Journal of Economic Dynamics & Control 192 (2026) 105412



Contents lists available at ScienceDirect

Journal of Economic Dynamics and Control

journal homepage: www.elsevier.com/locate/jedc





Neural networks vs. traditional methods: A hybrid nonlinear approach for solving large OLG models ⋆

Michael Reiter

Institute for Advanced Studies, Josefstädter Strasse 39, Vienna, A-1080, Austria

ARTICLE INFO

JEL classification:
C63
C68
E21

Keywords:
Perturbation methods
Neural networks
Machine learning
OLG Models

ABSTRACT

This paper develops a hybrid nonlinear solution method for large-scale dynamic macroeconomic models, combining insights from the deep learning literature with the strengths of traditional perturbation and projection techniques. The method embeds economically motivated functional forms and perturbation-based derivatives within an approximation that uses deep function composition. Applied to a large OLG model with borrowing constraints and multiple aggregate shocks, the solution method achieves high accuracy over a wide range of the relevant state space.

1. Introduction

With the advances in both hardware and software, researchers study quantitative macroeconomic models of ever higher complexity and larger dimension. Examples include large OLG models, heterogeneous agent (HANK) models, and models with disaggregated industry structure. Linearization around a stationary state is typically feasible for these models. However, the models often contain substantial nonlinearities such as occasionally binding constraints, and estimating the effect of uncertainty on behavior is a requirement for the solution. To cope with the increasing difficulty of the models, existing software packages, and most prominently Dynare (Juillard, 1996; Adjemian et al., 2024), steadily expand the toolkit of available solution techniques, typically based on perturbation or projection methods.

In addition, fundamentally new solution methods are developed based on deep learning, exploiting the dramatic recent advances in the field of artificial intelligence, supported by the availability of very fast GPUs. I use the term "Deep Learning" as in [Fernández-Villaverde (2025), Chapter 8] as the use of neural networks with many layers. "Neural network" (abbreviated "NN" in the following) is a very general concept, but in most applications it is a composition of linear (more precisely, affine) functions and low-dimensional (often element-by-element) nonlinear "activation functions" with desirable properties such as piecewise linearity or boundedness. Deep learning methods offer a high degree of generality: in principle, NNs can approximate virtually any nonlinear function and are relatively robust to high-dimensional inputs. Their flexibility derives from heavy parameterization and training by stochastic gradient descent with efficient backpropagation algorithms. Therefore, they seem to promise a powerful approximation that requires little researcher intervention in choosing functional forms or algorithms.

⋆ I am grateful to participants at the workshop "Celebrating Michel Juillard's Career and 30 Years of Dynare" for helpful discussions and suggestions. An anonymous referee provided very extensive and extremely insightful comments, which have greatly helped to improve the paper. Earlier work on the model was supported by the Austrian National Bank (Anniversary Fund project no. 17815).

E-mail address: michael.reiter@ihs.ac.at

https://doi.org/10.1016/j.jedc.2026.105412
Received 1 April 2025; Received in revised form 11 December 2025; Accepted 11 September 2026
Available online 13 September 2026
0165-1889/© 2026 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

The contribution of the present paper is to propose a hybrid solution methodology with the aim of combining the advantages of both the traditional solution methods and the deep learning approach. The method is motivated by the observation that most macroeconomic models have a special structure, which traditional perturbation and projection methods can easily exploit. In particular, the models have a stationary state, the solution is smooth in the neighborhood of the stationary state, and perturbation methods provide a reasonably precise solution in this neighborhood. Moreover, the nonlinearities are of a specific type, such as borrowing constraints or the zero-lower-bound constraint in monetary economics.

In general terms, the hybrid approach can be seen as a special case of a projection method, where equilibrium decision functions are approximated by parameterized functions, and parameters are chosen so as to minimize a set of equilibrium conditions (such as Euler residuals) on stochastic grids over the state space. Both the functional forms and the solution algorithm adopt ideas of the deep learning literature:

- Function composition is an important building block of the approximation. This approach promises to be both more flexible and more robust than the traditional approximations such as Chebyshev polynomials.
- Solving for the parameters of the approximation is understood as a stochastic process, similar to how stochastic gradient descent is used in deep learning.
- In high-dimensional models, very general methods can be used to select a reduced state vector.

The traditional components are the following:

- The first-order perturbation (linearization) around the deterministic steady state, which is retained as part of the overall approximation. It also serves as the starting point of the solution process.
- A nonlinear transformation of the reduced state variables and of the approximated variables, similar to Judd (2002) and Fernandez-Villaverde and Rubio-Ramirez (2006).
- A nonlinear approximation of the policy variables as a function of a reduced state vector. For example, this may include a quadratic component when the approximated function is assumed to be smooth, or a parsimonious spline approximation to account for changes in behavior close to occasionally binding constraints.

My approach is building on projection methods, pioneered by Judd (1992), and on perturbation methods, pioneered by Judd and Guu (1993), Juillard (1996). Perturbation methods have recently been applied to a very wide class of heterogeneous-agent models (Reiter, 2009; Boppart et al., 2018; Auclert et al., 2021; Reiter, 2023; Bilal, 2023; Bhandari et al., 2023), which can serve as a basis for applying hybrid methods to these models. The present paper can be seen as offering an alternative to a literature that uses sophisticated grid-based methods for high-dimensional models such as OLG models (Krueger and Kübler, 2004, 2006; Judd et al., 2014; Brumm and Scheidegger, 2017; Brumm et al., 2018).

The relationship of the present paper to the deep learning literature is explained in Section 2. The detailed description of the algorithm is given in Section 4, with a focus on the implementation for the example model of Section 3. Section 5 presents the numerical results.[1]

2. Outline of the solution approach

The literature on neural networks in economics and finance is rapidly evolving. Problems in dynamic optimization and derivative pricing can often be expressed as partial differential equations, for which the mathematical literature has provided powerful neural-network-based methods (E et al., 2017; Han et al., 2018; Sirignano and Spiliopoulos, 2018). Dynamic general equilibrium models have a more complicated structure, where the stability of solutions and the convergence of algorithms are serious concerns. Maliar et al. (2021) show that deep learning techniques are broadly useful for solving macroeconomic models. Fernandez-Villaverde et al. (2024) and Fernández-Villaverde (2025) provide a recent overview. It should be pointed out that the idea of combining theoretical knowledge with deep learning has also proven useful in the natural sciences; see, for example, Raissi et al. (2019).

Sections 2.2–2.6 explain in detail the relationship between the present paper and the deep learning literature. Section 2.1 prepares for that discussion by specifying the mathematical form of the problem to be solved. Finally, Section 2.7 uses the RBC model as a simple example to motivate some of the main ideas of the hybrid approach.

2.1. The objective function

We approximate the policy or equilibrium variables of the model as functions of the lagged state x_{t-1} and current shock vector ϵ_t, parameterized by the vector θ, written as $\mathcal{Y}(x_{t-1}, \epsilon_t; \theta)$. Parameters θ will be chosen so as to satisfy a set of equilibrium conditions in the form of (properly scaled) Euler residuals $\mathcal{E}(x_{t-1}, \epsilon_t; \theta)$. If $\mathcal{X}(.;\theta)$ denotes the elements of $\mathcal{Y}(.;\theta)$ that are state variables, Euler residuals can be written as

$$\mathcal{E}(x_{t-1}, \epsilon_t; \theta) \equiv \mathrm{E}_{\epsilon_{t+1}} \mathcal{F}\big(\mathcal{Y}(x_{t-1}, \epsilon_t; \theta), \mathcal{Y}(\mathcal{X}(x_{t-1}, \epsilon_t; \theta), \epsilon_{t+1}; \theta); \theta\big) \tag{1}$$

[1] The code to replicate the results is provided as a Julia package at https://myservice.ihs.ac.at/mreiter/research.html.

Section 3.4 lists the equation system that defines the Euler residuals for the example model of Section 3. If a parameter vector θ gives the exact solution of the model, then $\mathcal{E}\left(x_{t-1}, \epsilon_t; \theta\right) = 0$ for the set S of relevant values of (x_{t-1}, ϵ_t). Usually such a θ does not exist and we look for the θ that is best in the sense of minimizing the loss function

$$\int_S \|\mathcal{E}(x, \epsilon; \theta)\|^2 \omega(x, \epsilon)\, d(x, \epsilon) \tag{2}$$

where $\omega(x, \epsilon)$ is a weighting function that will reflect the relevance of the point (x, ϵ) in the state space.

2.2. Function composition

Traditional nonlinear methods approximate unknown functions as a (linear) combination of known basis functions (a seminal reference is Judd (1992)). Higher accuracy is achieved by increasing the number of basis functions. In multi-dimensional applications, some form of orthogonal polynomial is typically used. However, apart from becoming expensive to evaluate when the number of basis functions is large, higher-order polynomials also suffer from Runge's phenomenon: the approximation can oscillate near the edges of the interpolation interval, leading to very poor extrapolation.

In contrast, deep NNs are constructed by composition of "well-behaved" functions. For example, [Azinovic et al. (2022), Eq. (37)] write their neural network as

$$x \longmapsto \sigma_K\big(W_K \cdots \sigma_2\big(W_2\sigma_1(W_1 x + b_1) + b_2\big) \cdots + b_K\big),$$

where the W_k are matrices, b_k are vectors and σ_k are nonlinear "activation" functions acting element-wise on their vector arguments. Typical activation functions are piecewise linear or bounded functions, which serve to impose bound constraints. Both for interpolation and for extrapolation close to the training set, function composition therefore appears to give better control of the fit than polynomial-based approximation.

The idea of the hybrid approach is to use function composition, but use functional forms that are motivated by what is known about the model solution, and which nest a first order perturbation solution as a special case, which also serves as a starting point in the computation of the nonlinear solution. This is explained in detail in Section 4.1.

2.3. Parameter identification as a stochastic procedure

The loss function (2) contains two integrals: an expectation over the known distribution of ϵ_{t+1} and an integral over the model's state space. While the former can in principle be computed on a fixed quadrature grid (cf. Section 4.2), the latter is far more challenging. The relevant region of the state space is endogenous and not known a priori, so evaluating this integral accurately requires repeatedly simulating the model and exploring the set of states actually visited in equilibrium. Moreover, achieving high precision would require very long simulations, making the computation prohibitively expensive for realistic, high-dimensional models. Consequently, the objective function cannot be evaluated exactly in practice, even at a fixed θ.

Situations where the objective involves intractable or prohibitively costly expectations are common in machine learning and motivate treating optimization as a stochastic approximation problem. A substantial literature provides convergence results for methods of this type (e.g., Bellavia et al., 2018; Shao and Fan, 2025). In practice, stochastic gradient descent (SGD) and adaptive variants such as ADAM (Kingma and Ba, 2017) are widely used. Even with a fixed data set, computing the objective function and its gradient on the full sample is expensive; using smaller samples ("mini-batches"), particularly in early stages of the solution process, reduces computational burden. The noise in the gradient evaluation coming from small samples may even help to escape shallow local minima.

The same considerations apply here: since the integrals in (2) are approximated by simulation, each function and gradient evaluation is inherently noisy. Treating parameter identification as a stochastic procedure is therefore both natural and computationally efficient. Section 4.4 describes the sampling strategy used in the numerical example.

2.4. Endogenous state reduction

In heterogeneous agent models where the theoretical state space contains a whole distribution, which is an infinite-dimensional object, Krusell and Smith (1998) pushed the idea that economic agents can achieve approximately optimal behavior by making their actions a function of a few statistics (such as moments) that summarize the relevant information from the distribution. In their specific application, the cross-sectional mean of the distribution already turned out to convey enough information for an approximate equilibrium, although it is straightforward to allow for the use of higher cross-sectional moments. Reiter (2010a, 2023) outline a method to compute a set of linear functions of the distribution that allow optimization up to machine precision, in a linearized version of heterogeneous agent models. Both methods can be seen as a linear operation on the state space, $x^{reduc} = Hx$ for some matrix H. Han et al. (2026) and Payne et al. (2024) call x^{reduc} "generalized moments" and estimate H as part of the network. I will follow this approach in Section 4.1. A comparison to Reiter (2023) would be interesting but is left to future work.

2.5. Parsimonious parameterization

Fernandez-Villaverde et al. (2024) cite overparameterization as the first feature that distinguishes neural networks (NNs) from traditional methods. This property is made possible by regularization and motivated by the "double descent" phenomenon, the

seemingly paradoxical observation that overfitting the data can lead to improved model performance in many applications (Belkin et al., 2019). Kahou et al. (2024) discuss potential explanations for this in terms of "inductive bias" of machine learning solutions.

In contrast, the hybrid approach uses a highly parsimonious parameterization of the approximation, which I see as the first key deviation of the hybrid method from the NN approach. The motivation for this is threefold. First, the hope is that there is no need for a highly parameterized and flexible function if functional forms are informed by economic knowledge about the structure of the problem. Second, with a small set of parameters, some of the parameters can be assigned an interpretable meaning. This is useful when conditions must be imposed on the solution, cf. Section 4.5. The third motivation is technical. NNs with many parameters must rely on training methods that use only the gradient of the objective function. Typically, this will be stochastic gradient descent (SGD), as mentioned above, although limited-memory BFGS may be feasible if the problem is not too ill-conditioned. These methods benefit from the *cheap-gradient principle*, which states that computing the full gradient of a function costs only a small multiple of evaluating the function itself, regardless of the number of arguments. This property holds only for the reverse mode of automatic differentiation (AD), for which NNs are well structured and for which highly optimized software and hardware implementations exist.

However, for large and complex general equilibrium models in economics, reverse-mode AD cannot always be applied efficiently, partly due to memory restrictions (although this may change with further progress in software and hardware). I therefore employ forward-mode AD, which is easier to implement for general functions and requires less memory. Its drawback is that the cheap-gradient principle no longer applies. Nevertheless, it provides not only the gradient of the objective function (2) but also the gradients of each residual with respect to all parameters. This information allows the use of more powerful optimization algorithms such as Levenberg–Marquardt (LM), provided that the number of parameters is not too large.[2]

2.6. Functional forms

The second key difference to the deep learning literature is the more extensive use of a priori knowledge about a specific model. While the NN literature often tailors the approximation to reflect properties such as symmetry (Kahou et al., 2021), transversality conditions (Kahou et al., 2024) or the nature of the equilibrium conditions (Azinovic and Zemlicka, 2024), the hybrid approach uses economic knowledge about a specific model to choose functional forms. For variables that are expected to move smoothly with the states, it is natural to use quadratic approximations close to the stationary state.[3] For variables that may have kinks or even jumps, one can use spline functions where the knot points are determined endogenously. The approximation can nest the linearized or other traditional solutions as a special case and use it as a starting point of the iterative solution. Details are provided in Section 4.1.

2.7. A motivating example

A simple textbook model can serve to illustrate some of the key ideas of the hybrid method. Consider the RBC model with production function $Y_t = z_t K_{t-1}^{\alpha} L_t^{1-\alpha}$, utility function $U(c, L) = \log(c) + \eta \log(1 - L)$, TFP process $z_t = 1 + \rho(z_{t-1} - 1) + \epsilon_t$, and capital depreciation rate δ. It is convenient to approximate the labor input L_t, since it always lies between 0 and 1. Given past states (K_{t-1}, z_{t-1}) and the current shock ϵ_t, once an approximation for L_t is available, all other variables follow analytically:

$$
\begin{aligned}
z_t &= 1 + \rho(z_{t-1} - 1) + \epsilon_t \\
w_t &= (1 - \alpha)(z_t)(K_{t-1}/L_t)^{\alpha} \\
c_t &= w_t(1 - L_t)/\eta \\
r_t &= \alpha z_t (K_{t-1}/L_t)^{\alpha-1} - \delta \\
y_t &= z_t K_{t-1}^{\alpha} L_t^{1-\alpha} \\
I_t &= y_t - c_t \\
K_t &= I_t + (1 - \delta) K_{t-1}
\end{aligned}
$$

Using the scaled power transformation

$$
BC(x, \lambda) = \frac{1}{\lambda}\left(x^{\lambda} x_{stst}^{1-\lambda} - x_{stst}\right)
$$

which satisfies $BC(x_{stst}, \lambda) = 0$ and $dBC(x_{stst}, \lambda)/dx = 1$, we approximate $L(K, z)$ by the following composition of functions:

$$
\begin{aligned}
&\hat{K} = BC(K, \lambda_k), \qquad \hat{z} = BC(z, \lambda_z) \\
&\hat{L} = L_{stst} + \theta_0 + \theta_k \tilde{K} + \theta_z \tilde{z} + \theta_{kk} \tilde{K}^2 + \theta_{kz} \tilde{K} \tilde{z} + \theta_{zz} \tilde{z}^2 \\
&L = T(\hat{L}) \equiv L_{stst}(\tanh((\hat{L} - L^*)/L^*) + 1)
\end{aligned}
$$

The transformation $T(\hat{L})$ in the last line maps the real line onto the interval $(0, 2L_{stst})$, which works well since $L_{stst} = 0.365$ and $2L_{stst} < 1$. It satisfies $T(L_{stst}) = L_{stst}$, $T'(L_{stst}) = 1$, and $T''(L_{stst}) = 0$.

[2] For the case of very large parameter vectors, there is a literature on matrix-free LM methods (Curtis et al., 2009; Kandel et al., 2021; Martens, 2010).

[3] It appears that in NNs with standard activation functions such es ReLU, many parameters are required to approximate even some simple functions such as a quadratic with the commonly used activation functions (Fan et al., 2020).

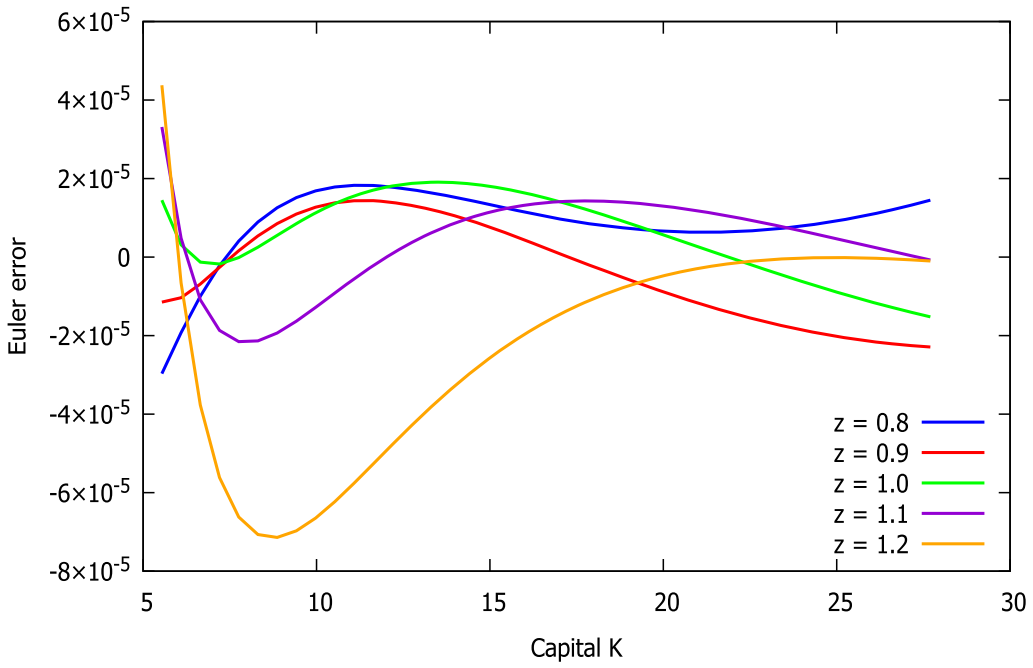


Fig. 1. Euler equation residuals in the RBC model.

The parameters θ_K and θ_z are taken directly from the linearization around the steady state. Because of the properties of $BC(\cdot)$ and $T(\cdot)$, the derivative at the deterministic steady state is preserved. The free parameters are $\lambda_k, \lambda_z, \theta_0, \theta_{kk}, \theta_{kz}, \theta_{zz}$, and they are chosen to minimize the sum of squared Euler residuals over a wide range of K and z around $K_{stst} = 13.85$ and $z_{stst} = 1$. Fig. 1 reports the Euler residual on a fine grid over the same domain. The Euler residual is scaled as $\mathcal{E}(x_t, z_t) = \beta \, \mathrm{E}_t \left[(1 + r_{t+1})(c_t/c_{t+1})\right] - 1$ so that it can be interpreted as a relative consumption error. With only six free parameters, the residuals are on the order of 10^{-5} even far from the deterministic steady state. This accuracy is achieved by exploiting structural properties of the solution: power transformations are useful in the presence of log-linear relationships, quadratic terms exploit the smoothness of the solution, and sigmoidal functions enforce known bounds on labor supply. Approximating only the labor function and computing all other quantities analytically reduces the number of parameters and guarantees that budget constraints and pricing equations hold exactly.[4] To solve the high-dimensional OLG model in Section 3, additional techniques are required; these will be explained in Section 4.

3. Example: A macroeconomic model of retirement saving

3.1. Overview

This is a model of retirement saving and of risk sharing between overlapping generations of worker households and a representative capitalist household. The model uses a combination of rather standard features of OLG models, and it was designed as an example that is hard to solve by global nonlinear methods. It features a large state space, many aggregate shocks, and occasionally binding constraints.

The model period is a year. There are two types of households: a representative capitalist dynasty, and $J = 60$ cohorts of worker households. Workers exogenously supply labor for $J^W = 40$ periods and are retired afterwards. When retired, they receive benefits from a public PAYGO pension system, financed by payroll taxes. The government runs the pension system as a defined-contribution scheme, always with a balanced budget. Capitalists issue bonds, hold the aggregate capital stock, and earn profits from monopolistically competitive firms. They can also save in bonds, but cannot borrow.

3.2. States and shocks

The model has 5 exogenous states, each following an AR(1) process:

[4] Azinovic and Zemlicka (2024) apply similar ideas with their NN approach to a model with several assets and find ways to enforce market clearing.

- Total factor productivity z_t, following

$$z_t = 1 + \rho_z\left(z_{t-1} - 1\right) + \epsilon_{z,t} \tag{3}$$

- The markup in goods markets μ_t, following

$$\mu_t = \bar{\mu} + \rho_\mu\left(\mu_{t-1} - \bar{\mu}\right) + \epsilon_{\mu,t} \tag{4}$$

- The efficiency ϕ with which investment is transformed into new capital, following

$$\phi_t = 1 + \rho_\phi\left(\phi_{t-1} - 1\right) + \epsilon_{\phi,t} \tag{5}$$

- The time discount factor of capitalists β_t^C, following

$$\beta_t^C = \bar{\beta}^C + \rho_\beta\left(\beta_{t-1}^C - \bar{\beta}^C\right) + \epsilon_{\beta,t} \tag{6}$$

- A productivity parameter ψ_t that shifts, if positive, the age-productivity profile in favor of the young, following

$$\psi_t = \rho_\psi \psi_{t-1} + \epsilon_{\psi,t} \tag{7}$$

The model has one endogenous aggregate state, namely productive capital. It follows

$$K_t = \Phi\left(K_{t-1}\left(1 - \delta_t\right), I_t, \phi_t\right) \tag{8}$$

where depreciation $\delta_t = \bar{\delta} + \epsilon_{\delta,t}$ is subject to the i.i.d. shock $\epsilon_{\delta,t}$ and the function Φ captures quadratic capital adjustment costs:

$$\Phi(K, I, \phi) = K + \phi I - \frac{1}{\eta_I \bar{\delta} K}\left(I - \frac{\bar{\delta} K}{1 - \bar{\delta}}\right)^2 \tag{9}$$

In addition, there are $J - 1$ cohort-specific states, namely the end-of-period savings (bond holdings) $b_{j,t}$ of all cohorts except the oldest cohort, which consumes all its resources. This yields a total number of $J + 5$ state variables.

In addition to $\epsilon_{z,t}$, $\epsilon_{\psi,t}$, $\epsilon_{\beta,t}$, $\epsilon_{\delta,t}$, $\epsilon_{\mu,t}$ and $\epsilon_{\phi,t}$, there are J^W i.i.d. shocks $\zeta_{j,t}$ to the labor productivity of each working cohort, cf. Eq. (10) below. This yields a total number of $J^W + 6$ shocks in each period.

3.3. Technical equations and accounting identities

The equations of this section are identities that can be solved analytically in order to reduce the number of variables that must be approximated. They will therefore be satisfied exactly in the nonlinear model solution. If a model contains static optimality conditions that determine endogenous labor supply, they can also be included here, if standard utility functions are used.

3.3.1. Worker households and labor supply

Workers in cohorts $j = 1, \ldots, J^W$ inelastically supply their labor endowment (hours) H_j. Workers in cohorts $j = J^W + 1, \ldots, J$ are retired and do not work.

Hourly productivity $\omega_{j,t}$ of cohort j is subject to the i.i.d. shock $\zeta_{j,t}$ and to the aggregate shock ψ_t, which tilts the productivity profile in favor of younger cohorts:

$$\omega_{j,t} = H_j\left(1 + \zeta_{j,t} + \psi_t \frac{J^W/2 - j}{J^W/2}\right), \qquad j = 1, \ldots, J^W \tag{10}$$

Aggregate labor supply in efficiency units is then given by

$$L_t = \frac{1}{J}\sum_{j=1}^{J^W} \omega_{j,t} \tag{11}$$

Bond holdings of cohort j at the end of period t are denoted by $b_{j,t}$. The gross return R_t^B on one unit of the bond held at the end of $t - 1$ is given by

$$R_t^B = \xi + r^c + (1 - \xi) p_t^B \tag{12}$$

where r^c is the coupon paid by the bond, p_t^B is the price of the bond, and ξ is the fraction of the bond that matures in each period. For a one-period bond, $\xi = 1$. The fraction $1 - \xi$ of the bond is carried to the next period. The budget constraint of cohort j is then

$$c_{j,t} = R_t^B b_{j-1,t-1} + inc(j,t) - p_t^B b_{j,t}, \qquad b_{J,t} = 0 \tag{13}$$

Household income $inc(j,t)$ is subject to a payroll tax at constant rate τ, which serves to finance lump sum pension benefits P_t, keeping the government budget balanced:

$$P_t = \frac{\tau w_t L_t}{J - J^W} \tag{14}$$

Household after-tax income is then given by

$$inc(j,t) = \begin{cases} P_t & \text{if } j > J^W \\ w_t(1 - \tau) H_j \omega_{j,t} & \text{otherwise} \end{cases} \tag{15}$$

3.3.2. Firms and production

To micro-found the capital dynamics in (9) we assume that new capital is produced in a competitive capital sector, using depreciated capital $\overline{K}_t \equiv K_{t-1}(1-\delta_t)$ and investment I as inputs. Since Φ is constant returns to scale in $\overline{K}$ and I, this sector generates zero profits. Because of capital adjustment costs, the value of installed capital differs from the price of output and is given by

$$Q_t = \frac{1}{\Phi_I\left(\overline{K}_t, I_t, \phi_t\right)} \tag{16}$$

Old capital, after being used for production but before depreciation, is sold to the capital producing sector at the price

$$Q_{o,t} = Q_t \Phi_K\left(\overline{K}_t, I_t, \phi_t\right) \tag{17}$$

The representative final good firm produces gross output $\hat{Y}_t$ with labor and capital using a Cobb-Douglas production function:

$$\hat{Y}_t = F\left(z_t, \overline{K}_t, L_t\right) = z_t \overline{K}_t^{\alpha} L_t^{1-\alpha} \tag{18}$$

We assume that factor markets are competitive, but firms charge a markup μ_t in the goods market, which varies exogenously over time, with the dynamics specified in (4).[5] Profit maximization on the part of the firms then implies that factor prices equal marginal productivity divided by the markup. The real wage is given by

$$w_t = \frac{1}{\mu_t} F_L\left(z_t, \overline{K}_t, L_t\right) \tag{19}$$

and the gross rental rate of capital as

$$r_t^K = \frac{1}{\mu_t} F_K\left(z_t, \overline{K}_t, L_t\right) \tag{20}$$

We assume that real GDP equals gross output net of fixed costs of production, which are set as a constant fraction $\overline{FC}$ of steady state monopoly rents, which gives

$$Y_t = \hat{Y}_t - FixCost = \hat{Y}_t - \hat{Y}^* \frac{\mu^* - 1}{\mu^*} \overline{FC} \tag{21}$$

where asterisks denote steady state values. If $\overline{FC} = 1$, fixed costs absorb all the monopoly rent in steady state. In the benchmark calibration, we choose $\overline{FC} = 0.5$.

The model contains three features that generate relatively large fluctuations in the real interest rate: adjustment costs to capital formation, shocks to depreciation and shocks to the effectiveness of new investment.

3.3.3. Capitalist households

We assume that both capital and firms are owned by a representative, infinitely lived dynasty. We refer to these households as capitalists. They make the decisions on physical investment, and their marginal utility of consumption serves as the stochastic discount factor for the firm decisions. Capitalists do not work. They obtain income from firm dividends and bond issuance. The dividend stream (cash flow) is given by

$$d_t = Y_t - w_t L_t - I_t \tag{22}$$

The capitalists' budget constraint can then be written as

$$\hat{C}_t = d_t - R_t^B B(t-1) + p_t^B B(t) \tag{23}$$

where

$$B_t = \frac{1}{J} \sum_{j=1}^{J-1} b_{j,t} \tag{24}$$

is the aggregate amount of bonds held by worker households at the end of period t, which amounts to debt of the capitalist households. The gross return on the stock of firms R_t^F is the sum of dividends d_t and the market price of firms p_t^F:

$$R_t^F = d_t + p_t^F \tag{25}$$

3.3.4. Market clearing

Aggregate consumption of worker households is

$$C_t = \frac{1}{J} \sum_{j=1}^{J} c_{j,t} \tag{26}$$

The labor market clears if labor demand equals effective labor supply L_t. The goods market clears if

$$Y_t = C_t + \hat{C}_t + I_t \tag{27}$$

GDP Y_t equals the sum of capitalist consumption $\hat{C}_t$, investment I_t and the aggregate consumption of workers C_t. The bond market clears if capitalists hold the offsetting position to aggregate household bonds B_t.

[5] This assumption is used as a simple short-cut. It could be micro-founded by assuming that firms produce a differentiated good, which is sold under monopolistic competition with exogenously varying demand elasticity, and then bundled into a final good using a Dixit-Stiglitz aggregator.

3.4. Euler equations and approximated equilibrium variables

A worker household maximizes discounted expected utility of consumption, with constant discount factor β. Marginal utility of consumption is given by $u'_j(c_{j,t}) = \nu_i c_{j,t}^{-\gamma}$ where the factor ν_i equals 1 for active workers and 0.7 for retirees. Workers can save, but not borrow. The first order condition is therefore

$$u'_j(c_{j,t}) \geq \beta \, \mathrm{E}\left[R^B_{t+1} u'_{j+1}(c_{j+1,t+1})\right] \tag{28}$$

with equality if the borrowing constraint is not binding.

The capitalist household makes three investment decisions: holding bonds, holding firm equity and making physical investments in the firm sector. The corresponding Euler equations are

$$\hat{U}_{c,t} p^B_t = \hat{\beta}^C_t \mathrm{E}_t\left(R^B_{t+1} \hat{U}_{c,t+1}\right) \tag{29}$$

$$\hat{U}_{c,t} p^F_t = \hat{\beta}^C_t \mathrm{E}_t\left(R^F_{t+1} \hat{U}_{c,t+1}\right) \tag{30}$$

$$\hat{U}_{c,t} Q_t = \hat{\beta}^C_t \mathrm{E}_t\left(\hat{U}_{c,t+1}(1-\delta_t)\left(r^K_{t+1} + Q_{o,t+1}\right)\right) \tag{31}$$

The Euler residuals $\mathcal{E}(x, \epsilon; \theta)$ in the loss function (2) are defined by Eqs. (28)–(31) in residual form. The occasionally binding constraint in (28) is transformed into an equation by the Fischer-Burmeister function [Miranda and Fackler (2002), Section 3.8].

3.5. Approximate solution

An approximate model solution consists of an approximation of equilibrium variables as a function of the state variables. Eqs. (28)–(31) define a total of $J+2$ equilibrium conditions. They can be used to pin down the following $J+2$ equilibrium variables:

- consumption of capitalists $\hat{C}_t$
- the price of the bond p^B_t
- the gross return on the firm value R^F_t
- the saving decision of worker cohorts $b_{j,t}$ for $j = 1, \ldots, J-1$. The last cohort does not save.

All other variables can be computed analytically as functions of the states, shocks and these approximated variables, using the model equations of the last section. The exact order in which the equations are used is given in Appendix A.

4. Solution method

The algorithm that I use in this paper is intended to illustrate the main ideas in a straightforward way, staying as close as possible to the literature that combines projection and simulation methods (Judd, 1992; Judd et al., 2012, 2011). The differences come from the way that the solution is approximated, and in the way that simulations are used.

We seek a recursive solution in which the relevant objects are approximated as functions of the model's state variables. There is some discretion in choosing which objects to approximate. A key decision is whether to approximate the model's policy variables directly or, instead, the conditional expectations in the model's forward-looking equations, as in the "parameterized expectations" approach of Den Haan and Marcet (1990). Both approaches have advantages and disadvantages. To simplify the exposition of the algorithm, I approximate the policy variables directly.

Solving the optimization problem (2) requires several components. First, we need a parameterized approximation of the equilibrium variables which, together with the model equations, defines the realized residuals $\mathcal{F}\left(x_{t-1}, \epsilon_t, \epsilon_{t+1}; \theta\right)$ (see Section 4.1). Second, we require a procedure to approximate integration over the distribution of shocks ϵ (Section 4.2). Third, we need an algorithm to minimize the loss function with respect to the parameters θ (Section 4.3). Finally, we require a sampling method to approximate the integral of the Euler equation over the state space (Section 4.4).

For parameter identification and to stabilize the solution, it is necessary to impose restrictions on the parameters. These restrictions are implemented through penalty functions, described in Section 4.5. Computational efficiency is discussed in Section 4.6.

4.1. The approximation network

I now describe the specialized neural network that is used to approximate the time-t equilibrium variables defined in Section 3.4, given a vector of lagged (time $t-1$) states x and a vector of shocks ε. We can write this in functional form as $\mathcal{A}(x, \varepsilon; \theta)$. Appendix A describes how to compute all the remaining time-t variables. Write this in functional form as $\mathcal{I}(x, \varepsilon, \mathcal{A}(x, \varepsilon; \theta); \theta)$. Combining the two we obtain the vector of all time-t variables $\mathcal{Y}(x, \varepsilon; \theta)$ and the subset of state variables $\mathcal{X}(x, \varepsilon; \theta)$ that enter the Euler equation (1).

The approximation scheme described here has a special structure, it is not a "basic neural network" as defined, for example, in [Fernández-Villaverde (2025), Chapter 3] or a "densely connected feedforward neural network" as defined in [Azinovic et al. (2022), Chapter 4.2.1]. It is a NN in the sense of using function composition with a flexible choice of states. It is deep (in the sense of several layers of nesting functions) but not broad (it does not have many variables within a given layer). Input to the network is the vector of predetermined state variables x and the vector of current shocks S. The example model has 65 states and 46 shocks, cf. Section 3.2. Output is the vector of approximated variables $Y = (Y^A, Y^I)$, where Y^A denotes aggregate variables and Y^I denotes cohort-specific variables. In the example model, 62 variables have to be approximated, cf. Section 3.4.

We approximate Y as a function of (x, S) through the composition of the following operations:

1. Compute the complete set of beginning-of-period states X as a function of x and S. X should be chosen such that in the recursive solution, all current variables are a function of X. For example, if the exogenous state x follows an AR(1) process, it is equivalent but more efficient to approximate variables as a function of x_t rather than x_{t-1} and the shock ϵ_t.

   Details for the OLG model: The relevant states at the beginning of period t are TFP z_t from Eq. (3), relative productivity level $z_{L,t}$ from Eq. (7), markup μ_t from Eq. (4), investment efficiency ϕ_t from Eq. (5), capitalists' discount factor β_t^C from Eq. (6), capital after depreciation $\overline{K}_t = K_{t-1}(1-\delta_t)$, as well as cohort assets b_{t-1} and the cohort-specific productivity shocks ϵ_t^L.
2. Compute a (medium- to low-dimensional) vector of aggregate variables $\hat{A}$ which are functions of X. These can either be known functions which are expected to be useful (such as K_t^α) or functions with parameters to be determined.

   Details for the OLG model: For the aggregate functions $\hat{A}$ the following variables were used: the aggregate states z_t, $z_{L,t}$, μ_t, ϕ_t, β_t^C and $\overline{K}_t$, the sum of assets held by active worker cohorts, the sum of assets held by retired worker cohorts, aggregate labor input L_t, the rental rate of capital r_t^K and the wage rate w_t.
3. Compute each element $Y_i \in Y^A$ by the following steps:
   (a) Apply the power transformation $T(x;\lambda) \equiv (x^\lambda - 1)/\lambda$ and $T(x;0) \equiv \log(x)$ to each element X_j of X:

   $$\hat{x}_{i,j} = \left(T\left(X_j; \lambda_{i,j}^X\right) - T\left(X_j{}^*; \lambda_{i,j}^X\right)\right)/T_x\left(X_j{}^*; \lambda_{i,j}^X\right) \tag{32}$$

   Here, $T_x(x;\lambda)$ denotes the partial derivative with respect to x.
   (b) Set q_i as the quadratic function in the variables $\hat{A} - \hat{A}^*$ with coefficients c_i^{quad}.
   (c) Set $\hat{y}_i = g_i \cdot \hat{x}_i + q_i$, where g_i is the gradient of Y with respect to X that is obtained from the linear perturbation around the steady state.
   (d) Apply the inverse power transformation

   $$\tilde{y}_i = T^{-1}\left(T_x\left(\hat{A}_i^*; \lambda_i^A\right)\hat{y}_i + T\left(\hat{A}_i^*; \lambda_i^A\right); \lambda_i^A\right) \tag{33}$$

   (e) Enforce non-negativity constraints by $Y_i = softmax(0, \tilde{y}_i)$.

   As a smoothed maximum function I used $softmax(x,y) = \log(\exp(ax) + \exp(ay))/a$ with $a = 10^5$.

   The use of the transformation $T(x;\lambda)$ in (32) and its inverse in (33) is such that the gradient of Y with respect to X is unchanged at the deterministic steady state X^*.[6]

   Details for the OLG model: The aggregate variables Y^A are consumption of capitalists $\hat{C}_t$, asset price p_t^B and the total return on firms R_t^F.
4. Compute additional aggregate variables M as linear functions of $(\hat{A}, Y^A)$ with coefficients determined in the training process.

   Details for the OLG model: Two linear combinations are used, denoted by M_1 and M_2.
5. Separately for each element $i \in Y^I$, compute a vector of cohort-specific variables $\hat{J}_i$ as a function of X, $\hat{A}$ and Y^A.

   Details for the OLG model: $\hat{J}_i$ for cohort i includes end-of-period savings of the last period b_{i-1} and cash-on-hand $\chi_i \equiv R_t^B b_{j-1,t-1} + inc(j,t)$, which can be computed from $\hat{A}$ and b_{i-1}, cf. Eq. (15). For working cohorts, the current productivity shock ζ_i is included in $\hat{J}_i$.
6. Compute each element $Y_i \in Y^I$ by the following steps:
   (a) Same as Step 3a above.
   (b) Define q_i as a parameterized function of the Y_i.

   Details for the OLG model: We set

   $$q_i = c_i^{const} + c_i^{lin} \cdot [\hat{A}; b_{i-1}; \zeta_i] + c_i^{shift} M_1 \chi_i + \int_0^{\chi_i} MPS_i(x)\, dx \tag{35}$$

   where the marginal propensity to save $MPS_i(x)$ is a piecewise linear function, with slopes given by c_i^{slope} and knot points given by $c_i^{knot} + c_i^{shiftknot} M_2$, where c_i^{knot} is a vector shifting each knot point separately.

   The (scalar or vector) parameters c_i^{lin}, c_i^{coh}, c_i^{slope}, c_i^{shift}, c_i^{knot} and $c_i^{shiftknot}$ are trained in the solution process.
   (c) Same as Step 3c above.
   (d) Same as Step 3d above.
   (e) Same as Step 3e above.

The critical part is Step 6b. This is where the understanding of the economic structure of the model comes in. The piecewise linear approximation of MPS is motivated by the fact that there are different saving regimes, zero MPS when a household is constrained, MPS of approximately 0.5 when the household expects to be constrained next period, high MPS depending on age when the asset level is high. Notice that (35) is an extremely parsimonious parameterization. Although the model has 65 states and 46 shocks, the approximation of cohort saving only has 30 free parameters for working and 29 free parameters for retired cohorts. To further

[6] Given invertible and differentiable scalar functions T and S and the constant c, define

$$\hat{x}(x) = \frac{T(x) - T(x^*)}{T'(x^*)}, \qquad \hat{y}(x) = c \cdot \hat{x}(x), \qquad y(x) = S^{-1}\left(S'(y^*)\hat{y}(x) + S(y^*)\right) \tag{34}$$

then $y(x^*) = y^*$ and $y'(x^*) = c$. If T operates element-wise on the vector x, then the calculation applies to the partial derivatives with respect to each component of x.

improve accuracy, the most natural route would be to include more knot points, and/or to model the knots and slopes as functions of the cohort-specific states. The example used here serves to illustrate a number of possible specifications. More experimentation can be used to find out what works best.

4.2. Computing expected values

Our model has 46 shocks; hence, the shock vector ϵ is 46-dimensional. A precise evaluation of conditional expectations, essentially an integration over a 46-dimensional domain, would be prohibitively costly, since classical quadrature does not work in such high dimensions, particularly because the integrands are not very smooth. An often recommended practice in this case is Quasi–Monte Carlo integration with low-discrepancy points such as Sobol points (Dick and Pillichshammer, 2010) as quadrature grids, but even this approach requires a very large number of points to be reliable.

I therefore use an ad hoc procedure to generate a pseudo-random quadrature grid, based on the specifics of the OLG model. The grid is constructed as follows. For each shock dimension j, we generate a collection of n values $\hat{\epsilon}_j(k)$ corresponding to the $(k-0.5)/n$-th quantiles of a normal distribution with mean zero and standard deviation σ_j. For each j, these values are randomly assigned to the j-th component of the n grid points $\hat{\epsilon}(i)$, independently across j.

In this way, the vectors $\hat{\epsilon}(i)$ approximate the marginal distribution of each shock ϵ_j very well. However, because the assignments are independent across components, the sample correlations between shocks $(\epsilon_{j_1}, \epsilon_{j_2})$ deviate from zero. To alleviate this problem, the points are reordered in a second step so as to make the correlations between aggregate shocks as close as possible to zero. In the same way, the correlation of each cohort-specific shock with the aggregate shocks, and with the shocks of the preceding and successor cohorts, is reduced to zero as closely as possible. The correlations of non-neighboring cohorts can differ substantially from zero. As a validity check, I have used a grid of $n = 200$ points constructed in this way, and compared Euler residuals computed with this grid to Euler residuals computed on a grid of 100,000 Sobol points, and the difference turns out to be very small.

4.3. Minimizing the loss function

We approximate the integral over the state space in the loss function (2) by summing over a finite sample obtained by (repeated) simulation, as will be explained in Section 4.4.

Given a sample of n points $(\bar{x}_j, \bar{\epsilon}_j)$ in the state space and a quadrature grid of m shock vectors $\hat{\epsilon}(i)$ defined in Section 4.2, the loss function (2) can be written as

$$\sum_{j=1}^{n}\sum_{i=1}^{m}\frac{1}{m}\mathcal{F}\big(\mathcal{Y}(\bar{x}_j,\bar{\epsilon}_j;\theta),\mathcal{Y}\big(\mathcal{X}(\bar{x}_j,\bar{\epsilon}_j;\theta),\hat{\epsilon}(i);\theta\big);\theta\big) \tag{36}$$

This expression is minimized over the parameter vector θ using the Levenberg-Marquardt method. For numerical robustness, it turned out to be useful to solve the linear system for the search direction by QR-decomposition.

For the calculations reported below, the Levenberg–Marquardt method was applied simultaneously to the full parameter vector θ, which contains 2262 elements. For a more richly parameterized specification, it is likely advantageous to partition the parameter vector into smaller components and, in each iteration, first solve for the oldest cohorts, then for the next younger cohorts, and so on. Alternatively, one could use a backward iteration approach similar to Reiter (2010b) or Cao et al. (2023).

4.4. Sampling

An important element of the solution process is the sampling and re-sampling of the grid on which the loss function is computed. Here is the rather simple strategy that I have used for the numerical example. Clearly, many variations are possible.

- Set **initial values** for the batch size n, the maximal batch size $n^{\max}$, and the parameter vector Θ. Furthermore, choose the size of the validation sample $N >> n$ and the size of the quadrature grid.
- **Outer loop**. For $k = 0, \ldots$
  - (a) Draw a new set of shocks S_k
  - (b) Use the current parameter values Θ and the shocks S_k to simulate a batch of $10n$ state and shock vectors $(\bar{x}_j, \bar{\epsilon}_j)$ for $j = 1, \ldots, 10n$; use every 10-th element of the simulation, to reduce the average correlation between grid points. In addition, simulate a set of $10N$ state and shock vectors as a validation sample.
  - (c) **Inner loop**. For $l = 0, \ldots$
    - i. As explained in Section 4.3, use the Levenberg–Marquardt (LM) algorithm to compute a new parameter vector $\hat{\Theta}$ that minimizes the loss function (36) on the current batch.
    - ii. Check whether the output of the LM algorithm reduces the loss function on the validation sample.
      - If it does, set $\Theta = \hat{\Theta}$
      - Otherwise:
        - If $n < n^{\max}$, set $n = \min\{2n, n^{\max}\}$ and continue with the next iteration of the inner loop.
        - Otherwise, stop the inner loop and go to the next iteration of the outer loop.
  - (d) Stop outer loop if no further progress is made in the outer loop on the validation sample.

The initial batch size was set to $n = 500$ and the maximal size to $n^{\max} = 2000$. The initial value of the parameter vector Θ is set such that the q_i in Section 4.1 are identically zero, which implies that the approximation is given by the first order perturbation around the steady state. The initial size of the validation sample was set to $N = 4000$, and increased to $N^{\max} = 10000$ during the iterations. The initial size of the quadrature grid was set to 50, and increased to 200.

4.5. Parameter restrictions

There is no theoretical guarantee that the computations converge. In the concrete example of Section 3, divergence can occur when the parameters imply that the marginal propensity to save (for at least some cohorts) exceeds unity. To prevent this, it is necessary to impose restrictions on the parameters.

These parameter restrictions are not strictly enforced; instead, violations are penalized. In particular, I impose penalties for the following events:

- A decrease in the slope of the spline for the marginal propensity to save between knot points, as suggested by economic theory.
- The sum of the coefficient on saving with respect to own wealth and the marginal propensity to save out of cash-on-hand exceeding 0.99.
- To ensure that the spline's knot points are strictly increasing and to slightly favor equally spaced knot points, I penalize
  - non-increasing (i.e., decreasing) knot points, and
  - the second difference of knot points (with a minimal penalty).

Moreover, for reasons of parsimony, the power transformation is applied only to aggregate variables, not to cohort-specific state or decision variables.

4.6. Computational efficiency

Parallelizing the algorithm is straightforward, and computation speed benefits substantially from additional cores. The key computations are performed at many grid points in arbitrary order, allowing each core to process a subset of states for which residuals are evaluated. Further efficiency gains may be possible in the computation of derivatives. The main bottleneck is the calculation of the Jacobian of all residuals with respect to all parameters being trained. This is done using forward-mode automatic differentiation, but existing packages are not always sufficiently efficient. Several aspects are crucial for fast computation of derivatives:

- Gradients should be stored as sparse vectors.
- Dot products and matrix–vector multiplications involving variables with sparse derivatives require efficient sparse-data handling.
- Jacobian computation can be accelerated by reducing dimensionality through intermediate variables. If $y = G(F(x))$ and $F(x)$ has lower dimension than x, introduce $z = F(x)$ as a new vector of independent variables and apply the chain rule $\frac{\partial y}{\partial x} = \frac{\partial y}{\partial z}\frac{\partial z}{\partial x}$. This is useful, for example, when residuals depend on model variables Y whose dimension is much smaller than that of the parameter vector on which they depend.

These components were hand-coded.

Calculations were done on a standard desktop equipped with a Ryzen 7 3700X processor (8 cores) and 16 GB of memory. Computing times are reported in Section 5.5.

5. Numerical results

5.1. Parameter values

The parameters of the economic model are reported in Table 1. Standard values were used whenever available. The standard deviations of the shocks were not chosen to match empirical data, but rather to generate large variation in the state variables (cf. Table 2), thereby making the model's nonlinearities more relevant. The cohort-specific shock to labor productivity was set sufficiently large so that even young cohorts occasionally save. The time endowment of households is set so as to generate a hump-shaped profile with a maximum at around age 50. [7]

5.2. Nonlinearity in impulse responses

Fig. 2 shows impulse response functions that illustrate the effects of nonlinearity in the model. The upper four panels compare results obtained from three solution methods. "Linear" refers to the standard linearization around the deterministic steady state. "Semi-Linear" denotes the solution in which the directly approximated variables $\hat{C}_t$, p_t^B, R_t^F, and $b_{j,t}$ follow the same linear approximation

[7] Following Hansen (1993), I set

$$H_j = 1 + \left(0.044929 * (age(j) - 20) - 0.000741 * (age(j) - 20)^2\right)/0.732586$$

where $age(j) \equiv 20 + (j - 0.5)$ for $j = 1, \ldots, 40$.

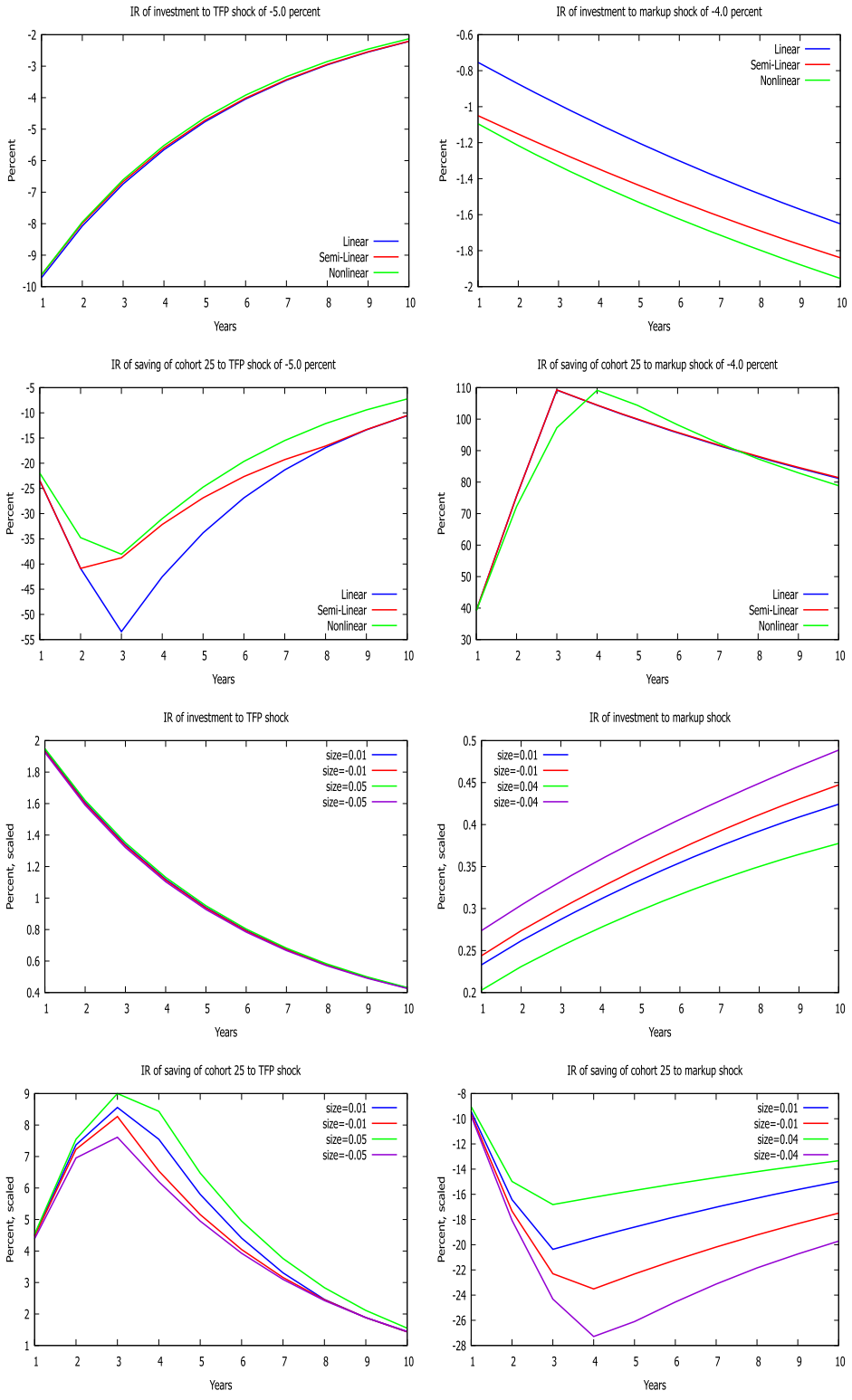


Fig. 2. Impulse responses.

Table 1
Parameter values.

Parameter	Symbol	Value		
Preferences				
Discount factor Capitalists	$\hat{\beta}$	0.9800		
Discount factor Workers	β	0.9600		
Risk aversion	γ	2.0000		
Firms				
Average depreciation rate	$\bar{\delta}$	0.100		
Capital share	α	0.3600		
Quadratic adj.cost	η_I	10.0000		
Average markup	$\bar{\mu}$	1.1667		
Fraction fixed costs	$\overline{FC}$	0.5		
Finance				
Bond coupon	κ	0.0204		
Average tax rate	$\bar{\tau}$	0.2200		
Fraction of bonds maturing	ξ	0.200		
Shocks		ρ	σ_{shock}	$\sigma_{variable}$
TFP	ϵ_z	0.8000	0.01200	0.020
Productivity profile	ϵ_ψ	0.5000	0.01732	0.020
Markup	ϵ_μ	0.9333	0.00720	0.020
Investment efficiency	ϵ_ϕ	0.8000	0.03000	0.050
Discount factor	ϵ_β	0.9333	0.00072	0.002
Depreciation rate	ϵ_δ		0.00500	
Cohort labor productivity	ζ_j		0.05000	

(but satisfy the borrowing constraints), while all other variables are computed to satisfy the exact nonlinear model equations (cf. Appendix A). This solution shares the same deterministic steady state as the linear one. "Nonlinear" refers to the fully nonlinear solution obtained via the hybrid approach.[8]

The figure displays the responses of investment and the savings of Cohort 25 (age 45) to TFP and markup shocks. While the investment response to TFP shocks is very similar across the three solution methods, markup shocks generate large redistributive effects, which in turn induce strong nonlinear behavior and substantial differences even between the linear and semi-linear approaches. Because Cohort 25 is close to the borrowing limit in the deterministic steady state, its responses exhibit pronounced nonlinearities, especially in reaction to the markup shock. Note that, by construction, the impact effect on cohort savings is identical for the linear and semi-linear solutions, because cohort savings is a directly approximated variable, unlike investment. Thereafter, general equilibrium effects drive a wedge between the linear and semi-linear solutions.

The lower four panels display impulse responses of investment and cohort savings from the nonlinear solution for shocks of varying magnitudes. Each response is normalized by the corresponding shock size, so the lines would coincide if the model were linear in the shock size. Again, sizeable nonlinearities emerge, particularly in the case of a markup shock.

5.3. The saving behavior of worker cohorts

Fig. 3 plots various quantiles of each cohort's asset distribution over a long simulation of the nonlinear model and compares them to asset holdings in the deterministic steady state. Workers accumulate substantial assets only from middle age onward and begin to decumulate during retirement. Asset levels exhibit considerable dispersion throughout the simulation. The gap between the median (green line) and the deterministic steady state (black line) reflects the equilibrium effect of precautionary behavior. This effect is small and visible primarily for the younger cohorts, because the reduction in the wage rate and the increase in average pension benefits offset the direct precautionary motive, as will be shown in Section 5.4.

Fig. 4 shows, for each cohort, the fraction of time that households are at the borrowing constraint during a long simulation, under both the semi-linear and the nonlinear solutions. In the semi-linear solution, fractions are in the interior only for cohorts that are close to the constraint in the steady state. In the nonlinear model, however, even young cohorts face a significant probability of being unconstrained, owing to the large cohort-specific productivity shock. Since retired households face little uncertainty, their borrowing constraints almost never bind.

5.4. Business cycle statistics

Table 2 reports business-cycle statistics: sample means, standard deviations expressed as a percent of the mean, autocorrelations, and correlations with output Y. For each statistic, the table first lists the value produced by the linear solution, followed by the value

[8] The nonlinear solution has a different deterministic steady state. To make its impulse responses comparable to those of the other two solution methods, they are computed as the difference between two paths starting from the steady state of the linearized model, one following the initial impulse, and the other in the absence of a shock.

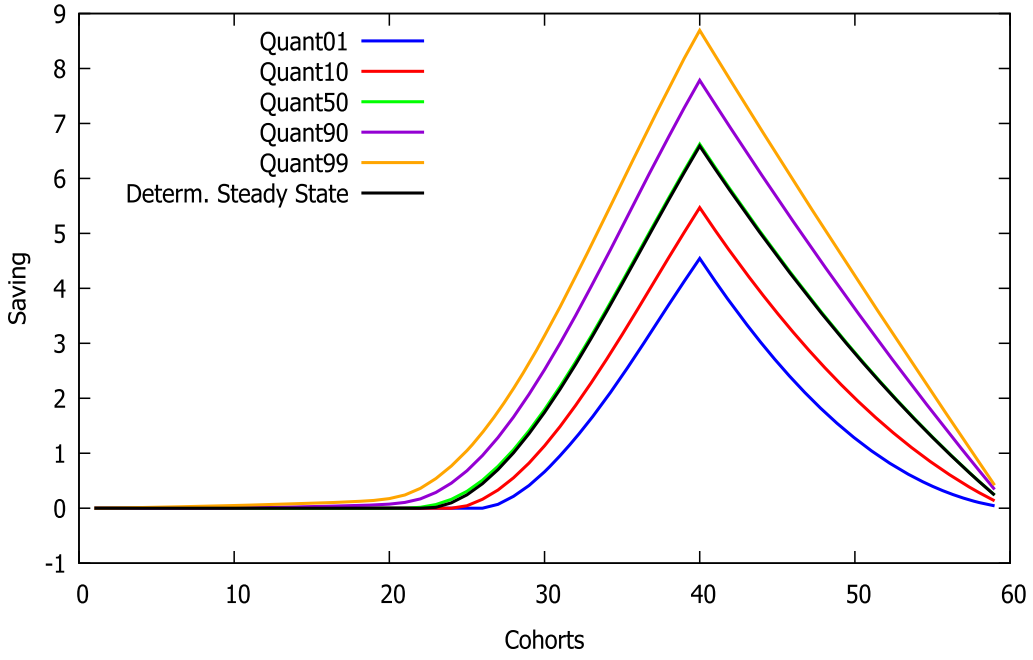


Fig. 3. Distribution of cohort savings in long simulation.

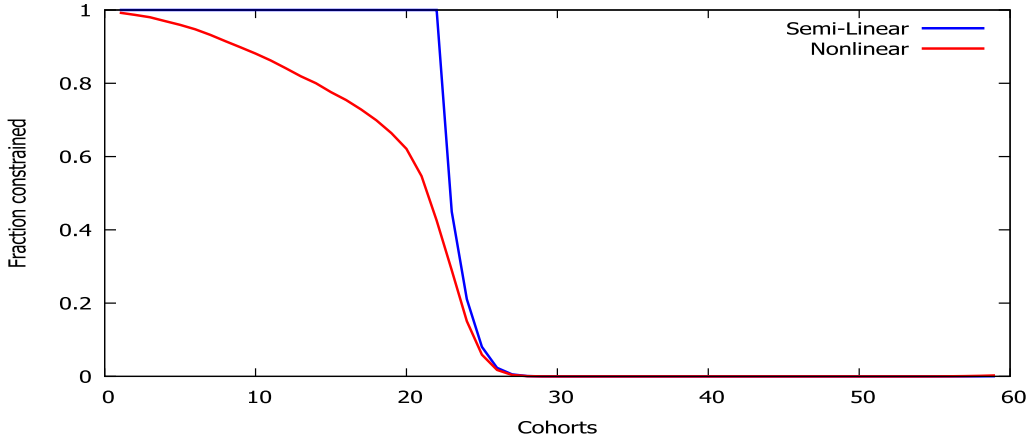


Fig. 4. Fraction of households at borrowing constraint in long simulation.

of the nonlinear solution relative to the linear one. All values are computed from a simulation of 100,000 periods, without detrending. The upper part of the table refers to the main calibration.

A key result is the increase in both the aggregate capital stock K and the value of aggregate workers' bond holdings $B \cdot p^B$ in the nonlinear solution, driven by precautionary saving. The resulting rise in wages and decline in asset returns lead to a redistribution from capitalists to workers. This is reflected in a slight increase in worker consumption and a decrease in capitalist consumption. The rate of return both on firms (R^F) and on bonds (R^B) decreases. This and the rise in pension benefits P reduce the saving motive of workers, partially offsetting the precautionary saving motive.

Both capital and aggregate bond holdings are very volatile and persistent, and even more so for the bond holdings of individual cohorts. This is consistent with the large differences between quantiles of saving levels in Fig. 3. Notably, capitalist consumption is also highly volatile, reflecting that capitalists live solely off capital income and hold a leveraged position in the aggregate capital stock. The increase in capital in the nonlinear solution helps stabilize their consumption. GDP, investment, and most other variables are also slightly more stable in the nonlinear solution, which again appears to result from precautionary behavior captured by the nonlinear but not by the linear solution.

Table 2
Business Cycle Statistics.

	Mean		Std. dev.		Autocorr.		Corr.(Y)	
	Lin	NL/Lin	Lin	NL/Lin	Lin	NL/Lin	Lin	NL/Lin
Benchmark case $FC = 0.5$								
Y	3.068	1.0027	6.28	0.984	0.90	0.998	1.00	1.000
C	1.848	1.0030	4.52	1.003	0.93	1.000	0.82	1.002
$\hat{C}$	0.279	0.9861	3.92	0.952	0.96	1.000	0.43	1.039
I	0.941	1.0071	5.83	0.980	0.83	0.996	0.89	0.998
L	1.946	1.0000	1.25	1.008	0.08	1.000	0.18	1.007
w	0.931	1.0028	3.43	0.996	0.91	0.998	0.84	0.997
P	1.196	1.0028	3.93	0.995	0.90	0.998	0.88	0.998
d	0.315	0.9893	5.65	0.964	0.81	0.996	0.06	1.277
Q	1.000	1.0027	5.07	1.026	0.83	1.001	-0.22	0.989
r^K	0.120	0.9971	1.62	0.963	0.92	0.999	-0.37	0.977
K	9.409	1.0080	18.95	0.977	0.99	1.000	0.82	0.997
$B \cdot p^B$	1.756	1.0124	24.38	0.811	0.99	0.999	0.23	0.927
R^F	1.021	0.9998	3.53	0.967	-0.07	0.982	0.13	1.027
R^B	1.021	0.9993	1.94	0.991	-0.00	1.252	0.06	0.989
c_{10}	1.616	1.0025	7.89	0.928	0.32	1.057	0.49	1.026
c_{30}	2.343	1.0067	4.65	1.064	0.85	0.995	0.88	0.994
c_{50}	1.595	1.0025	6.40	0.958	0.90	1.017	0.46	0.996
c_{60}	1.439	1.0021	8.13	0.975	0.93	1.008	0.40	1.005
b_{30}	1.743	1.0374	44.98	0.818	0.71	0.967	0.22	0.920
b_{50}	2.817	0.9982	38.37	0.962	0.91	1.022	0.05	0.697
Zero-profit case $FC = 1$								
Y	2.832	1.0108	7.62	0.961	0.93	0.996	1.00	1.000
C	1.848	1.0071	4.63	1.003	0.93	0.998	0.83	1.016
$\hat{C}$	0.043	0.8122	3.83	0.631	0.98	0.999	0.59	1.034
I	0.941	1.0270	7.99	0.946	0.91	0.993	0.92	0.997
w	0.931	1.0095	3.80	0.965	0.93	0.995	0.88	0.992
P	1.196	1.0094	4.34	0.966	0.92	0.995	0.90	0.994
d	0.079	0.8481	10.48	0.841	0.82	0.977	-0.28	0.901
Q	1.000	1.0029	5.13	1.034	0.83	1.000	-0.17	0.992
r^K	0.120	0.9862	2.07	0.924	0.95	0.999	-0.57	0.962
K	9.409	1.0277	25.24	0.950	0.99	1.000	0.87	0.991
$B \cdot p^B$	1.756	0.9372	33.26	0.646	0.99	1.000	-0.19	1.115
R^F	1.021	0.9992	3.67	0.933	-0.03	0.979	0.03	1.230
R^B	1.021	0.9975	2.05	0.984	0.05	0.787	-0.07	0.833

Table 3
Accuracy of capitalists' decisions.

	RMSE				Quantile 95			
	Semi-lin	Nonlin1	Nonlin2	Nonlin3	Semi-lin	Nonlin1	Nonlin2	Nonlin3
Cons. Cap.	0.070	0.033	0.031	0.031	0.115	0.065	0.062	0.061
Price Bond	0.057	0.022	0.019	0.019	0.100	0.047	0.040	0.038
Return Firms	0.621	0.068	0.061	0.060	1.378	0.138	0.122	0.122

The second part of the table refers to an alternative calibration in which the fixed cost of production equals monopoly rents in steady state ($\overline{FC} = 1$ in Eq. (21)). Without monopoly rents, capitalist consumption falls to less than 2 percent of output on average and becomes even more volatile. This increases the capitalists' precautionary saving motive but lowers the interest rate even more. In this case, the combined effect of the lower interest rate and the higher benefit rate reduces aggregate worker saving, more than offsetting the precautionary saving motive.

5.5. Measuring accuracy

The model features 65 states and 46 shocks, and the nonlinear solution uses only 2262 free parameters for 62 approximated functions, on average about 36.5 parameters per function. The question is what level of accuracy can be achieved with such an extremely parsimonious parameterization.

Table 3 reports results for the decisions of capitalists, and Fig. 5 shows results for worker saving by cohort. Accuracy is measured in terms of the residuals of the Euler equations (28)–(31), using the root mean squared error (RMSE) and the quantiles of the absolute error. Residuals are scaled as fractions of the corresponding bond price, firm value, and consumption value, respectively. In addition to the semi-linear solution, three versions of the nonlinear solution are compared, obtained after approximately 25 minutes,

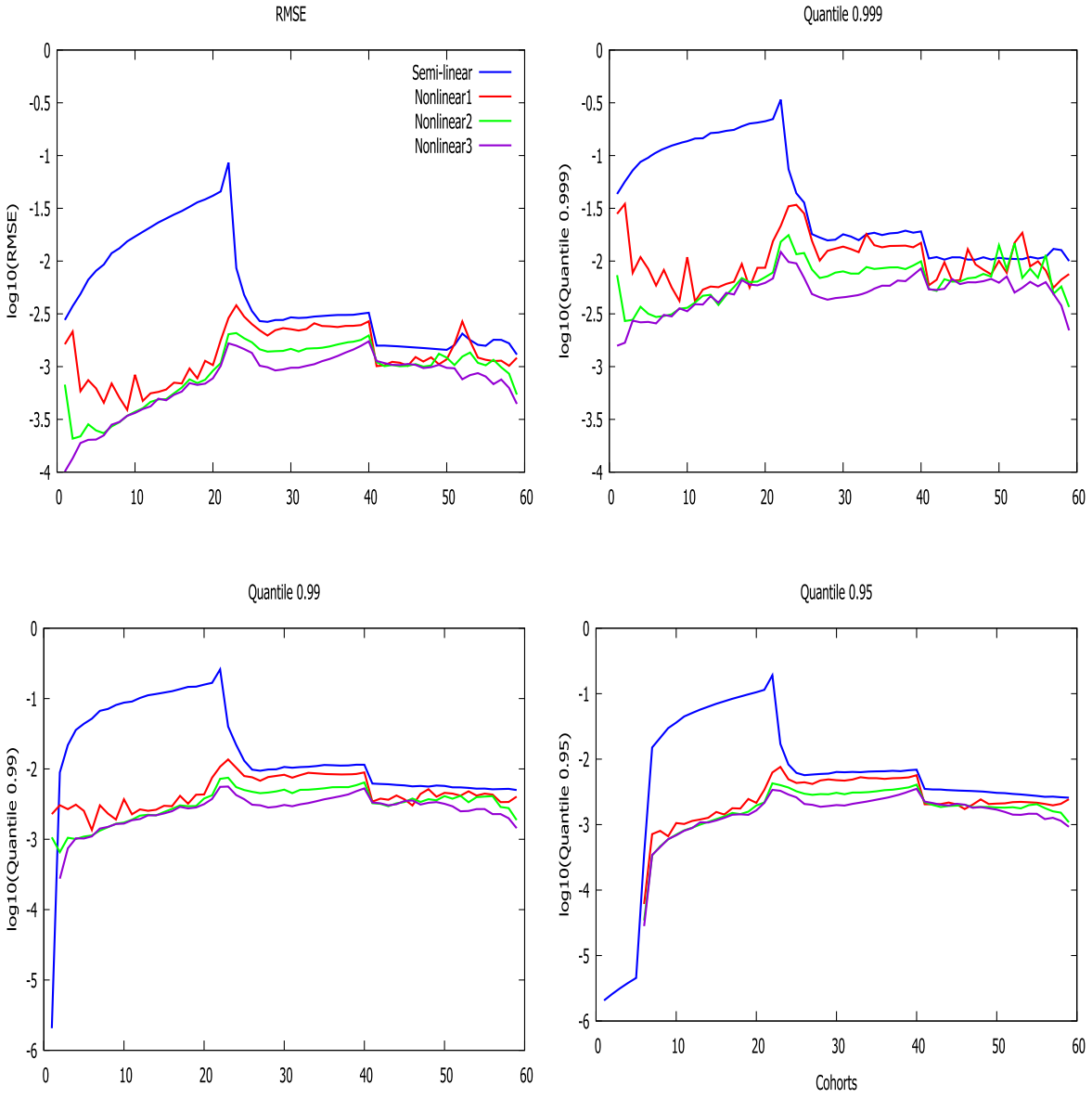


Fig. 5. Accuracy of worker decisions.

3 hours, and 7.5 hours of computation time, respectively. Beyond this point, improvements in accuracy are small. The most precise solution, denoted "Nonlinear 3," is also the one used for the results reported in the preceding subsections. Accuracy is evaluated using every 10th observation along an independently generated simulation of 200,000 periods. Moreover, the accuracy check employs a quadrature grid of shock vectors of size $n = 200$, whereas the training relied on smaller grids of 50 or 100 points (cf. Section 4.2).

The reduction in the capitalists' Euler residuals delivered by the best nonlinear solution relative to the semi-linear one is by a factor between 2 and 10, both in terms of RMSE and in the 95th percentile of the absolute error. For the saving decisions of workers, improvements of nearly two orders of magnitude in RMSE occur for the cohorts that frequently switch between saving and being borrowing-constrained. For workers who save in almost every period, the semi-linear solution already attains reasonable accuracy, and the nonlinear solution improves accuracy by roughly a factor of three. The quantiles exhibit a similar pattern as the RMSE. The RMSE averaged across all cohorts is approximately 0.0009, equivalent to 0.09 percent of consumption, and reaches about 0.17 percent for cohorts 22 and 40.

The nonlinear solution method achieves remarkably high accuracy despite its parsimonious parameterization, delivering substantial improvements over the semi-linear benchmark across both aggregate and cohort-level decisions.

6. Discussion and conclusions

The aim of this paper is to present a hybrid method that combines elements of deep learning with traditional solution methods for DSGE models. The approach was tested on a large OLG model that was designed so as to be difficult to solve nonlinearly. The numerical results are encouraging. The method can handle the occasionally binding borrowing constraints of households, and uncovers the precautionary effects of large shocks. Although the model has 65 states and 46 shocks and the approximation is very parsimoniously parameterized, the hybrid method improves the accuracy of the solution, measured as the RMSE of the Euler equations, by one to two orders of magnitude compared to the semi-linear approximation. Scaled in terms of percentage consumption deviations, the RMSE is on average below 0.1 percent over a very wide range of the state space. To improve accuracy beyond the level achieved in Section 5, it will be necessary to increase the number of parameters in the hybrid approach, for example by making the parameters of the spline approximation depend on the states in a more general way.

The algorithm presented in this paper should not be regarded as a fixed template, but rather as a flexible framework that can be modified and extended along multiple dimensions to exploit a priori information about the structure of the economic model and the expected properties of its solution. This flexibility leaves considerable scope for researcher judgment and requires a solid understanding of model characteristics. At present, implementation demands a higher coding effort than standard solution environments such as Dynare. Looking forward, an important objective will be to reduce the need for manual intervention in the solution process. This could be achieved by automating key algorithmic choices through advanced AI tools that analyze fundamental model properties, such as convexity, smoothness etc., and adapt the numerical strategy accordingly.

Finally, it would be very useful to have a more rigorous evaluation, ideally by a “horse race” similar to Den Haan (2010), where different researchers apply their favorite method to an interesting model of sufficient complexity, and accuracy and computational speed can be compared.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used ChatGPT 5.1 for literature search and in order to check for spelling, grammar and some stylistic improvements. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Declaration of competing interest

The author declares that he has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Computing all equilibrium variables

- Get $\omega_{j,t}$ from (10)
- Get $\delta_t = \bar{\delta} + \epsilon_{\delta,t}$
- Get $\overline{K}_t = K_{t-1}(1-\delta_t)$
- Get L_t from (11)
- Get w_t from (19)
- Get rK_t from (20)
- Get R_t^B from (12)
- Get $\hat{Y}_t$ from (18)
- Get Y_t from (21)
- Get P_t from (14)
- Get d_t from (23)
- Get p_t^F from (25)
- Get I_t from (22)
- Get $c_{j,t}$ from (13); set $b_{J,t} = 0$ for the last cohort.
- Get C_t from (26)
- Get Q_t from (16)
- Get $Q_{o,t}$ from (17)

References

Adjemian, S., Juillard, M., Karamé, F., Mutschler, W., Pfeifer, J., Ratto, M., Rion, N., Villemot, S., 2024. Dynare: Reference Manual, Version 6. Dynare Working Papers 80. CEPREMAP.

Auclert, A., Bardóczy, B., Rognlie, M., Straub, L., 2021. Using the sequence-space jacobian to solve and estimate heterogeneous-agent models. Econometrica 89 (5), 2375–2408. https://doi.org/10.3982/ECTA17434

Azinovic, M., Gaegauf, L., Scheidegger, S., 2022. Deep equilibrium nets. Int. Econ. Rev. 63 (4), 1471–1525. https://doi.org/10.1111/iere.12575

Azinovic, M., Zemlicka, J., 2024. Intergenerational consequences of rare disasters. http://dx.doi.org/10.2139/ssrn.4386477.

Belkin, M., Hsu, D., Ma, S., Mandal, S., 2019. Reconciling modern machine-learning practice and the classical bias-variance trade-off. Proc. Natl. Acad. Sci. 116 (32), 15849–15854. https://doi.org/10.1073/pnas.1903070116

Bellavia, S., Gratton, S., Riccietti, E., 2018. A Levenberg–Marquardt method for large nonlinear least-squares problems with dynamic accuracy in functions and gradients. Numer. Math. 140 (3), 791–824. https://doi.org/10.1007/s00211-018-0977-z
Bhandari, A., Bourany, T., Evans, D., Golosov, M., 2023. A Perturbational Approach for Approximating Heterogeneous Agent Models. NBER Working Papers 31744. National Bureau of Economic Research, Inc. https://ideas.repec.org/p/nbr/nberwo/31744.html.
Bilal, A., 2023. Solving heterogeneous agent models with the master equation. Manuscript, Harvard. https://doi.org/10.3386/w31103.
Boppart, T., Krusell, P., Mitman, K., 2018. Exploiting MIT shocks in heterogeneous-agent economies: the impulse response as a numerical derivative. J. Econ. Dyn. Control 89 (C), 68–92. https://doi.org/10.1016/j.jedc.2018.01.002.
Brumm, J., Scheidegger, S., 2017. Using adaptive sparse grids to solve high-Dimensional dynamic models. Econometrica 85, 1575–1612. https://doi.org/10.3982/ECTA12216.
Brumm, J., Scheidegger, S., Schenk, O., 2018. Multi-dimensional dynamic programming with adaptive sparse grids. Comput. Econ. 52 (1), 1–34. https://doi.org/10.1007/s10614-017-9705-7
Cao, D., Luo, W., Nie, G., 2023. Global DSGE models. Rev. Econ. Dyn. 51, 199–225. https://doi.org/10.1016/j.red.2023.01.001
Curtis, F.E., Nocedal, J., Wächter, A., 2009. A matrix-free algorithm for equality constrained optimization problems with rank-deficient jacobians. SIAM J. Optim. 20, 1224–1249. https://doi.org/10.1137/08072471X
Den Haan, W.J., 2010. Comparison of solutions to the incomplete markets model with aggregate uncertainty. J. Econ. Dyn. Control 34 (1), 4–27. https://doi.org/10.1016/j.jedc.2008.12.010.
Den Haan, W.J., Marcet, A., 1990. Solving the stochastic growth model by parameterizing expectations. J. Bus. Econ. Stat. 8 (1), 31–34. https://doi.org/10.1080/07350015.1990.10509770.
Dick, J., Pillichshammer, F., 2010. Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration. Cambridge University Press. https://doi.org/10.1017/CBO9780511761188.
E, W., Han, J., Jentzen, A., 2017. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Commun. Math. Stat. 5 (4), 349–380. https://doi.org/10.1007/s40304-017-0117-6
Fan, F., Xiong, J., Wang, G., 2020. Universal approximation with quadratic deep networks. Neural Netw. 124, 383–392. https://doi.org/10.1016/j.neunet.2020.01.007
Fernández-Villaverde, J., 2025. Deep Learning for Solving Economic Models. NBER Working Paper w34250. National Bureau of Economic Research. https://doi.org/10.2139/ssrn.5487075
Fernandez-Villaverde, J., Nuno, G., Perla, J., 2024. Taming the Curse of Dimensionality: Quantitative Economics with Deep Learning. NBER Working Papers 33117. National Bureau of Economic Research, Inc. https://ideas.repec.org/p/nbr/nberwo/33117.html.
Fernandez-Villaverde, J., Rubio-Ramirez, J.F., 2006. Solving DSGE models with perturbation methods and a change of variables. J. Econ. Dyn. Control 30 (12), 2509–2531. https://doi.org/10.1016/j.jedc.2005.07.009.
Han, J., Jentzen, A., E, W., 2018. Solving high-dimensional partial differential equations using deep learning. Proc. Natl. Acad. Sci. 115 (34), 8505–8510. https://doi.org/10.1073/pnas.1718942115
Han, J., Yang, Y., E, W., 2026. DeepHAM: A global solution method for heterogeneous agent models with aggregate shocks. Quant. Econ. 17, 297–341. https://doi.org/10.3982/qe2190.
Hansen, G., 1993. The cyclical and secular behavior of the labor input: comparing efficiency units and hours worked. J. Appl. Econom. 8, 71–80. https://doi.org/10.1002/jae.3950080106.
Judd, K.L., 1992. Projection methods for solving aggregate growth models. J. Econ. Theory 58 (2), 410–452. https://doi.org/10.1016/0022-0531(92)90061-L.
Judd, K.L., 2002. Perturbation methods with nonlinear changes of variables. Manuscript, Hoover Institution.
Judd, K.L., Guu, S.-M., 1993. Perturbation solution methods for economic growth models. In: Varian, H.R. (Ed.), Economic and Financial Modeling with Mathematica. Springer, New York. https://doi.org/10.1007/978-1-4757-2281-9_4.
Judd, K.L., Maliar, L., Maliar, S., 2011. Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models. Quant. Econ. 2 (2), 173–210. https://ideas.repec.org/a/ecm/quante/v2y2011i2p173-210.html.
Judd, K.L., Maliar, L., Maliar, S., 2012. Merging Simulation and Projection Approaches to Solve High-Dimensional Problems. NBER Working Papers 18501. National Bureau of Economic Research, Inc. https://doi.org/10.3386/w18501.
Judd, K.L., Maliar, L., Maliar, S., Valero, R., 2014. Smolyak method for solving dynamic economic models: lagrange interpolation, anisotropic grid and adaptive domain. J. Econ. Dyn. Control 44 (C), 92–123. https://ideas.repec.org/a/eee/dyncon/v44y2014icp92-123.html.
Juillard, M., 1996. Dynare : a program for the resolution and simulation of dynamic models with forward variables through the use of a relaxation algorithm. CEPREMAP Working Papers (Couverture Orange) 9602. CEPREMAP. https://ideas.repec.org/p/cpm/cepmap/9602.html.
Kahou, M.E., Fernández-Villaverde, J., Gómez-Cardona, S., Perla, J., Rosa, J., 2024. Spooky Boundaries at a Distance: Inductive Bias, Dynamic Models, and Behavioral Macro. CESifo Working Paper Series 11292. CESifo. https://ideas.repec.org/p/ces/ceswps/_11292.html.
Kahou, M.E., Fernández-Villaverde, J., Perla, J., Sood, A., 2021. Exploiting Symmetry in High-Dimensional Dynamic Programming. CESifo Working Paper Series 9161. CESifo. https://ideas.repec.org/p/ces/ceswps/_9161.html.
Kandel, S., Maddali, S., Nashed, Y. S.G., Hruszkewycz, S.O., Jacobsen, C., Allain, M., 2021. A matrix-free Levenberg-Marquardt algorithm for efficient ptychographic phase retrieval. arXiv:2103.01767.
Kingma, D.P., Ba, J., 2017. Adam: A method for stochastic optimization. arXiv:1412.6980.
Krueger, D., Kübler, F., 2004. Computing equilibrium in OLG models with stochastic production. J. Econ. Dyn. Control 28 (7), 1411–1436. https://doi.org/10.1016/S0165-1889(03)00111-8
Krueger, D., Kübler, F., 2006. Pareto-improving social security reform when financial markets are incomplete!? Am. Econ. Rev. 96 (3), 737–755. https://doi.org/10.1257/aer.96.3.737
Krusell, P., Smith, A.A., 1998. Income and wealth heterogeneity in the macroeconomy. J. Polit. Econ. 106 (5), 867–896. https://doi.org/10.1086/250034.
Maliar, L., Maliar, S., Winant, P., 2021. Deep learning for solving dynamic economic models. J. Monet. Econ. 122, 76–101. https://doi.org/10.1016/j.jmoneco.2021.07.004
Martens, J., 2010. Deep learning via hessian-free optimization. In: Proceedings of the 27th International Conference on Machine Learning (ICML). Omnipress, Haifa, Israel, pp. 735–742. https://www.cs.toronto.edu/~jmartens/docs/Deep_HessianFree.pdf.
Miranda, M.J., Fackler, P.L., 2002. Applied Computational Economics and Finance. MIT Press.
Payne, J., Rebei, A., Yang, Y., 2024. Deep learning for search and matching models. Swiss Finance Institute Research Paper No. 25-05. https://doi.org/10.2139/ssrn.4768566.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045
Reiter, M., 2009. Solving heterogeneous agent models by projection and perturbation. J. Econ. Dyn. Control 33 (3), 649–665. https://doi.org/10.1016/j.jedc.2008.08.010.
Reiter, M., 2010a. Approximate and almost-exact aggregation in dynamic stochastic heterogeneous-agent models. IHS Working Paper 258. https://irihs.ihs.ac.at/id/eprint/2021/.
Reiter, M., 2010b. Solving the incomplete markets model with aggregate uncertainty by backward induction. J. Econ. Dyn. Control 34 (1), 28–35.
Reiter, M., 2023. Solving heterogeneous agent models by second-order perturbation. IHS Working Paper. http://irihs.ihs.ac.at/id/eprint/6641/.
Shao, W., Fan, J., 2025. Convergence of a stochastic variance reduced Levenberg-Marquardt method. Comput. Optim. Appl. 90 (2), 417–444.
Sirignano, J., Spiliopoulos, K., 2018. Dgm: a deep learning algorithm for solving partial differential equations. J. Comput. Phys. 375, 1339–1364. https://doi.org/10.1016/j.jcp.2018.08.029