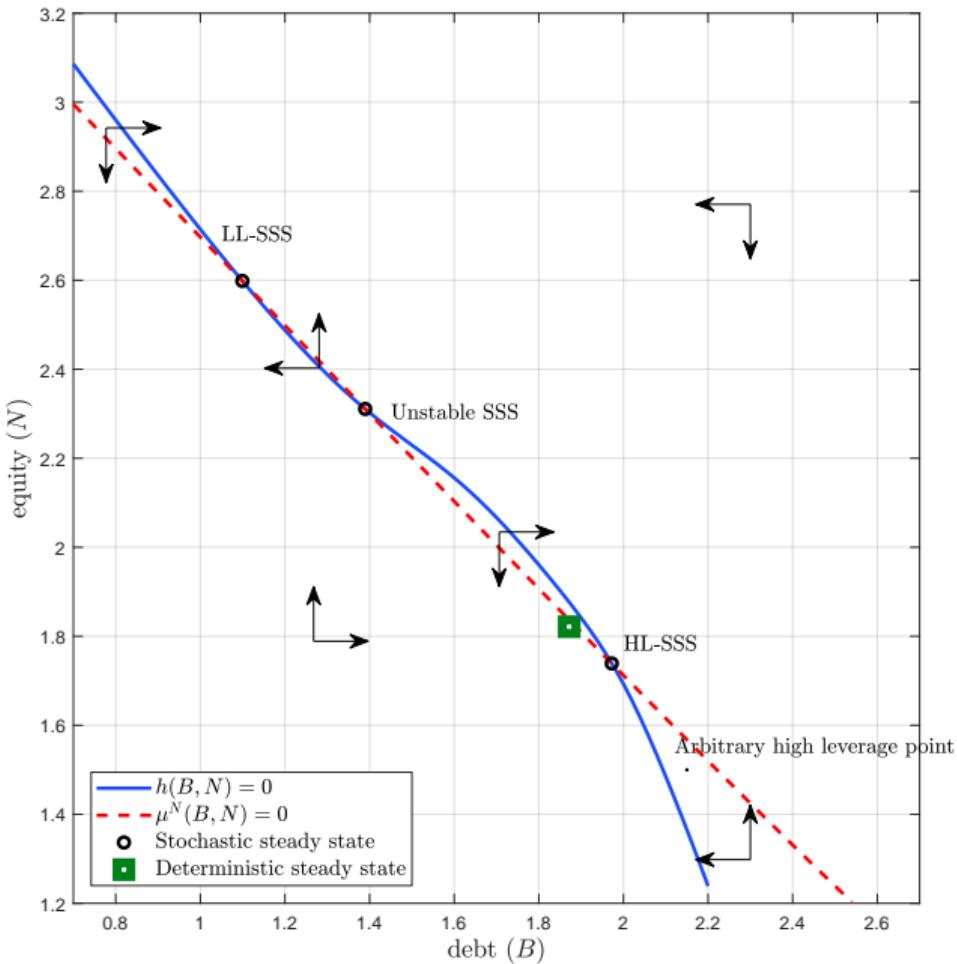
Parameter	Value	Description	Source/Target
$\alpha$	0.35	capital share	standard
$\delta$	0.1	yearly capital depreciation	standard
$\gamma$	2	risk aversion	standard
$\rho$	0.05	households' discount rate	standard
$\lambda_1$	0.986	transition rate u.-to-e.	monthly job finding rate of 0.3
$\lambda_2$	0.052	transition rate e.-to-u.	unemployment rate 5 percent
$y_1$	0.72	income in unemployment state	Hall and Milgrom (2008)
$y_2$	1.015	income in employment state	$\mathbb{E}(y) = 1$
$\hat{\rho}$	0.0497	experts' discount rate	$K/N = 2$

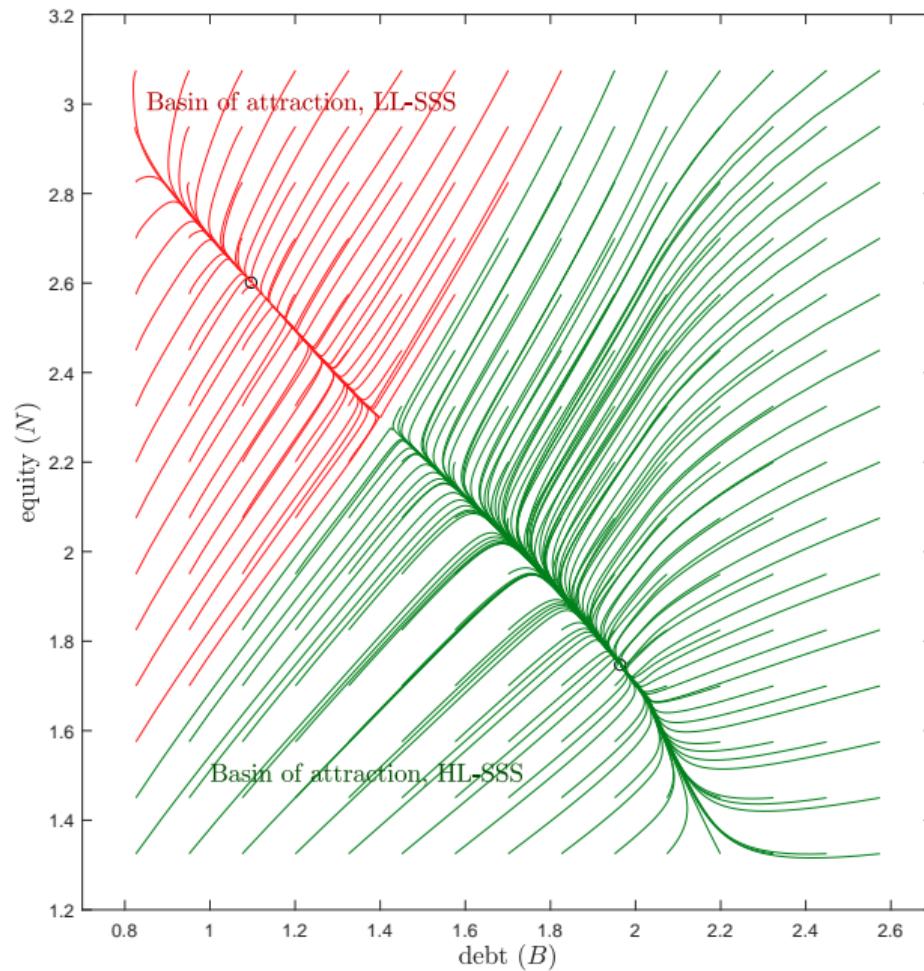


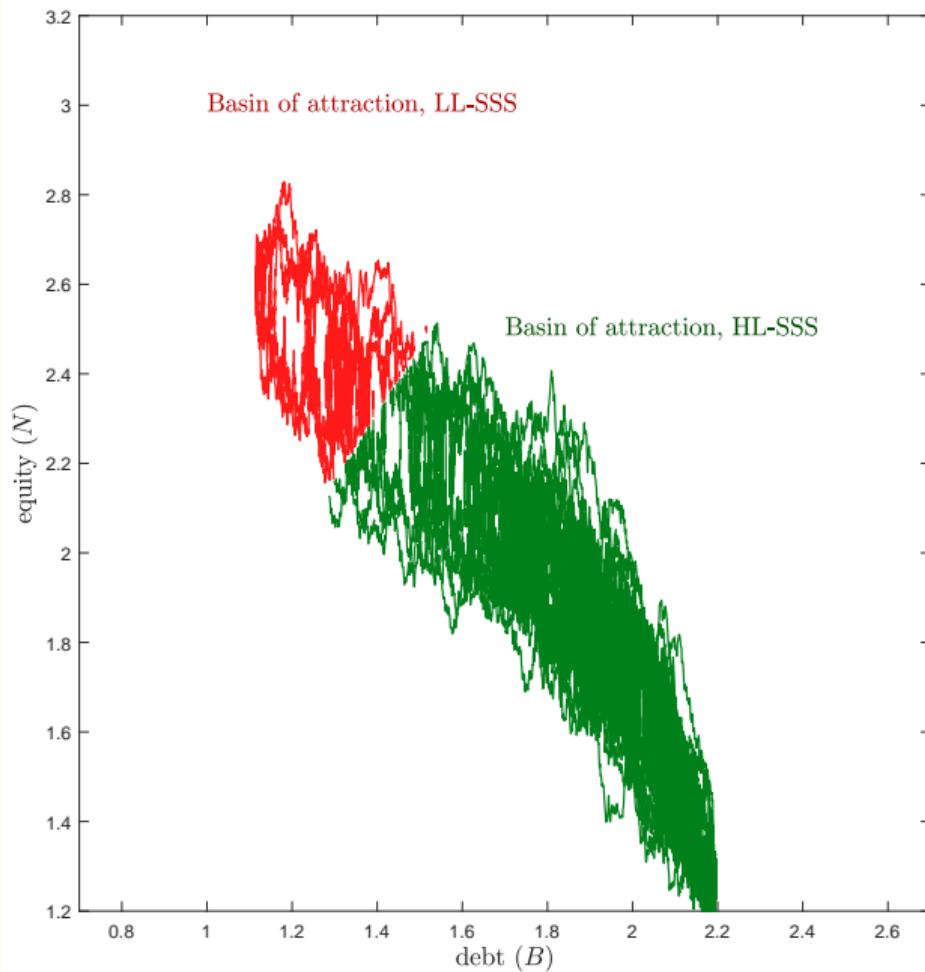


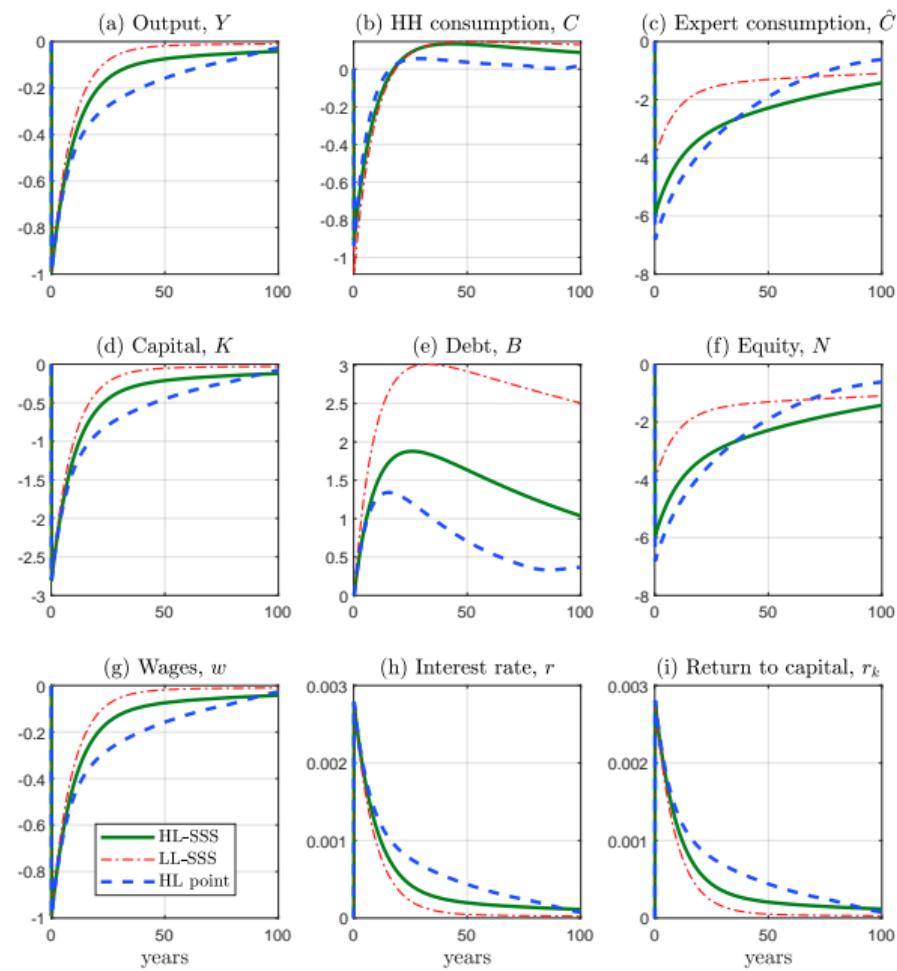






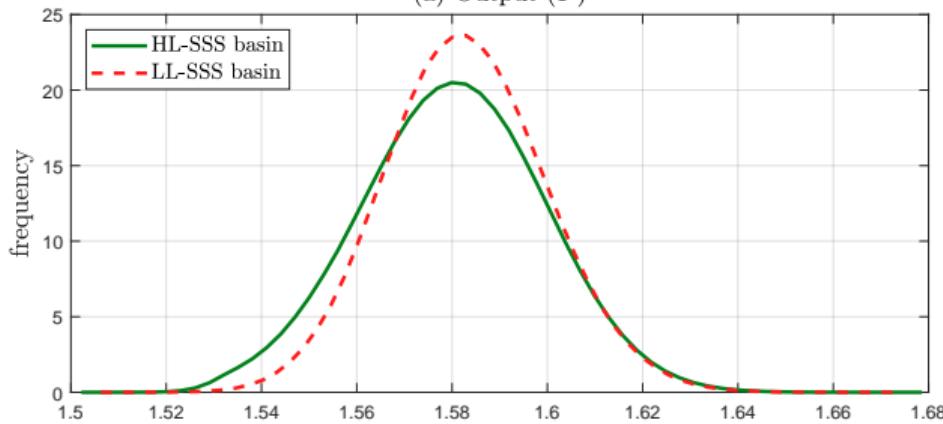
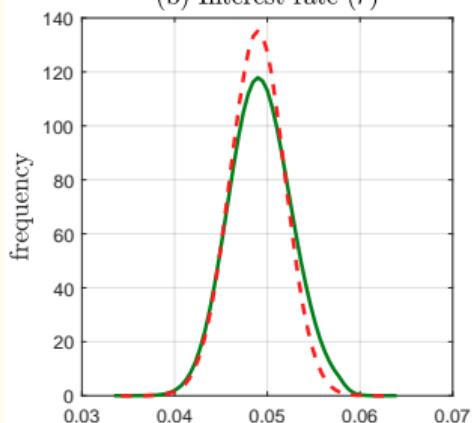
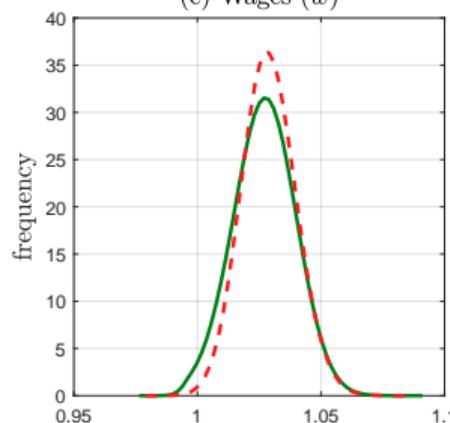




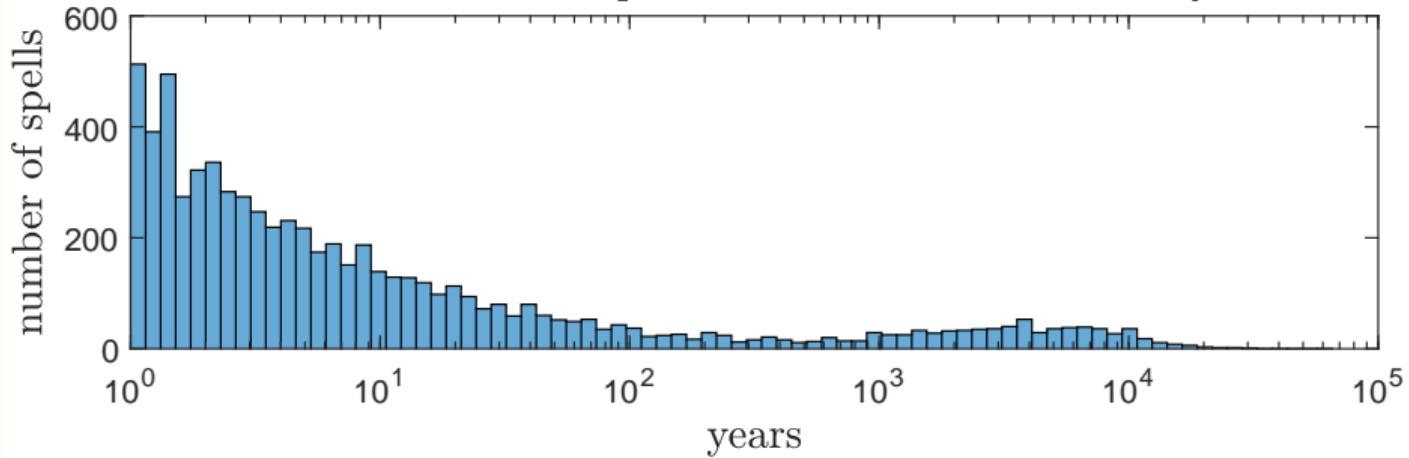


	Mean	Standard deviation	Skewness	Kurtosis
$\gamma^{\text{basin } HL}$	1.5802	0.0193	0.0014	2.869
$\gamma^{\text{basin } LL}$	1.5829	0.0169	0.1186	3.0302
$r^{\text{basin } HL}$	4.92	0.3364	0.0890	2.866
$r^{\text{basin } LL}$	4.89	0.2947	-0.0282	3.0056
$w^{\text{basin } HL}$	1.0271	0.0125	0.0014	2.8691
$w^{\text{basin } LL}$	1.0289	0.0111	0.1186	3.0302

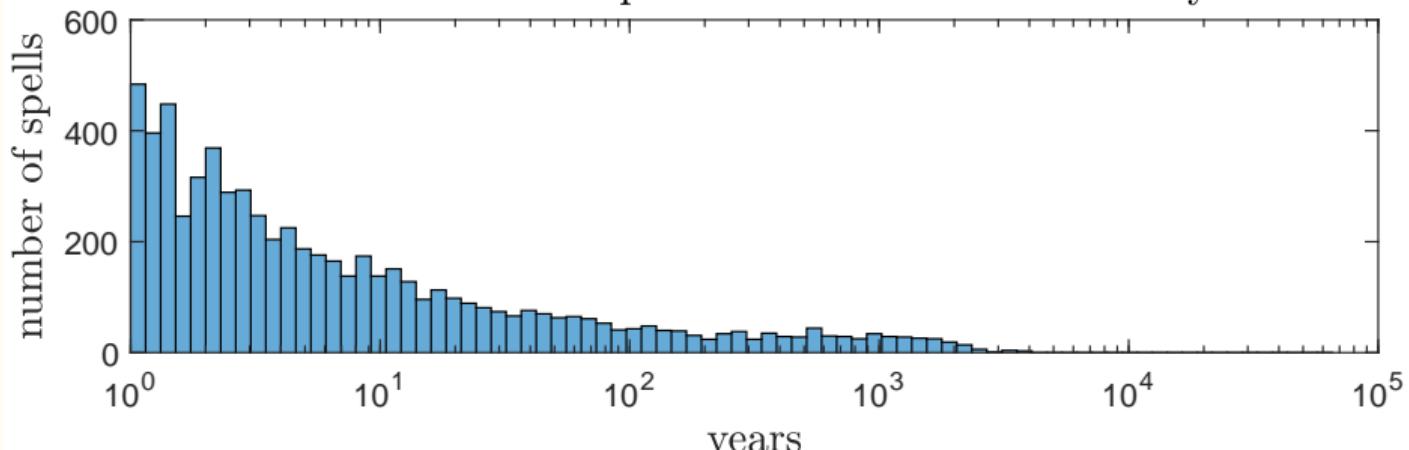
**Table 1:** Moments conditional on basin of attraction.

(a) Output ( $Y$ )(b) Interest rate ( $r$ )(c) Wages ( $w$ )

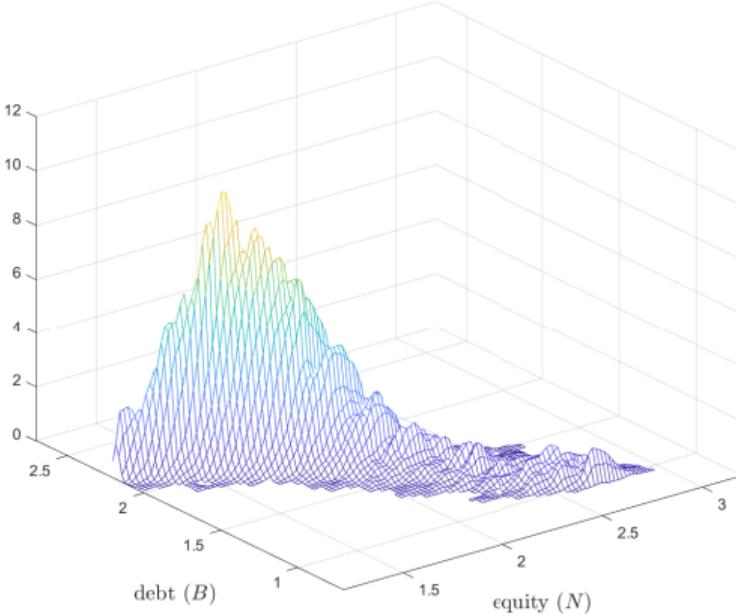
Median duration of spells at HL-SSS basin: 4.1667 years



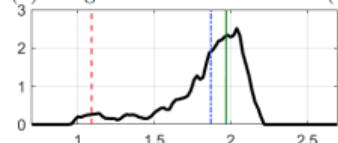
Median duration of spells at LL-SSS basin: 3.9167 years



(a) Ergodic density  $f(B, N)$

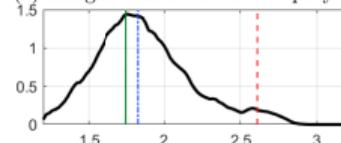


(b) Marginal distribution of debt ( $B$ )

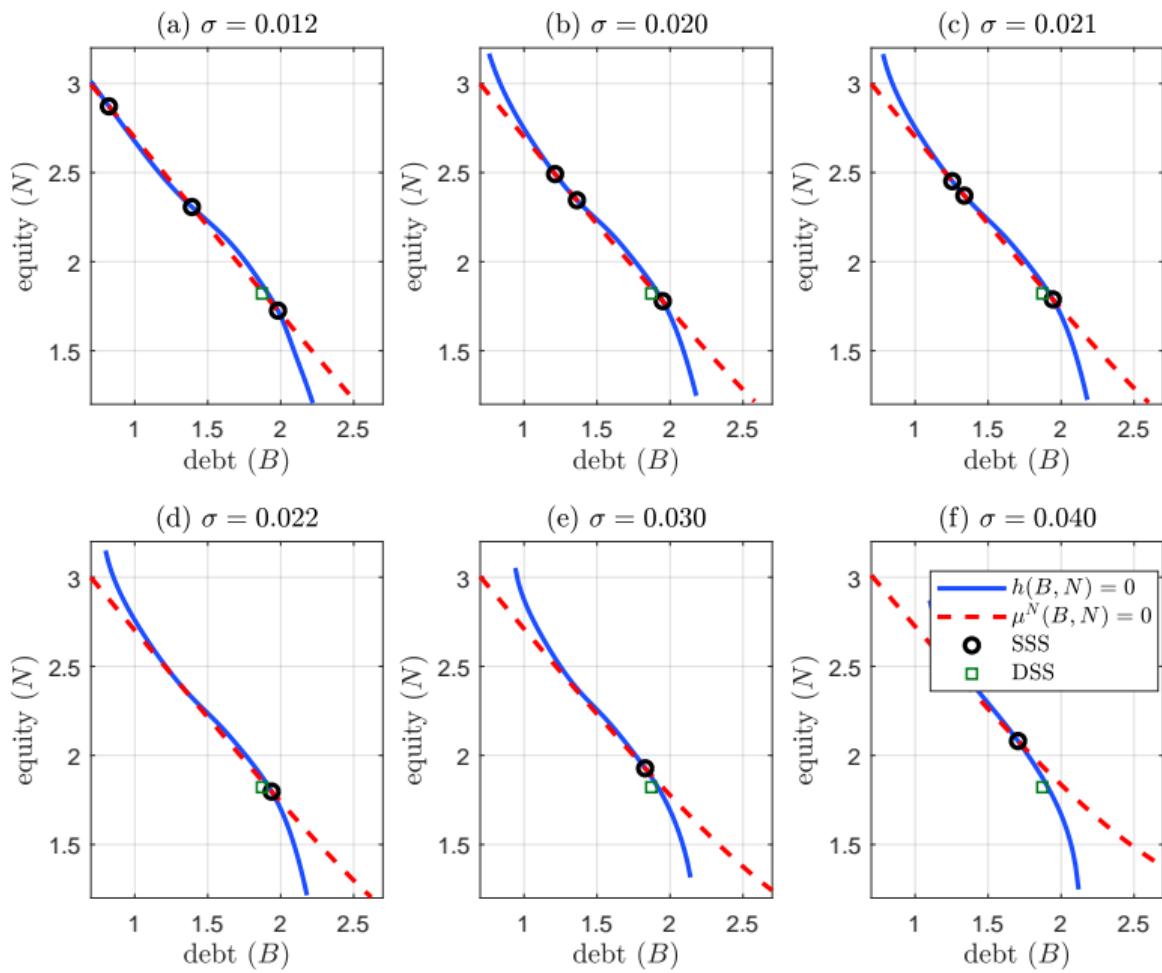


Marginal distribution  
— LL-SSS  
— HL-SSS  
— DSS

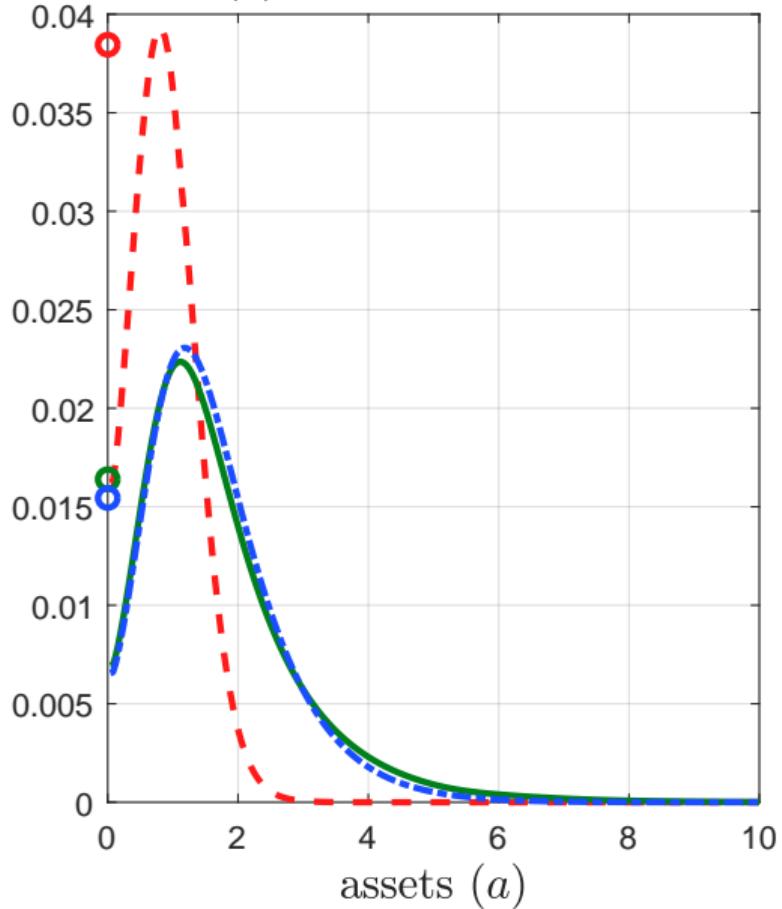
(c) Marginal distribution of equity ( $N$ )



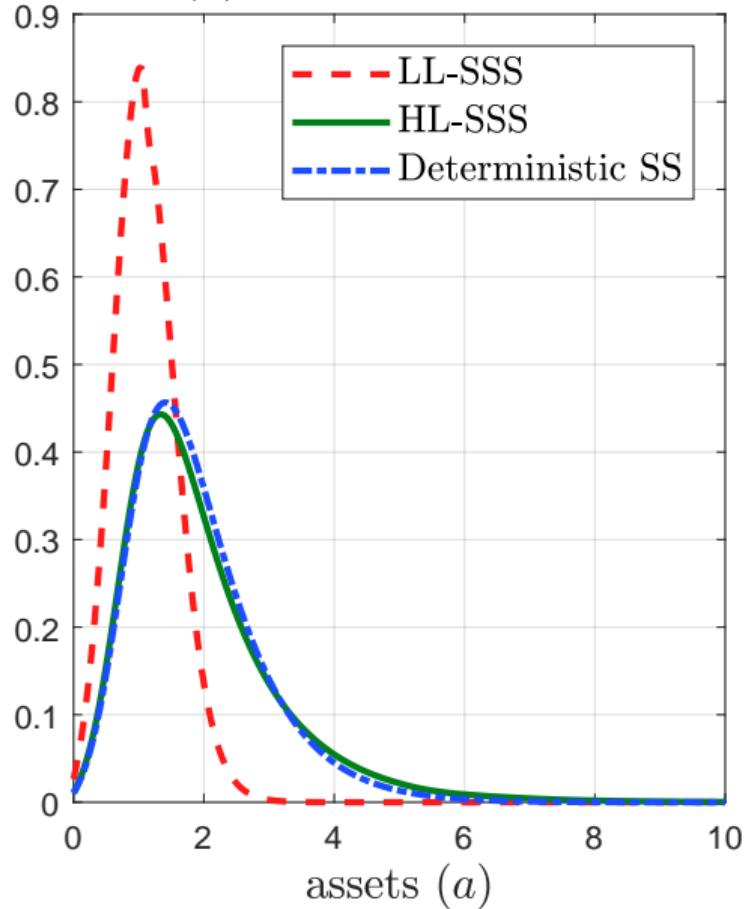
Marginal distribution  
— LL-SSS  
— HL-SSS  
— DSS

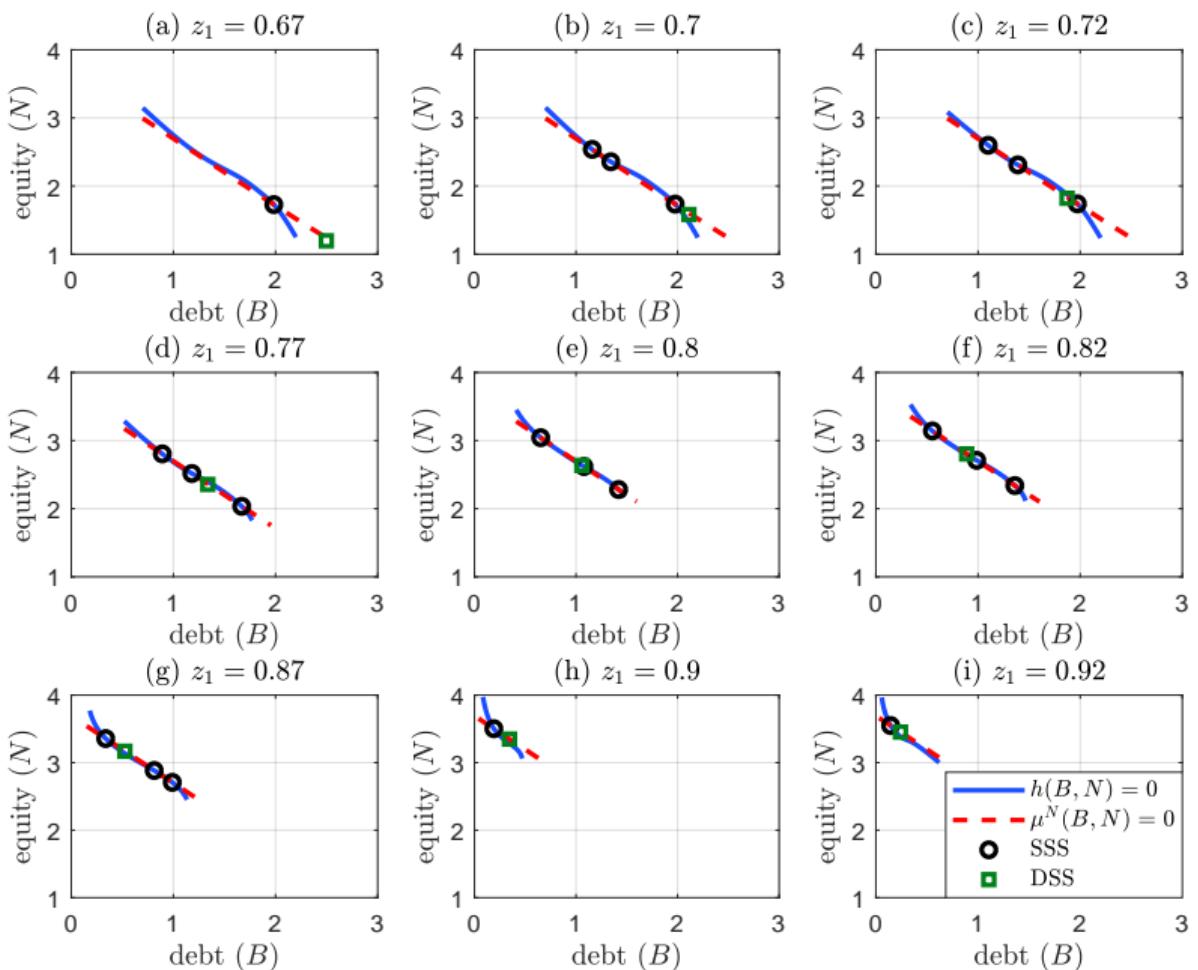


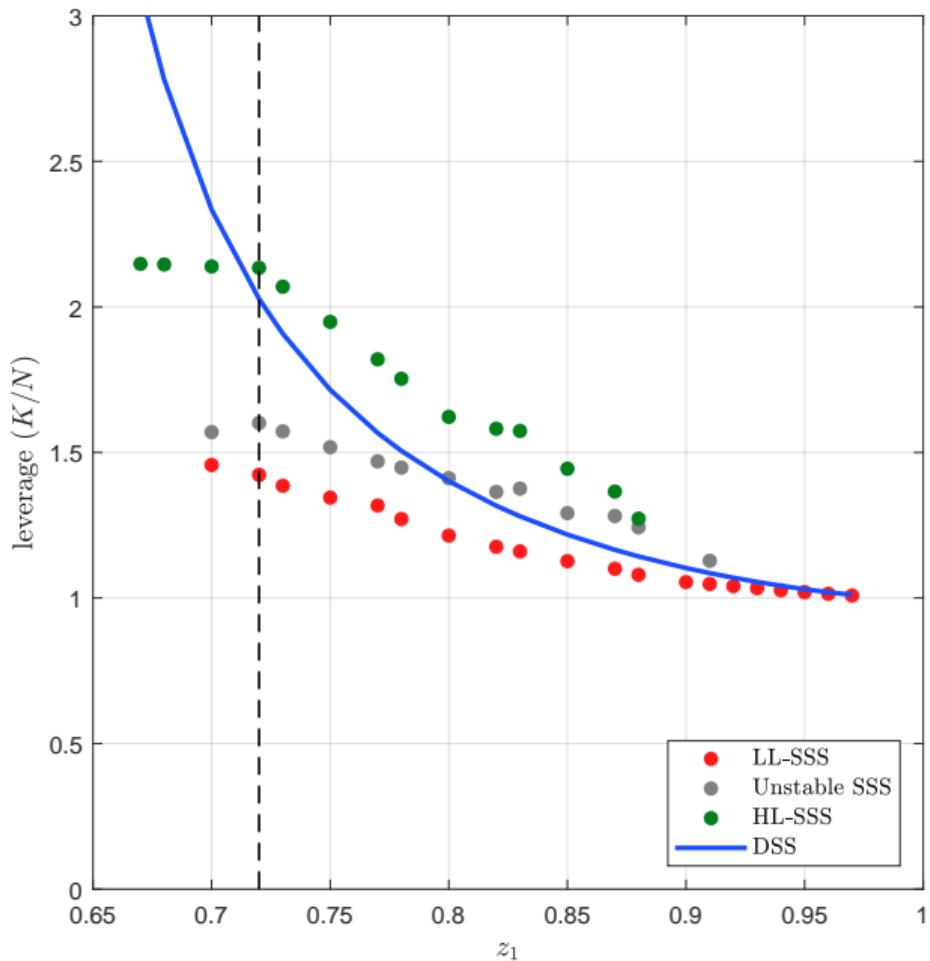
(a) Low- $z$  households

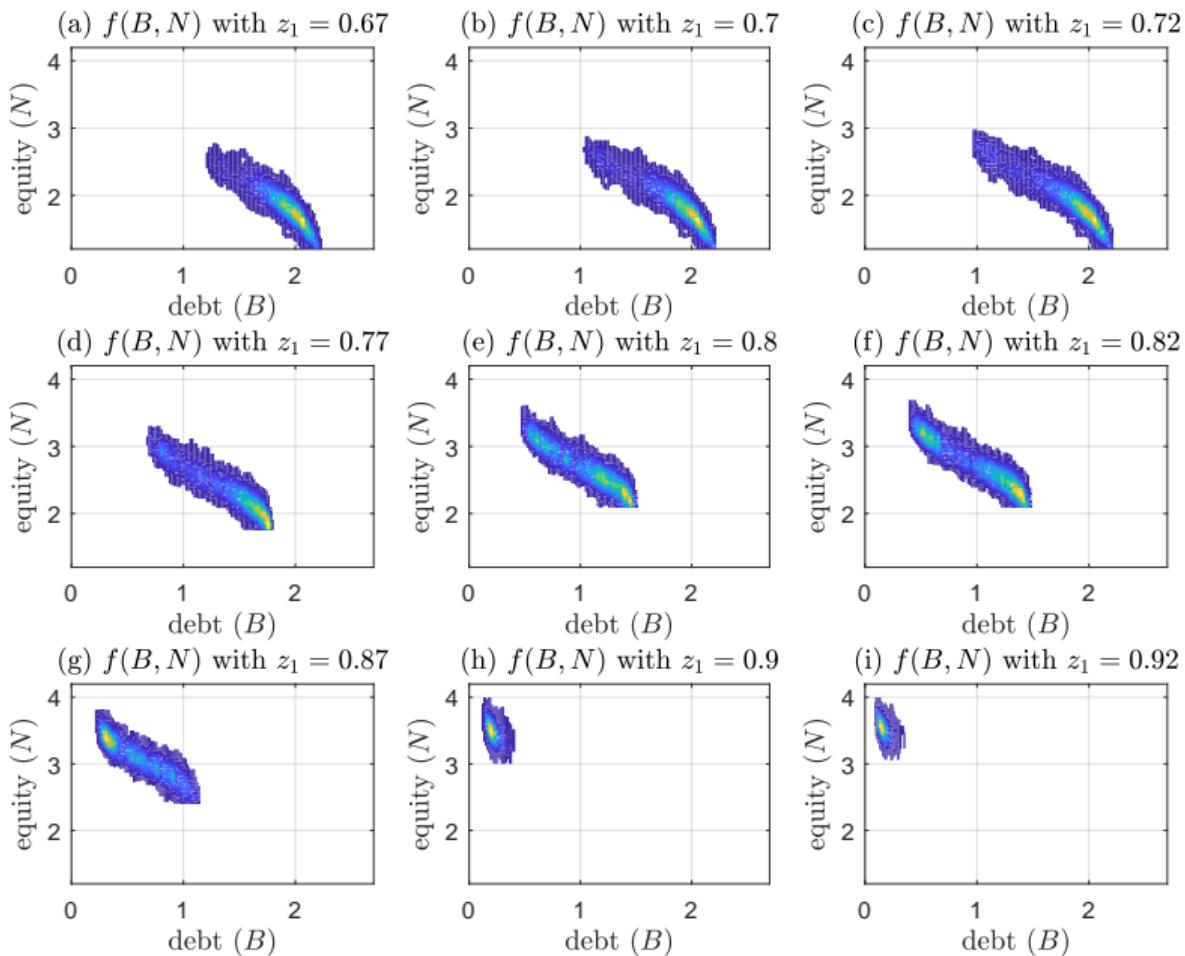


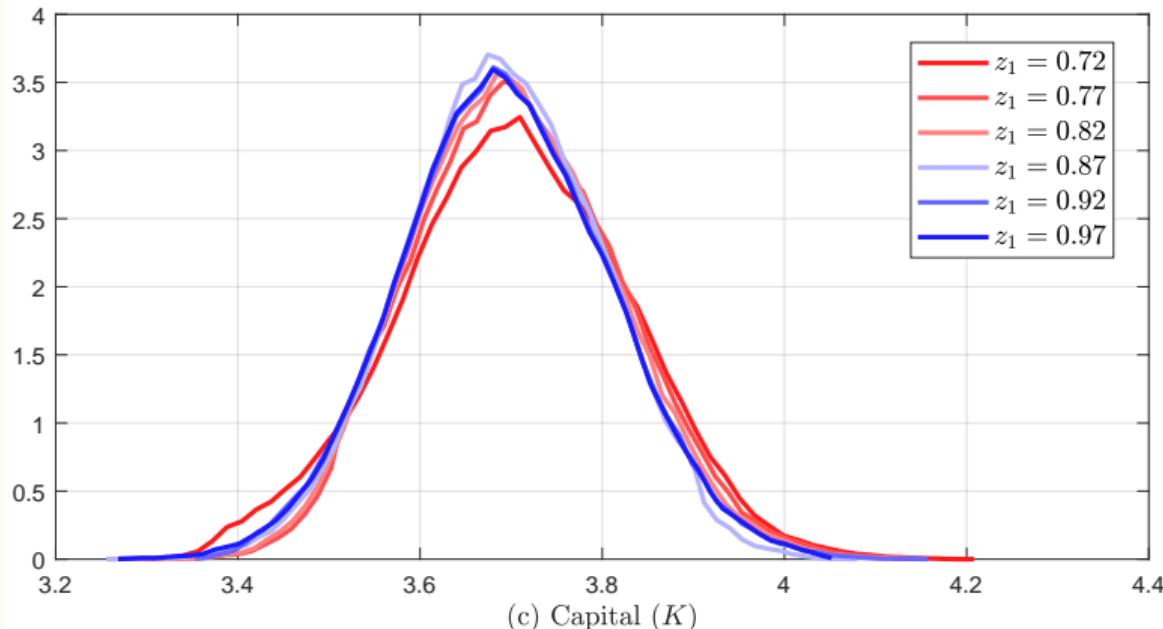
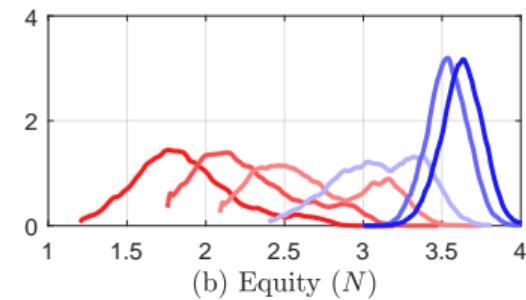
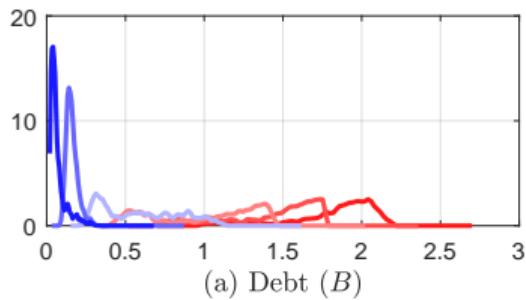
(b) High- $z$  households

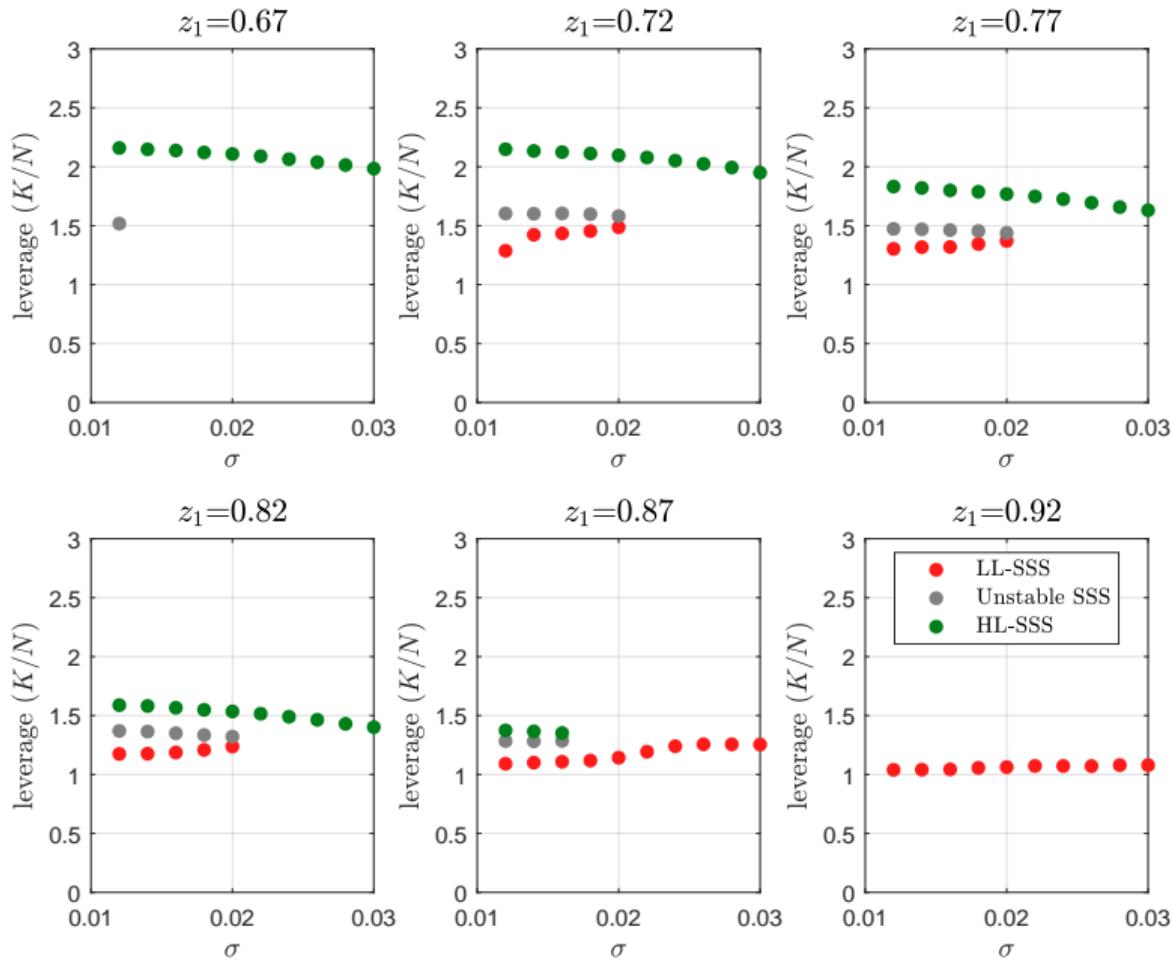


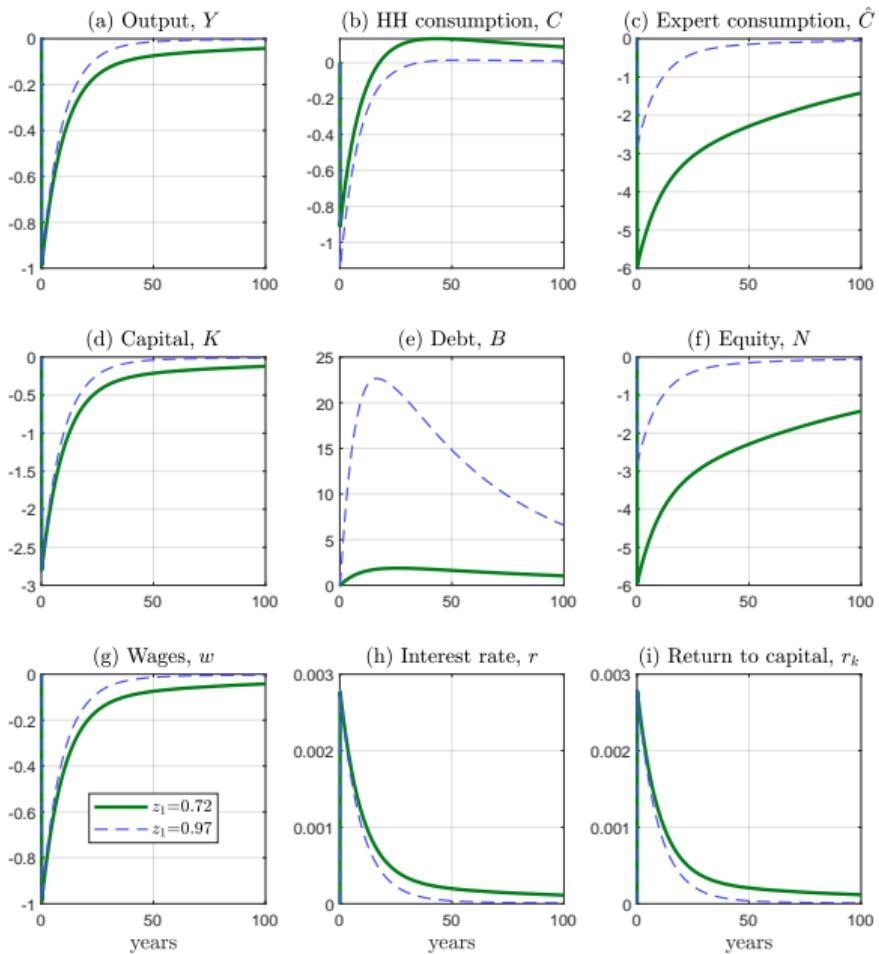




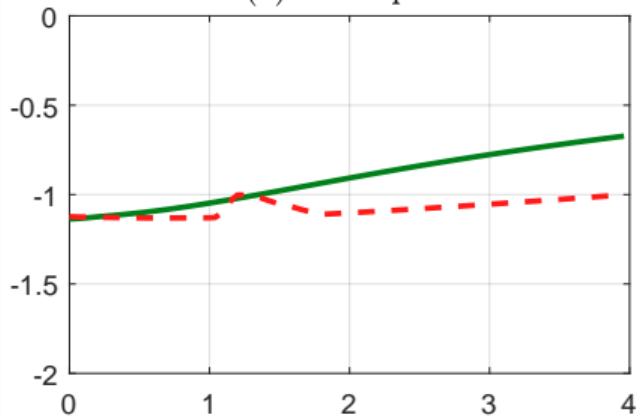




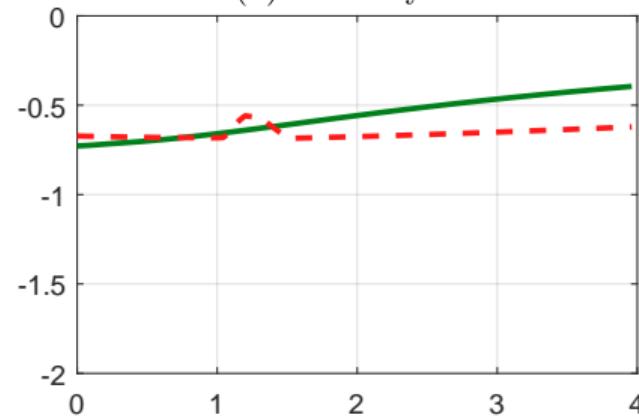




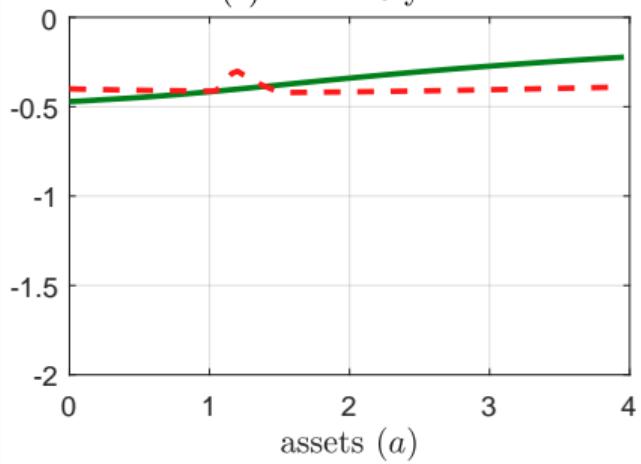
(a) On impact



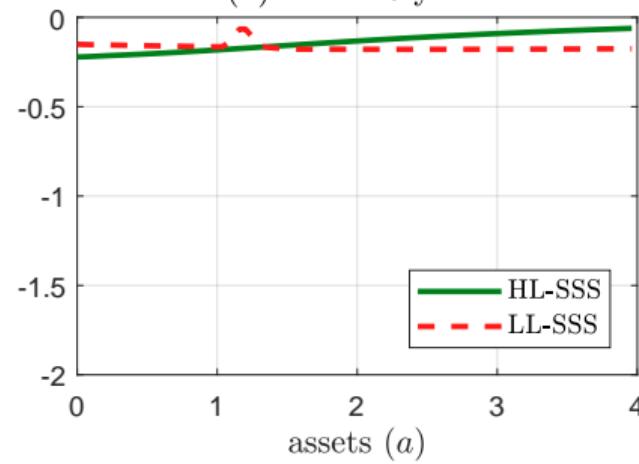
(b) After 5 years



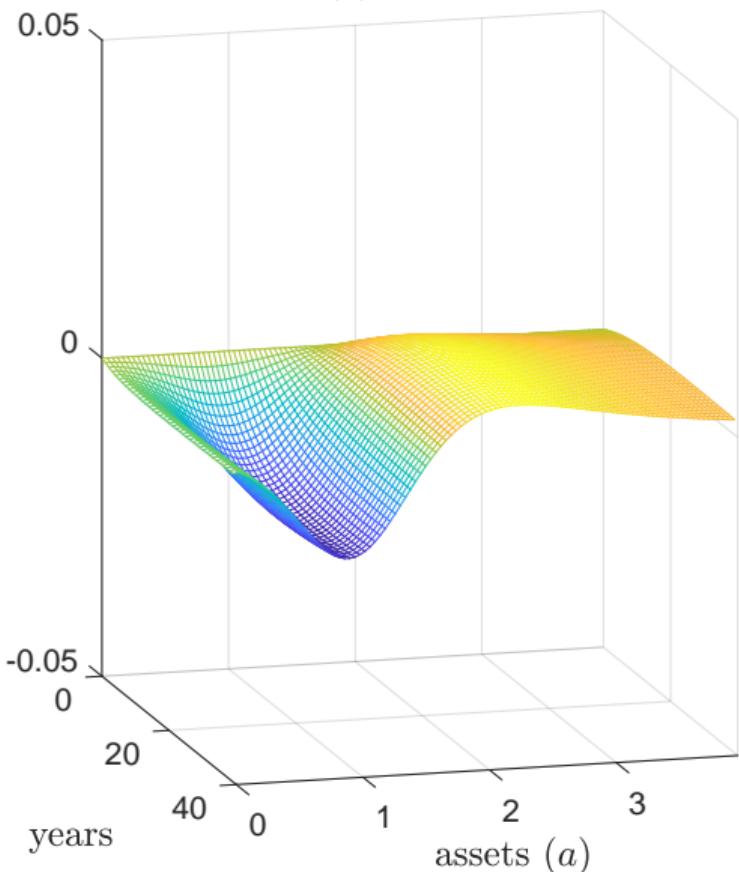
(c) After 10 years



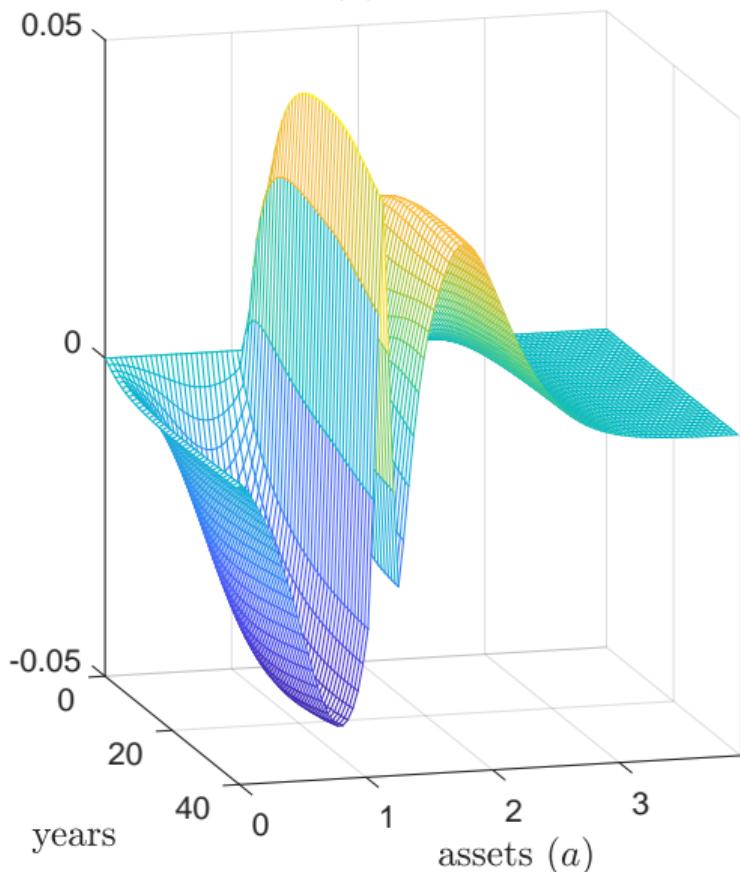
(d) After 20 years



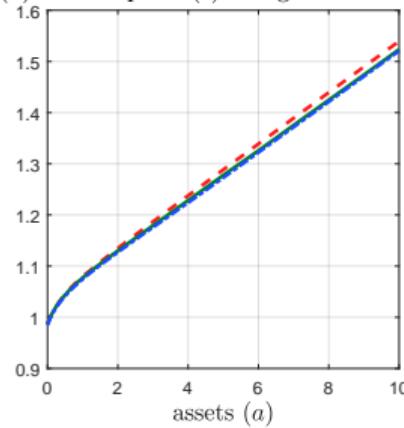
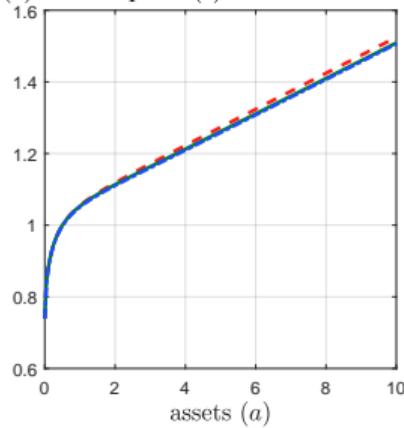
(a) HL-SSS



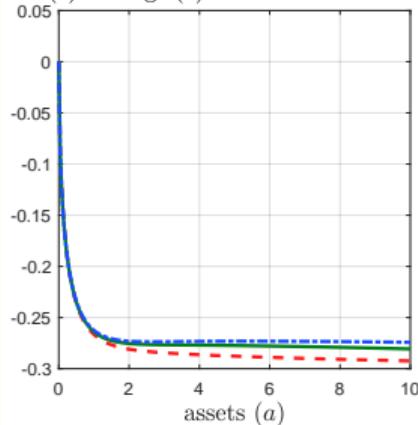
(b) LL-SSS



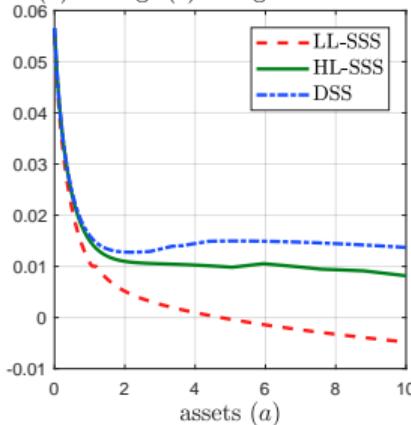
(a) Consumption ( $c$ ) of low- $z$  households (b) Consumption ( $c$ ) of high- $z$  households



(c) Savings ( $s$ ) of low- $z$  households



(d) Savings ( $s$ ) of high- $z$  households



## Recent advances

- **Maliar, Maliar, and Winant (2021)**: discrete time, finite number of agents, approximation of Euler/Bellman equations.
- **Azinovic, Gaegauf, and Scheidegger (2019)**: discrete time, continuum of agents (finite histogram), approximation of Euler equations.
- **Duarte (2018)**: continuous time, finite number of agents, reinforcement learning.
- **Fernández-Villaverde et al. (2022)**: continuous time, finite number of agents, approximation of HJBs.
- **Ebrahimi-Kahou et al. (2021)**: discrete time, finite number of agents, symmetry + concentration of measure.