PASC19 Zurich Switzerland 12–14 June 2019

Deep Equilibrium Nets

Marlon Azinovic[1,2] <marlon.azinovic@bf.uzh.ch> Luca Gaegauf[1] <luca.gaegauf@bf.uzh.ch> Simon Scheidegger[3] <simon.scheidegger@unil.ch>

1 University of Zurich 2 Swiss Finance Institute 3 University of Lausanne

Universität Zürich UZH — UNIL | Université de Lausanne

Summary

Goal:
Developing economic models, which are consistent with findings at the micro-economic level. Therefore, we want to be able to study models, which include heterogenous agents, uncertainty and financial frictions.

Problem:
Equilibria in such models are hard to compute! Because they simultaneously feature:

1. stochasticity,
2. high-dimensional state spaces,
3. strong non-linearities in the equilibrium functions,
4. irregular goemetries of the ergodic set of states.

Our solution:
Approximate a **recursive equilibrium**: $\{f_i\}_{i=1}^{N_{\text{out}}}$:

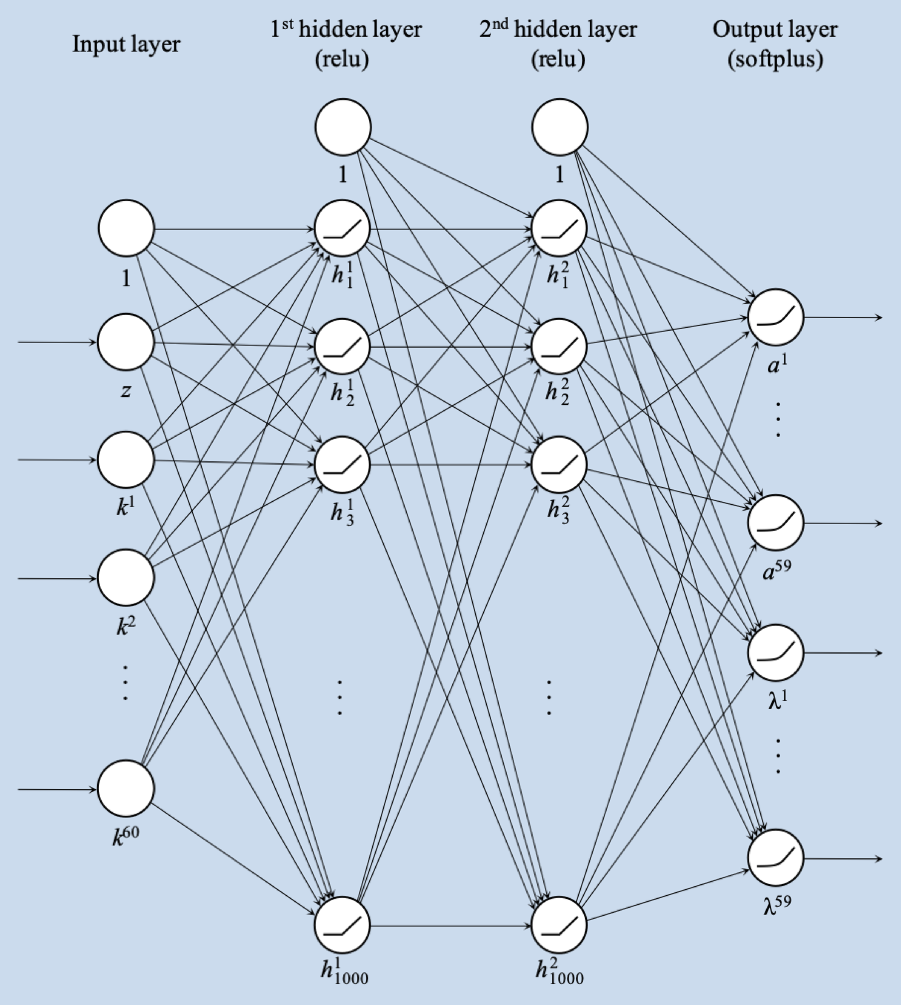
$$f_i : \underbrace{\mathbf{x}}_{\text{state}} \rightarrow \underbrace{f_i(\mathbf{x})}_{\text{endogenous variables}}$$

by a **deep equilibrium net**: $\mathcal{N}_\rho$, where

$$\mathcal{N}_\rho : \underbrace{\mathbf{x}}_{\text{state}} \rightarrow \underbrace{\mathcal{N}_\rho(\mathbf{x})}_{\text{approx. endogenous variables}} \approx \begin{bmatrix} f_1(\mathbf{x}) \\ \dots \\ f_{N_{\text{out}}}(\mathbf{x}) \end{bmatrix}.$$

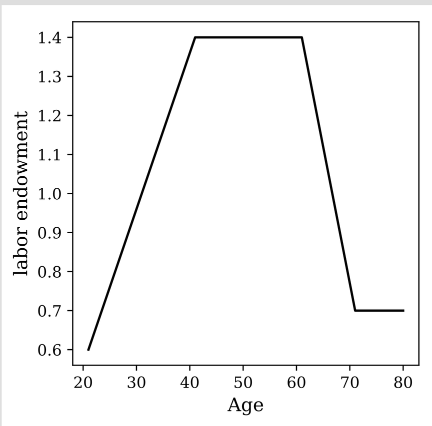
Key ideas:

1. use the errors in the equilibrium conditions as loss function.
2. learn the equilibrium functions with stochastic gradient descent.
3. take the data points from a simulated path.



Algorithm for a benchmark problem

Benchmark problem: OLG economy, with 60 age-groups, capital, borrowing constraints and stochastic production.



Recursive equilibrium: a functional rational expectations equilibrium is given by:

exogenous shocks (z) to depreciation (δ) and total factor productivity (ξ) — number of agents — agents die without savings — agents are "born" without capital

$$\mathbf{f} : \underbrace{\{0,1,2,3\}} \times \mathbb{R}^{60} \rightarrow \mathbb{R}^{59\cdot 2} \ : \ \mathbf{f}\left(\begin{bmatrix} z_t \\ \mathbf{k}_t \end{bmatrix}\right) = \mathbf{f}\left(\begin{bmatrix} z_t \\ 0_t^1 \\ k_t^2 \\ \dots \\ k_t^{59} \\ k_t^{60} \end{bmatrix}\right) = \begin{bmatrix} a_t^1 \\ \dots \\ a_t^{59} \\ \lambda_t^1 \\ \dots \\ \lambda_t^{59} \end{bmatrix}$$

0: δ=0.5, ξ=0.85
1: δ=0.5, ξ=1.15
2: δ=0.9, ξ=0.85
3: δ=0.9, ξ=1.15

such that: $\forall h = 1, \dots, 59:$

$$0 = \beta \mathbb{E}_t \left[\frac{R_{t+1} u'(c_{t+1}^{h+1}) + \lambda_t^h}{u'(c_t^h)}\right] - 1$$

$$0 = \lambda_t^h (a_t^h - \underline{a})$$

$$0 \leq \lambda_t^h$$

$$\underline{a} \leq a_t^h$$

marginal utility — consumption — borrowing constraint — KKT multiplier

where $a_t^h = k_{t+1}^{h+1}$, and

$$c_t^h = k_t^h R_t + l_t^h w_t - a_t^h$$

$$R_t = \xi_t \alpha K_t^{\alpha - 1} L_t^{1-\alpha} + (1 - \delta_t)$$

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

$$K_t = \sum_{h=1}^{60} k_t^h, \ L_t = \sum_{h=1}^{60} l_t^h$$

capital — labor — savings — capital return — wage — aggregate capital and labor

Loss function: our loss function is defined as the mean-squared error of the equilibrium conditions:

network parameters

$$\ell_{\mathcal{D}_{\text{train}}}(\boldsymbol{\rho}) := \frac{1}{|\mathcal{D}_{\text{train}}|} \frac{1}{N-1} \sum_{\mathbf{x}_j \in \mathcal{D}_{\text{train}}} \sum_{i=1}^{N-1} \left(\left(e_{\text{REE}}^i(\mathbf{x}_j)\right)^2 + \left(e_{\text{KKT}}^i(\mathbf{x}_j)\right)^2 \right)$$

number of agents — relative Euler equation error (i.e., consumption error) — KKT slackness errors

where

$$e_{\text{REE}}^i(\mathbf{x}_j) := \frac{u'^{-1}\left(\beta \mathbb{E}_{z_j}\left[r(\hat{\mathbf{x}}_{j,+}) u'(\hat{c}^{i+1}(\hat{\mathbf{x}}_{j,+}))\right] + \hat{\lambda}^i(\mathbf{x}_j)\right)}{\hat{c}^i(\mathbf{x}_j)} - 1$$

$$e_{\text{KKT}}^i(\mathbf{x}_j) := (\hat{a}^i(\mathbf{x}_j) - \underline{a}) \hat{\lambda}^i(\mathbf{x}_j).$$

Next period's state generated by neural network $\begin{bmatrix} z_+ \\ 0 \\ \hat{a}^{[1:N-1]}(x_j) \end{bmatrix}$

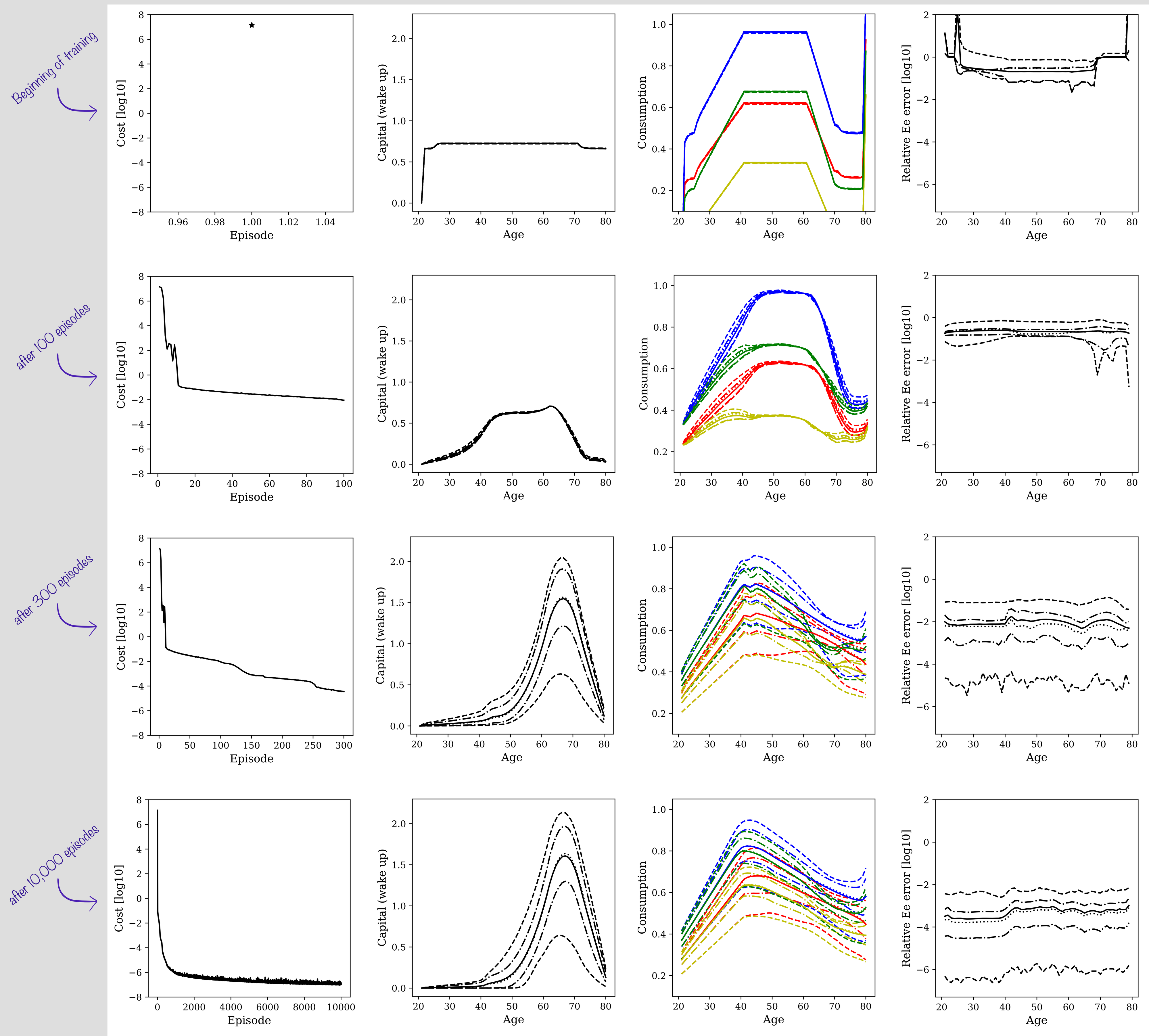
Estimated savings — Estimated KKT multiplier

Sampling the most relevant states: we generate new training sets $\mathcal{D}_{\text{train}}$ by simulating the economy. This can be done at virtually zero computational cost.

Finding good parameters: we update the parameters of the neural network using mini-batch gradient descent on newly simulated states.

Learning process

Evolution of the cost (1st column), capital holding (2nd clolumn), consumption (third column), and relative error in the Euler equations (fourth column) after 1 (first row), 100 (second row), 300 (third row), and 10,000 (fourth row) episodes of training.



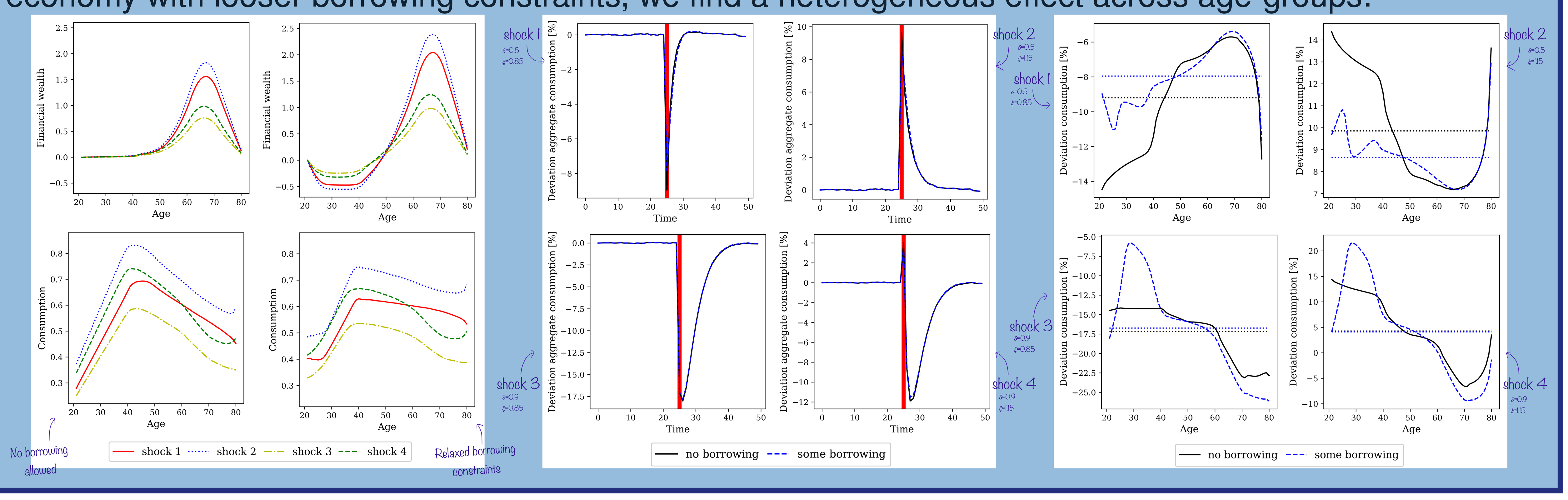
Final errors: Final errors in the equilibrium conditions on a path of 10,000 simulated states:

	mean	max	0.1	10	50	90	99.9
Rel Ee [%]	0.06	1.31	0.00	0.01	0.04	0.13	0.54
KKT [$\times 10^{-2}$]	0.01	0.25	0.00	0.00	0.00	0.02	0.15

Run-time: 2.1 seconds per episode, running on a single Cray XC50 compute node of the "Piz Daint" system that is installed at the Swiss National Supercomputer Centre (CSCS).

Economic results

Effect of looser borrowing constraints: comparing the benchmark economy without borrowing to an economy with looser borrowing constraints, we find a heterogeneous effect across age-groups:



Acknowledgements:
This work was generously supported by grants from the Swiss National Supercomputing Centre (CSCS) under project ID s885 and the Swiss Platform for Advanced Scientific Computing (PASC) under project ID "Computing equilibria in heterogeneous agent macro models on contemporary HPC platforms".

We thank Felix Kübler for his valuable comments support. We acknowledge helpful suggestions by Karl Schmedders and participants at the *Advances in Computational Economics and Finance* Seminar and the *Brown bag lunch seminar* at the University of Zurich.

Simon Scheidegger gratefully acknowledges support from the Cowles Foundation at Yale University.

Full paper available on SSRN: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3393482`

Presented at PASC 2019 in Zurich, Switzerland on Thursday 13th June 2019