Global Solutions of Heterogeneous Agent Models with Non-linear Aggregate Dynamics[*]

Jeppe Druedahl[†]
Emil Holst Partsch[‡]

Abstract

In this paper, we present and evaluate global solution methods for heterogeneous agent models with non-linear aggregate dynamics. We consider models with weak approximate aggregation, where the aggregate dynamics can be summarized with a small number of states. We allow the perceived law-of-motion the agents use for now- and forecasting to be non-linear, and do not impose any parametric restrictions on it. Specifically, we derive the perceived law-of-motion using either a *neural net* or *radial basic function interpolation*. Both methods deliver precise global solutions, but radial basis function interpolation is faster because of a slow training step for the neural net, and more stable in terms of ensuring convergence. We are able to globally solve our benchmark model with an aggregate non-linearity and period-by-period market clearing in less than 15 minutes on a desktop computer.

[*] We are grateful for helpful comments from Patrick Moran, Søren Hove Ravn and Nicolai Waldstrøm. Center for Economic Behavior and Inequality (CEBI) is a center of excellence at the University of Copenhagen, founded in September 2017, financed by a grant from the Danish National Research Foundation, Grant DNRF134. All errors are our own.

[†] University of Copenhagen and CEBI. jeppe.druedahl@econ.ku.dk.

[‡] University of Copenhagen and Danmarks Nationalbank. ehp@econ.ku.dk.

1 Introduction

Recently, a burgeoning literature incorporating explicit heterogeneity into business cycle model has come to the forefront of modern macroeconomics. This literature shows that introducing such heterogeneity may radically alter transmission mechanisms and dynamics relative to comparable representative agent economies. The standard approach is to use *local* solution methods. These are generally fast, but can miss central model dynamics due to exogenous non-linearities or strong internal propagation and feedback loops. *Global* solution methods are typically more complicated and slower, but allow us to study a broader range of economic questions. For example, a global solution is necessary to study both the response of precautionary saving and portfolio choice to aggregate risk, and the state dependence of impulse responses form shocks and economic policies.

The first methods for solving heterogeneous agent models globally were presented in Den Haan (1996, 1997) and Krusell and Smith (1997, 1998). The resulting literature was later evaluated in a special issue of *Journal of Economics Dynamics and Control* (Den Haan, 2010b)[1] and surveyed in Algan et al. (2014).[2] A central insight from this line of work was that the dynamics of the macroeconomic aggregates could be modeled in terms of only the mean of the wealth distribution. This is generally referred to as »approximate aggregation«. In this paper, we rely on a *weaker* version of approximate aggregation, where we only assume that the aggregate dynamics can be modeled in terms of a small number of aggregate states. We also deviate from the standard assumption of linear or log-linear aggregate dynamics, allowing the perceived law-of-motion agents use for forecasting to be non-linear. This is important because a full global solution is not needed as soon as aggregate dynamics are approximately linear. If aggregate dynamics are instead linear, one can instead rely on early contributions in Reiter (2002, 2009, 2010) and efficient first order solution methods developed in state-space form (Bayer and Luetticke, 2020) and in sequence-space form (Auclert et al., 2021).[3]

Our proposed solution method is a variant of the standard Krusell-Smith algorithm.

[1] For this paper, the two step approach used in Maliar et al. (2010), and the non-stochastic simulation method developed in Young (2010) (see also Tan (2020)), are in particular relevant.

[2] See Terry (2017) for a survey on methods for solving heterogeneous firm models.

[3] See also Ahn et al. (2018); Boppart et al. (2018); Winberry (2018); Bilal (2021).

The major difference is that we allow the perceived laws-of-motion (PLMs) the agents use for now- and forecasting to be non-linear, and do not impose any parametric restrictions on them. Specifically, we derive the perceived law-of-motion using either a neural net or radial basic function interpolation. We formulate the perceived-laws-of-motion such that it should hold exactly in the limit of no approximation errors.[4] This allows us to reap the full gains of the local flexibility of our function approximation methods. Additionally, we formulate the household problem such that period-by-period market clearing is a pure interpolation problem on already derived policy functions. This provides additional speed-up. We apply our method to a Heterogeneous Agents Neo-Classical (HANC) model with adjustable technology utilization subject to an upper bound and capital adjustment costs.

Our paper is complementary to Fernández-Villaverde et al. (2021), who use the method from Fernández-Villaverde et al. (2020) to solve a Heterogeneous Agent New Keynesian (HANK) model with an occasionally binding zero lower bound using a neural net for the perceived laws-of-motion. In line with their results, assuming linear laws of motion results in higher than acceptable forecast errors. Instead, we propose using radial basis function interpolation for the PLMs to solve the model globally, which is stable, accurate, and fast. Specifically, using our benchmark model, we can find the full global solution in less than 15 minutes on a desktop computer using radial basic function interpolation. This is slightly slower than linear PLMs, but orders of magnitude more precise. Instead, using a neural net to generate PLMs is slower and less precise than using radial basis function interpolation, and in some cases, we face a lack of convergence.

The proposed solution method is easy to implement and Python code for all results in the paper is provided at github.com/JeppeDruedahl/GlobalHA.[5]

Finally, our paper is related to the recent literature on using artificial intelligence algorithms for solving general equilibrium models with heterogeneous agents (Maliar and Maliar, 2020; Gorodnichenko et al., 2021; Hill et al., 2021; Maliar et al., 2021;

[4] This differs from the implementation of the Krusell-Smith method for a Heterogeneous Agent New Keynesian (HANK) model in Bayer et al. (2019), where one of the PLMs is for a variable with an expectation term. Errors should then only be zero *on average*.

[5] We build on top of the Python package GEModelTools, which we have developed for computing local solutions to heterogeneous agent models using the sequence-space method from Auclert et al. (2021).

Valaitis and Villa, 2021; Azinovic et al., 2022; Curry et al., 2022; Han et al., 2022). These methods are designed to also alleviate the cost of solving each agent's dynamic programming problem. This is particularly relevant when there are many aggregate states, as even our weak version of approximate aggregation fails in that case. Our paper focuses on non-linear but lower dimensional problems and can therefore rely on much simpler algorithms.

The paper is structured as follows. Section 2 presents the overall solution method. Section 3 presents our benchmark model. The main results are presented in Section 4 and additional results for a model without capital adjustment costs are presented in Section 5. Section 6 concludes.

2 Solution Method

Setup. To present our solution method in quasi-general terms, before turning to our benchmark model in the following section, we consider a heterogeneous agents model with a single endogenous state and a single continuous choice. The approximate household problem then is

$$\begin{aligned}
\tilde{v}(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, \boldsymbol{z}_t, a_{t-1}) &= \max_{a_t, c_t} u(c_t) + \beta \mathbb{E}_t \left[\tilde{v}(\boldsymbol{Z}_{t+1}, \boldsymbol{S}_t, \boldsymbol{z}_{t+1}, a_t) \right] \qquad (1) \\
&\text{s.t.} \\
\boldsymbol{S}_t, \boldsymbol{P}_t &= \text{PLM}(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}) \\
a_t + c_t &= m(\boldsymbol{z}_t, a_{t-1}, \boldsymbol{P}_t) \\
\boldsymbol{z}_{t+1} &\sim \Gamma_z(\boldsymbol{z}_t) \\
\boldsymbol{Z}_{t+1} &\sim \Gamma_Z(\boldsymbol{Z}_t) \\
a_t &\geq -b,
\end{aligned}$$

where

1. $\boldsymbol{Z}_t$ are exogenous aggregate shocks.
2. $\boldsymbol{S}_{t-1}$ are pre-determined (finite dimensional) aggregate states.
3. $\boldsymbol{P}_t$ are »prices«.
4. PLM($\bullet$) is the *Perceived-Law-of-Motion*.

5. $\boldsymbol{z}_t$ is stochastic and exogenous idiosyncratic states.

6. c_t is consumption providing utility $u(c_t)$ discounted by β.

7. a_t is end-of-period assets (borrowing constraint given by b).

8. $m(\bullet)$ is cash-on-hand with $\frac{\partial m(\bullet)}{\partial a_{t-1}} > 0$.

This problem is approximate because the distribution of households, $\boldsymbol{D}_t$, over the idiosyncratic states $\boldsymbol{z}_t$ and a_{t-1}, are *not* included as a state. In the true model, the distribution matters for determining both current and future aggregate states and prices. In the approximation, this is instead captured by a finite number of aggregate states, $\boldsymbol{S}_t$. These can, however, be functions of the distribution.

The PLM must be specified to let the household update the aggregate states, $\boldsymbol{S}_t$, and infer all the prices, $\boldsymbol{P}_t$, in their budget constraint. The PLM must be consistent with all aggregate model relationship, where expectation terms can be evaluated with e.g. a quadrature method and using that $\boldsymbol{Z}_{t+1} \sim \Gamma_Z(\boldsymbol{Z}_t)$ is known. In general, the PLM is therefore allowed to take the form

$$\text{PLM}(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}) = \mathbb{E}\left[f(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, \boldsymbol{Z}_{t+1}; \Psi) \,|\, \boldsymbol{Z}_t, \boldsymbol{S}_{t-1}\right], \tag{2}$$

where Ψ are parameters to be determined. In the benchmark model, we specify the PLM without an expectation term, but when searching for market clearing prices we use that an expectation term can be evaluated.

Importantly, the law-of-motion the PLM is approximating is fully *deterministic*, and there is therefore no bias-variance trade-off when estimating it from data on states, $\boldsymbol{S}_t$, shocks, $\boldsymbol{Z}_t$, and prices $\boldsymbol{P}_t$. In other word, there is no fundamental over-fitting risk in letting $f(\bullet)$ have infinitely many degrees of freedom, and it can therefore be modeled with a neural net or radial basis function interpolation.[6]

An *equilibrium condition* is that the PLM is (approximately) self-consistent in terms of the implied dynamics. I.e. as the household observes simulated data, it should not be able to improve forecast errors by updating Ψ.

[6] In the presence of approximation errors some over-fitting is possible. A stochastic simulation might result in differences in outputs, $\boldsymbol{P}_t$ and $\boldsymbol{S}_t$, for (almost) the same inputs, $\boldsymbol{Z}_t$ and $\boldsymbol{S}_{t-1}$. In this case it might therefore be valuable to introduce some smoothing, by restricting $f(\bullet)$ in some way to even this out.

Household solution and simulation. The approximate household problem can be solved with an endogenous-grid-method as

$$
\begin{aligned}
q(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, a_t) &= \mathbb{E}\left[v_a(\boldsymbol{Z}_{t+1}, \boldsymbol{S}_t, z_{t+1}, a_t)\right] \\
\tilde{c}(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, a_t) &= (\beta q(\bullet))^{-\frac{1}{\sigma}} \\
\tilde{m}(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, a_t) &= a_t + c(\bullet) \\
c^\star(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, a_{t-1}) &= \text{interp } \tilde{m}(\bullet) \to \tilde{c}(\bullet) \text{ at } m(z_t, a_{t-1}, \boldsymbol{P}_t) \\
a^\star(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, a_{t-1}) &= m(\bullet) - c^\star(\bullet) \\
v_a(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, a_{t-1}) &= \frac{\partial m(\bullet)}{\partial a_{t-1}} c^\star(\bullet)^{-\sigma}.
\end{aligned} \tag{3}
$$

Next, define the associated cash-on-hand savings function by

$$
\begin{aligned}
a^\star(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, m_t) &= a^\star(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, a_{t-1}) \\
a_{t-1} &= m^{-1,a}(m_t, z_t, \boldsymbol{P}_t).
\end{aligned} \tag{4}
$$

The distribution of households can be simulated forward using either Monte Carlo simulation or the non-stochastic histogram method from Young (2010), which we prefer here.

Aggregate simulation. For pre-determined $\boldsymbol{D}_0$, $\boldsymbol{Z}_{-1}$ and $\boldsymbol{S}_{-1}$, and the savings policy function $a^\star(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, m_t)$, the model can be simulated for $t \in \{0, \ldots, T\}$ by the following period-by-period three step procedure

1. Draw $\boldsymbol{Z}_t$ given $\boldsymbol{Z}_{t-1}$;
2. Find $a_t^\star(z_t, m_t) = a^\star(\boldsymbol{Z}_t, \boldsymbol{S}_{t-1}, z_t, m_t)$ by interpolation over $\boldsymbol{Z}_t$ and $\boldsymbol{S}_{t-1}$;
3. Search for $\boldsymbol{P}_t$ so $\int a_t^\star(z_t, m(z_t, a_{t-1}, \boldsymbol{P}_t)) d\boldsymbol{D}_t$ clears the savings market.

A potential computational bottleneck is step 3. A central benefit of the formulation we use here is that the search for market clearing prices only involves interpolating an already known policy function. Typically, a large number of simulation periods is needed, say $5,000$, and therefore small computational costs quickly accumulate.[7]

[7] In models with additional endogenous states and continuous and discrete choices, clearing the market easily becomes a computational bottleneck. See Bakota (2022) for a method to avoid finding

Aggregate solution. The PLM can be found from a fixed-pointed iteration with relaxation as

1. Draw shocks to be used in all simulations
2. Solve and simulate a linearized version of the model
3. Estimate the PLM on the simulated data
4. Given the PLM compute $\breve{\boldsymbol{S}}^0$ and $\breve{\boldsymbol{P}}^0$ on the grid of $\boldsymbol{Z}_t$ and $\boldsymbol{S}_{t-1}$
5. Set the PLM convergence iteration counter $n = 0$
6. Solve the approximate household problem given $\breve{\boldsymbol{S}}^n$ and $\breve{\boldsymbol{P}}^n$
7. Simulate the model given household behavior
8. Estimate the PLM on the simulated data
9. Given the PLM compute $\breve{\boldsymbol{S}}_{NEW}$ and $\breve{\boldsymbol{P}}_{NEW}$ on the grid of $\boldsymbol{Z}_t$ and $\boldsymbol{S}_{t-1}$
10. Stop if $|\breve{\boldsymbol{S}}_{NEW} - \breve{\boldsymbol{S}}^n|_\infty <$ tol. and $|\breve{\boldsymbol{P}}_{NEW} - \breve{\boldsymbol{P}}^n|_\infty <$ tol.
11. Update $\breve{\boldsymbol{S}}$ and $\breve{\boldsymbol{P}}$ by relaxation with $\omega \in (0, 1)$

$$
\begin{aligned}
\breve{\boldsymbol{S}}^{s+1} &= \omega \breve{\boldsymbol{S}}_{NEW} + (1-\omega)\breve{\boldsymbol{S}}^s \\
\breve{\boldsymbol{P}}^{s+1} &= \omega \breve{\boldsymbol{P}}_{NEW} + (1-\omega)\breve{\boldsymbol{P}}^s.
\end{aligned}
$$

12. Increment n and return to step 6

The convergence criteria in step 10 and the relaxation in step 11 is done in terms of the evaluated values of the PLM on the grid. This ensures that the relaxation is done on the inputs to the household problem. The standard approach of iterating on the parameters *inside* the PLM is less appealing when the PLM is non-linear in these parameters. It would then not be ensured that a relaxation scheme in terms of averages of current and past parameters would imply smooth changes in the inputs to the household problem.

market clearing prices period-by-period.

The potential computational bottlenecks in the algorithm are in steps 6-8. The computational challenges in finding the market clearing prices in step 7 were discussed above. The computational complexity of the household problem in step 6 is increasing in the number of aggregate shocks and states due to the curse of dimensionality, and their combined number must therefore be small. For simplicity, we here consider tensor product grids and multi-linear interpolation.[8] The computational complexity of the estimation in step 8 will depend on the chosen function approximation method for the PLM. It will be negligible, when the parameters in the PLM can be estimated with a simple linear regression, but can increase substantially if training of a neural net is required, as discussed later.

Functional approximation methods. The choice of functional approximation method used for the PLM is central. We consider tree different approaches:

1. Linear regression (short: OLS)
2. Neural net with a single layer (short: NN)
3. Radial basis function interpolation (short: RBF)

To describe these, let $X_{it} \in \mathbf{Z}_t, \mathbf{S_{t-1}}$ denote the i'th input to the PLM for $i \in \{1, \ldots, \#_{ZS}\}$ and let $Y_{jt} \in \boldsymbol{S}_t, \boldsymbol{P}_t$ denote the j'th output for $j \in \{1, \ldots, \#_{SP}\}$.

For *linear regression*, we have

$$Y_{jt} = \Psi_{j0} + \sum_{i=1}^{\#_{ZS}} \Psi_{ji} X_{it}. \tag{5}$$

Estimation of Ψ is straightforward with OLS.[9]

For the *neural net*, we have

$$Y_{jt} = \Psi_{j0} + \sum_{q=1}^{Q} \Psi_{jq} \phi \left(\Psi_{jq0} + \sum_{i=1}^{\#_{ZS}} \Psi_{jqi} s(X_{it}, \Psi_{js}) \right), \tag{6}$$

[8] The presence of aggregate non-linearities makes global interpolation methods problematic. A speed-up could, however, probably be achieved with adaptive sparse grids (see e.g. Brumm and Scheidegger, 2017; **?**; Eftekhari and Scheidegger, 2022).

[9] Technically, it is straightforward to include non-linear terms. For a specific model in question, it is, however, not a priori clear, which terms to include.

where $\phi(\bullet)$ is the activation function, $s(\bullet)$ is the scaling function and Q is the number of neurons. We found best behavior with $Q = 5,000$ neurons, and *ReLu* activation functions, $\phi(x) = \max\{0, x\}$, on data scaled to have zero mean and unitary variance.[10] We train the neural net (i.e. find Ψ) using stochastic gradient descend with a Nesterov momentum of 0.90 and mean squared errors as the loss function. We aim for a perfect fit and over-fitting is therefor not an issue. We terminate when no improvement has been achieved for 5 epochs. We use the implementation in *tensorflow* in Python.

For the *radial basis functions*, we have

$$Y_{jt} = \Psi_{j00} + \sum_{i=1}^{\#_{ZS}} \Psi_{j0i} X_{it} + \sum_{\tau=1}^{\mathcal{T}} \Psi_{jk} \phi\left(\sum_{i=1}^{\#_{ZS}} \sqrt{\left(X_{it} - X_{i\tau}^{\text{sim}}\right)^2} \right), \tag{7}$$

where $\phi(\bullet)$ is the kernel, and $X_{i\tau}^{\text{sim}}$is the i'th input to the PLM from the τ'th period in the simulation for $\tau \in \{0, \ldots, T\}$. We use a *thin plate function* kernel, $\phi(x) = x^2 \log x$. The parameters are chosen by solving the equation system implied by exactly fitting all the data points from the simulation. We use the implementation in *scipy* in Python.

The choice of function approximation method must be evaluated both in terms of its ability to secure the stability of the solution method, its accuracy at convergence, and the implied solution time. Specifically, we measure the accuracy of the derived aggregate laws of motion using the long-run dynamic forecast errors as suggested by Den Haan (2010a).

A central issue for stability of the solution method is, that the complicated geometry of inputs, $\boldsymbol{Z}_t$ and $\boldsymbol{S}_{t-1}$, change between iterations, even when keeping the shocks used for simulation fixed. Therefore extrapolation beyond the data set where the PLM was estimated is unavoidable. All the three chosen function approximation methods deliver good results in this respect. We have also experimented with global interpolation using Chebyshev polynomials and local interpolation using nearest or natural neighbor interpolation or Barycentric interpolation from a Delaunay-triangulation. These function approximation methods, however, all suffer from boundary issues, and we have often found stability problems in terms of lack of convergence.

[10] It is straightforward to use multi-layered neural nets as well. We have found no benefit from doing this.

Regarding *implementation complexity*, the three function approximation methods are similar, as they all rely on standard packages.

3 Benchmark model: HANC

In this section, we present a baseline heterogeneous agent neo-classical (HANC) model with aggregate risk. Essentially, it is an extension of Krusell and Smith (1998) to have i) technology utilization costs and ii) capital adjustment costs. These extensions introduce an aggregate non-linearity and a forward looking term, such that we need to search for prices at each period along the simulation.

Firms. A representative firm rents capital, K_{t-1}, and hires labor, L_t, to produce goods, with the production function

$$Y_t = u_t Z_t K_{t-1}^{\alpha} L_t^{1-\alpha}, \tag{8}$$

where Z_t is exogenous technology, and u_t is technology utilization. Technology follows an AR(1):

$$Z_{t+1} - Z_{ss} = \rho_Z (Z_t - Z_{ss}) + \epsilon_{t+1}^Z,\ \ \epsilon_t^Z \sim \mathcal{N}(0, \sigma_Z^2). \tag{9}$$

Changing u_t involves virtual adjustment costs. Capital depreciates with the rate δ. The representative firms objective is given as

$$\begin{gathered} \max_{L_t, K_{t-1}, u_t} u_t Z_t K_{t-1}^{\alpha} L_t^{1-\alpha} - w_t L_t - r_t^k K_{t-1} - \chi_1 \left(u_t - \tilde{u}\right) - \tfrac{\chi_2}{2} \left(u_t - \tilde{u}\right)^2 \\ \text{s.t. } u_t \leq \bar{u}. \end{gathered}$$

where r_t^k is rental rate of capital and w_t is the wage rate, χ_1 and χ_2 are parameters determining the size of the linear and convex adjustment costs, and $\tilde{u}$ is zero-cost level of technology utilization. Finally, $\bar{u}$ is an upper bound on u_t introducing an aggregate non-linearity. The problem yields standard pricing equations plus an equation pinning down technology capacity utilization

$$r_t^k = \alpha u_t Z_t (K_{t-1}/L_t)^{\alpha\text{-}\mathbf{1}} \equiv r^k(u_t, Z_t, K_{t-1}, L_t) \tag{10}$$

$$w_t = (1-\alpha) u_t Z_t (K_{t-1}/L_t)^{\alpha} \equiv w(u_t, Z_t, K_{t-1}, L_t) \tag{11}$$

$$u_t = \max\left[\frac{Z_t K_{t-1}^{\alpha} L_t^{1-\alpha} - \chi_1 + \chi_2 \tilde{u}}{\chi_2}, \bar{u}\right] \equiv u(Z_t, K_{t-1}, L_t). \tag{12}$$

The implied (real) interest rate is

$$r_t = r_t^k - q_t\delta = r^k(u_t, Z_t, K_{t-1}, L_t) - \delta \equiv r(u_t, Z_t, K_{t-1}, L_t).$$

Capital producers. Capital producers choose investment, I_t,and take the price of capital, q_t, as given. Investments are subject to quadratic adjustment costs, and capital producers objective is given as

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t I_t \left\{ q_t \left[1 - \frac{\phi}{2}\left(\log \frac{I_t}{I_{t-1}}\right)^2 \right] - 1 \right\}.$$

Optimality implies (dropping higher-order terms)

$$q_t \left[1 - \phi \log \frac{I_t}{I_{t-1}}\right] = 1 - \beta \mathbb{E}_t \left[q_{t+1} \phi \log\left(\frac{I_{t+1}}{I_t}\right)\right]. \tag{13}$$

Assuming that capital adjustment costs are virtual, the law of motion for aggregate capital is trivially given by

$$K_t = I_t + (1-\delta) K_{t-1} - \tfrac{\phi}{2}\left(\log \tfrac{I_t}{I_{t-1}}\right)^2 . \tag{14}$$

Implies zero adjustment costs in steady-state.

Households. Households are heterogeneous ex post with respect to their productivity, z_t, and assets, a_{t-1}. The distribution of households is denoted D_t. Each period household exogenously supply z_t units of labor, and choose consumption c_t subject to a no-borrowing constraint. The household problem is

$$
\begin{aligned}
v(Z_t, K_{t-1}, I_{t-1}, z_t, a_{t-1}) &= \max_{a_t, c_t} \frac{c_t^{1-\sigma}}{1-\sigma} + \beta \mathbb{E}_t \left[v(Z_{t+1}, K_t, I_t, z_{t+1}, a_t) \right] \qquad (15) \\
&\text{s.t.} \\
L_t &= 1 \\
K_{t-1} &= \int a_{t-1} dD_t \\
r_t, w_t, K_t, I_t &= \text{PLM}(Z_t, K_{t-1}, I_{t-1}) \\
a_t + c_t &= (1 + r_t) a_{t-1} + w_t z_t \\
\log z_{t+1} &= \rho_z \log z_t + \psi_{t+1} \ \ , \psi_{t+1} \sim \mathcal{N}(\mu_\psi, \sigma_\psi),\ \mathbb{E}[z_{t+1}] = 1 \\
Z_{t+1} &= \rho_Z (Z_t - Z_{ss}) + \epsilon_{t+1}^Z \\
D_{t+1} &= \Lambda(Z_t, D_t) \\
a_t &\geq 0.
\end{aligned}
$$

Calibration. The calibration is arbitrary, but implies aggregate and idiosyncratic dynamics within reasonable bounds.

1. **Preferences:** $\sigma = 2$, $\beta = 0.995$
2. **Income:** $\rho_z = 0.96$, $\sigma_\psi = 0.15$
3. **Production and investment:** $\alpha = 0.33$, $\delta = 0.05$, $\varphi = 0.05$
4. **Technology utilization:** $\overline{u} = 1.0$, $\chi_1 = 1$, $\chi_2 = 1$, $\tilde{u} = 0.99$
5. **Aggregate technology:** $\rho_Z = 0.80$, $\sigma_Z = 0.01$

3.1 Global solution

The general solution method can be used cf. Section 2 as follows

1. The shocks are $\boldsymbol{Z}_t = \{Z_t\}$.
2. The aggregate states are $\boldsymbol{S}_t = \{K_t, I_t\}$.
3. The »prices« are $\boldsymbol{P}_t = \{r_t, w_t\}$,

4. The PLM is

$$\begin{aligned}
K_t &= \mathrm{PLM}_K(Z_t, I_{t-1}, K_{t-1}; \Psi) \\
q_t &= \mathrm{PLM}_q(Z_t, I_{t-1}, K_{t-1}; \Psi) \\
u_t &= u(Z_t, K_{t-1}, 1) \\
w_t &= w(u_t, Z_t, K_{t-1}, 1) \\
r_t^k &= r(u_t, Z_t, K_{t-1}, 1) \\
r_t &= r_t^k - q_t \delta_t.
\end{aligned}$$

5. The cash-on-hand function is

$$m(z_t, a_{t-1}, \boldsymbol{P}_t) = (1 + r_t) a_{t-1} + w_t z_t.$$

6. The market clearing condition is

$$\int a_t^{\star}(\boldsymbol{z}_t, m(\boldsymbol{z}_t, a_{t-1}, w_t, r_t)) dD_t = K_t,$$

where we guess on I_t and get r_t from

$$\begin{aligned}
q_t &= \frac{1 - \beta \mathbb{E}_t \left[q_{t+1} \phi \log \left(\frac{I_{t+1}}{I_t} \right) \right]}{1 - \phi \log \left(\frac{I_t}{I_{t-1}} \right)} \\
K_{t+1} &= \mathrm{PLM}_K(Z_{t+1}, I_t, K_t; \Psi) \\
I_{t+1} &= K_{t+1} - (1 - \delta) K_t \\
q_{t+1} &= \mathrm{PLM}_q(Z_{t+1}, I_t, K_t; \Psi) \\
u_t &= u(Z_t, K_{t-1}, 1) \\
w_t &= w(u_t, Z_t, K_{t-1}, 1) \\
r_t^k &= r(u_t, Z_t, K_{t-1}, 1) \\
r_t &= r_t^k - q_t \delta_t,
\end{aligned}$$

and where expectations are evaluated using Gauss-Hermite quadrature on the actual law of motion, $Z_{t+1} = \rho_Z (Z_t - Z_{ss}) + \epsilon_{t+1}^Z$.

Numerical implementation. For the *household problem*, we use $\#_a = 80$ grid points for $a_t \in [0, 100]$ and $\#_z = 3$ grid points for z_t discretized using the Rouwenhorst-method. For the deterministic steady state we solve and simulate the household problem with a tolerance of 10^{-12}. When solving the household problem globally, we lower the tolerance to 10^{-4}.

For the *aggregate states* we use, 10 grid points for Z_t, 20 grid points for K_{t-1}, 15 grid points for I_{t-1} and 3 Gauss-Hermite nodes for Z_{t+1}. The grid spans are chosen relative to the mean values of the PLM states from initial linear simulation.

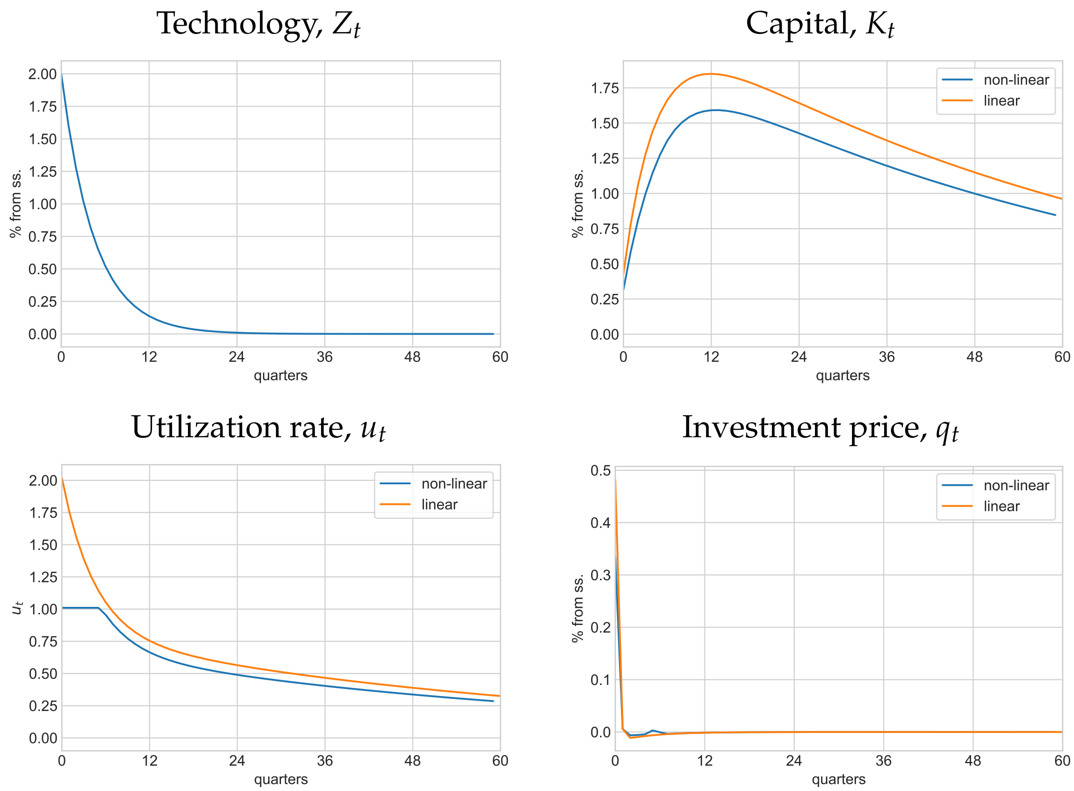
We simulate the model for $5,000$ periods and use a market clearing tolerance of 10^{-6}. For the *PLMs*, we use a relaxation weight of $\omega = 0.65$ and a tolerance of 10^{-4} whenever possible.

Timings were computed on a Windows 10 desktop computer with a i7-4770 3.4 GHz CPU with 4 cores and 32 GB of RAM.

3.2 Perfect foresight IRFs

To get intuition on the aggregate non-linearites present in the model, we first consider a 2 percent shock to technology and compare the linear and non-linear perfect foresight impulse responses computed with the sequence space method in Auclert et al. (2021). In Figure 1, we see that the linearized solution does *not* take the non-linearity of technology utilization into account, and implies a utilization rate way above the allowed upper limit of $\overline{u} = 1$. The positive effect on capital accumulation is consequently overstated relative to the nonlinear impulse response. This shows that a simulation of the model with aggregate risk using linearization will be very inaccurate. Therefore a global solution is needed.

Figure 1: Perfect foresight and linearized impulse responses


Technology, Z_t

Capital, K_t

Utilization rate, u_t

Investment price, q_t

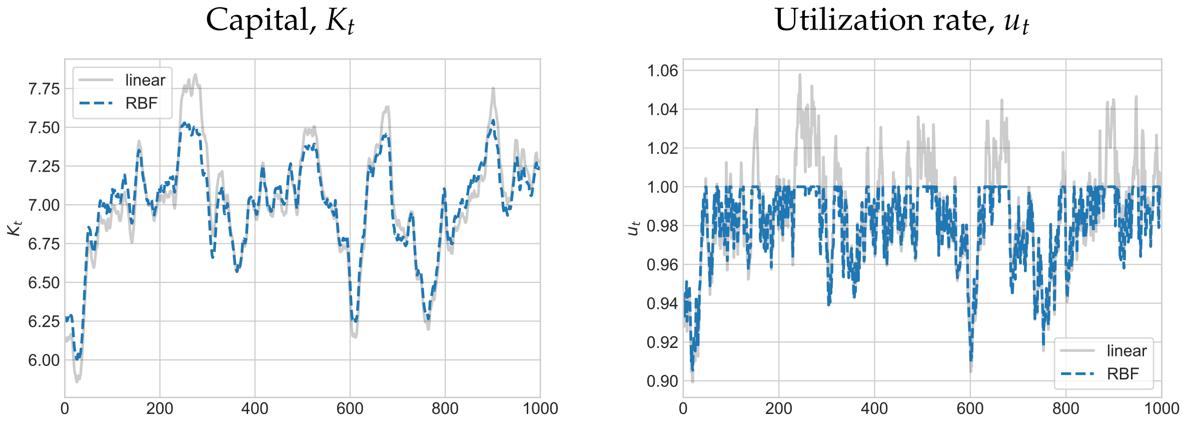
4 Results

In this Section, we present results from using the global solution presented in Section 3 and how the choice of function approximation method in the PLMs affects the accuracy and solution time. This section does not contain results with a neural net for function approximation due to convergence issues. In Section 5, we consider the limit of no capital adjustment costs, $\varphi \to 0$, where all the function approximation methods yields converged global solutions.

4.1 Simulations

As explained in Section 2, we begin our global solution methods from a linearized simulation given a fixed draw of aggregate shocks. At convergence, our simulation method implies a new simulation with these aggregate shocks. Figure 2 shows the last 1,000 periods from these two simulations, where we use RBF for function approximation in the PLMs. We see that the upper limit of the utilization rate is violated in the linearized simulation, but respected in the global simulation. This naturally affects the dynamics of capital, which is markedly different, but still similar enough to make the linearized model reasonable to start from.

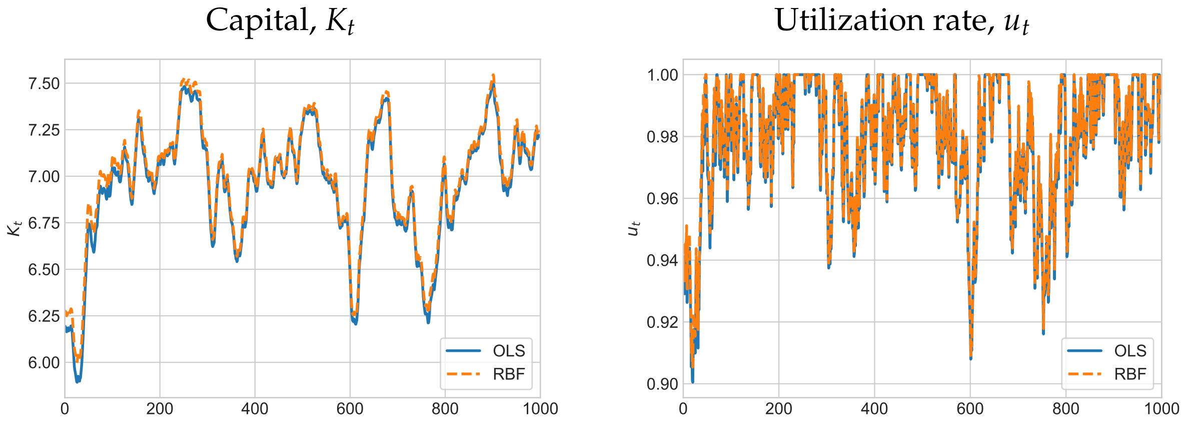
Figure 2: Simulation (in-sample): RBF vs. linear


Capital, K_t

Utilization rate, u_t

Next, we consider out-of-sample simulations with new draws of shocks not used in the solution. In Figure 3, we show global simulations with the OLS and RBF function approximation methods for the PLMs. The OLS based simulation is consistently beneath the RBF based one, though the differences are not large.

Figure 3: Simulation (out-of-sample): PLM methods

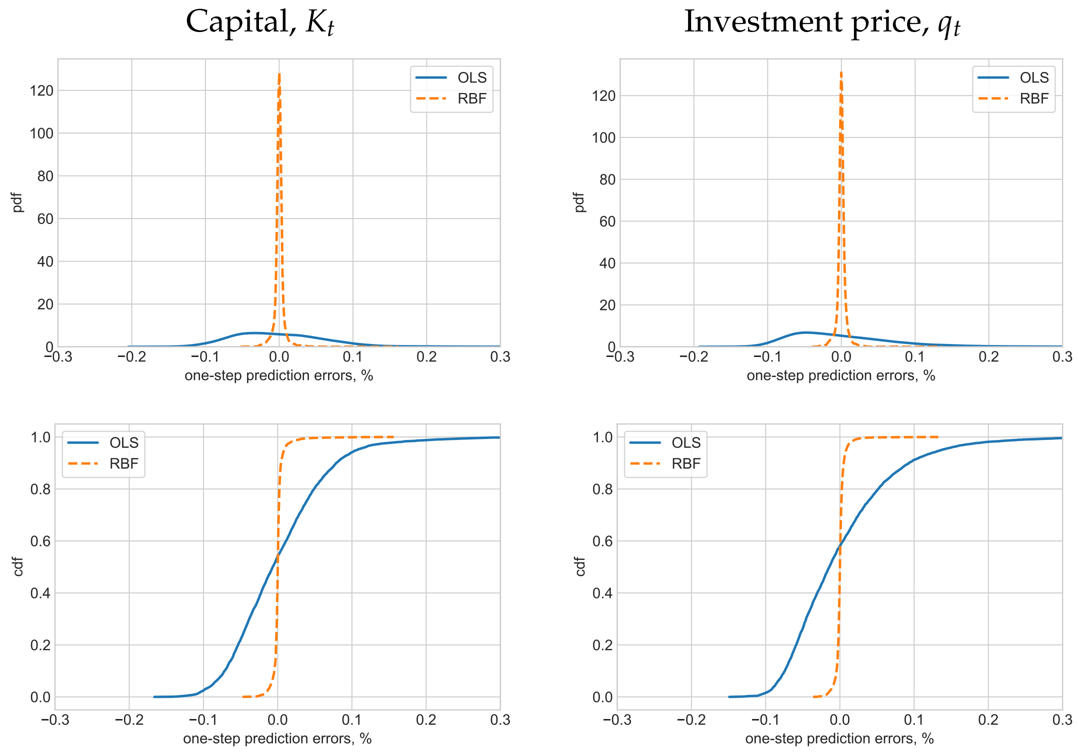

Capital, K_t

Utilization rate, u_t

4.2 Accuracy

To choose between the different function approximation methods, we now consider their accuracy. In Figure 4, we first show the one-step prediction errors for capital and the investment price in percent of the true realizations in an out-of-sample simulation. Using RBF delivers much more precise predictions than using OLS. With RBF, the errors are nicely centered around zero. With OLS, the errors are fat-tailed and left-skewed. The skewness is likely due to the upper bound on the technology utilization rate.

Figure 4: One-step ahead PLM errors



Capital, K_t

Investment price, q_t

A stronger accuracy measure is given by the dynamic Den Haan errors, cf. Den Haan (2010a). We use a standard out-of-sample simulation and a simulation based on the PLMs alone to calculate these. Specifically, the path of capital and the investment price is generated by the PLMs alone as

$$
\begin{aligned}
K_t^{\text{PLM}} &= \text{PLM}_K(Z_t, K_{t-1}^{PLM}, I_{t-1}^{PLM}) \\
q_t^{\text{PLM}} &= \text{PLM}_K(Z_t, K_{t-1}^{PLM}, I_{t-1}^{PLM}) \\
I_t^{\text{PLM}} &= K_t^{\text{PLM}} - (1-\delta)K_{t-1}^{PLM}.
\end{aligned}
\tag{16}
$$

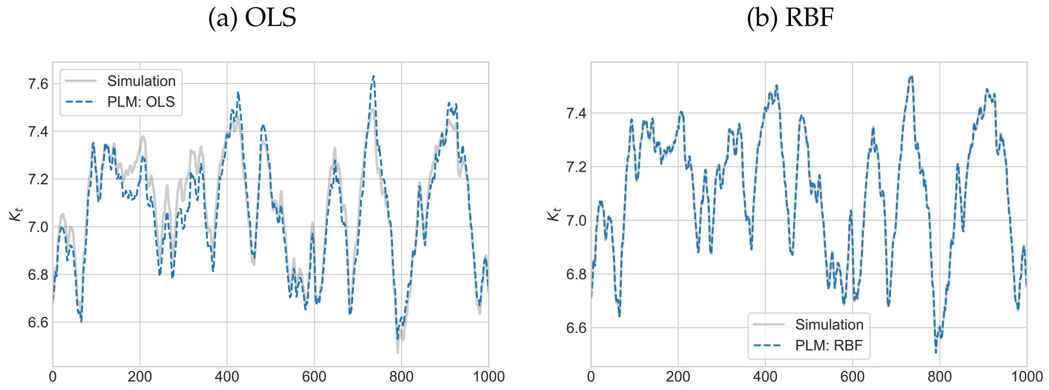
The dynamic errors are then calculated in percentage terms as

$$
\begin{aligned}
&100 \times \left|\log K_t^{\text{PLM}} - \log K_t\right| \\
&100 \times \left|\log q_t^{\text{PLM}} - \log q_t\right|.
\end{aligned}
\tag{17}
$$

In Figure 5, we plot the dynamic PLM forecasts against the actual model simulation. We see that with RBF the PLM simulation lies almost exactly on top of the model simulation, while with OLS it is consistently below.

This observation is also confirmed in Table 1. For both the maximum and the average, the dynamic errors are more than an order of magnitude larger with OLS than RBF. This is moreover achieved with a minimal increase in the solution time of about 10 percent due to an increased time to estimate the PLMs. The RBF errors are very small and economically acceptable. Using RBF takes slightly longer due to a more complicated estimation of the PLM. With both methods, the solution time is less than 15 minutes. RBF is therefore clearly preferred.[11]

Figure 5: Dynamic PLM errors


(a) OLS

(b) RBF

[11] The low number of required simulations shows the benefits of starting from the linearized solution.

Table 1: HANC: Prediction Errors

	OLS	RBF	NN
	dynamic log prediction errors $\times$ 100		
max	3.75	0.26	
mean	0.88	0.04	
median	0.79	0.03	
99th perc.	3.16	0.18	
90th perc.	1.62	0.07	
		timings (secs.)	
total	666.3	722.2	
- solve household problem	356.2	315.4	
- simulate with market clearing	310.0	356.3	
- estimate PLMs	0.0	50.5	
iterations	13	14	

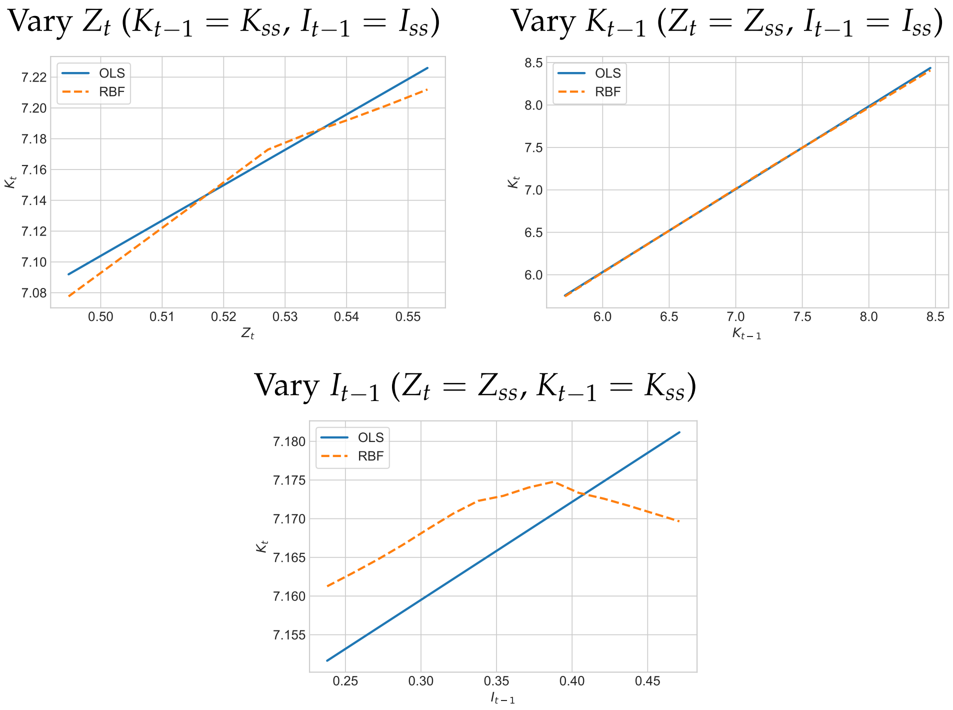
4.3 PLMs

To understand the differences in accuracy, we now investigate how the PLMs actually look like.

In Figures 6 and 7, we plot how the PLMs for capital, K_t, and the investment price, q_t, vary along each input dimension keeping the values of the other inputs fixed at their (deterministic) steady-state values.

We see that RBF interpolation captures non-linearities in the model, especially along the Z_t and the I_{t-1} dimensions, which cannot be captured with linear OLS.

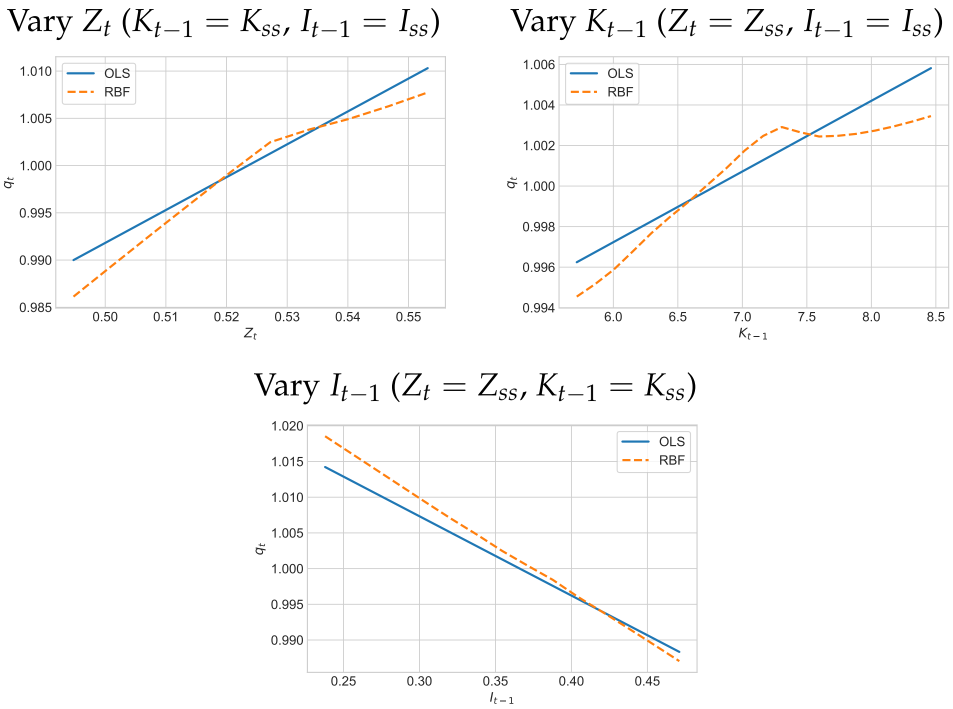
Figure 6: PLM for capital, K_t – 2D slices


Vary Z_t ($K_{t-1} = K_{ss}$, $I_{t-1} = I_{ss}$)

Vary K_{t-1} ($Z_t = Z_{ss}$, $I_{t-1} = I_{ss}$)

Vary I_{t-1} ($Z_t = Z_{ss}$, $K_{t-1} = K_{ss}$)

Figure 7: 2PLM for investment price, q_t – 2D slices


Vary Z_t ($K_{t-1} = K_{ss}$, $I_{t-1} = I_{ss}$)

Vary K_{t-1} ($Z_t = Z_{ss}$, $I_{t-1} = I_{ss}$)

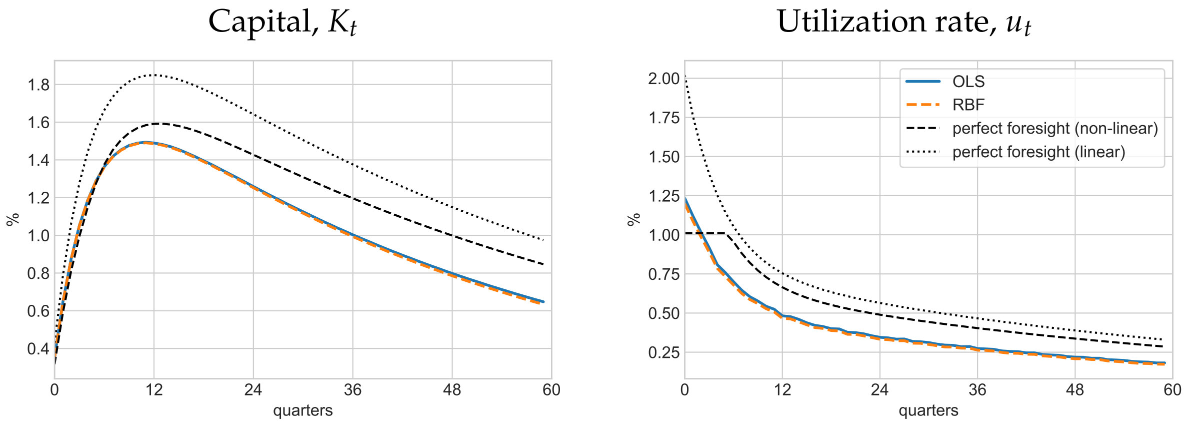
Vary I_{t-1} ($Z_t = Z_{ss}$, $K_{t-1} = K_{ss}$)

4.4 Global solution IRFs

In this sub-section, we present impulse responses functions (IRFs) from the global solutions and compare them with IRFs from the linear and non-linear perfect foresight solutions. We again consider a 2 percent TFP shock as in Section 3.

To obtain IRFs from the global models, we add the shock to the baseline technology path, Z_t, and simulate forward. We then subtract the baseline simulation (the model simulation without the added shock) and divide it by the initial value. We do this for about 500 evenly spaced different starting points and take the mean to obtain global impulse responses. Results are presented in Figure 8.

Figure 8: Impulse-responses: Global vs. perfect foresight

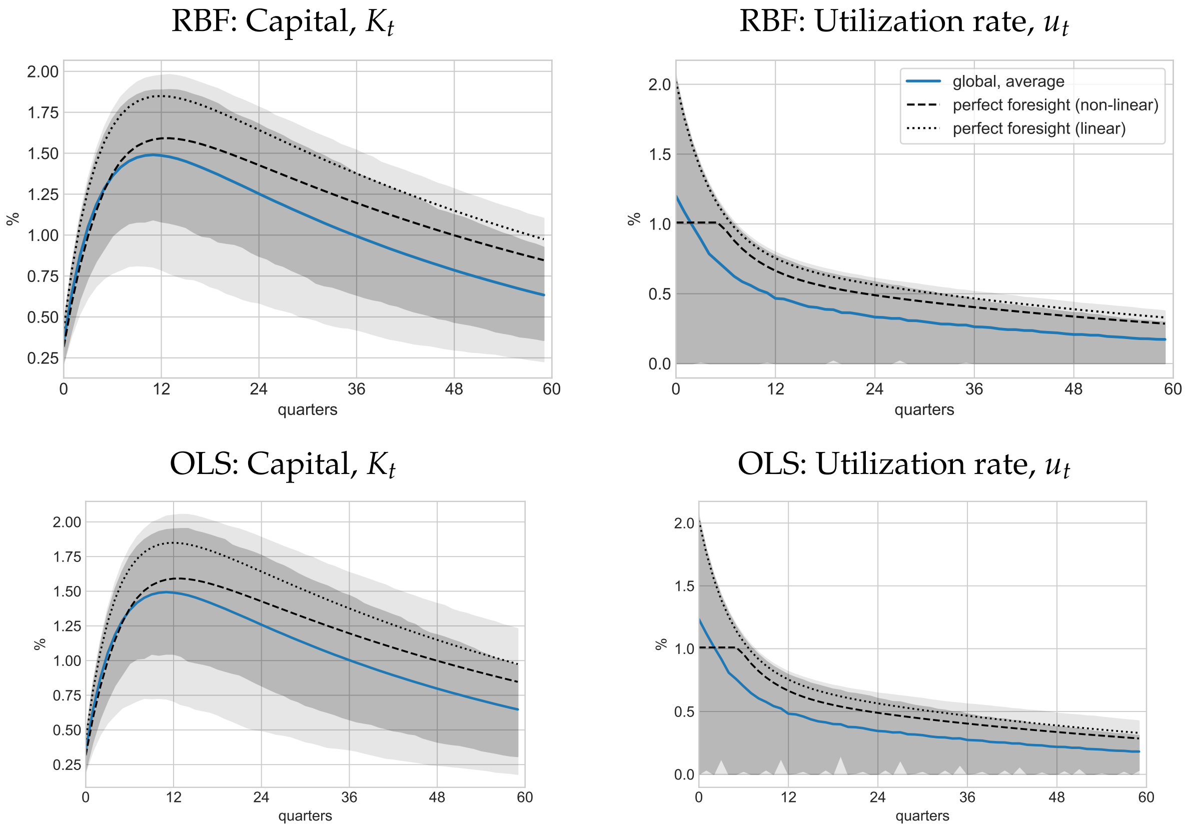


The main takeaway is that the impulses from global non-linear models differ quite a bit from perfect foresight-based solutions. Remember, that the linearized perfect foresight solution is a linearized global solution, and is thus directly comparable with the non-linear global solution. We also note that the OLS and RBF based global solutions are very close. Thus, even if the RBF based PLM is more precise and yields different simulation paths, for shocks of this size, IRFs are pretty similar.

Note that the reason the technology utilization rate, u_t, is never capped off in the global solution IRFs is that the IRFs are means over many IRFs, some of which hit the upper bound and some do not, whereas the perfect foresight IRFs are relative to the deterministic steady-state.

Finally, we also show global IRFs with 90 (light grey) and 70 percentile (dark grey) bands in Figure 9. This shows state dependence; initial values matter for the resulting IRFs. We note minor differences between the OLS and RBF PLM bands; the OLS bands are wider. Thus the OLS based model seems more volatile than the RBF based one.

Figure 9: Impulse-responses: State-dependence



RBF: Capital, K_t

RBF: Utilization rate, u_t

OLS: Capital, K_t

OLS: Utilization rate, u_t

5 Results with no capital adjustment costs

In this section, we consider a simpler variant of the model without capital adjustment costs, $\varphi \to 0$. Then there is only one aggregate state, $S_t = K_t$, and we do not need to search over I_t for market clearing, as the pricing equations give use price, r_t and w_t, directly from the pre-determined stock of capital.

To give an overview of the viability of the different function approximation methods for the PLMs, we again calculate dynamic Den Haan errors and the solution time in Table 2. When using OLS and RBF the results are similar to those obtained in the full model. The NN -based solution now converges. The implied accuracy is satisfactory and, on average, almost as good as with RBF. The maximum errors are, however, much larger. This is likely because the neural net overestimates curvature along the endpoints of the capital grid, a problem not present in the radial basis function solution, see Appendix Figure A.2. Additionally, the solution time is an order of magnitude longer with NN. This primarily results from a very slow training step, which makes it time-consuming to estimate the PLMs. With NN, the number of required iterations is also higher.

Additional figures are include in Appendix A.

Table 2: HANC: Prediction Errors (no capital adjustment costs)

	OLS	RBF	NN
	dynamic log prediction errors × 100		
max	4.86	0.37	3.17
mean	0.99	0.03	0.08
median	0.84	0.02	0.02
99th perc.	3.52	0.25	1.69
90th perc.	1.99	0.06	0.09
	timings (secs.)		
total	255.7	335.6	1943.0
- solve household problem	230.0	218.9	513.0
- simulate (no market clearing)	25.7	28.0	39.7
- estimate PLMs	0.0	88.7	1390.3
iterations	10	15	22

6 Conclusion

We have presented an extension of the basic Krusell-Smith algorithm to account for non-linear dynamics. Our preferred choice is to model the perceived laws-of-motion (PLMs) with radial basis function interpolation. This delivers stable, accurate, and precise results and is easy to implement. To the best of our knowledge, the proposed global solution method is state-of-the-art for heterogeneous agent models, which can be approximated by a low number of aggregate states, but have non-linear dynamics.

Multiple further lines of inquiry are possible and valuable. The internal robustness of the proposed solution method can be investigated by varying grid sizes, tolerances and simulation lengths. Additional activation functions for the neural net and additional kernels for the radial basis function interpolation can also be investigated. A method related to radial basis function interpolation, gaussian process regression (implemented in e.g. *scikit-learn*) could also be considered. To investigate the convergence issues further, additional models should be considered. Including models with New Keynesian features.

Code speed-up is achievable by using parallelization and graphics cards (especially for the neural net). Or by solving the household problem using a Howard improvement step as in Rendahl (2022) (or a modified policy iteration version hereof).

Finally, the accuracy and speed of radial basis function interpolation make it a good candidate for use in a parameterized expectation based approach.

References

Ahn, S., Kaplan, G., Moll, B., Winberry, T., and Wolf, C. (2018). When Inequality Matters for Macro and Macro Matters for Inequality. *NBER Macroeconomics Annual*, 32:1–75.

Algan, Y., Allais, O., Den Haan, W. J., and Rendahl, P. (2014). Chapter 6 - Solving and Simulating Models with Heterogeneous Agents and Aggregate Uncertainty. In Schmedders, K. and Judd, K. L., editors, *Handbook of Computational Economics*, volume 3 of *Handbook of Computational Economics Vol. 3*, pages 277–324. Elsevier.

Auclert, A., Bardóczy, B., Rognlie, M., and Straub, L. (2021). Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models. *Econometrica*, 89(5):2375–2408.

Azinovic, M., Gaegauf, L., and Scheidegger, S. (2022). Deep Equilibrium Nets. SSRN Scholarly Paper 3393482.

Bakota, I. (2022). Avoiding Root-Finding in the Krusell-Smith Algorithm Simulation. Technical report.

Bayer, C. and Luetticke, R. (2020). Solving discrete time heterogeneous agent models with aggregate risk and many idiosyncratic states by perturbation. *Quantitative Economics*, 11(4):1253–1288.

Bayer, C., Luetticke, R., Pham-Dao, L., and Tjaden, V. (2019). Precautionary Savings, Illiquid Assets, and the Aggregate Consequences of Shocks to Household Income Risk. *Econometrica*, 87(1):255–290.

Bilal, A. (2021). Solving Heterogeneous Agent Models with the Master Equation. Working Paper.

Boppart, T., Krusell, P., and Mitman, K. (2018). Exploiting MIT shocks in heterogeneous-agent economies: the impulse response as a numerical derivative. *Journal of Economic Dynamics and Control*, 89:68–92.

Brumm, J. and Scheidegger, S. (2017). Using Adaptive Sparse Grids to Solve High-Dimensional Dynamic Models. *Econometrica*, 85(5):1575–1612.

Curry, M., Trott, A., Phade, S., Bai, Y., and Zheng, S. (2022). Finding General Equilibria in Many-Agent Economic Simulations Using Deep Reinforcement Learning. *arXiv:2201.01163 [cs, econ, q-fin]*.

Den Haan, W. J. (1996). Heterogeneity, Aggregate Uncertainty, and the Short-Term Interest Rate. *Journal of Business & Economic Statistics*, 14(4):399–411. Publisher: [American Statistical Association, Taylor & Francis, Ltd.].

Den Haan, W. J. (1997). Solving dynamic models with aggregate shocks and heterogenous agents. *Macroeconomic Dynamics*, 1(2):355–386. Publisher: Cambridge University Press.

Den Haan, W. J. (2010a). Assessing the accuracy of the aggregate law of motion in models with heterogeneous agents. *Journal of Economic Dynamics and Control*, 34(1):79–99.

Den Haan, W. J. (2010b). Comparison of solutions to the incomplete markets model with aggregate uncertainty. *Journal of Economic Dynamics and Control*, 34(1):4–27.

Eftekhari, A. and Scheidegger, S. (2022). High-Dimensional Dynamic Stochastic Model Representation. Number: arXiv:2202.06555 arXiv:2202.06555 [cs, econ, q-fin].

Fernández-Villaverde, J., Hurtado, S., and Nuño, G. (2020). Financial Frictions and the Wealth Distribution. Working Paper 26302, National Bureau of Economic Research. Series: Working Paper Series.

Fernández-Villaverde, J., Marbet, J., Nuño, G., and Rachedi, O. (2021). Inequality and the Zero Lower Bound. Technical report.

Gorodnichenko, Y., Maliar, L., Maliar, S., and Naubert, C. (2021). Household Savings and Monetary Policy under Individual and Aggregate Stochastic Volatility. Technical report. Publisher: CEPR Discussion Paper No. DP15614.

Han, J., Yang, Y., and E, W. (2022). DeepHAM: A Global Solution Method for Heterogeneous Agent Models with Aggregate Shocks. Number: arXiv:2112.14377 arXiv:2112.14377 [cs, econ, q-fin].

Hill, E., Bardoscia, M., and Turrell, A. (2021). Solving Heterogeneous General Equilibrium Economic Models with Deep Reinforcement Learning. *arXiv:2103.16977 [cs, econ, q-fin]*.

Krusell, P. and Smith, A. A. (1997). Incoem and wealth heterogeneity, portfolio choice, and equilibrium asset returns. *Macroeconomic Dynamics*, 1(02):387–422.

Krusell, P. and Smith, A. A. (1998). Income and wealth heterogeneity in the macroeconomy. *Journal of Political Economy*, 106(5):867–896.

Maliar, L. and Maliar, S. (2020). Deep Learning: Solving HANC and HANK Models in the Absence of Krusell-Smith Aggregation. Technical report.

Maliar, L., Maliar, S., and Valli, F. (2010). Solving the incomplete markets model with aggregate uncertainty using the Krusell–Smith algorithm. *Journal of Economic Dynamics and Control*, 34(1):42–49.

Maliar, L., Maliar, S., and Winant, P. (2021). Deep learning for solving dynamic economic models. *Journal of Monetary Economics*, 122:76–101.

Reiter, M. (2002). Recursive computation of heterogeneous agent models. Working Paper.

Reiter, M. (2009). Solving heterogeneous-agent models by projection and perturbation. *Journal of Economic Dynamics and Control*, 33(3):649–665.

Reiter, M. (2010). Approximate and Almost- Exact Aggregation in Dynamic Stochastic Heterogeneous-Agent Models. IHS Economics Series Working Paper 258.

Rendahl, P. (2022). Continuous vs. Discrete Time: Numerical Gains from Trade. Technical report.

Tan, E. (2020). A fast and low computational memory algorithm for non-stochastic simulations in heterogeneous agent models. *Economics Letters*, 193:109285.

Terry, S. J. (2017). Alternative Methods for Solving Heterogeneous Firm Models. *Journal of Money, Credit and Banking*, 49(6):1081–1111.

Valaitis, V. and Villa, A. T. (2021). A Machine Learning Projection Method for Macro-Finance Models. Working Paper.

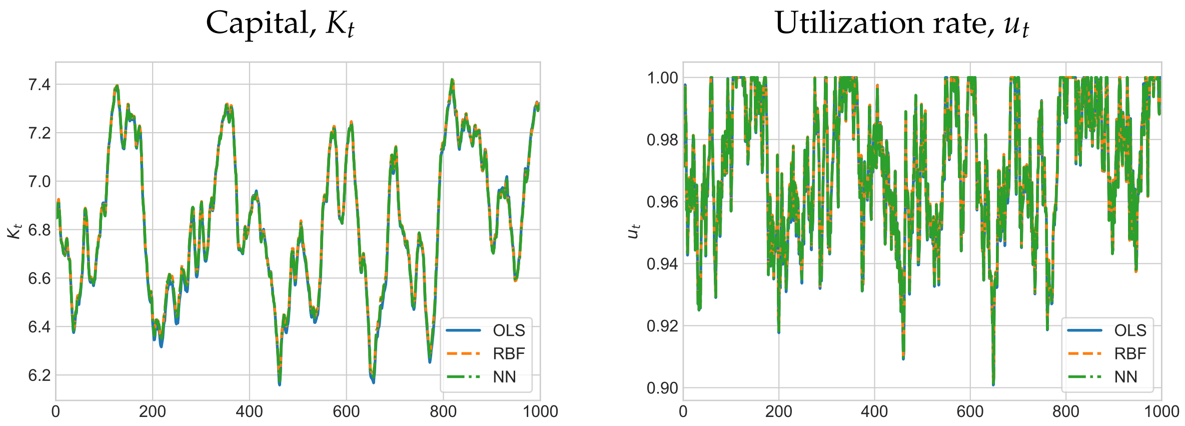
Winberry, T. (2018). A method for solving and estimating heterogeneous agent macro models. *Quantitative Economics*, 9(3):1123–1151–1151.

Young, E. R. (2010). Solving the incomplete markets model with aggregate uncertainty using the Krusell–Smith algorithm and non-stochastic simulations. *Journal of Economic Dynamics and Control*, 34(1):36–41.

A Additional Tables and Figures

A.1 No capital adjustment costs

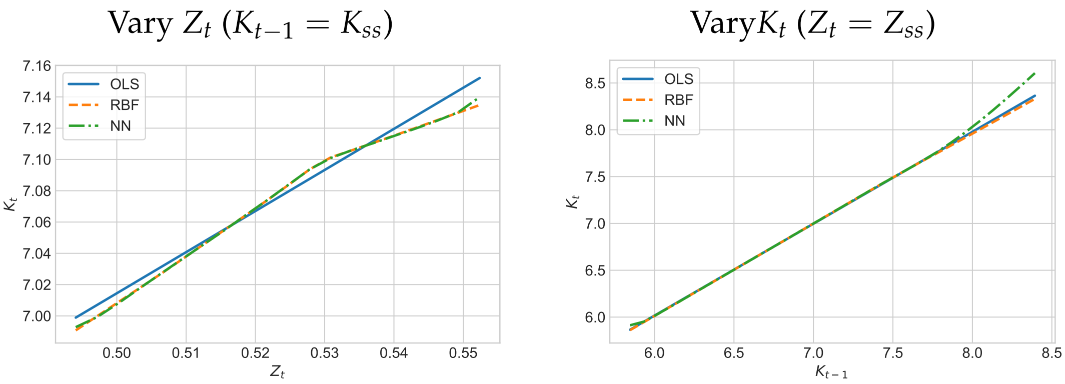
Figure A.1: Simulation (out-of-sample): PLM methods



Capital, K_t

Utilization rate, u_t

Figure A.2: PLM for capital, K_t – 2D slices



Vary Z_t ($K_{t-1} = K_{ss}$)

VaryK_t ($Z_t = Z_{ss}$)

Figure A.3: One-step ahead PLM errors

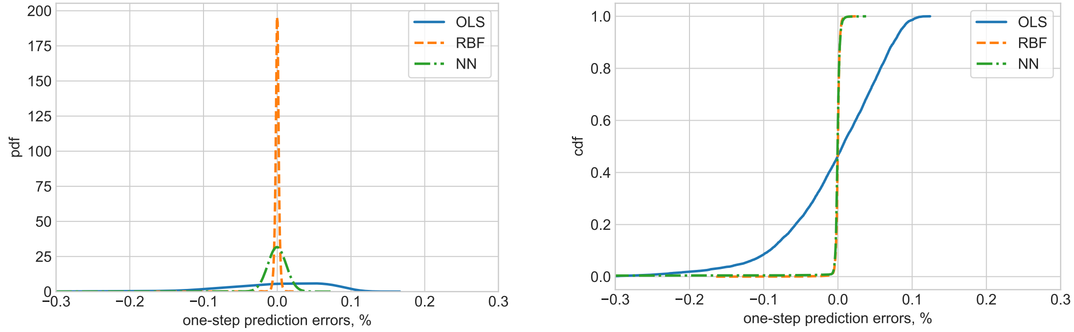
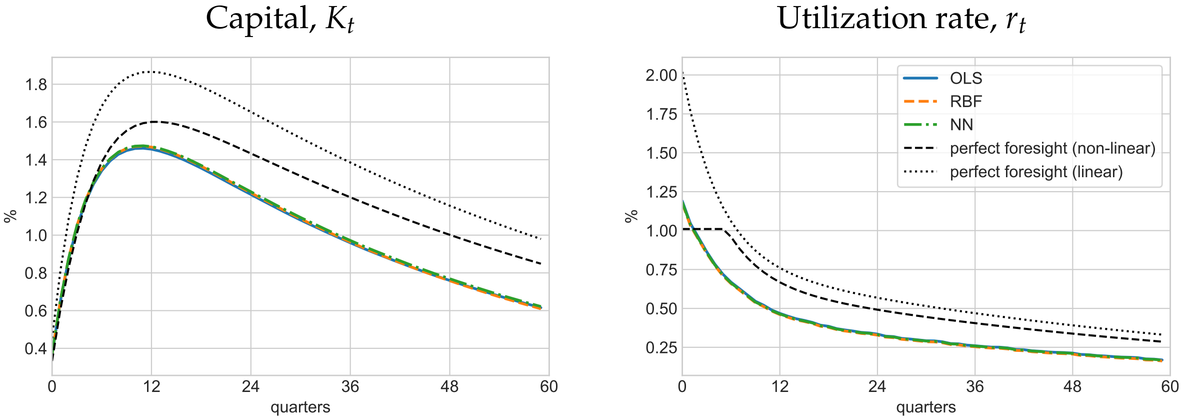


Figure A.4: Impulse-responses: Global vs. perfect foresight


Capital, K_t

Utilization rate, r_t