HANK on Speed:
Robust Nonlinear Solutions using Automatic Differentiation

Gregor Boehl

University of Bonn

November 7, 2024

Abstract

Building on automatic differentiation, I propose a solution method for heterogeneous agents models with many aggregate equations which allows to account even for strong nonlinearities. A powerful open source reference implementation is provided which typically solves the canonical HANK model within a few seconds, including the nonlinear transition dynamics of the complete distribution. I study a permanent shift in redistribution policy in a medium-scale two-asset HANK model featuring many aggregate frictions. Since firms wish to deplete their capital stock, the transition path is characterized by a long-lasting deflationary episode, which may be intensified by the interest rate lower bound.

Keywords: Heterogeneous agents, Computational methods, General equilibrium, Nonlinear systems
JEL: C63, C32, E52, E47

1 Introduction

Over the last decade, heterogeneous agent models have emerged as an important new class of macroeconomic models.[1] They allow to account for the heterogeneity of agents in their wealth, abilities, or other characteristics, thereby permitting economists to better understand the role of different groups of agents for the economy. Heterogeneous agent models provide more accurate predictions about how different agents respond to changes in the economy, such as shifts in policy, changes in market conditions, or unexpected shocks, and in turn give insight on how these changes impact on the distribution of agents. This allows these models to capture important new economic channels, which is useful for designing policies that are more effective and equitable, and provide a more nuanced and realistic view of the economy, ultimately leading to a deeper understanding of how the economy works.

This paper shows how to solve and simulate heterogeneous agent models with strong nonlinearities and many aggregate equations. The current frontier of macroeconomic research assigns a pivotal role to

⋆Address: Institute for Macroeconomics and Econometrics, University of Bonn, Adenauerallee 24-42, 53113 Bonn, Germany. An earlier version of this draft was circulating under the title "Robust Nonlinear Transition Dynamics in HANK". I am grateful to Christian Bayer, Flora Budianto, Marten Hillebrand, Keith Kuester, Alexander Meyer-Gohde, an anonymous referee, and participants of several conferences and seminars for discussions and helpful comments on the contents of this paper. I gratefully acknowledge financial support by the Deutsche Forschungsgemeinschaft (DFG) under CRC-TR 224 (projects C01 and C05) and under project number 441540692.

Email address: gboehl@uni-bonn.de

URL: https://gregorboehl.com

[1] Seminal applications include, e.g., Kaplan et al. (2018); Gornemann et al. (2016); McKay et al. (2016); Auclert and Rognlie (2017, 2018); Ahn et al. (2018); Auclert (2019); De Ferra et al. (2020); Hagedorn et al. (2019); Krueger et al. (2015); Bayer et al. (2020, 2023) and Achdou et al. (2022).

such nonlinearities. Important examples include occasionally binding constraints (e.g. the lower bound on nominal rates), asymmetries (e.g. downwards nominal wage rigidity or asymmetric disaggregated pricing decisions), and *aggregate* nonlinearities such as severe financial frictions or labor markets with search-and-matching.[2] The literature stresses that such nonlinearities are likely to play a central role in the propagation of policies and economic shocks through the economy, or to different groups of economic agents. However, due to the high complexity of heterogeneous agent models, current advances have focussed on methods that involve linearization techniques, thereby discarding the nonlinear features of the model. Linearization was in particular necessary for models that feature many aggregate equations. This paper fills this gap by finding the perfect foresight solution of heterogeneous agent models while fully accounting for the underlying nonlinearities.[3]

As the basis of my methodological contribution I develop a modular representation of nonlinear heterogeneous agent models. This representation allows to combine the disaggregated decisions of the cross-section of agents with an arbitrary number of – potentially highly nonlinear – aggregated equilibrium conditions. I then show that the model's steady state and the corresponding stationary distribution can be identified by means of a robust and generic routine based on the technique of *automatic differentiation* (AD).[4] At the heart of the paper, I then develop a numerical method that, based on Newton's method, solves for the perfect foresight path. The solution is represented in sequence space, i.e. represents the truncated trajectory up to a distant horizon, and fully accounts for the nonlinearity of the model. More precisely, I show that the sub-problem of solving the system of linear equations associated with each Newton-step can be tackled efficiently using a novel iterative procedure based on Jacobian-vector-products (JVPs). The efficiency comes from explicitly exploiting several features that are specific to economic models, and by leveraging that JVPs can be evaluated quickly using AD. The method not only allows to solve for the nonlinear aggregate dynamics, but also to obtain the full nonlinear transition sequence of the cross-sectional distribution between different steady states.

Along with the method, I provide a high-level reference implementation that I propose as a blueprint of best-practices for the provision of codes and numerical routines in economics: the *econpizza* package. The package consequently follows the open-source paradigm and comes with an extensive online documentation.[5] A core concept is the strict separation of economic model (provided by the user), underlying simulation code (provided by the implementation), and the analysis of the results (left to the user). To achieve this, I introduce a generic and standardized modelling syntax for the representation of heterogeneous agent models, which levers the modular structure of these models: the aggregate equations on one side, and the disaggregated decision problem on the other. The implementation further showcases how to provide generic, reusable code and to adhere to the principles of modern software development. As I argue, this helps to make these methods accessible to a larger group of researchers while allowing and fostering continuous progress in the field.

I apply the proposed methods to a fundamental economics question: the macroeconomic costs and benefits of redistributive policy. I propose a medium-scale heterogeneous agent model which features the full set

[2] E.g., see Gust et al. (2017) for the role of the interest rate lower bound on the empirical dynamics and Lindé and Trabandt (2018) for the effects of nonlinearities on fiscal multipliers. Petrosky-Nadeau et al. (2018) document that nonlinear labor search frictions can induce endogenous disasters in otherwise standard models. Klenow and Kryvtsov (2008) study the role of asymmetric state-dependent heterogeneous firms pricing on inflation dynamics.

[3] While this paper focuses on models with heterogeneous households, the presented methods can equally well be applied to models with heterogeneity across other types of agents, such as firms or banks.

[4] Automatic differentiation is a computational technique for efficiently computing the derivatives of a function without having to write out the derivative by hand or use numerical methods. Section 3 gives a short primer on AD and its virtues.

[5] The documentation can be found at `https://econpizza.readthedocs.io`.

of frictions of contemporary DSGE models. Agents may hold two classes of assets and face idiosyncratic income risk. This gives rise to a precautionary savings motive and a non-trivial distribution of assets. If transfers are financed by labor income taxes, the distortionary effects of these taxes decreases output significantly relative to an equilibrium without transfers. My methodology allows to study the transition dynamics of the complete cross-sectional distribution between the two equilibria. As I show, these dynamics are strongly deflationary in the short and medium term due to the depletion of the capital stock, and their recessionary nature may be intensified by severe nonlinearities such as a binding lower bound on the nominal interest rate or downwards nominal wage rigidity.

Literature

The idea of solving representative agent models in sequence space dates back to the seminal paper of Fair and Taylor (1980). Building on this idea, Laffargue (1990) and Juillard et al. (1996) show that the block tridiagonal structure of the sequence space Jacobian of representative agent models can be exploited to efficiently solve the system of linear equations during each single Newton step. Juillard et al. (1998) further show that Newton-based methods are advantageous over first-order algorithms in terms of robustness and speed. This type of algorithm (but using AD to calculate the single Jacobian blocks) is also provided in the reference implementation for representative agent models. Appendix D contains details. Unfortunately, the sequence space Jacobian of *heterogeneous* agent models does not inherit such handy block tridiagonal structure, nor can we safely ex-ante assume its sparsity.[6] For this reason I here propose a method that uses the existing knowledge about the dynamic system of heterogeneous agent models – manifested in the steady state sequence space Jacobian – to efficiently solve the system of linear equations during each Newton step iteratively.

The sequence space approach to heterogeneous agent models was introduced by Boppart et al. (2018) and further advanced by Auclert et al. (2021). Both methods focus on linearized solutions in the direct neighborhood of the steady state. Boppart et al. (2018) propose a small-scale model which features very few aggregated variables and equations. They show that the underlying heterogeneous agent model can be solved giving a guess on the trajectory of these variables without having to keep track of the disaggregated variables. The authors use the nonlinear impulse responses in the immediate neighborhood of the steady state to generate general linear impulse responses. Building on this, Auclert et al. (2021) provide an elegant and efficient method for calculating the steady state sequence space Jacobian which provides linear impulse response functions for models with many aggregated variables. Auclert et al. (2021) also use the steady state sequence space Jacobian in the context of a Newton method, which allows them to find the nonlinear perfect foresight solution in the direct neighborhood of the steady state.[7]

An alternative approach for solving heterogeneous agent models is based on the state-space representation and goes back to Reiter (2009). This approach as well returns impulse responses to the linearized model. Since the disaggregated state space of heterogeneous agent models may generally be very large, such state-space representation usually makes a state-space reduction necessary. Such reduction routines are given, e.g., by Algan et al. (2008), Winberry (2018), Ahn et al. (2018), or Bayer et al. (2020), where they are also applied to the Bayesian estimation of linearized heterogeneous agent models. Similarly, Reiter (2023) provides a method that also allows for second-order perturbation solutions. While these methods require

[6] Indeed, when adding the distribution and agents' decisions as variables to the root finding problem, the block tridiagonal structure would persist. However, this would render the problem prohibitory large.

[7] The strategy to use the same Jacobian for each subsequent iteration is known as the *chord method*. While it may provide good results for systems with mild nonlinearities and close to the steady state, it often does not converges for more complicated models or when simulating dynamics further away from the steady state.

the approximation or compression of the distribution,[8] they allow for very general functional relationships between the distribution and the aggregated economy. In contrast, the sequence state approach of Auclert et al. (2021) does not require approximation or compression of distribution but requires the existence of sufficiently good linear approximation of these functional forms. Other than that, my method does neither require any of the functional forms of the model to be linear, nor any approximation or compression of the distribution. Centrally, the method also allows to find the fully nonlinear perfect foresight solution even if the trajectory is very far from the steady state, whereas the nonlinear solver is robust even to strong nonlinearities due to the use of the true Jacobian during the Newton iterations. My method further allows to fully trace the nonlinear perfect foresight transition of the complete cross-sectional distribution.

By applying automatic differentiation to solve economic models, this paper also adds to a very recent branch of the literature which introduces machine learning tools for quantitative economics. Examples include Scheidegger and Bilionis (2019); Maliar et al. (2021); Kahou et al. (2021); Bianchi et al. (2021); Azinovic et al. (2022) and in particular Fernández-Villaverde et al. (2023). These papers use deep learning networks to solve nonlinear high-dimensional macroeconomic models including aggregate uncertainty. Notably, solving this type of models was deemed impossible only a few years ago and the current progress in this area is very impressive. The current state of this line of research documents that it is well possible to successfully apply deep learning to solve economic models, but that these methods (and their convergence properties) are not yet well understood, may take a long time to train and require additional expert knowledge. As machine learning techniques are currently very actively researched across many fields, it is very likely that they are the path forward to circumvent the curse of dimensionality and to solve complex nonlinear models with aggregate uncertainty. This paper can be seen as an intermediary step contributing to this overarching goal.[9]

My method is based on iteratively solving the system of linear equations of each Newton step. A class of related numerical methods are the well-known Krylov subspace methods, see e.g. the generalized minimal residual method (GMRES, Saad and Schultz, 1986) and the biconjugate gradient stabilized method (Van der Vorst, 1992, BiCGSTAB). Relative to these methods the algorithm introduced here has a considerably lower computational and memory overhead and is thus much faster.[10] However, while the conditions for convergence of my method are generic to the type of nonlinear systems found in economic models, they may not be generalizable to be used outside this subclass.

The rest of this paper is structured as followed. Section 2 lays out a medium-scale HANK model with two assets. Section 3 presents the main methodological contributions. Section 4 discusses details and concepts of the reference implementation with a particular focus on repoducible and extensible code. In Section 5 the method is applied to the dynamics of a change in government redistribution, whereas Section 6 gives concluding remarks.

2 A Class of Medium Scale Heterogeneous Agent Models

The results of this paper are based on three different models which share the same aggregate economy but differ in the degree of heterogeneity of households: a model with a single representative household (RANK),

[8] In fact, Ahn et al. (2018) show that the reduction of the distribution can actually be achieved at machine level accuracy. However, this cannot be accomplished for agents' decisions.

[9] Similarly, the online appendix of Achdou et al. (2022) also applies automatic differentiation in the context of a Newton method. Other than in the method introduced here, they calculate the full Jacobian matrix which is very costly for larger models.

[10] E.g. GMRES requires an Arnoldi iteration during each step, which can be computationally costly, and needs to store the results from previous iterations.

an heterogeneous agent model with one asset, and an heterogeneous agent model where households have access to a liquid and an illiquid asset (two-asset HANK). For the sake of brevity I here only sketch the setup of the two-asset HANK model and those aggregate equations that are non-standard, and redirect the exposition of the RANK and the one-asset-HANK model to Appendix A.

I combine the two-asset HANK model (e.g. Auclert et al. (2021)) with the standard medium-scale DSGE model in the tradition of Smets and Wouters (2007).[11] This results in a rich dynamic model with many aggregate state variables that demonstrates the potential of the method. The model hence contains a disaggregated part with the households decisions and the distribution dynamics, and the aggregated part of a medium-scale DSGE model. The modular setup of the method and implementation presented in Sections 3 and 4 allows to specify both parts separately.

In the disaggregated part of the model, households can hold liquid bonds b_{it} and illiquid assets a_{it}, the latter pay higher returns but are subject to convex portfolio adjustment costs. Households face idiosyncratic labor income risk e_{it} and a borrowing constraint on both assets. They wish to accumulate net worth for the purpose of consumption smoothing and to insure against the associated idiosyncratic income risk. Their Bellman equation is given by

$$V_t(e_{it}, b_{i,t-1}, a_{i,t-1}) = \max_{c_{it}, b_{it}, a_{it}} \left\{ \frac{c_{it}^{1-\sigma_c}}{1-\sigma_c} - \chi \frac{n_t^{1+\sigma_l}}{1+\sigma_l} + \beta_t E_t V_{t+1}(e_{i,t+1}, b_{it}, a_{it}) \right\} \tag{1}$$

such that

$$c_{it} + a_{it} + b_{it} = \frac{(1-\tau_t) w_t n_t}{\int P(e_{jt}) e_{jt}^{1-\Xi} dj} e_{it}^{1-\Xi} + (1 + r_t^a) a_{i,t-1} + (1 + r_t^b) b_{i,t-1} - \Phi_t(a_{it}, a_{i,t-1}) + T_t, \tag{2}$$

$$a_{it} \geq 0, \tag{3}$$

$$b_{it} \geq \bar{b}, \tag{4}$$

where n_t denotes labor supply, c_{it} the consumption of household i, and e_{it} is their household-specific labor productivity which follows an AR(1) process in logs,

$$\log e_{it} = \rho_e \log e_{i,t-1} + \epsilon_{it}^e. \tag{5}$$

$\Phi_t(\cdot)$ is the function specifying portfolio adjustment costs for the illiquid asset

$$\Phi_t(a_{it}, a_{i,t-1}) = \frac{\chi_1}{\chi_2} \left| \frac{a_{it} - (1 + r_t^a) a_{i,t-1}}{(1 + r_t^a) a_{i,t-1} + \chi_0} \right|^{\chi_2} [(1 + r_t^a) a_{i,t-1} + \chi_0], \tag{6}$$

with $\chi_0, \chi_1 > 0$ and $\chi_2 > 1$. T_t is a government lump-sum transfer specified further below.

The aggregate economy features all the bells and whistles of the medium-scale workhorse model of Smets and Wouters (2007). To avoid linear approximations, the design of the price and wage Phillips curves follows a Rotemberg (1982) setup as in Gust et al. (2012) rather than the Calvo (1983) price setting of Smets and Wouters (2007). The conventional parts of the model, including labor unions, the firm side and the government are presented in detail in Appendix A.1. Deviating from this, the setup of labor unions when

[11] This goes beyond the work of Bayer et al. (2020) as I additionally add various forms of inertia that are present in Smets and Wouters (2007), such as, e.g., inflation and wage indexation.

combined with heterogeneous households gives rise to a wage Phillips curve that takes the form

$$\psi_w \left(\frac{\pi_t^w}{\tilde{\pi}_t^w} - 1 \right) \frac{\pi_t^w}{\tilde{\pi}_t^w} = \psi_w \beta_t E_t \left\{ \left(\frac{\pi_{t+1}^w}{\tilde{\pi}_{t+1}^w} - 1 \right) \frac{\pi_{t+1}^w}{\tilde{\pi}_{t+1}^w} \right\} + \frac{\mu_t^w}{\mu_t^w - 1} \left(\chi n_t^{1+\sigma_l} - \frac{(1-\tau_t) w_t n_t}{\mu_t^w} \int e_{it} c_{it}^{-\sigma} di \right), \tag{7}$$

with wage indexation term $\tilde{\pi}_t^w$ and where $\pi_t^w = \frac{w_t^n}{w_{t-1}^n} \pi_t$ denotes wage inflation. Wages can be subject to downwards nominal wage rigidity governed by rigidity parameter ι_w,

$$w_t = \max \left\{ \iota_w \frac{w_{t-1}}{\pi_t}, w_t^n \right\}. \tag{8}$$

The setup of firms includes capital formation, capital adjustment costs, capital utilisation costs and price indexation. Dividends are given by

$$\Pi_t = y_t - w_t n_t - i_t - \frac{\psi}{2} \left(\frac{\pi_t}{\tilde{\pi}_t} - 1 \right)^2 y_t. \tag{9}$$

No arbitrage on financial markets requires that

$$\frac{R_t}{E_t \pi_{t+1}} = \frac{E_t \{ \Pi_{t+1} + s_{t+1} \}}{s_t} = E_t R_{t+1}^a = E_t R_{t+1}^b + \zeta, \tag{10}$$

where R_t is the policy rate and s_t is the stock price. ζ parameterizes the cost for liquidity transformation charged by the financial intermediary. Ex-post returns are subject to surprise inflation,

$$R_t^b = \frac{R_{t-1}}{\pi_t} - \zeta, \tag{11}$$

and capital gains are given by

$$R_t^a = \Theta_t^s \left(\frac{\Pi_t + s_t}{s_{t-1}} \right) + (1 - \Theta_t^s) \frac{R_{t-1}}{\pi_t}, \tag{12}$$

with Θ_p denoting the share of equity in the illiquid portfolio. A balanced government budget requires

$$\tau_t w_t n_t = \left(\frac{R_{t-1}}{\pi_t} - 1 \right) B^g + g_t + T_t, \tag{13}$$

where τ_t is the tax rate (rather than the tax volume), and government transfers T_t are an exogenous policy decision which is assumed to follow an AR(1) process in logs

$$\ln T_t = (1 - \rho_T) \ln \bar{T} + \rho_T T_{t-1} + \varepsilon_t^T. \tag{14}$$

Importantly, the government taxes labor only and adjusts the labor tax rate from period to period to run a balanced budget. The central bank sets the policy rate R_t following a conventional monetary policy rule with interest rate inertia.

$$\ln R_t^n = \rho \ln R_{t-1}^n + (1 - \rho) \left(\ln R_t^* + \phi_\pi \left[\ln \pi_t - \ln \bar{\pi} \right] + \phi_y \left[\ln y_t - \ln \bar{y} \right] \right) + \ln v_t, \tag{15}$$

that may be subject to the zero lower bound on nominal interest rates (ZLB)

$$R_t = \max\{1,\ R_t^n\}. \tag{16}$$

Market clearing requires

$$y_t = \int c_{it} di + g_t + i_t + \psi_t + \zeta b_{it} di, \tag{17}$$

$$s_t + b^g = \int a_{it} + b_{it} di. \tag{18}$$

The calibration – for the aggregate and disaggregate part of the model – is quite conventional and anchored around at the values reported in Boehl (2022), where a similar model is estimated under inclusion of the households' preference parameters. To allow for transition dynamics from one steady state to another, only parameters are fixed ex-ante but no steady state values. The inflation target is set to 2% annually and the initial level of transfers is zero. Appendix A contains further details on the calibration.

3 Solving Nonlinear Heterogeneous Agent Models

This section presents the main methodological innovations. I first provide a general aggregate representation of heterogeneous agent models. Then I review the principals of automatic differentiation (AD), which are important for understanding the innovations of the main method. I then present an iterative procedure that uses AD to find the dynamic equilibrium transition path. Finally, I show how the steady state and the steady state Jacobian, the latter being an important ingredient to the main method, can be calculated efficiently using AD.

3.1 An aggregate representation of heterogeneous agent models

Let $(x_t)_{t\geq 0} \in \mathbb{X} \subset \mathbb{R}^n$ be the *aggregated* variables in period t including aggregate shocks and denote the *disaggregated* state space of heterogeneous agents by $\mathbb{S} \subset \mathbb{R}^{n_s}$.[12] For the two-asset example from Section 2, $\mathbb{S} \subset \mathbb{R}^3$ is the space spanning over the domains of the two types of assets a_{it} and b_{it} and the domain of the household-specific productivity level e_{it}.

A large class of heterogeneous agent models can be cast in the form

$$(a_t, w_t) = W(w_{t+1}, x_{t-1}, x_t, x_{t+1}), \tag{19}$$

$$d_t = D(a_t, d_{t-1}), \tag{20}$$

$$\mathbf{0} = f(x_{t-1}, x_t, x_{t+1}, d_t, a_t), \tag{21}$$

where $(w_t, a_t, d_t)_{t\geq 0}$ are time-t functions defined on $\mathbb{S}$ as follows. $w_t : \mathbb{S} \to \mathbb{W} \subset \mathbb{R}^{n_w}$ denotes recursive valuations of points in $\mathbb{S}$, $a_t : \mathbb{S} \to \mathbb{A} \subset \mathbb{R}^{n_a}$ are agents' actions (or functions thereof) for a given state, and $d_t : \mathbb{S} \to [0, 1]$ is the distribution of agents across $\mathbb{S}$.[13] Since $\mathbb{S}$ is of infinite dimensionality, it must be discretized on a grid S_g with suitably chosen grid size g. Thus, in practice (w_t, a_t, d_t) are grid-based

[12] Any aggregate shock $\varepsilon^{\ddagger}$ can be included in the set of aggregate variables by shifting the timing of $\varepsilon_t^{\ddagger}$ one period backwards to $\varepsilon_{t-1}^{\ddagger}$ and adding an auxiliary equation $\varepsilon_t^{\ddagger} = 0$. Impulses can then be simulated by setting $\varepsilon_{t-1}^{\ddagger} \neq 0$. Expected shocks can be treated equivalently.

[13] Without loss of generality, there could be several distributions. I will here use the singular term for the sake of simplicity.

representations of the underlying functions where w_t is a $n_w \times g$ matrix, a_t is $n_a \times g$ and d_t is the distribution over S_g represented by the g-dimensional unit hypercube $[0,1]^g$ with $\sum_j^g d_{t,j} = 1$. Notably, the only type of models not captured by this representation are models in which the agents' actions depend on the *full* distribution.

In terms of the dynamic programming problem given in Section 2, w_t would represent the value function $V_t(e_{it}, b_{i,t-1}, a_{i,t-1})$ and $W(\cdot)$ represents the Bellman operator over Equations (1) to (6). This takes prices such as wages and the return rates r_t^a and r_t^b as given, which, in terms of the formulation above, are part of the set of aggregate variables x_t. In practice however, w_t are often the *marginal* values as required when using variants of the endogenous grid method (EGM) of Carroll (2006), which usually are more efficient computationally.[14] I will therefore also refer to W as the EGM-step which maps expected marginal values into current marginal values. The above representation is without loss of generality as it also includes cases where W is a more complicated recursion, e.g. tracing default probabilities over time in models where agents can default on their debt.

The object a_t are the decisions implied by solving the Bellman equation, e.g. for the two-asset HANK the choices of $\{a_{it}, b_{it}, c_{it}\}$ for each node on the grid S_g. Note that a_t is defined on S_g but not directly related to the distribution d_t over S_g. The role of the *evolution* of the distribution, $D(\cdot)$ is then to map the agents' current decisions a_t and the last distribution d_{t-1} into the current distribution d_t.[15] The function $f(\cdot) = z_t$ with $f : \mathbb{R}^n \to \mathbb{R}^n$ contains the n nonlinear equations describing the law-of-motion of the aggregate economy, formulated such that each equation (i.e. each residual in z_t) must equal zero in equilibrium, i.e. $z_t = \mathbf{0}$. For the model from Section 2 this corresponds to all equations following (7) additional to the respective aggregate relationships from Appendix A.1. Future or past aggregate variables beyond $t+1$ or before $t-1$ can easily be included by introducing auxiliary variables. In the following it is assumed that W, D, and f are differentiable and that this property is retained throughout the discretization and interpolation routines.

Omitting the expectations operator on $t+1$-objects implies to abstract from aggregate uncertainty and focus on the perfect foresight path. For an initial state $(x_0, d_0) \in (\mathbb{X}, [0,1]^g)$, an equilibrium consists of $\langle W, D, f\rangle$ satisfied by sequences of $\{x_t, w_t, a_t, d_t\}_{t=1}^{\infty}$ for all $t = 1, 2, \ldots$ periods. The above specification nests representative agent models if (w_t, a_t, d_t) are empty sets. The model then solely consist of the n aggregate variables x_t and the n aggregate equations in $f : \mathbb{X} \to \mathbb{R}^n$.

This representation generalizes the specification of Auclert et al. (2021) in two regards: first, the effect of idiosyncratic variables on the aggregated economy can take arbitrary functional forms and is not restricted to be linear in the distribution. In particular, it is not necessary to explicitly cast idiosyncratic variables into aggregate output variables before supplying them to f. Instead, the distribution and the agents' actions enter f directly and the mapping from (w_t, a_t, d_t) in f can take arbitrary functional forms.[16] Second, the aggregate economy does not require a representation as a directed acyclic graph.

Take (x_0, d_0) as given and fix a terminal period T sufficiently large. Assume that (x_T, w_T) in period T are known (e.g. because T is very large and the economy is thus ϵ-close to a steady state in T). Starting with w_T and a guess for the sequence of aggregated variables $\{x_t\}_{t=1}^{T-1}$ the function $W(\cdot)$ can be iterated backwards in time, thereby providing the sequence of decisions $\{a_t\}_{t=1}^{T-1}$. Then starting with d_0, this sequence can be used

[14] A generalization of EGM for multiple dimensions and portfolio choice models is given by Hintermaier and Koeniger (2010). For the two-asset HANK it is necessary to track the marginal values of liquid and illiquid assets and, hence, $w_t \in S_g^2$.

[15] It is conceptually straightforward to let D also be a function of the aggregate variables x_t. I am here abstracting from this because I am unaware of applications where this would be necessary in practice. d_t can usually be constructed from the agents' idiosyncratic actions and d_{t-1} by, e.g., using the lottery method of Young (2010).

[16] To be precise, the appendix of Auclert et al. (2021) provides a extension for this case which uses the Jacobians of D and f w.r.t. d_{ss}. While this is conceptionally solid, these Jacobians are very large in practice and their evaluation is very costly.

to iterate the function $D(\cdot)$ forwards in time until T, resulting in the sequence of distributions $\{d_t\}_{t=1}^{T-1}$. Using boldface notation for time-sequences, $\mathbf{x} = \{x_t\}_{t=1}^{T-1}$, let backwards and forwards iteration and aggregation be represented by the functions F_a, F_d and F_x as

$$F_a : \mathbf{x} \underset{W}{\to} \mathbf{a}, \tag{22}$$

$$F_d : \mathbf{a} \underset{D}{\to} \mathbf{d}, \tag{23}$$

$$F_x : (\mathbf{x}, \mathbf{d}, \mathbf{a}) \underset{f}{\to} \mathbf{z}, \tag{24}$$

where $\mathbf{z} = \{z_t\}_{t=1}^{T-1} = \{f(x_{t-1}, x_t, x_{t+1}, d_t, a_t)\}_{t=1}^{T-1}$ is the sequence of residuals from the aggregated equations. It is then straightforward to define a function $F : \mathbb{R}^{n(T-1)} \to \mathbb{R}^{n(T-1)}$ by

$$F(\mathbf{x}) = F_x(\mathbf{x}, F_a(\mathbf{x}), F_d(F_a(\mathbf{x}))) = \mathbf{z}, \tag{25}$$

which is, thus, defined in aggregate terms only.[17] It follows that a perfect foresight equilibrium trajectory $\mathbf{x}^*$ is given by

$$F(\mathbf{x}^*) = \mathbf{0}, \tag{26}$$

implying that a fully nonlinear period-T truncated solution to the model in Eqns. (19) to (21) can be expressed as a $(T-1 \times n)$-dimensional root finding problem. A solution to this type of problems can be found using Newton's method. Starting with an initial guess on the equilibrium trajectory $\mathbf{x}_0$, Newton's method is given by the iteration

$$\mathbf{x}_{i+1} = \mathbf{x}_i - J(\mathbf{x}_i)^{-1} F(\mathbf{x}_i), \tag{27}$$

until $\|\mathbf{x}_{i+1} - \mathbf{x}_i\| < \epsilon$, where $J(\mathbf{x}_i)$ is the Jacobian matrix of F evaluated at $\mathbf{x}_i$ and ϵ is a given (very small) stopping criterion. While Newton's method is known for quadratic convergence, applying this method directly is usually impractical (if not impossible) because the calculation of the Jacobian and its inverse is normally very expensive. To circumvent this problem, I below present an iterative method to solve the linear system of equations associated with $J(\mathbf{x}_i)^{-1} F(\mathbf{x}_i)$ in (27) very efficiently using automatic differentiation.

3.2 A primer on automatic differentiation

Automatic differentiation (AD) is a computational technique for efficiently computing derivatives of a function. AD is heavily used in machine learning, where, given a model such as, e.g., a neural network, the gradient of a loss function with respect to the model's parameters is typically used to optimize the parameters and train the model. With AD, the gradient can be computed efficiently and accurately, even for complex models. This can be especially useful in large-scale optimization problems, where the number of variables and constraints can be very large.

Many explanations of AD start by pointing out that AD works by using the chain rule of calculus to build up the derivative of a function from the derivatives of its constituent parts. While this is technically true, it is sometimes mistaken to imply that AD can magically and at almost zero computational costs provide the Jacobian of any multivariate function. This, importantly, is not the case.

AD knows two distinct modes: *forward mode* and *reverse mode*. Forward accumulation is accomplished by augmenting the algebra of real numbers and obtaining a new arithmetic: the algebra of *dual numbers*. An additional component is added to every number to represent the derivative of a function at the number, and

[17] Note again that for a representative agent problem it simply holds that $F = F_x$.

all arithmetic operators are extended for the augmented algebra. A dual number x is given by

$$a = b + c \cdot \epsilon, \tag{28}$$

such that $\epsilon > 0$ and $\epsilon^2 = 0$. The algebra of dual numbers has hence some similarities to the algebra of complex numbers, with the difference that $\epsilon^2 = 0 \neq -1$.

Using the vector of dual numbers $\mathbf{a} = \mathbf{b}+\mathbf{c}\epsilon$ the algebra can be extended to multivariate analytic functions such that

$$g(\mathbf{a}) = g(\mathbf{b} + \mathbf{c}\epsilon) = g(\mathbf{b}) + J(\mathbf{b})\mathbf{c}\epsilon, \tag{29}$$

where g is a function and $J(\mathbf{b})$ its Jacobian evaluated at $\mathbf{b}$. This implies that AD – via dual numbers – allows to efficiently calculate Jacobian-vector products (JVPs) such as $J(\mathbf{b})\mathbf{c}$ just by a single forward-pass. It does however by no way imply that the Jacobian itself is cheap to obtain. To see this, denote the JVP as $J(\mathbf{b})\mathbf{c} = \Lambda(\mathbf{b}, \mathbf{c})$. Assuming $g : \mathbb{R}^m \to \mathbb{R}^n$, the Jacobian at $\mathbf{b}$ is given by

$$J(\mathbf{y}) = J(\mathbf{y})\mathbf{I}_n = J(\mathbf{y})\begin{bmatrix}\mathbf{e}_1^\intercal & \mathbf{e}_2^\intercal & \cdots & \mathbf{e}_n^\intercal\end{bmatrix} = \begin{bmatrix}\Lambda(\mathbf{y}, \mathbf{e}_1^\intercal) & \Lambda(\mathbf{y}, \mathbf{e}_2^\intercal) & \cdots & \Lambda(\mathbf{y}, \mathbf{e}_n^\intercal)\end{bmatrix}, \tag{30}$$

where $\mathbf{I}_n$ is the n-dimensional identity matrix and $\mathbf{e}_i$ is the ith vector in the standard basis of $\mathbb{R}^n$. This implies that calculating the Jacobian of a function with domain $\mathbb{R}^n$ requires exactly n evaluations of f, which is only one evaluation less than needed for the one-sided finite difference approximation of the Jacobian. Notably for our application, this makes the evaluation of objects such as the Jacobian of f w.r.t. d_t (as e.g. suggested in the appendix of Auclert et al. (2021) to generalize their model specification) prohibitory expensive for many applications.[18] Thus and so far, the only clear advantage of AD over finite difference methods is precision.

In *reverse mode* automatic differentiation, the computational graph of the function is traversed in reverse order, and the derivative of each node is computed by using the derivatives of its outputs. The derivative of each node is then used to update the derivative of its inputs. The process continues until the derivatives of the inputs are computed. Importantly, this allows to cheaply evaluate the vector-Jacobian product (VJP) $\mathbf{c}^\intercal J(\mathbf{b}) = \Gamma(\mathbf{b}, \mathbf{c})$ at a single pass of the function. Continuing to assume that the codomain of g is $\mathbb{R}^m$, the Jacobian can therefore also be evaluated by

$$J(\mathbf{y}) = \mathbf{I}_m J(\mathbf{y}) = \begin{bmatrix}\mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_m\end{bmatrix} J(\mathbf{y}) = \begin{bmatrix}\Gamma(\mathbf{y}, \mathbf{e}_1) \\ \Gamma(\mathbf{y}, \mathbf{e}_2) \\ \vdots \\ \Gamma(\mathbf{y}, \mathbf{e}_m)\end{bmatrix}, \tag{31}$$

which requires m evaluations. The use of reverse mode AD over finite differences is hence beneficial either for evaluating VJPs at low costs or for calculating J if $m < n$. The next two subsections show how to apply these insights efficiently to solve macroeconomic heterogeneous agent models.

3.3 An extended Newton's method based on JVPs

Summarizing the last subsection, the particular strength of AD is to evaluate JVPs, VJPs, and Jacobians of functions with either very small domain or codomain. Unfortunately, the latter is clearly not the case for

[18] The finite difference approximation of a JVP is given by $\Lambda(\mathbf{y}, \mathbf{z}) \approx \frac{f(\mathbf{y}+\sigma\mathbf{z})-f(\mathbf{y})}{\sigma}$ where σ is the step size. The evaluation of a JVP using dual numbers requires one evaluation of f versus two evaluations when using a finite difference approximation. Note that the dual number evaluation of f comes with a computational overhead.

the function F from Eqn. (25) – Newton's method – which comprises a square Jacobian of size $n(T-1) \times n(T-1)$. Calculating a *single* Jacobian for the two-asset HANK model and a truncation horizon of $T = 300$ takes more than 5 minutes on a standard laptop, thereby rendering the calculation of a single transition path prohibitory costly.[19] Thus, AD alone cannot solve the problem of finding nonlinear transition paths for heterogeneous agent models.

Rather than actually calculating and inverting the Jacobian matrix for Newton's method in Eq. (27), we may use the solution to the system of linear equations

$$J(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i) = -F(\mathbf{x}_i), \tag{32}$$

i.e. during each iteration we are looking for an $\mathbf{y} = \mathbf{x}_i - \mathbf{x}_{i+1}$ such that $\Lambda(\mathbf{x}_i, \mathbf{y}) = F(\mathbf{x}_i)$. Denote by $\bar{\mathbf{x}}$ the sequence of aggregate variables in the steady state and by $\bar{J} = J(\bar{\mathbf{x}})$ the steady state Jacobian. Then

$$J(\mathbf{x}_i)\mathbf{y} = F(\mathbf{x}_i), \tag{33}$$

$$(J(\mathbf{x}_i) - \alpha^{-1}\bar{J} + \alpha^{-1}\bar{J})\mathbf{y} = F(\mathbf{x}_i), \tag{34}$$

$$\alpha^{-1}\bar{J}\mathbf{y} = F(\mathbf{x}_i) - (J(\mathbf{x}_i) - \alpha^{-1}\bar{J})\mathbf{y}, \tag{35}$$

$$\mathbf{y} = \mathbf{y} + \alpha\bar{J}^{-1}(F(\mathbf{x}_i) - J(\mathbf{x}_i)\mathbf{y}), \tag{36}$$

where $\alpha > 0$ is a scalar dampening factor to ensure convergence.

This last equation can be used as the starting point for an iterative procedure. Proposition 1 provides a first convergence result for the neighborhood of the steady state.

Proposition 1. *Given an initial guess* $\mathbf{y}_0$ *and fixing* $\alpha = 1$*, the iterative scheme*

$$\mathbf{y}_{j+1} = \mathbf{y}_j + \alpha\bar{J}^{-1}(F(\mathbf{x}_i) - \Lambda(\mathbf{x}_i, \mathbf{y}_j)), \tag{37}$$

converges to

$$\lim_{j\to\infty} \mathbf{y}_j = J(\mathbf{x}_i)^{-1}F(\mathbf{x}_i) \tag{38}$$

if $\mathbf{x}_i$ *is sufficiently close to* $\bar{x}$ *and* $\bar{J}$ *and* $J(\mathbf{x}_i)$ *are invertible.*

Proof. Given invertibility of $\bar{J}$ it is clearly the case that if $\mathbf{y}_{j+1} = \mathbf{y}_j$, then $F(\mathbf{x}_i) = \Lambda(\mathbf{x}_i, \mathbf{y}_j) = J(\mathbf{x}_i)\mathbf{y}_j$. This means we have to prove convergence of $\mathbf{y}_j$ in (37). It is well known that the iterative procedure

$$\mathbf{y}_{j+1} = \mathbf{c} + A\mathbf{y}_j \tag{39}$$

converges for a square matrix A if the spectral radius $\rho(A)$ of A is less than unity, i.e. if the modulus of all eigenvalues of A lie within the unit circle.

Define $\Sigma_i = J(\mathbf{x}_i) - \bar{J}$ to be the deviation of the Jacobian at iteration i from the steady state Jacobian. For $\alpha = 1$ we have that

$$\mathbf{y}_{j+1} = \mathbf{y}_j + \bar{J}^{-1}(F(\mathbf{x}_i) - \Lambda(\mathbf{x}_i, \mathbf{y}_j)), \tag{40}$$

$$= \bar{J}^{-1}F(\mathbf{x}_i) + \left(\mathbf{I} - \bar{J}^{-1}J(\mathbf{x}_i)\right)\mathbf{y}_j, \tag{41}$$

$$= \bar{J}^{-1}F(\mathbf{x}_i) - \bar{J}^{-1}\Sigma_i\mathbf{y}_j, \tag{42}$$

[19] Achdou et al. (2022) discuss this case in their online appendix but rule it out as being too costly in practice.

and convergence thus depends on the spectral radius $\rho(\bar{J}^{-1}\Sigma_i)$ of the second term.

In an ϵ-close neighborhood of the steady state we have, for very small ϵ and any norm $\|\cdot\|$, $\|\mathbf{x}_i - \bar{\mathbf{x}}\| < \epsilon$ and thus $\|J(\mathbf{x}_i) - \bar{J}\| = \|\Sigma_i\| < \epsilon$. Since the determinant of a matrix equals the product of its eigenvalues it follows that because $\epsilon \ll 1$, the determinant $\det(\Sigma_i) < \epsilon$ is also very small. Recall that for any square matrices B and C it holds that $\det(BC) = \det(B)\det(C)$ and we thus have

$$\det(\bar{J}^{-1}\Sigma_i) = \det(\bar{J}^{-1})\det(\Sigma_i) < \epsilon \implies \rho(\bar{J}^{-1}\Sigma_i) < 1. \tag{43}$$

■

The above result is useful because it holds as long as $\rho(\bar{J}^{-1}\Sigma_i) < 1$, which may not only be true in the direct neighborhood of $\bar{x}$. However, since we are in particular interested in those cases where the deviation from the steady state is large, we can go one step further and refine the iterative procedure for more general cases by adding a dampening factor α_j. For this, the following Lemma 1 will be useful, which allows us to arrive at Proposition 2.

Lemma 1. *The Rayleigh quotient of a real matrix M and vector $\mathbf{z}$ is given by*

$$R(M, \mathbf{z}) = \frac{\mathbf{z}^\intercal M \mathbf{z}}{\mathbf{z}^\intercal \mathbf{z}}. \tag{44}$$

It holds that

i) *If $\mathbf{v}$ is an eigenvector of M with associated eigenvalue λ, then*

$$R(M, \mathbf{v}) = \frac{\mathbf{v}^\intercal M \mathbf{v}}{\mathbf{v}^\intercal \mathbf{v}} = \frac{\mathbf{v}^\intercal \lambda \mathbf{v}}{\mathbf{v}^\intercal \mathbf{v}} = \lambda. \tag{45}$$

ii) *For any iterative procedure $\mathbf{y}_{j+1} = \mathbf{z} + M\mathbf{y}_j$ with square matrix M, $\mathbf{y}_j$ grows (or shrinks) along the eigenvector associated with the eigenvalue of M with largest magnitude (Mises and Pollaczek-Geiringer, 1929). Together with i) this implies*

$$\lim_{j\to\infty} R(M, \mathbf{y}_j) = \rho(M). \tag{46}$$

iii) *For a square matrix M and a vector $\mathbf{z}$ with $\|\mathbf{z}\| > 0$ it holds that*

$$|R(M, \mathbf{z})| \in [0, \sigma_{\max}], \tag{47}$$

where $\sigma_{\max}$ is the largest singular value of M (see Appendix B).

Proposition 2. *Given an initial guess $\mathbf{y}_0$, a scaling parameter $\gamma \in (1, 2)$ and initializing $\alpha_0 = 1$, the iterative scheme*

$$\mathbf{y}_{j+1} = \mathbf{y}_j + \alpha_j \bar{J}^{-1}(F(\mathbf{x}_i) - \Lambda(\mathbf{x}_i, \mathbf{y}_j)), \tag{48}$$

$$\alpha_j = \min\left\{\alpha_{j-1}, \gamma / \left|R\left(\bar{J}^{-1}J(\mathbf{x}_i), \mathbf{y}_j\right)\right|\right\}, \tag{49}$$

with the Rayleigh quotient $R(\cdot) = \frac{\mathbf{y}_j^\intercal \bar{J}^{-1}\Lambda(\mathbf{x}_i, \mathbf{y}_j)}{\mathbf{y}_j^\intercal \mathbf{y}_j}$, converges to

$$\lim_{j\to\infty} \mathbf{y}_j = J(\mathbf{x}_i)^{-1} F(\mathbf{x}_i) \tag{50}$$

if all generalized eigenvalues of $\bar{J}$ and $J(\mathbf{x}_i)$ are real, positive and finite.

Proof. Positivity and finiteness of the generalized eigenvalues implies that $\bar{J}$ and $J(\mathbf{x}_i)$ are nonsingular. We again have to prove convergence of $\mathbf{y}_j$ in (48) for which $\rho(A_j) < 1$ given $A_j = \mathbf{I} - \alpha_j B$ with $B = \bar{J}^{-1} J(\mathbf{x}_i)$ is a sufficient condition. Denote by $\lambda_k(A_j)$ the (unordered) kth eigenvalue of A_j and define $\lambda_k(B)$ respectively. It holds that

$$\lambda_k(A_j) = 1 - \alpha_j \lambda_k(B), \tag{51}$$

$$|\lambda_k(A_j)| = \sqrt{(1 - \alpha_j \Re(\lambda_k(B)))^2 + (\alpha_j \Im(\lambda_k(B)))^2}, \tag{52}$$

$$= \sqrt{1 - 2\alpha_j \Re(\lambda_k(B)) + \alpha_j^2 |\lambda_k(B)|^2}. \tag{53}$$

Imposing $|\lambda_k(A_j)| < 1$ for all k requires

$$\alpha_j < \frac{2\Re(\lambda_k(B))}{|\lambda_k(B)|^2} \quad \forall k = 1, 2, \ldots, n(T-1), \tag{54}$$

which is an upper bound on α_j. Under the given assumption that all eigenvalues of B are real and positive, this reduces to

$$\alpha_j < \frac{2}{\rho(B)}, \tag{55}$$

where the spectral radius $\rho(B)$ of B is unknown. Eqn. (49) defines the recursion

$$\alpha_j = \min\left\{\alpha_{j-1}, \gamma / \left|R(B, \mathbf{y}_j)\right|\right\}, \tag{56}$$

which uses the Rayleigh quotient

$$R(B, \mathbf{y}_j) = \frac{\mathbf{y}_j^\intercal B \mathbf{y}_j}{\mathbf{y}_j^\intercal \mathbf{y}_j} = \frac{\mathbf{y}_j^\intercal \bar{J}^{-1} \Lambda(\mathbf{x}_i, \mathbf{y}_j)}{\mathbf{y}_j^\intercal \mathbf{y}_j}. \tag{57}$$

from Lemma 1 to also iteratively approximate $\rho(B)$ with each iteration on $\mathbf{y}_j$.

The remaining task is to show the convergence of α_j. Since from Lemma 1 we know that $|R(B, \mathbf{y}_j)| \leq \sigma_{\max}(B)$ it follows that α_j is bounded by

$$\alpha_j \in \left[\frac{\gamma}{\sigma_{\max}(B)}, 1\right], \tag{58}$$

and in the limit it holds

$$\lim_{j\to\infty} \alpha_j \in \left[\frac{\gamma}{\sigma_{\max}(B)}, \min\left\{1, \frac{\gamma}{\rho(B)}\right\}\right], \tag{59}$$

that is, α_j and thereby $A_j = \mathbf{I} - \alpha_j B$ converges with given bounds. As the eigenvalues of B are positive and real, we have that

$$\lim_{j\to\infty} \rho(A_j) = \max\left\{\lim_{j\to\infty}\left(1 - \alpha_j \lambda_{\min}(B)\right), \lim_{j\to\infty}\left(\alpha_j \rho(B) - 1\right)\right\} \tag{60}$$

with

$$\lim_{j\to\infty}\left(1 - \alpha_j \lambda_{\min}(B)\right) \leq 1 - \gamma \frac{\lambda_{\min}(B)}{\sigma_{\max}(B)} < 1 \tag{61}$$

and

$$\lim_{j\to\infty}\left(\alpha_j\rho(B)-1\right)\in\left\{\begin{array}{ll}\left[\gamma\frac{\rho(B)}{\sigma_{\max}(B)}-1,\gamma-1\right] & \text{if } \rho(B)\geq\gamma\\ \left[\gamma\frac{\rho(B)}{\sigma_{\max}(B)}-1,\rho(B)-1\right] & \text{if } \rho(B)<\gamma\end{array}\right\}<1, \tag{62}$$

where in the second line $\rho(B)<\gamma$ implies $\rho(B)-1<\gamma-1<1$. Thus, $\lim_{j\to\infty}\rho(A_j)<1$ and $\mathbf{y}_j$ converges to a solution to the linear system of equations in (32). ■

Note that Proposition 2 contains *two* simultaneous iterative procedures: the procedure of solving for the increment $\mathbf{y}$ associated with each Newton step, and the procedure finding a feasible dampening factor α associated with each Newton step. Both procedures operate at the same time, which is unproblematic since the underlying problem within one Newton step is linear.

Intuitively, for each new Newton step we set α_j to equal the inverse of the approximation of the spectral radius of B scaled by γ and thereby dampen the spectral radius of A to lie inside the unit circle, which guarantees convergence to a solution of the system of linear equations under the given assumptions. Numerically, the beauty in Proposition 2 lies in the fact that both, $\mathbf{y}_j$ and α_j can be calculated by just one forward sweep on F. $F(\mathbf{x}_i)-\Lambda(\mathbf{x}_i,\mathbf{y}_j)$ is a vector and $\bar{J}^{-1}(F(\mathbf{x}_i)-\Lambda(\mathbf{x}_i,\mathbf{y}_j)$ can hence be evaluated by the LU-factorization of $\bar{J}$, which only needs to be calculated once for a given model. All other operations are simple calculations in vector space.

The two conditions for Proposition 2 – positivity and realness of the generalized eigenvalues – are sufficient, but not necessary conditions. Importantly, this implies that the iterative procedure may converge even if these conditions are not satisfied. The condition that the generalized eigenvalues are real is a rather weak condition, which is related to the fact that the Rayleigh approximation cannot distinguish between the real and imaginary part of the largest eigenvalue. If the eigenvalues are complex, (54) is still likely to be satisfied whenever $\Re(\lambda_k(B))>0$ because, in practice, $\alpha_j\leq\frac{\gamma}{\rho(B)}$ is only an upper bound (c.f. eqn. (58)).[20]

The value of $\gamma\in(1,2)$ determines the speed of convergence and can be chosen freely within the given bounds: a value close to 2 will shift the largest eigenvalue of B close to 2, resulting in a spectral radius $\rho(A)$ of A close to one. A value of α_j close to 0 will shift the *smallest* eigenvalue of B close to 0 (c.f. eqn. (60)), which also results in $\rho(A)$ close to one. Since convergence speed depends on the magnitude of $\rho(A)$ (smaller in magnitude is better) an optimal value will lie somewhere in the mid-range. While a sufficiently small γ could compensate for the rare cases in which $\frac{2\Re(\lambda_k(B))}{|\lambda_k(B)|^2}\ll\frac{2}{\rho(B)}$ for complex eigenvalues, a value of $\gamma=1.5$ has proven very reliable for a wide range of applications.

3.4 Finding the steady state and its Jacobian

Continue to denote steady state objects by a bar, i.e. $\bar{x}$, $\bar{d}$, $\bar{a}$ and $\bar{w}$. In the common cases when the steady state is not unique some variables must be fixed ex-ante via an additional set of restrictions $x=b(\mathfrak{x})$, where

[20] A path to ensure that both conditions are guaranteed to be satisfied is to pre-multiply $F(\mathbf{x}_i)$ and $\Lambda(\mathbf{x}_i,\mathbf{y}_j)$ in (48) and (49) by $J(\mathbf{x}_i)^\intercal(\bar{J}^{-1})^\intercal\bar{J}^{-1}$ instead of $\bar{J}^{-1}$. These expressions can efficiently be calculated using AD via

$$J(\mathbf{x}_i)^\intercal(\bar{J}^{-1})^\intercal\bar{J}^{-1}F(\mathbf{x}_i)=\Gamma\left(\mathbf{x}_i,\left((\bar{J}^{-1})^\intercal\bar{J}^{-1}F(\mathbf{x}_i)\right)^\intercal\right)^\intercal \tag{63}$$

and ensures that the central matrix in the iterative procedure is positive definite. However, convergence would be relatively slow because the largest and smallest eigenvalues of $J(\mathbf{x}_i)^\intercal(\bar{J}^{-1})^\intercal\bar{J}^{-1}J(\mathbf{x}_i)$ are $\sigma_{\min}(B)^2$ and $\sigma_{\max}(B)^2$ which implies eigenvalues of the iterative procedure with modulus considerably closer to unity.

$\mathfrak{x}$ is the subset of x necessary to evaluate $\bar{x}$.[21] The steady state must satisfy

$$\bar{x} = b(\bar{\mathfrak{x}}), \tag{64}$$

$$(\bar{a}, \bar{w}) = W(\bar{w}, \bar{x}, \bar{x}, \bar{x}), \tag{65}$$

$$\bar{d} = D(\bar{a}, \bar{d}), \tag{66}$$

$$\mathbf{0} = f(\bar{x}, \bar{x}, \bar{x}, \bar{d}, \bar{a}). \tag{67}$$

For a given guess on $\mathfrak{x}$ calculate the corresponding guess on $\bar{x}$ using $b(\cdot)$. $\bar{w}$ can then be found by iterating on Eqn. (65) until it converges. Denote the function that does so as $\bar{w} = \bar{W}(\bar{x})$. Given $\bar{x}$ and $\bar{w}$, the steady-state distribution $\bar{d}$ can also be found by iterating on Eqn. (66) until convergence (or, alternatively, via the unit-eigenvector). Denote this solver as $\bar{d} = \bar{D}(\bar{a})$ and define $\bar{f}$ equivalently for $f(\cdot)$. Combining those three, $\bar{\mathfrak{x}}$ must satisfy $H(\bar{\mathfrak{x}}) = \mathbf{0}$ with H defined as

$$H(\bar{\mathfrak{x}}) = \bar{f}\left(b(\bar{\mathfrak{x}}), \bar{D}\left(b(\bar{\mathfrak{x}}), \bar{W}(b(\bar{\mathfrak{x}}))\right), \bar{W}(b(\bar{\mathfrak{x}}))\right). \tag{68}$$

Since $\mathfrak{x} \in \mathbb{R}^m$ with m small we can this time calculate the complete Jacobian of H using forward mode automatic differentiation and, starting with some guess $\mathfrak{x}_i$ on $\bar{\mathfrak{x}}$, the root of H can be found using a modified Newton's method

$$\mathfrak{x}_{i+1} = \mathfrak{x}_i - J_H(\mathfrak{x}_i)^+ H(\mathfrak{x}_i), \tag{69}$$

where $J_H(\mathfrak{x}_i)^+$ denotes the Moore–Penrose inverse. Using the latter is necessary because $J_H(\mathfrak{x}_i)$ typically does not have full rank since the codomain of f is $\mathbb{R}^n$ and $n \leq m$. Proofs of convergence for the modified Newton procedure are, e.g., given by Ben-Israel (1965). Despite not requiring any manual input, the procedure turns out very robust in practice even for relatively bad initial guesses. In particular, it is usually not necessary to manually provide some of the steady state relationships as an additional input to the dynamic system of the economic model.

Given $\bar{x}$, the steady state Jacobian $\bar{J} = J(\bar{\mathbf{x}}) = J(\{\bar{x}\}_0^T)$ of F can also be found using AD. To be clear on notation, let $J_{x \to y}$ be the matrix whose (i, j)th entry is $J_{ij} = \frac{\partial y_i}{\partial x_j}$. The (i, j)th entry of the steady state Jacobian is then for $i > 1$ given by

$$\bar{J}_{ij} = \frac{\partial z_i}{\partial x_j} = \frac{\partial f_i}{\partial x_j} + \frac{\partial f_i}{\partial d_i}\frac{\partial d_i}{\partial x_j} = \frac{\partial f_i}{\partial x_j} + \frac{\partial f_i}{\partial d_i}\left(\frac{\partial d_i}{\partial a_i}\frac{\partial a_i}{\partial x_j} + \frac{\partial d_i}{\partial d_{i-1}}\frac{\partial d_{i-1}}{\partial x_j}\right) \tag{70}$$

$$= \frac{\partial f_i}{\partial x_j} + \frac{\partial f_i}{\partial d_i}\sum_k^{T-j}\frac{\partial d_i}{\partial d_k}\frac{\partial d_k}{\partial a_k}\frac{\partial a_k}{\partial x_j} = \frac{\partial f_i}{\partial x_j} + \sum_k^{T-j}\frac{\partial f_i}{\partial a_k}\frac{\partial a_k}{\partial x_j}, \tag{71}$$

which is a n-by-n matrix.

Recall that the function $F_a(\cdot)$ from Eqn. (22) is defined on the complete sequence $\mathbf{x}$. We can calculate the Jacobian with respect to x_{T-1} by stacking the JVPs of the transpose of the last n vectors of the standard

[21] A typical example for New-Keynesian models is that the steady state inflation needs to be fixed ex-ante since it is a policy choice variable of the central bank.

basis of $\mathbb{R}^{(T-1)n}$:

$$\bar{J}_{x_{t-1}\to\mathbf{a}} = \begin{bmatrix} \Lambda_{F_a}\left(\bar{\mathbf{x}}, \mathbf{e}_{(T-1)}^\intercal\right)^\intercal \\ \Lambda_{F_a}\left(\bar{\mathbf{x}}, \mathbf{e}_{(T-1)2}^\intercal\right)^\intercal \\ \vdots \\ \Lambda_{F_a}\left(\bar{\mathbf{x}}, \mathbf{e}_{(T-1)(n-1)}^\intercal\right)^\intercal \\ \Lambda_{F_a}\left(\bar{\mathbf{x}}, \mathbf{e}_{(T-1)n}^\intercal\right)^\intercal \end{bmatrix}^\intercal = \begin{bmatrix} \frac{\partial a_0}{\partial x_{T-1}} \\ \frac{\partial a_1}{\partial x_{T-1}} \\ \vdots \\ \frac{\partial a_{T-2}}{\partial x_{T-1}} \\ \frac{\partial a_{T-1}}{\partial x_{T-1}} \end{bmatrix} = \begin{bmatrix} \frac{\partial a_0}{\partial x_{T-1}} \\ \frac{\partial a_0}{\partial x_{T-2}} \\ \vdots \\ \frac{\partial a_0}{\partial x_1} \\ \frac{\partial a_0}{\partial x_0} \end{bmatrix} = \bar{J}_{\mathbf{x}\to a_0}, \tag{72}$$

where the equivalence in the second step holds because in the steady state we have $\frac{\partial a_0}{\partial x_k} = \frac{\partial a_l}{\partial x_{k+l}}$ for any l. Note that the calculation of this object requires only n evaluations of $F_a(\cdot)$.

Similarly, the Jacobian $\bar{J}_{\mathbf{a}\to z_{T-1}}$ of z_{T-1} (i.e. the last output element of F_x) w.r.t. the sequence $\mathbf{a}$ can be evaluated by using reverse mode automatic differentiation:

$$\bar{J}_{\mathbf{a}\to z_{T-1}} = \begin{bmatrix} \Gamma_{F_x\circ F_d}\left(\bar{\mathbf{x}}, \mathbf{e}_{(T-1)}\right) \\ \Gamma_{F_x\circ F_d}\left(\bar{\mathbf{x}}, \mathbf{e}_{(T-1)2}\right) \\ \vdots \\ \Gamma_{F_x\circ F_d}\left(\bar{\mathbf{x}}, \mathbf{e}_{(T-1)(n-1)}\right) \\ \Gamma_{F_x\circ F_d}\left(\bar{\mathbf{x}}, \mathbf{e}_{(T-1)n}\right) \end{bmatrix} = \begin{bmatrix} \frac{\partial z_{T-1}}{\partial a_0} \\ \frac{\partial z_{T-1}}{\partial a_1} \\ \vdots \\ \frac{\partial z_{T-1}}{\partial a_{T-2}} \\ \frac{\partial z_{T-1}}{\partial a_{T-1}} \end{bmatrix}^\intercal = \begin{bmatrix} \frac{\partial z_{T-1}}{\partial a_0} \\ \frac{\partial z_{T-2}}{\partial a_0} \\ \vdots \\ \frac{\partial z_1}{\partial a_0} \\ \frac{\partial z_0}{\partial a_0} \end{bmatrix}^\intercal = \bar{J}_{a_0\to\mathbf{y}}, \tag{73}$$

which, as above, uses the fact that $\frac{\partial z_k}{\partial a_0} = \frac{\partial z_{k+l}}{\partial a_l}$ in steady state. Note again that independently of the complexity of the functions F_x and F_d this requires only n evaluations of $F_x \circ F_d$.

Finally, initialize a helper matrix $\hat{J}$ with the tensor (outer) product $J(\bar{\mathbf{x}})_{\mathbf{a}\to z_{T-1}} \otimes J(\bar{\mathbf{x}})_{x_{t-1}\to\mathbf{a}}$ and add $\frac{\partial f}{\partial x_t}$ to $\hat{J}_{T-1,T-1}$, $\frac{\partial f}{\partial x_{t+1}}$ to $\hat{J}_{T-1,T-2}$ and $\frac{\partial f}{\partial x_{t-1}}$ to $\hat{J}_{T-2,T-1}$. Then

$$\hat{J} = \begin{bmatrix} \frac{\partial z_{T-1}}{\partial a_0}\frac{\partial a_0}{\partial x_{T-1}} & \cdots & \frac{\partial z_{T-1}}{\partial a_0}\frac{\partial a_0}{\partial x_1} & \frac{\partial z_{T-1}}{\partial a_0}\frac{\partial a_0}{\partial x_0} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial z_1}{\partial a_0}\frac{\partial a_0}{\partial x_{T-1}} & \cdots & \frac{\partial z_1}{\partial a_0}\frac{\partial a_0}{\partial x_1} & \frac{\partial f}{\partial x_{t-1}} + \frac{\partial z_1}{\partial a_0}\frac{\partial a_0}{\partial x_0} \\ \frac{\partial z_0}{\partial a_0}\frac{\partial a_0}{\partial x_{T-1}} & \cdots & \frac{\partial f}{\partial x_{t+1}} + \frac{\partial z_0}{\partial a_0}\frac{\partial a_0}{\partial x_1} & \frac{\partial f}{\partial x_t} + \frac{\partial z_0}{\partial a_0}\frac{\partial a_0}{\partial x_0} \end{bmatrix} = \begin{bmatrix} \frac{\partial z_{T-1}}{\partial x_{T-1}} & \cdots & \frac{\partial z_{T-1}}{\partial x_0} \\ \vdots & \ddots & \vdots & \vdots \\ \frac{\partial z_1}{\partial x_{T-1}} & \cdots & \frac{\partial f}{\partial x_t} + \frac{\partial z_0}{\partial x_0} \end{bmatrix}, \tag{74}$$

and $\bar{J}$ can be expressed as the recursion over its block components

$$\bar{J}_{ij} = \bar{J}_{i-1,j-1} + \hat{J}_{T-i,T-j} = \sum_{k=0}^{\min\{i,j\}} \hat{J}_{T-i+k,T-j+k}, \tag{75}$$

which corresponds to the expression in Eqn. (70) and therefore yields $\bar{J}$ for only n evaluations of F_a, F_d and F_x each. Since $\bar{J}$ is sparse for most applications, a sparse implementation of the incomplete LU decomposition can be used to pre-calculate $\bar{J}^{-1}$. For any model, this only needs to be done once.

4 Conceptual Approach and Implementation

The methods used in macroeconomics in general, and in particular the methods presented in this paper, have reached a degree of complexity that not only requires expert knowledge on numerical methods but also on computational and programming tools. Two implications follow from this insight: First, for a methodological contribution to be useful for the general economics community it is important to provide

high-level reference implementations that do not require expert computational knowledge. Such a high-level implementation is a program with strong abstraction from the computational details of the implementation. Second, it calls for a design of software that is comprehensible and extensible, but yet allows for reproducible research that can be conducted efficiently.

This section discusses these design and implementation requirements in detail. The methodological contributions of this paper are implemented in the *econpizza* software package, which I propose as a blueprint to address these issues. Together with the package, I suggest a generic high-level syntax to express heterogeneous agent models, which is presented thereafter. I then give speed benchmarks for the reference implementation. Concrete details on the econpizza software – including guides and tutorials – are redirected to the extensive online documentation of the package.[22]

4.1 Design choices, open-source software, and the importance of reusable code

The reference implementation addresses five requirements, which I regard as central for the future progress of macroeconomic research:

i.) **Strict separation between the input representing the economic model, the code of the solution routines, and routines for economic analysis.** A primary objective is to organize frequently used numerical routines into software libraries, which are then reusable *across* models. The use of such standardized and reusable packages – instead of relying on large blobs of user-written routines – has the potential to largely reduce the complexity of individual codes.[23] Consequently, it is necessary to strictly separate the model from the solution routines. To this end, the econpizza package not only is a high-level library for solving heterogeneous agent models, but also implements a syntax to generically represent heterogeneous agent models, which is discussed further below.

ii.) **Adherence to the open-source paradigm.** With the rising complexity of numerical methods, their performance increasingly depends on the quality of their implementation. This means that software libraries should improve over time, must allow for corrections or usability enhancements, and should adapt to computational advances. Although tempting, it is not a fruitful approach for PhD students and young scholars to write simulation programs from scratch but, rather, to build and elaborate on an existing codebase. To allow this, the *econpizza* is publicly developed on the version control platform GitHub, which allow users to suggest changes to the code, point out potential bugs, or to propose new functionalities.[24] Such version control systems play a central role in modern software development and are also widely used in, e.g., physics or engineering.

iii.) **Integration in a modern software development workflow.** Another objective is to maintain a clean, working, and well-documented code base. To satisfy this requirements, for the reference implementation tools for versioning (see above), automated unit testing, automatic code linting, and automated module documentations are employed. Automated unit tests ensure that, after any changes, the publicly available code produces economically correct results.[25] Linting is an automated procedure to format

[22] The online documentation can be found at `https://econpizza.readthedocs.io`.

[23] A different dimension of this problem is that macroeconomic research is currently struggling with issues of insufficient replicability of numerical work. It can not be assumed that journal referees have the time budget or the ability to check and verify the large amounts of codes that are necessary for a single contemporary research project. A reduction in complexity can thus increase the transparency of macroeconomic research.

[24] This is the fundamental concept behind open-source software. It is hence somewhat counterproductive, in the sense of points i and ii, to publish a blob of inseparable model, simulation, and analysis codes on a private website where they can not evolve over time.

[25] Unit testing is implemented through pytest, a widely used testing framework in Python, running in GitHub Actions. The latter is a free GitHub service that automatically employs the tests on a remote server whenever updates are pushed to the GitHub repository.

code such that it adheres to official coding style guidelines. This greatly improves readability of code and thus supports extensibility and reproducibility.[26] Automated module documentation are webpages that are automatically generated from the documentation strings of classes and functions, which help to increase the transparency and traceability of the software.[27]

iv.) **Maintenance of a high-level information flow between the user and the software.** In practice, low-level routines may – for various reasons not provide the expected results. As only a cursory understanding of the underlying implementation can be expected from the user, an insufficient information flow bears the risk of unintentional misuse and false results. It is thus fatal if internal errors are not sufficiently propagated and communicated. For this reason it is crucial to implement reliable checks and informative warning and error messages for the underlying routines and potential pitfalls, as it is covered in the reference implementation.

v.) **Use of a modular programming language that fully supports the functional programming paradigm.** It is highly challenging to write (and maintain) code libraries written in languages with limited support for functional programming.[28] This complicates writing and sharing function libraries that are reusable across frameworks and models because the libraries quickly becomes intractable in size and functions are inflexible. Furthermore, a programming language should integrate well with versioning systems (such as, e.g., GitHub) and feature a straightforward packaging system.[29] For these reasons, the community has recently started to adapt free and open-source languages such as Python and Julia. In particular, Python is highly flexible, simple to use, and is well integrated into modern development workflows.[30] The reference implementation is written in Python using the JAX framework, which provides just-in-time compilation and automatic differentiation. Additional details on JAX are given further below.

A fundamental design principle (cf. Point i.) above) of the econpizza package is to separate the input of the economic model (provided by the user), the underlying solution routines (the software package), and the economic analysis of the simulation outcomes. The reference implementation provides a simple syntax for expressing heterogeneous agent models, which is based on the widely used YAML format.[31] Details can be found in Appendix C.

[26] In econpizza, automated linting is implemented through the autopep8 package (which enforces the PEP 8 style guide) and pre-commit hooks. Such hooks are running automatically before code is uploaded to GitHub.

[27] Automated documentation for econpizza is implemented through Sphinx, a documentation generator widely used by the Python community, and employed on Read the Docs, which is an open-sourced free software documentation hosting platform used by many open-source projects.

[28] For example, the Matlab software widely used in economics restricts the number of function definition per file to one, and function definitions do not allow for default arguments. A default argument is an argument to a function that a programmer is not required to specify because a default value is provided.

[29] A packaging system allows to easily install additional modules/packages which provide specific functions and classes. Python and R, and more recently also Julia, have very rich ecosystem of packages for a large variety of applications, the large majority of which are well-tested and developed in the public domain. The econpizza package can be directly installed via the official Python repositories.

[30] Python is an object-oriented general purpose language used in a large field of applications. It is the de-facto industry standard in data science and machine learning and supported by mayor big-tech firms. Python is free and open source with no limiting licences or additional costs, has a huge active user base and its design simplicity allows for a high code quality.

[31] YAML ("Yet Another Markup Language") and is a standardized human-readable data-serialization language. The format is similar to XML but has a minimal syntax in order to be easily usable. It is useful to provide data input in a clear and simple way across programming languages, and is widely used in applications that require a high level human-computer interaction, such as configuration files or data storage.

4.2 Speed benchmarks

The implementation in the econpizza package heavily levers on just-in-time ("jit") compilation via the Python framework JAX. JAX is an open-source machine learning framework developed by Google which supports high-level automatic differentiation and jiting while providing the same syntax as NumPy, which is the primary Python library for numerical computing. The execution speed of JAX-jitted functions is en-par with execution speed of compiled code from languages as Fortran or C.

	tentative		**full**	
Model	once	subsequent	once	subsequent
RANK	2.102	0.094	2.191	0.184
HANK1	10.977	1.299	16.559	3.726
HANK2 (no capital)	26.438	2.139	57.390	10.708
HANK2	28.380	1.532	80.568	8.383

Table 1: Speed benchmarks for the three models provided in Appendix A. All numbers are in seconds. The HANK2 model without capital is the small-scale HANK model with two assets as described in Section 5.3.

Table 1 provides speed benchmarks for the four baseline models presented in Appendix A.[32] Note that RANK models are not solved using the method from Section 3 but rely on a simpler procedure that exploits the block tridiagonal structure of the sequence space Jacobian, which is outlined in Appendix D. The simulations listed under "tentative" use a truncation horizon of 150 (vs. 300 for "full") and a slightly smaller grid (50 grid points for the asset grid of the HANK1 model and 10 and 20 grid points for the liquid and illiquid assets in the HANK2 models) than reported in Appendix A. The simulation results from these tentative simulations are very similar to the results obtained when using the full grid as specified in Appendix A.

The table documents a large speed difference between the first simulation ("once") and each successive simulation ("subsequent"). Subsequent simulations allow to use different shocks, initial conditions, or parameters that do not alter the steady state. These simulations are much faster for two reasons. First, for each new steady state the steady state sequence space Jacobian must be recomputed including its LU decomposition. The computational load of the Jacobian and the LU decomposition vary substantially with the length of the truncation horizon. Second, the model functions must be compiled (which is done automatically by JAX). Both steps are not necessary if the steady state remains unaltered when the objects can simply be reused.

Runtimes of the RANK model are much faster since solving the RANK model does not require the calculation of the steady state Jacobian and its LU decomposition. Additionally, automatic differentiation does not have to traverse through the distribution and value functions. Comparing the HANK1 model to the HANK2 model, calculation and compilation times roughly double while the number of grid points remains roughly the same. Further, calculation and compilation times of the medium-scale HANK2 model with capital compared to the small-scale HANK2 model without capital are only marginally larger although the complexity of the aggregate system of equations of the medium-scale model is considerably larger than the complexity of the small scale model. Since the disaggregated problem is exactly the same for both models, this suggests that the complexity of the aggregated system of equations is of second order importance for the computational complexity.

[32] All benchmarks are done on a standard laptop with 8 Intel(R) Core(TM) i7-8650U CPUs (1.90GHz). The package does not make explicit use of parallel computing.

5 Nonlinear Transition Dynamics

A key feature of models with heterogeneous households is that they allow to study the dynamic effects of government transfers, taxes, and redistribution over the cross-sectional distribution of households and over business cycle aggregates. This section applies the introduced method to study the nonlinear transition dynamics of – permanent or transitory – changes in government redistribution policy. I first discuss the dynamic responses of a change in steady state transfers. I then study the effects of two severe nonlinearities on the transition dynamics, the first being downwards nominal wage rigidity and the second being the zero lower bound on nominal interest rates. Finally, I analyse the role of the distribution of wealth for the transmission of policy shocks. To allow for a realistic account of the distribution of assets while being en-par with the business cycle literature, simulations in this section are based on the two-asset HANK model from Section 2.

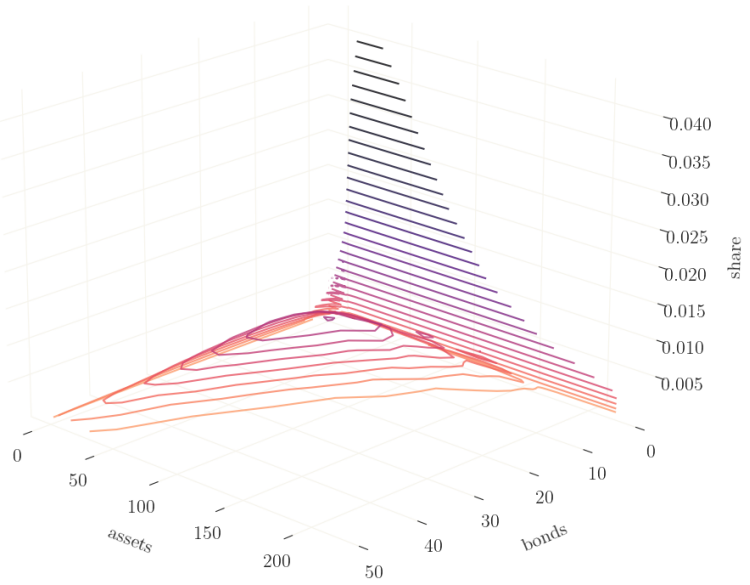

Figure 1: The stationary distribution of the heterogeneous agent New Keynesian model with two-assets. Note that for the sake of clearer display, quantities are given as shares of nodes on a log-grid (rather than true densities), meaning that shares for larger values on the grid are overrepresented.

5.1 A permanent increase in government redistribution

Assume a permanent increase in government transfers from zero to 10% of (old) GDP. After announcement, the government gradually increases the volume of transfers with an autocorrelation of $\rho_T = 0.8$. Since the government is running a balanced budget, the labor tax rate must adjust simultaneously and the policy thus redistributes income of high labor income earners to those with low labor income. Importantly, this permanent shift in transfers and taxes implies the transition from one steady state to another.

Before looking at the dynamics it is useful to study and compare the two steady states in detail. Figure 1 shows the stationary distribution of assets in the steady state without redistribution. The distribution is bimodal: the majority of agents holds a fair amount of assets and bonds, where agents with many bonds

tend to hold fewer assets and vice versa. Roughly one-third (34.3%) of all agents do not hold any bonds but yet a considerable volume of assets. These are the famous wealthy hand-to-mouth households of Kaplan et al. (2018). These households have experienced a series of negative income shocks and thus depleted their stock of liquid bonds. Since they only hold illiquid bonds and liquidations of these bonds are subject to portfolio adjustment cost, they face limited insurance and have a higher marginal propensity to consume out of income.

Table 2 presents the redistributive steady state relative to the steady state without redistribution. Clearly, the distorting effect of the necessary additional 20% in labor tax rate is massive. Production in the new steady state is more than ten percent below the old level, with the capital stock almost 20% below the previous level and an accompanying decline in real wages. The fall in output reflects in lower dividends and an associated drop in equity. Since households are better insured through the new system of government transfers, the demand for liquid bonds falls by almost 50%. Since poor households wish to insure less, inequality as measured by top-10-percentiles increases for both types of assets while it falls slightly for consumption.

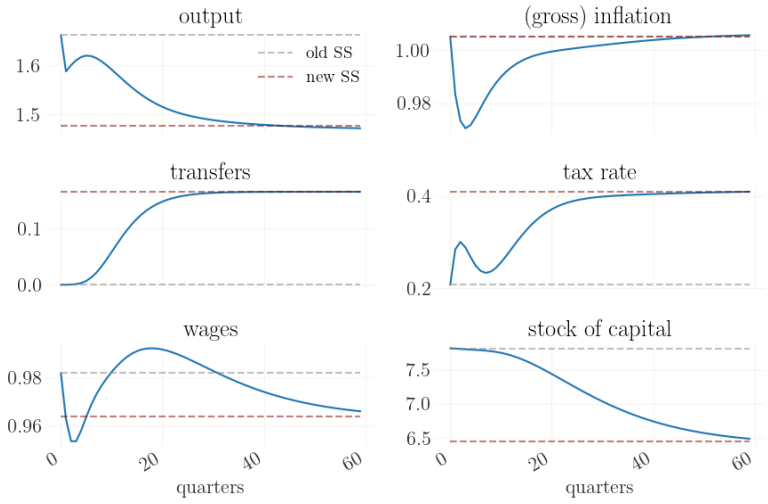

Figure 2: Nonlinear transition dynamics for a permanent increase in government transfers. All measures are given in levels. Interest and inflation rates are given in quarterly gross-rates. The dashed gray and red lines represent the old steady state without and with redistribution, respectively.

The magnitude of these very negative effects may, at first, be surprising. A common intuition for HANK models prescribes rather positive effects of transfers as households have different marginal propensities to consume. The main driver of the negative response documented by the simulations in this section is the response of labor supply determined by the wage Phillips curve in (7). In its core and absent any nominal wage rigidities (thus, for $\psi_w = 0$ and $\mu_t^w = 1$) the collective bargaining decision would read

$$\chi n_t^{\sigma_l} = (1 - \tau_t) w_t \int e_{it} c_{it}^{-\sigma} di, \tag{76}$$

which includes the aggregated individual marginal utilities of consumption $\int e_{it} c_{it}^{-\sigma} di$ as well as the tax rate τ_t, which acts as a wedge between wage and labor supply. Since in the steady state, wages, via marginal costs, are closely tied to the steady state markup, labor supply must fall in response to an increase in τ_t.

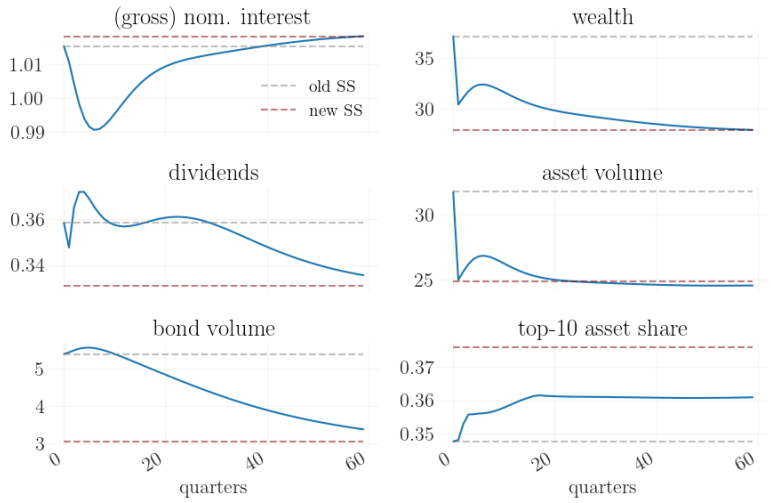

Figure 3: Nonlinear transition dynamics for a permanent increase in government transfers. All measures are given in levels. Interest and inflation rates are given in quarterly gross-rates. The dashed gray and red lines represent the old steady state without and with redistribution, respectively.

Aggregated marginal utility of consumption act as an additional amplifier since they are dominated by the negative consumption response of productive households.[33] The overall responses to a redistributive shock are thus, both in the short and long run, mainly determined by the supply-side.

	relative change		relative change
output	-11.1%	top-10% share assets	+8.1%
wages	-1.8%	top-10% share bonds	+11.8%
labor hours	-9.4%	top-10% share consumption	-3.7%
capital	-17.5%	assets	-21.6%
consumption	-11.6%	bonds	-43.5%
interest rates	+0.27%	equity	-26.8%
dividends	-7.6%	tax rate	+20.2%

Table 2: Difference of the steady states with redistribution relative to the old steady state without redistribution.

Turn next to the aggregate dynamic effects of such policy. As we have seen in Table 2, the new steady state comes with a higher real interest rate because with higher lump-sum transfers, the households wish less insurance and thus demand higher bond and asset returns. The central bank leaves the inflation target unchanged but immediately adjusts its target rate to the new steady state real rate. Yet, the adjustment of the nominal rate is gradual due to interest rate smoothing.

Figures 2 and 3 show the aggregate dynamics. Most centrally, after the announcement of the new policy

[33]The use of Greenwood et al. (1988) preferences allows to eliminate the wealth effect manifested by aggregated marginal utilities of consumption. The distortionary effect of labor taxation still prevails, even if the increase in transfers would be largely debt financed.

firms strive to downsize their capital stock. They wish to do so smoothly due to the presence of capital adjustment costs, resulting in a relatively high ratio of capital to output and a fall of the marginal productivity of capital. The firms' new choice of optimal factor inputs then leads to a dampening of labor demand and causes wages to decrease, leading to a decline in marginal costs and inflation. Monetary policy responds with lowering the nominal interest rate to stimulate consumption. When the increase in transfers gains momentum, so does the surge in the labor tax rate. Consequently, pre-tax wages increase which, in turn, stimulates inflation. This effect is, however, dampened by price and wage inertia.

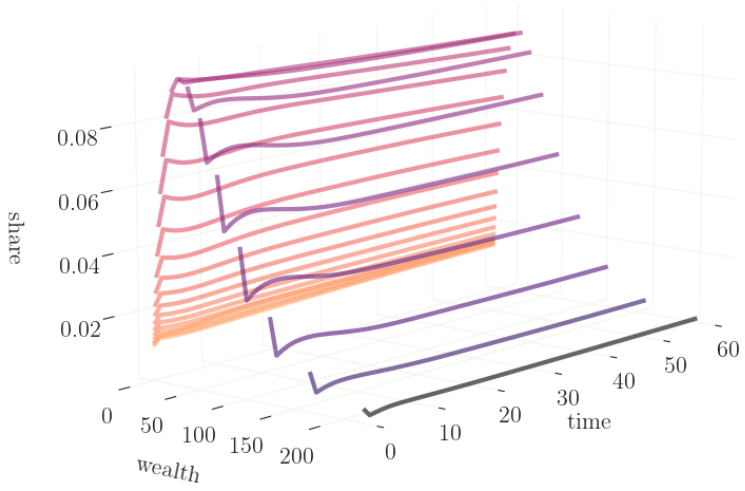

Figure 4: Nonlinear transition dynamics of the distribution of illiquid asset for a permanent increase in government transfers. Each line presents the transition over time of the share associated with one grid node, meaning that the shares of larger grid values are overrepresented.

Firms reduce their capital stock by distributing disproportionally large dividends and, consequently, dividends fluctuate around their old level for an extended period of time. However, the anticipation of lower dividends in the future causes a large contraction in the volume of equity which leads to a crash in the asset market that is reflected by a massive one-time devaluation of assets. While the transition to the new system of transfers is by large completed after 20 quarters (i.e. five years), the response of the economy shows a high degree of persistence. As such, the new level of the capital stock and the new wage rate is only reached after about 60 quarters or, respectively, 15 years.

Figure 4 shows the transition dynamics of the distribution of assets. Wealthy households experience the announcement of the new policy as a strong negative news shock. The large change in the expected flow of future dividends triggers an ample change in the volume of equity, which causes assets to devaluate heavily. This destroys a considerable amount of wealth in the economy and the weak positive effect of a larger return to assets due to higher dividend payments can not offset this effect. Consequently, the finding that the new steady state supply of assets is significantly below the old steady state (c.f. Table 2) reflects in a shrinkage of the tail-mass of the distribution over time, as documented in Figure 4. This effect – the convergence of the distribution – is considerably slower than the convergence of aggregate variables. This can also be seen when converging the convergence speed of inequality dynamics in asset holdings as measured in top-10-percentiles in Figure 3.

5.2 The role of DNWR and the ZLB during transition

The presence of strong nonlinearities such as the zero lower bound (ZLB) on nominal interest rate an downwards nominal wage rigidity (DNWR) can have large effects on the transition dynamics. At the ZLB, the central bank is reluctant to set the nominal rate below zero to avoid that households start hoarding cash. DNWR implies that, either due to regulatory reasons or for incentive considerations, firms are unable to lower *nominal* wages below a certain threshold. Both are known to pose severe challenges to numerical solution techniques.

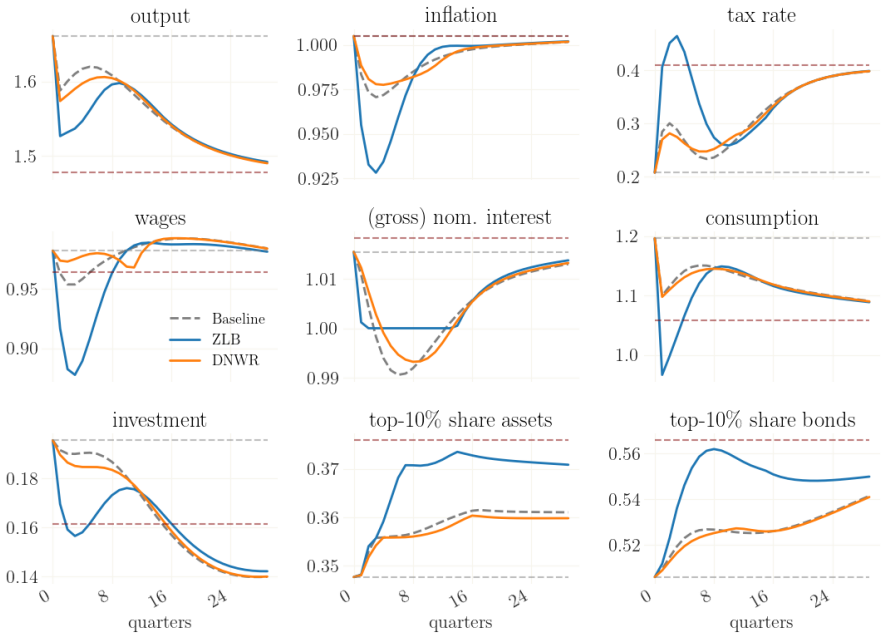

Figure 5: Nonlinear transition dynamics for a permanent increase in government transfers. The blue and orange lines show the transition dynamics with an active ZLB and DNWR, respectively. Both are inactive for the dashed black line. All measures are given in levels. Interest and inflation rates are given in quarterly gross-rates. The dashed gray and red lines represent the old steady state without and with redistribution, respectively.

Figure 5 again shows the transition dynamics after the announced gradual and permanent increase in transfers from the previous subsection. The gray dashed line is the same as in Figures 2 and 3 where I abstract from the ZLB and DNWR. The blue line shows responses where the ZLB is present and active. The dynamic effects of the ZLB are qualitatively quite similar to those implied by a representative agent model with the ZLB: since the central bank can no longer respond and stimulate the economy by decreasing the real rate, consumption falls. Households are willing to accept lower wages because the marginal utility of consumption increases. The associated fall in marginal costs leads to an additional drop in inflation which causes a further increase in the real rate. The response of inequality – as measured by top percentiles – is striking: both measures double in the initial spike relative to the scenario without the ZLB. Clearly, only the wealthy benefit from relatively higher interest rates while the lack of an additional monetary stimulus harms those households with low net worth.

The orange line in Figure 5 shows simulations where nominal wages are downward-rigid (DNWR). The setup is calibrated to allow wages to fall by a maximum of 2% quarterly, which corresponds to realistic

estimates. The constraint keeps wages from falling in the beginning of the transition period, which in turn dampens the response of inflation. While this has some negative impact on output during the transition phase, the overall effects are rather moderate compared to the simulations with ZLB.

5.3 The role of the distribution for the transition of shocks

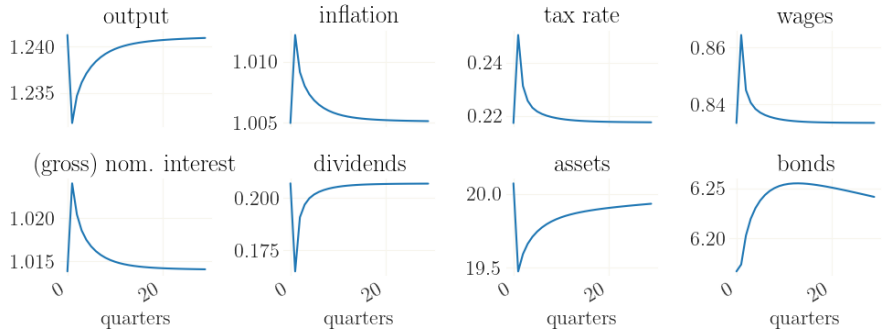

Figure 6: Nonlinear impulse responses to a one-time increase in government transfers. All measures are given in levels. Interest and inflation rates are given in quarterly gross-rates.

As a final exercise, let me focus on the role of the distribution of wealth for the transmission of transitory shocks, i.e. to one-time shocks without any persistence. For this purpose, let me simplify the setup considerably by abstracting from capital (i.e., $\alpha = 0$), labor unions ($\psi_w = 0$, $\mu_t^w = 1$) and price and wage indexation ($\omega_p = \omega_w = 0$). Additionally, I set the interest rate smoothing parameter ρ to zero. The model then collapses to the two-asset HANK pendant of the canonical 3-equation model and all persistence in response to economic shocks comes from temporary shifts in the distribution of wealth.

Consider again a one-time transfer of the size of 10% of GDP. As before, the government runs a balanced budget and the increase in transfers must be compensated by an increase in taxes. By assumption, all taxes in the model are labor taxes and the shock can thus right away be interpreted as a redistribution shock. Figure 6 shows the impulse responses to such redistributive shock. When taxes increase, households see their after-tax income decreasing and respond by lowering their labor supply. To counteract this, firms have to increase pre-tax wages. The increase in the tax rate thus acts like a classic distortionary wedge between supply and demand, causing hours worked and output to decline.

Additionally, the increase in real wages triggers an increase in prices, thereby causing inflation. This is different to the previous results since the effect coming from the depletion of the capital stock is absent in this model. The central bank reacts instantaneously by raising the nominal (and real) interest rate, which in turn causes two effects. First, the government has to further increase taxes to finance the higher interest payments on government debt. Second, the increase in the real rate triggers an increase in savings. Since dividends are falling due to lower revenues and higher costs, the return on assets falls and bonds become relatively more attractive than assets. Correspondingly the total volume of bonds increases while asset demand falls. However, the effect of lower dividends on equity causes aggregate wealth to decline overall.

Since the effect of the shock is short lived, the inequality effects of this measure are of second order. Households with a larger share of assets loose a small fraction of their wealth due to the fall in dividends. Since the transfer is only transitory, less productive households with no liquid assets save parts of their transfer. Overall, this causes a slight increase in inequality in asset holdings (top-10% hold more) and a

slight decrease in inequality in bond holdings (top-10% hold less). Households are slowly melting off their excess savings after the one-time transfer.

6 Conclusion

This paper introduces an iterative method to find nonlinear solutions to macroeconomic models with heterogeneous agents. The method is based on Newton iterations and levers the techique of automatic differentiation. I provide an easy-to-use reference implementation and suggest a series of central requirements of such an implementation. These are, among others, the consequent separation of economic model, solution code, and analysis, the generation of reusable code (across models), and adherence to the open-source philosophy.

The solution method is applied to study the nonlinear transition dynamics of a gradual but permanent change in government redistribution. The overall effects of such policy, both in the short and in the long run, are contractionary and can be aggravated by strong nonlinearities such as a lower bound on interest rates or downwards nominal wage rigidity.

References

Achdou, Y., Han, J., Lasry, J.M., Lions, P.L., Moll, B., 2022. Income and wealth distribution in macroeconomics: A continuous-time approach. The review of economic studies 89, 45–86.

Ahn, S., Kaplan, G., Moll, B., Winberry, T., 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., 2008. Solving heterogeneous-agent models with parameterized cross-sectional distributions. Journal of Economic Dynamics and Control 32, 875–908.

Auclert, A., 2019. Monetary policy and the redistribution channel. American Economic Review 109, 2333–67.

Auclert, A., Bardóczy, B., Rognlie, M., Straub, L., 2021. Using the sequence-space jacobian to solve and estimate heterogeneous-agent models. Econometrica 89, 2375–2408.

Auclert, A., Rognlie, M., 2017. Aggregate demand and the top 1 percent. American Economic Review 107, 588–92.

Auclert, A., Rognlie, M., 2018. Inequality and aggregate demand. Technical Report. National Bureau of Economic Research.

Azinovic, M., Gaegauf, L., Scheidegger, S., 2022. Deep equilibrium nets. International Economic Review 63, 1471–1525.

Bayer, C., Born, B., Luetticke, R., 2020. Shocks, Frictions, and Inequality in US Business Cycles. CEPR Discussion Papers 14364.

Bayer, C., Born, B., Luetticke, R., 2023. The liquidity channel of fiscal policy. Journal of Monetary Economics 134, 86–117.

Ben-Israel, A., 1965. A modified newton-raphson method for the solution of systems of equations. Israel journal of mathematics 3, 94–98.

Bianchi, F., Melosi, L., Rottner, M., 2021. Hitting the elusive inflation target. Journal of Monetary Economics 124, 107–122.

Boehl, G., 2022. An Ensemble MCMC Sampler for Robust Bayesian Inference. Available at SSRN 4250395. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4250395.

Boppart, T., Krusell, P., 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.

Calvo, G.A., 1983. Staggered prices in a utility-maximizing framework. Journal of Monetary Economics 12, 383–398.

Carroll, C.D., 2006. The method of endogenous gridpoints for solving dynamic stochastic optimization problems. Economics letters 91, 312–320.

De Ferra, S., Mitman, K., Romei, F., 2020. Household heterogeneity and the transmission of foreign shocks. Journal of International Economics 124, 103303.

Fair, R.C., Taylor, J.B., 1980. Solution and Maximum Likelihood Estimation of Dynamic Nonlinear RationalExpectations Models. Technical Report. National Bureau of Economic Research.

Fernández-Villaverde, J., Hurtado, S., Nuno, G., 2023. Financial frictions and the wealth distribution. Econometrica 91, 869–901.

Gornemann, N., Kuester, K., Nakajima, M., 2016. Doves for the rich, hawks for the poor? distributional consequences of monetary policy .

Greenwood, J., Hercowitz, Z., Huffman, G.W., 1988. Investment, capacity utilization, and the real business cycle. The American Economic Review 78, 402–417.

Gust, C., Herbst, E., López-Salido, D., Smith, M.E., 2017. The empirical implications of the interest-rate lower bound. American Economic Review 107, 1971–2006.

Gust, C.J., Herbst, E.P., López-Salido, J.D., Smith, M.E., 2012. The Empirical Implications of the Interest-Rate Lower Bound. Technical Report. Board of Governors of the Federal Reserve System (US).

Hagedorn, M., Luo, J., Manovskii, I., Mitman, K., 2019. Forward guidance. Journal of Monetary Economics 102, 1–23.

Hintermaier, T., Koeniger, W., 2010. The method of endogenous gridpoints with occasionally binding constraints among endogenous variables. Journal of Economic Dynamics and Control 34, 2074–2088.

Juillard, M., Laxton, D., McAdam, P., Pioro, H., 1998. An algorithm competition: First-order iterations versus newton-based techniques. Journal of Economic Dynamics and Control 22, 1291–1318.

Juillard, M., et al., 1996. Dynare: A program for the resolution and simulation of dynamic models with forward variables through the use of a relaxation algorithm. volume 9602. Citeseer.

Kahou, M.E., Fernández-Villaverde, J., Perla, J., Sood, A., 2021. Exploiting symmetry in high-dimensional dynamic programming. Technical Report. National Bureau of Economic Research.

Kaplan, G., Moll, B., Violante, G.L., 2018. Monetary policy according to HANK. NBER Working Papers 3. National Bureau of Economic Research, Inc. URL: https://ideas.repec.org/p/nbr/nberwo/21897.html.

Klenow, P.J., Kryvtsov, O., 2008. State-dependent or time-dependent pricing: Does it matter for recent us inflation? The Quarterly Journal of Economics 123, 863–904.

Krueger, D., Mitman, K., Perri, F., 2015. Macroeconomics and heterogeneity, including inequality. Handbook of Macroeconomics (forthcoming) .

Laffargue, J.P., 1990. Résolution d'un modèle macroéconomique avec anticipations rationnelles. Annales d'Economie et de Statistique , 97–119.

Lindé, J., Trabandt, M., 2018. Should we use linearized models to calculate fiscal multipliers? Journal of Applied Econometrics 33, 937–965.

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

McKay, A., Nakamura, E., Steinsson, J., 2016. The power of forward guidance revisited. American Economic Review 106, 3133–3158.

Mises, R., Pollaczek-Geiringer, H., 1929. Praktische verfahren der gleichungsauflösung. ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 9, 58–77.

Petrosky-Nadeau, N., Zhang, L., Kuehn, L.A., 2018. Endogenous disasters. American Economic Review 108, 2212–45.

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

Reiter, M., 2023. State reduction and second-order perturbations of heterogeneous agent models. Technical Report. IHS Working Paper.

Rotemberg, J.J., 1982. Monopolistic price adjustment and aggregate output. The Review of Economic Studies 49, 517–531.

Saad, Y., Schultz, M.H., 1986. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on scientific and statistical computing 7, 856–869.

Scheidegger, S., Bilionis, I., 2019. Machine learning for high-dimensional dynamic stochastic economies. Journal of Computational Science 33, 68–82.

Smets, F., Wouters, R., 2007. Shocks and frictions in us business cycles: A bayesian dsge approach. American Economic Review 97, 586–606.

Van der Vorst, H.A., 1992. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. SIAM Journal on scientific and Statistical Computing 13, 631–644.

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

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, 36–41.

Appendix A Models

This section presents the three main models used throughout the paper and which are provided with the package. A fourth model, applied in Section 5, is the two-asset-model without capital, which is a special case of the two-asset-model reseted below. Wherever possible, the three models share the same parameters, which are given in Table A.3.

Appendix A.1 A medium scale RANK model

This model is based on Gust et al. (2012), which is an early working paper version of Gust et al. (2017). Relative to this reference it contains a series of simplifications, e.g. no growth in steady state. The model features all the bells and whistles of the medium-scale workhorse model of Smets and Wouters (2007) but uses Rotemberg pricing instead of Calvo pricing. This model also forms the basis for the aggregate equations for the two HANK models.

Households

$$\Lambda_t = \frac{1}{c_t - hc_{t-1}} - \frac{h\beta_t}{E_t c_{t+1} - hc_t} \tag{A.1}$$

$$\Lambda_t = \beta_t \epsilon_t^{\Lambda} E_t \left\{ \Lambda_{t+1} \frac{R_t}{\pi_{t+1}} \right\} \tag{A.2}$$

$$d_t + c_t + \tau_t + \frac{\psi_w}{2}\left(\frac{\pi_t^w}{\tilde{\pi}_t^w} - 1\right)^2 = w_t n_t + R_{t-1}/\pi_t d_{t-1} + \Pi_t + \Pi_t^b \tag{A.3}$$

The last equation is the households budget constraint which not necessary for the aggregate dynamics due to Walras' law.

Labor Unions

$$\psi_w \left(\frac{\pi_t^w}{\tilde{\pi}_t^w} - 1\right)\frac{\pi_t^w}{\tilde{\pi}_t^w} = \psi_w \beta_t E_t \left\{ \left(\frac{\pi_{t+1}^w}{\tilde{\pi}_{t+1}^w} - 1\right)\frac{\pi_{t+1}^w}{\tilde{\pi}_{t+1}^w} \right\} + n_t \frac{\mu_t^w}{\mu_t^w - 1}\left(\chi n_t^{\sigma_l} - \Lambda_t w_t/\mu_t^w\right) \tag{A.4}$$

$$\pi_t^w = \frac{w_t^n}{w_{t-1}^n}\pi_t \tag{A.5}$$

$$\ln \tilde{\pi}_t^w = \omega_w \ln \bar{\pi}^w + (1 - \omega_w) \ln \pi_{t-1}^w \tag{A.6}$$

$$w_t = \max\left\{ \iota_w \frac{w_{t-1}}{\pi_t}, w_t^n \right\} \tag{A.7}$$

Firms

$$\psi\left(\frac{\pi_t}{\tilde{\pi}_t}-1\right)\frac{\pi_t}{\tilde{\pi}_t} = \frac{1}{1-\mu_t}+\frac{\mu_t}{\mu_t-1}mc_t+\psi E_t\left\{\beta_{t+1}\frac{\Lambda_{t+1}}{\Lambda_t}\left(\frac{\pi_{t+1}}{\tilde{\pi}_t}-1\right)\frac{\pi_{t+1}}{\tilde{\pi}_t}\frac{y^f_{t+1}}{y^f_t}\right\} \tag{A.8}$$

$$\ln\tilde{\pi}_t = \omega\ln\bar{\pi}+(1-\omega)\ln\pi_{t-1} \tag{A.9}$$

$$y^f_t = (u_t k_{t-1})^{\alpha}(z_t n_t)^{1-\alpha} \tag{A.10}$$

$$k_t = (1-\delta)k_{t-1}+\epsilon^i_t\left(1-\frac{\psi_i}{2}\left(\frac{i_t}{i_{t-1}}-1\right)^2\right)i_t \tag{A.11}$$

$$1 = q_t\epsilon^i_t\left(1-\frac{\psi_i}{2}\left(\frac{i_t}{i_{t-1}}-1\right)^2-\psi_i\left(\frac{i_t}{i_{t-1}}-1\right)\frac{i_t}{i_{t-1}}\right)+\beta_t\epsilon^i_{t+1}\frac{\Lambda_{t+1}}{\Lambda_t}q_{t+1}\psi_i\left(\frac{i_{t+1}}{i_t}-1\right)\left(\frac{i_{t+1}}{i_t}\right)^2 \tag{A.12}$$

$$q_{t-1}\frac{R_t}{\pi_{t+1}} = MPK_t u_t+(1-\delta)q_t-C(u_{t-1}) \tag{A.13}$$

$$w_t = (1-\alpha)mc_t\frac{y^f_t}{n_t} \tag{A.14}$$

$$MPK_t = \alpha mc_t\frac{y^f_t}{(u_t k_{t-1})} \tag{A.15}$$

$$C(u_t) = \bar{MPK}(u_t-1)+\frac{1}{2}\frac{\psi_u}{1-\psi_u}(u_t-1)^2 \tag{A.16}$$

$$\psi_u(u_t-1) = (1-\psi_u)(MPK_t-\bar{MPK}) \tag{A.17}$$

$$\Pi_t = \left(1-mc_t-\frac{\psi_p}{2}\left(\frac{\pi_t}{\tilde{\pi}_t}-1\right)^2\right)y^f_t-\left(1-q_t\left(1-\frac{\psi_i}{2}\left(\frac{i_t}{i_{t-1}}-1\right)^2\right)\right)i_t \tag{A.18}$$

Financial sector

$$d_t = q^b_t b_t+q_t k_t \tag{A.19}$$

$$R_t = \frac{(1+\kappa q^b_{t+1})}{q^b_t} \tag{A.20}$$

$$\Pi^b_t = \left((1+\kappa q^b_t)b_{t-1}+R_{t-1}q_{t-1}k_{t-1}-R_{t-1}d_{t-1}\right)/\pi_t \tag{A.21}$$

Government

$$q^b_t b_t+\tau_t = g_t+\frac{(1+\kappa q^b_t)}{\pi_t}b_{t-1} \tag{A.22}$$

$$b_t = \frac{\bar{y}}{\bar{q}^b} \tag{A.23}$$

$$\ln R^n_t = \rho\ln R^n_{t-1}+(1-\rho)\left(\ln R^*_t+\phi_\pi\left[\ln\pi_t-\ln\bar{\pi}\right]+\phi_y\left[\ln y_t-\ln\bar{y}\right]\right)+\ln v_t \tag{A.24}$$

$$R_t = \max\left\{1,\ R^n_t\right\} \tag{A.25}$$

Clearing conditions

$$c_t + i_t + g_t + u_t k_{t-1} + \frac{\psi_w}{2}\left(\frac{\pi_t^w}{\tilde{\pi}_t^w} - 1\right)^2 = \left(1 - \frac{\psi}{2}\left(\frac{\pi_t}{\pi_{ss}} - 1\right)^2\right) y_t^f \tag{A.26}$$

$$c_t + i_t + g_t = y_t \tag{A.27}$$

Exogenous processes

$$\ln \beta_t = (1-\rho_\beta)\ln\bar{\beta} + \rho_\beta \ln \beta_{t-1} + \varepsilon_t^\beta \tag{A.28}$$

$$\ln z_t = (1-\rho_z)\ln\bar{z} + \rho_z \ln z_{t-1} + \varepsilon_t^z \tag{A.29}$$

$$\ln \mu_t^w = (1-\rho_w)\ln\bar{\mu}^w + \rho_w \ln \mu_{t-1}^w + \varepsilon_t^w \tag{A.30}$$

$$\ln \mu_t = (1-\rho_p)\ln\bar{\mu} + \rho_p \ln \mu_{t-1} + \varepsilon_t^p \tag{A.31}$$

$$\ln g_t = (1-\rho_g)\ln(0.2\bar{y}) + \rho_g \ln g_{t-1} + \varepsilon_t^g \tag{A.32}$$

$$\ln \epsilon_t^i = \rho_i \ln \epsilon_{t-1}^i + \varepsilon_t^i \tag{A.33}$$

$$\ln \epsilon_t^\Lambda = \rho_\Lambda \ln \epsilon_{t-1}^\Lambda + \varepsilon_t^\Lambda \tag{A.34}$$

$$\ln R_t^* = (1-\rho_r)\ln\bar{R} + \rho_r \ln R_{t-1}^* + \varepsilon_t^r \tag{A.35}$$

$$\tag{A.36}$$

Parameters

The parameters are set as in Table A.3.

Appendix A.2 A small scale HANK model with one asset

Households can hold one type of assets a_{it} and face idiosyncratic income risk and a borrowing constraint. They have GHH preferences with the composite good $x_{i,t}$, and the Bellman equation is given by

$$V_t(e_{it}, a_{i,t-1}) = \max_{c_{it}, n_{it}, a_{it}} \left\{ \frac{x_{it}^{1-\sigma_c}}{1-\sigma_c} + \beta \mathbb{E}_t \left[V_{t+1}(e_{i,t+1}, a_{it}) | e \right] \right\} \tag{A.37}$$

$$x_{it} = c_{it} - e_{it} \frac{n_{it}^{1+\sigma_l}}{1+\sigma_l} \tag{A.38}$$

$$c_{it} + a_{it} = \frac{R_{t-1}}{\pi_t} a_{i,t-1} + w_t e_{it} n_{it} - \tau_t \bar{\tau}(e_{it}) + \Pi_t \bar{\Pi}(e_{it}) \tag{A.39}$$

$$a_{it} \geq 0 \tag{A.40}$$

where e_{it} is i's household-specific productivity which follows an AR(1) process in logs as in (5). $\bar{\tau}(e)$ and $\bar{\Pi}(e)$ are skill-specific incidence rules for taxes and dividends.

The aggregate model is as in Appendix A.1 but parameters are chosen such that labor is the only production factor (i.e. no capital accumulation with $\alpha = 0$) and such that there are no price inertia in the Phillips curve ($\omega = 0$). Dividends are given by (9). The government is running a balanced budget with

$$\tau_t = \left(\frac{R_{t-1}}{\pi_t} - 1 \right) \bar{B} + g_t. \tag{A.41}$$

Parameter		Value	Target
σ_l	inverse Frisch elasticity of labour supply	2	
χ	weight on the disutility of labour	–	$\bar{n} = 0.33$
β	steady state discount factor	0.995	
θ	elasticity of substitution	6	
θ_w	elasticity of substitution for wages	11	
κ	decay parameter for coupon payments of perpetual bonds	0.975	
δ	depreciation rate	0.025	
h	habit formation parameter	0.74	
ψ_i	parameter on the costs of investment adjustment	5.6	
ψ_p	parameter on the costs of price adjustment	60.	
ψ_w	parameter on the costs of wage adjustment	96.	
ψ_u	parameter on the capital utilisation costs	0.8	
α	capital income share	0.33	
π^*	inflation target	$1.02^{\frac{1}{4}}$	
$\phi_p i$	Monetary policy rule coefficient on inflation	1.5	
ϕ_y	Monetary policy rule coefficient on output	0.1	
ρ	persistence in (notional) nominal interest rate	0.8	
ω_p	coefficient on steady state inflation in price indexation	0.44	
ω_w	coefficient on steady state wage inflation in wage indexation	0.66	
ι_w	degree of dowards nominal wage rigidity	1.	
ρ_β	persistence of discount factor shock	0.9	
ρ_z	persistence of technology shocks	0.9	
ρ_p	persistence of price MU shock	0.9	
ρ_w	persistence of wage MU shock	0.9	
ρ_g	persistence of government spending shock	0.9	
ρ_i	persistence of MEI shock	0.9	
ρ_r	persistence of MP shock	0.9	
ρ_u	persistence of wage MU shock	0.9	

Table A.3: Joint parameters

I further abstract from labor unions and thus, due to GHH preferences, labor supply simplifies to

$$n_t^{\sigma_l} = w_t. \tag{A.42}$$

Markets clear with

$$\int c_{it} di = C_t = \left(1 - \frac{\psi}{2}\left(\frac{\pi_t}{\bar{\pi}} - 1\right)^2\right) y_t, \tag{A.43}$$

$$\int a_{it} di = \bar{B}, \tag{A.44}$$

and the parameters specific to this model are given in Table A.4.

Parameter		Value
σ_c	intertemporal elasticity of substitution	2
$\bar{\beta}$	discount factor	0.98
$\bar{B}$	bond supply	5.6
α	capital factor share	0
ω_p	coefficient on steady state inflation in price indexation	1
$\bar{a}$	borrowing constraint	0
σ_e	standard error of earnings	0.6
ρ_e	autocorrelation of earnings	0.966
n_e	points for Markov chain of e	4
n_a	points for asset grid	50

Table A.4: Parameters specific to the one-asset-HANK model.

Appendix A.3 A medium scale HANK model with two assets

The two-asset HANK model shares many of the aggregate features with the representative agent model in Appendix A.1 and is presented in Section 2. A central difference is the setup of households. Based on the endogenous grid method of Carroll (2006), the appendix of Auclert et al. (2021) describes an efficient algorithm to solve the two-asset household problem with convex adjustment costs. All equations that are not stated in 2 are as in the RANK model, including the exogenous processes from Equations (A.28)-(A.35). Parameters specific to the two-asset HANK model are given in Table A.5.

Parameter		Value
σ_c	intertemporal elasticity of substitution	2
σ_l	inverse Frisch elasticity of labour supply	2.9
χ	weight on the disutility of labour	0.5
ψ_p	parameter on the costs of price adjustment	60
ψ_w	parameter on the costs of price adjustment	96
$\psi_a 0$	parameter on portfolio adjustment no.1	0.25
$\psi_a 1$	parameter on portfolio adjustment no.2	15
$\psi_a 2$	parameter on portfolio adjustment no.3	2
ζ	liquidity premium	0.005
B_G	government bond supply	2.8
$\bar{\beta}$	discount factor	0.98
$\bar{T}$	steady state government transfers	1e-5
ρ_T	autocorrelation government transfers	0.8
$\bar{b}$	borrowing constraint	0
σ_e	standard error of earnings	0.92
ρ_e	autocorrelation of earnings	0.966
ζ	steady state liquidity premium	0.1
n_e	points for Markov chain of e	3
n_b	points for liquid asset grid	20
n_a	points for illiquid asset grid	25

Table A.5: Parameters specific to the two-asset HANK model.

Appendix B Proof of part iii) of Lemma 1

The Lemma states that for a real square matrix M and a vector $\mathbf{z}$ with $\|\mathbf{z}\| > 0$ it holds that

$$|R(M, \mathbf{z})| \in [0, \sigma_{\max}], \tag{B.1}$$

where $\sigma_{\max}$ is the largest singular value of M.

Proof. It is well known that

$$R(M^\intercal M, \mathbf{z}) \in [\sigma^2_{\min}, \sigma^2_{\max}], \tag{B.2}$$

with $\sigma_{\min}$ as the respective smallest singular value of M. The result from the Lemma follows immediately if we can show that

$$R(M^\intercal M, \mathbf{z}) \geq R(M, \mathbf{z})^2. \tag{B.3}$$

Define $\mathbf{w} = M\mathbf{z}$. Then the above is is equivalent to

$$\frac{\mathbf{z}^\intercal M^\intercal M \mathbf{z}}{\mathbf{z}^\intercal \mathbf{z}} - \frac{\mathbf{z}^\intercal M \mathbf{z} \mathbf{z}^\intercal M \mathbf{z}}{(\mathbf{z}^\intercal \mathbf{z})^2} = \frac{\mathbf{w}^\intercal \mathbf{w}}{\mathbf{z}^\intercal \mathbf{z}} - \frac{\mathbf{z}^\intercal \mathbf{w} \mathbf{z}^\intercal \mathbf{w}}{(\mathbf{z}^\intercal \mathbf{z})^2} \geq 0, \tag{B.4}$$

$$\mathbf{w}^\intercal \mathbf{w} - \frac{\mathbf{w}^\intercal \mathbf{z} \mathbf{z}^\intercal \mathbf{w}}{\mathbf{z}^\intercal \mathbf{z}} \geq 0, \tag{B.5}$$

$$\mathbf{w}^\intercal \left(\mathbf{I} - \frac{\mathbf{z}\mathbf{z}^\intercal}{\mathbf{z}^\intercal \mathbf{z}} \right) \mathbf{w} \geq 0, \tag{B.6}$$

which uses the fact that $R(M, \mathbf{z}) = R(M^\intercal, \mathbf{z})$. It is further that

$$\mathbf{w}^\intercal \mathbf{w} - \frac{\mathbf{w}^\intercal \mathbf{z}\mathbf{z}^\intercal \mathbf{w}}{\mathbf{z}^\intercal \mathbf{z}} = \mathbf{w}^\intercal \mathbf{w} \left(1 - \frac{(\mathbf{w}^\intercal \mathbf{z})^2}{(\mathbf{z}^\intercal \mathbf{z})(\mathbf{w}^\intercal \mathbf{w})}\right). \tag{B.7}$$

Since from the Cauchy-Schwarz inequality we have

$$(\mathbf{w}^\intercal \mathbf{z})^2 \geq (\mathbf{z}^\intercal \mathbf{z})(\mathbf{w}^\intercal \mathbf{w}), \tag{B.8}$$

and thus

$$\frac{(\mathbf{w}^\intercal \mathbf{z})^2}{\mathbf{z}^\intercal \mathbf{z} \mathbf{w}^\intercal \mathbf{w}} < 1, \tag{B.9}$$

it follows that

$$\mathbf{w}^\intercal \left(\mathbf{I} - \frac{\mathbf{z}\mathbf{z}^\intercal}{\mathbf{z}^\intercal \mathbf{z}}\right) \mathbf{w} \geq 0. \tag{B.10}$$

■

Appendix C A generic syntax to express heterogeneous agent models

Using the formalization from Section 3, the necessary user input to describe a heterogeneous agent model can be reduced to two elements: the EGM step manifesting in the function $W(\cdot)$ from Equation (19) and the n aggregate equations in $f(\cdot)$ from Equation (21). In contrast, it is typically not necessary to explicitly specify the mapping from agents' decisions to the distribution, $D(\cdot)$, from (20) since this function is generic and standard routinen such as, e.g., the lottery method of Young (2010) can be used.

```
# EGM stage: marginal values & decisions
decisions:
  inputs: [WaPrime,WbPrime]
  calls: |
    z_grid = income(skills_grid, tax, w, n, transfers)
    Psi = marginal_cost_grid(a_grid, Ra−1, psi_a0, psi_a1, psi_a2)
    WaPrimeExp = expect_transition(skills_transition, WaPrime)
    WbPrimeExp = expect_transition(skills_transition, WbPrime)
    Wa, Wb, a, b, c, uce = egm_step(WaPrimeExp, WbPrimeExp, a_grid, b_grid, z_grid, skills_grid, kappa_grid, \
       beta, sigma_c, Rb−1, Ra−1, psi_a0, psi_a1, psi_a2, Psi)
  outputs: [a,b,c,uce]

# intermediate stage: aggregation
aux_equations: |
    # calculate asset share of top−10%
    top10a = 1 - percentile(a, dist, .9)
    # aggregation
    UCE = sum(dist*uce, axis=(0,1,2))
    ...

# main stage: aggregate equations
equations:
    ~ psi_w*(piwn/piwntilde - 1)*piwn/piwntilde = wage_markup/(wage_markup−1)*chi*n**(1+sigma_l) + \
          1/(1−wage_markup)*(1 - tax)*w*n*UCE + \
          psi_w*beta*(piwnPrime/piwntildePrime - 1)*piwnPrime/piwntildePrime # wage Phillips curve
    ~ piwn = wn/wnLag*pi # wage inflation
    ~ w = max(iota*wLag/pi, wn) # dowards nominal wage rigidity

    ~ div = (1 - psi_p/2*(pi/pitilde - 1)**2)*y - w * n - i # dividends
    ~ Rb = Rr - zeta # real bond returns
    ~ Ra = assetshareLag * (div + equity) / equityLag + (1 - assetshareLag) * Rr # real asset returns
    ...
```

Figure C.7: Part of the YAML-file which specifies the two-asset HANK from 2. The block `decisions` represents the function $W(\cdot)$ which depends on w_{t+1} (here: `WaPrime` and `WbPrime`) and aggregate variables such as wages `w`, labor hours `n`, and parameters such as σ_c (here `sigma_c`). The outputs are disaggregated savings a_{it} and b_{it}, consumption c_{it} and marginal utilities. The equations block shows the first aggregate equations starting with Equation (7).

The reference implementation provides a simple syntax for expressing heterogeneous agent models, which is based on the widely used YAML format.[34] Similar as the mod-file in Dynare, the file allows to specify variables and parameters as well as meta-parameters such as, e.g., the grids used to represent the distribution of idiosyncratic states across agents. Importantly, the package – via the YAML file – permits a standardized way to specify the function $W(\cdot)$ including its inputs and outputs. This is illustrated in Figure C.7 (key: "`decisions`") for a part of the specification of the one-asset-HANK model from Appendix A. Subfunctions of $W(\cdot)$ (e.g. the function `egm_step`) can be described as a conventional Python function that is defined in an external functions file and referenced in the YAML. In the example, `Wa` and `Wb` are the recursive decision object w_t from Section 3.1 while `a,b,c,uce` are the agents actions in a_t.

[34] YAML ("Yet Another Markup Language") and is a standardized human-readable data-serialization language. The format is similar to XML but has a minimal syntax in order to be easily usable. It is useful to provide data input in a clear and simple way across programming languages, and is widely used in applications that require a high level human-computer interaction, such as configuration files or data storage.

The figure further shows a part of the system of aggregate equations ("`equations`"), which constitute the function $f(\cdot)$.[35] Together with the specification of $W(\cdot)$, this allows the combination of the disaggregated parts of various heterogeneous agents models with arbitrary systems of aggregate equations. Additionally, the file further allows to stage parameter values and the inputs required for the steady state search (not shown in the figure).

Appendix D Representative agent models and automatic differentiation

This appendix is merely given for completeness and documents how the software implementation handles nonlinear *representative* agent models. Analog to the representation in Section 3 for heterogeneous agent models, a representative agent model can simply be written as

$$z_t = f(x_{t-1}, x_t, x_{t+1}), \tag{D.1}$$

where the solution requires $z_t = \mathbf{0}$. For simplicity, denote the three Jacobians of this function as

$$f_A(x) = \partial f / \partial x_{t-1}, \tag{D.2}$$
$$f_B(x) = \partial f / \partial x_t, \tag{D.3}$$
$$f_C(x) = \partial f / \partial x_{t+1}. \tag{D.4}$$

For each $f(x_{t-1}, x_t, x_{t+1}) = z_t$, these three Jacobians can be obtained at low costs via backwards propagation with only n evaluations of f. Since $dim(z_t) = n \ll dim(\mathbf{z}) = n(T-1)$, each evaluation is computationally cheap. The sequence space Jacobian is then given by the block tridiagonal matrix

$$J(\mathbf{x}) = \begin{bmatrix} f_B(x_1) & f_C(x_2) & & & & & \\ f_A(x_1) & f_B(x_2) & f_C(x_3) & & & & \\ & f_A(x_2) & f_B(x_3) & f_C(x_4) & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & f_A(x_{T-3}) & f_B(x_{T-2}) & f_C(x_{T-1}) \\ & & & & f_A(x_{T-2}) & f_B(x_{T-1}) \end{bmatrix}. \tag{D.5}$$

During each Newton iteration we seek to solve

$$J(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i) = -F(\mathbf{x}_i) = -\mathbf{z}_i. \tag{D.6}$$

Following the ideas of Laffargue (1990) and Juillard et al. (1996), this can be solved efficiently by bringing $J(\mathbf{x})$ in a block bidiagonal form

$$\hat{J} = \begin{bmatrix} I & M_1 & & & & \\ & I & M_2 & & & \\ & & I & M_3 & & \\ & & & \ddots & \ddots & \\ & & & & I & M_{T-2} \\ & & & & & I \end{bmatrix}. \tag{D.7}$$

[35] For example, expressing the aggregate relationships in $f(\cdot)$ is, in its core, what the "model" block in Dynare's mod-file does.

This can be done by initializing $M_0 = \mathbf{0}$ and $\hat{z}_0 = \mathbf{0}$ and for $j \in 1, 2, \ldots, T-1$ setting

$$K_j = f_B(x_j) - f_A(x_j)M_{j-1}, \tag{D.8}$$

$$M_j = K_j^{-1} f_C(x_j), \tag{D.9}$$

$$\hat{z}_j = K_j^{-1} z_j - f_A(x_j)\hat{z}_{j-1}, \tag{D.10}$$

which is equivalent to recursively solving for and subtracting each a pair of equations in $J(\mathbf{x})$. Each y_t in $\mathbf{y}_i = \mathbf{x}_{j+1} - \mathbf{x}_i$ can then simply be found via the recursion

$$y_t = \hat{z}_t - M_t y_{t+1}. \tag{D.11}$$