NBER WORKING PAPER SERIES

# USING THE SEQUENCE-SPACE JACOBIAN TO SOLVE AND ESTIMATE HETEROGENEOUS-AGENT MODELS

Adrien Auclert
Bence Bardóczy
Matthew Rognlie
Ludwig Straub

Working Paper 26123
http://www.nber.org/papers/w26123

NATIONAL BUREAU OF ECONOMIC RESEARCH
1050 Massachusetts Avenue
Cambridge, MA 02138
July 2019, Revised November 2020

For helpful comments, we thank the editor Gianluca Violante, four anonymous referees, as well as Riccardo Bianchi-Vimercati, Luigi Bocola, Michael Cai, Jesus Fernández-Villaverde, Joao Guerreiro, Kurt Mitman, Ben Moll, Laura Murphy, Martin Souchier, and Christian Wolf. Martin Souchier also provided outstanding research assistance. This research is supported by the National Science Foundation grant SES-1851717. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

NBER working papers are circulated for discussion and comment purposes. They have not been peer-reviewed or been subject to the review by the NBER Board of Directors that accompanies official NBER publications.

Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models
Adrien Auclert, Bence Bardóczy, Matthew Rognlie, and Ludwig Straub
NBER Working Paper No. 26123
July 2019, Revised November 2020
JEL No. C63,E21,E32


## ABSTRACT

We propose a general and highly efficient method for solving and estimating general equilibrium heterogeneous-agent models with aggregate shocks in discrete time. Our approach relies on the rapid computation of sequence-space Jacobians—the derivatives of perfect-foresight equilibrium mappings between aggregate sequences around the steady state. Our main contribution is a fast algorithm for calculating Jacobians for a large class of heterogeneous-agent problems. We combine this algorithm with a systematic approach to composing and inverting Jacobians to solve for general equilibrium impulse responses. We obtain a rapid procedure for likelihood-based estimation and computation of nonlinear perfect-foresight transitions. We apply our methods to three canonical heterogeneous-agent models: a neoclassical model, a New Keynesian model with one asset, and a New Keynesian model with two assets.



Adrien Auclert
Department of Economics
Stanford University
579 Serra Mall
Stanford, CA 94305-6072
and NBER
aauclert@stanford.edu

Matthew Rognlie
Department of Economics
Northwestern University
2211 Campus Drive
Evanston, IL 60208
and NBER
matthew.rognlie@northwestern.edu

Bence Bardóczy
Northwestern University
Department of Economics
2211 Campus Drive
Evanston, IL 60208
bardoczy@u.northwestern.edu

Ludwig Straub
Department of Economics
Harvard University
Littauer 211
Cambridge, MA 02139
and NBER
ludwigstraub@fas.harvard.edu


A repository for code that automates and applies methods is available at https://github.com/shade-econ/sequence-jacobian

# 1 Introduction

A rapidly expanding literature in macroeconomics incorporates rich heterogeneity into dynamic general equilibrium models. A central challenge in this literature is that equilibrium involves the time-varying, high-dimensional distribution of agents over their state variables.

In this paper, we propose a general, systematic, and highly efficient method to deal with this challenge. Our method follows Reiter (2009) by perturbing the model to first order in aggregates. But, while the Reiter method writes equilibrium as a system of linear equations in the *state space*, we instead write it as a system of linear equations in the space of perfect-foresight sequences—the *sequence space*. Since the size of this system is independent of the size of the state space, it becomes feasible to solve and estimate models that feature very rich heterogeneity. Our sequence-space approach builds on Boppart, Krusell and Mitman (2018), who solve for nonlinear impulse responses to small shocks, but we obtain a much faster solution by directly exploiting linearity.

We demonstrate the power of our method by solving and estimating three models, with increasing degrees of complexity, at unparalleled speed. A code repository accompanies this paper and provides general-purpose routines that automate the new algorithms we introduce.[1]

The central objects in our method are *sequence-space Jacobians*: the derivatives of equilibrium mappings between aggregate sequences around the steady state. These Jacobians summarize every aspect of the model that is relevant for general equilibrium. For example, consider a standard incomplete markets model. That model features a Jacobian $\mathcal{J}^{C,r}$ that maps, to first order, changes in the sequence of real interest rates $\{r_t\}$ to changes in the sequence of aggregate consumption $\{C_t\}$. Under the hood, this mapping includes the heterogeneous responses of households to changes in $r$, as well as the evolution over time of the distribution of agents that follows from this change in $r$. But to know the aggregate effect of $r$ on $C$, all we need to know is $\mathcal{J}^{C,r}$: it is a *sufficient statistic*. Our method exploits this property. We compute all relevant sequence-space Jacobians, and then compose and invert these Jacobians to obtain the model's full set of impulse responses.

Our main contribution is a fast algorithm for computing Jacobians for a large class of heterogeneous-agent problems, truncated to a horizon of $T \times T$. A direct approach to calculating these Jacobians is quite costly. For instance, to calculate $\mathcal{J}^{C,r}$, this direct approach requires iterating backward to obtain the consumption policy at every date, then iterating forward on the distribution of assets to date $T-1$, given a shock to $r_s$ at each of the $T$ dates $s = 0 \dots T-1$. Our method, by contrast, exploits the structure of the linearized heterogeneous-agent problem around the steady state, which we capture formally in proposition 1. It requires only a single backward iteration from $s = T-1$ to obtain the consumption policy and impulses to the distribution. These objects are then efficiently combined with information from the steady-state solution to form the full Jacobian, lowering the cost by a factor of about $T$ relative to the direct approach. Our algo-

[1] See `https://github.com/shade-econ/sequence-jacobian`, which provides routines written in Python, as well as pedagogical notebooks. A separate replication archive uses these routines to produce all figures and tables presented in this paper.

rithm therefore provides a dramatic speed improvement, since $T$ is typically at least equal to 300 in practice, and sometimes as large as 1000.

We combine this algorithm with a systematic approach to composing and inverting Jacobians to solve for general equilibrium impulse responses. Equilibrium in the sequence space can always be expressed as a solution to a certain nonlinear system

$$\mathbf{H}(\mathbf{U}, \mathbf{Z}) = 0, \tag{1}$$

where $\mathbf{U}$ represents the time path of unknown aggregate sequences (usually aggregate prices and quantities) and $\mathbf{Z}$ represents the time path of exogenous shocks. Obtaining the impulse responses of unknowns to shocks, $d\mathbf{U} = -\mathbf{H}_{\mathbf{U}}^{-1}\mathbf{H}_{\mathbf{Z}}d\mathbf{Z}$, requires computing the Jacobians $\mathbf{H}_{\mathbf{U}}$ and $\mathbf{H}_{\mathbf{Z}}$, which are formed by combining Jacobians from different parts of the model. Starting from the heterogeneous-agent Jacobians computed using our fast algorithm, this can be achieved by any method that systematically applies the chain rule. We propose one such method, forward accumulation along a directed acyclic graph (DAG). This procedure can be automated, and it usually only takes a few milliseconds.

We verify that our method is accurate by showing that it delivers exactly the same solution as the Reiter method, for models where the Reiter method is feasible. Like all perturbation methods, both our method and the Reiter method are subject to error in taking derivatives; to allow for a precise comparison, we therefore use automatic differentiation to take error-free derivatives in both methods. We then demonstrate accuracy in two ways. First, we show that in response to specific 1% shocks, impulse responses under the two methods differ everywhere by less than $10^{-9}$%. Second, we provide a method to recover the state-space law of motion from our sequence-space solution, and show that matrices in this law of motion differ from the same matrices obtained using the Reiter method by a maximum of less than $10^{-8}$. With accuracy established, we additionally discuss how varying the truncation horizon $T$, or replacing automatic with numerical differentiation, can affect these errors.

In sum, our method enables researchers to obtain model Jacobians and linearized general equilibrium impulse responses, accurately and rapidly, in models with heterogeneous agents that can potentially be very complex. To show how these objects can be used in practice, we cover two applications that are very common in applied research: estimation on time-series data, and computation of nonlinear perfect-foresight transitions.

To build toward estimation, we first summarize the Boppart, Krusell and Mitman (2018) simulation procedure. As they point out, the linearized impulse responses to shocks truncated to a horizon of $T$ form an $MA(T-1)$ representation of the model with aggregate shocks, which yields a straightforward simulation procedure. These sample paths can be used to calculate approximate time-series moments, which in principle can then be used for estimation.

We next provide an alternative route, which is to use analytical formulas to calculate the model's time-series moments directly from impulse responses. From here, we can directly compute the likelihood function of any empirical time series. Given a prior over parameters, we can

Table 1: Summary of computing times.

| Computing times for: | req. | Krusell-Smith | HD Krusell-Smith | one-asset HANK | two-asset HANK |
|---|---|---|---|---|---|
| Steady state (s.s.) | | 0.42 s | 52.08 s | 1.88 s | 16.16 s |
| Heterogeneous-agent Jacobians ($\mathcal{J}$) | s.s. | 0.09 s | 10.47 s | 0.32 s | 3.50 s |
| One impulse response | $\mathcal{J}$ | 0.0009 s | 0.0009 s | 0.0136 s | 0.0263 s |
| All impulse responses (**G**) | $\mathcal{J}$ | 0.0033 s | 0.0033 s | 0.0506 s | 0.1735 s |
| Simulation (100,000 periods) | **G** | 0.0040 s | 0.0050 s | 0.0219 s | 0.1047 s |
| Bayesian estimation (shocks) | **G** | | | | |
| single likelihood evaluation | | 0.00064 s | 0.00068 s | 0.00235 s | 0.0140 s |
| obtaining posterior mode | | 0.06 s | 0.06 s | 0.66 s | 16.22 s |
| RWMH (100,000 draws) | | 66 s | 66 s | 284 s | 1450 s |
| Bayesian estimation (shocks + model) | $\mathcal{J}$ | | | | |
| single likelihood evaluation | | — | — | 0.056 s | 0.227 s |
| obtaining posterior mode | | — | — | 14 s | 522 s |
| RWMH (100,000 draws) | | — | — | 5609 s | 21282 s |
| Nonlinear impulse responses | $\mathcal{J}$ | 0.32 s | 27.85 s | 1.17 s | 14.63 s |
| No. of idiosyncratic states | | 3,500 | 250,000 | 3,500 | 10,500 |
| Time horizon ($T$) | | 300 | 300 | 300 | 300 |
| No. of shock parameters in estimation | | 3 | 3 | 6 | 14 |
| No. of model parameters in estimation | | 0 | 0 | 3 | 5 |

*Notes.* The times given are incremental, with the "req." column denoting the prerequisite step for each computation. RWMH refers to Random Walk Metropolis Hastings. Our Krusell-Smith model and its "high-dimensional" (HD) version are described in Section 2 and Appendix B.1. Our one-asset HANK model is described in Appendix B.2. Our two-asset HANK model is described in Appendix B.3. All calculations in this paper were performed on a laptop with a 2.6GHz Intel Core i7-10750H processor with six cores.

then find the posterior mode and trace out the posterior distribution via Markov Chain Monte Carlo methods. Here, a critical benefit of our sequence-space method is that it makes it easy to reuse Jacobians, especially heterogeneous-agent Jacobians, across multiple computations of the likelihood function. This dramatically speeds up estimation, especially for the parameters of shock processes.

Finally, we demonstrate how to solve equation (1) nonlinearly by using our sequence-space Jacobians in a quasi-Newton method. We consider two types of nonlinear transitions: large temporary shocks, and transitions to a new steady state. We show how, for the examples we consider, sequence-space Jacobians allow convergence to the nonlinear solution in just a few iterations.

Throughout the paper, we apply our methods to three canonical heterogeneous-household models of increasing complexity: a neoclassical model in the spirit of Krusell and Smith (1998), a one-asset New Keynesian model in the spirit of McKay, Nakamura and Steinsson (2016), and a two-asset New Keynesian model in the spirit of Kaplan, Moll and Violante (2018). Table 1 illustrates the speeds that our algorithms are able to achieve on a laptop computer.[2] For each of our three models (including a high-dimensional version of the Krusell-Smith model), it takes less than

[2] All computations were performed on a laptop with a 2.6GHz Intel Core i7-10750H processor with six cores.

11 seconds to compute the heterogeneous-agent Jacobians $\mathcal{J}$. Once these Jacobians are known, it is almost immediate to calculate impulse responses. Posterior-mode estimation takes less than 9 minutes for every model, and is, for simpler models, a matter of seconds or milliseconds. A standard Random Walk Metropolis Hastings algorithm that traces out the posterior distribution of parameters with 100,000 draws takes less than six hours for our most complex, two-asset HANK model. By contrast, the leading computational techniques existing today find it challenging to estimate a two-asset HANK model at all.

**Related literature.** Since the early breakthroughs of Krusell and Smith (1998) and den Haan (1997), the literature on solution methods for heterogeneous-agent models has grown tremendously. Part of the literature has developed nonlinear methods, which are well-suited to address questions that inherently involve higher-order aggregate moments, such as the aggregate implications of risk premia or volatility shocks.[3] However, when it comes to the distribution of agents, these methods typically require either limited heterogeneity, or "approximate aggregation" (where only a few moments of the distribution matter for forecasting general equilibrium dynamics).

Our paper, by contrast, follows Reiter (2009) by linearizing with respect to aggregates but preserving nonlinearities with respect to idiosyncratic shocks. The Reiter method can be used to solve models that do not feature approximate aggregation, and instead capture the rich interactions between the distribution of agents and macroeconomic outcomes that are the hallmark of the recent heterogeneous-agent literature (see, for example, Krueger, Mitman and Perri 2016 and Kaplan and Violante 2018). Its main limitation is that it involves a linear system that grows with the dimension of the state space of the heterogeneous-agent model. For many complex models, the Schur (or equivalent) decomposition required to solve these models becomes too costly. This has led the literature to turn to "model reduction" methods, which involve approximating the equilibrium distribution, and sometimes also the value function.[4] How accurately these methods match the solution without model reduction varies depending on the application.[5] Our method, by contrast, solves the unreduced model, leaving all heterogeneity intact.

Boppart, Krusell and Mitman (2018) also propose a sequence-space method that solves the unreduced model and avoids the need for a large state-space system. They solve nonlinearly for impulse responses to one-time, unanticipated aggregate shocks ("MIT shocks"); when the shocks are small, this gives approximately the model's linear impulse responses. This method, however, requires some way to solve for nonlinear impulse responses in the first place. Boppart, Krusell and Mitman (2018) follow the typical approach by iterating over guesses for aggregate sequences,

[3] See the survey by Algan, Allais, Den Haan and Rendahl (2014) and recent work by Brumm and Scheidegger (2017), Mertens and Judd (2018), Proehl (2019), and Fernández-Villaverde, Hurtado and Nuño (2019), among many others.

[4] See, for instance, Reiter (2010), Ahn, Kaplan, Moll, Winberry and Wolf (2018b), Winberry (2018), and Bayer and Luetticke (2020).

[5] For instance, Ahn et al. (2018b) show that their model reduction technique works well for a one-asset model, but that it is more difficult to achieve a good fit for a two-asset model; they are able to reduce the size of the state-space system for the latter to 2445-by-2445, but further reduction degrades accuracy.

but there is no general method for updating these guesses, nor any guarantee of convergence.[6] By exploiting linearity instead, we avoid the need for any iteration, achieving the stability and speed required for advanced applications such as estimation.[7]

**Layout.** The rest of the paper proceeds as follows. Section 2 introduces our computational method with an example. Section 3 provides our fast algorithm for computing the Jacobians of heterogeneous-agent problems. Section 4 shows how to efficiently combine these Jacobians to compute general equilibrium impulse responses. Section 5 provides our application to estimation, and section 6 our application to nonlinear transitions. Section 7 concludes.

## 2 The sequence-space Jacobian: an example

We introduce our methods by means of an example: Krusell and Smith (1998)'s celebrated extension of the real business cycle model to heterogeneous households. This model is a natural starting point, since it well-known and there exist many well-established methods for solving it.

We set up the model in the sequence space, that is, assuming perfect foresight with respect to aggregates. We then show how to use the sequence-space Jacobian to solve for the impulse response of the model to a total factor productivity (TFP) shock in a fraction of a second.

### 2.1 Model description

The economy is populated by a mass 1 of heterogeneous households that maximize the time-separable utility function $\mathbb{E}\left[\sum \beta^t u\left(c_t\right)\right]$, where $u$ has the standard constant relative risk aversion form, $u\left(c\right) = \frac{c^{1-\sigma}}{1-\sigma}$. There exist $n_e$ idiosyncratic states, and in any period $t$, agents transition between any two such states $e$ and $e'$ with exogenous probability $P\left(e, e'\right)$. We denote by $\pi$ the stationary distribution of $P$ and assume that the mass of agents in each state $e$ is always equal to $\pi\left(e\right)$.[8] Agents supply an exogenous number of hours $l$, and earn wage income $w_t el$, where $w_t$ is the wage per efficient hour. Agents can only trade in capital $k$, which pays a rental rate $r_t$ net of depreciation, and are subject to a no-borrowing constraint. The value function of an agent entering

[6] In section 6, we propose such a method for updating guesses using sequence-space Jacobians. It would be redundant, however, to use this method to obtain linear impulse responses, since we can solve directly for these responses from the Jacobians.

[7] We share with all aggregate linearization methods the drawback that the model does not generate risk premia, portfolio choice is indeterminate, and optimal Ramsey policy is ill-defined. For these applications, higher-order perturbations or global solution methods are more appropriate (see for example Fernández-Villaverde, Rubio-Ramírez and Schorfheide 2016.)

[8] In the original Krusell and Smith (1998) model, the transition probabilities depend on the aggregate state, that is, $P$ takes the form $P\left(e, e', Z_t\right)$. Our methods can be applied to this case as well (see the general formulation in appendix A).

the period in income state $e$ and with capital $k_-$ at time $t$ is therefore

$$\begin{aligned} V_t(e, k_-) = \quad & \max_{c,k} \quad u(c) + \beta \sum_{e'} V_{t+1}(e', k) P(e, e') \\ & \text{s.t.} \quad c + k = (1 + r_t) k_- + w_t e l \\ & \qquad k \geq 0 \end{aligned} \tag{2}$$

Denote by $c_t^*(e, k_-)$ and $k_t^*(e, k_-)$ the policy functions that solve the Bellman equation (2). Also denote by $D_t(e, K_-) \equiv \Pr(e_t = e, k_{t-1} \in K_-)$ the measure of households in state $e$ that own capital in a set $K_-$ at the start of date $t$. The distribution $D_t$ has law of motion

$$D_{t+1}(e', K) = \sum_e D_t\left(e, k_t^{*-1}(e, K)\right) P(e, e') \tag{3}$$

where $k_t^{*-1}(e, \cdot)$ denotes the inverse of $k_t^*(e, \cdot)$. We assume that prior to $t = 0$, the economy is in a steady state with constant wage $w_{ss}$ and net rental rate $r_{ss}$, corresponding to a steady state of the general equilibrium economy discussed momentarily. In this steady state, there is a unique value function and decision rule solving (2), and a unique stationary distribution $D_{ss}$ solving (3). We suppose that agents start in this stationary distribution at date 0, so that $D_0 = D_{ss}$.

Equation (2) shows that, for any $t$, the policy $k_t^*(e, k_-)$ is a function of the future path $\{r_s, w_s\}_{s \geq t}$. Given $D_0 = D_{ss}$, through (3), the distribution $D_t(e, K)$ at any $t$ is a function of the entire path $\{r_s, w_s\}_{s \geq 0}$.[9] It follows that aggregate household capital holdings are characterized by a *capital function* $\mathcal{K}_t(\{r_s, w_s\}_{s \geq 0})$, where

$$\mathcal{K}_t(\{r_s, w_s\}) = \sum_e \int_{k_-} k_t^*(e, k_-) D_t(e, dk_-) \tag{4}$$

The ability to reduce interactions between heterogeneous agents to functions such as $\mathcal{K}_t$, which map aggregate sequences into aggregate sequences, is key to the sequence-space Jacobian method. We now combine this $\mathcal{K}_t$ function with equations describing production and market-clearing conditions to describe the entire Krusell-Smith economy. Production is carried out by a competitive representative firm, which has a Cobb-Douglas technology $Y_t = Z_t K_{t-1}^{\alpha} L_t^{1-\alpha}$, rents capital and labor from workers at rates $r_t + \delta$ and $w_t$, and faces the sequence of total factor productivity $Z_t$. The firm's first-order conditions

$$r_t = \alpha Z_t \left(\frac{K_{t-1}}{L_t}\right)^{\alpha - 1} - \delta \tag{5}$$

$$w_t = (1 - \alpha) Z_t \left(\frac{K_{t-1}}{L_t}\right)^{\alpha} \tag{6}$$

[9] This can be shown recursively: given $D_0 = D_{ss}$, $D_1$ is a function of $\{r_s, w_s\}_{s \geq 0}$, and therefore so is $D_2$, through its dependence on $D_1$. In section 3, we elicit explicitly the first-order dependence of $D_t$, $k_t^*$, and $\mathcal{K}_t$ on the sequence $\{r_s, w_s\}_{s \geq 0}$.

relate the paths of prices $\{r_t, w_t\}$ to the exogenous paths $\{Z_t, L_t = \sum \pi(e)\, el\}$ and the endogenous path for capital $\{K_t\}$. Combining (4)–(6), we can express the capital market clearing condition at each point in time as a function $H$,

$$H_t(\mathbf{K}, \mathbf{Z}) \equiv \mathcal{K}_t\left(\left\{\alpha Z_s\left(\frac{K_{s-1}}{\sum \pi(e)\, el}\right)^{\alpha-1} - \delta, (1-\alpha) Z_s\left(\frac{K_{s-1}}{\sum \pi(e)\, el}\right)^{\alpha}\right\}\right) - K_t = 0 \tag{7}$$

where $\mathbf{K} = (K_0, K_1, \ldots)'$. Given initial capital $K_{-1}$ and the exogenous path for productivity, $\mathbf{Z} = (Z_0, Z_1, \ldots)'$, equation (7) pins down the equilibrium path of capital.

## 2.2 Impulse responses

Applying the implicit function theorem to (7), the linear impulse response of capital to a transitory technology shock $d\mathbf{Z} = (dZ_0, dZ_1, \ldots)'$ is given by

$$d\mathbf{K} = -\mathbf{H}_{\mathbf{K}}^{-1}\mathbf{H}_{\mathbf{Z}} d\mathbf{Z} \tag{8}$$

where $\mathbf{H}_{\mathbf{K}}$ and $\mathbf{H}_{\mathbf{Z}}$ denote the Jacobians of $\mathbf{H}$ with respect to $\mathbf{K}$ and $\mathbf{Z}$, evaluated at the steady state. Given $d\mathbf{K}$, the impulse responses of other variables, e.g. $\{Y_s, L_s, r_s, w_s\}$, follow immediately. In practice, (8) is solved up to a given (large) horizon $T$ such that $K$ and $Z$ have approximately returned to steady state by time $T$.

We use the chain rule to relate the Jacobians $\mathbf{H}_{\mathbf{K}}$ and $\mathbf{H}_{\mathbf{Z}}$ to the derivatives of the $\mathcal{K}$ function defined in equation (4), evaluated at the steady state. For example, differentiating equation (7) with respect to $K_s$, we find that the $t, s$ entry of $\mathbf{H}_{\mathbf{K}}$ is

$$[\mathbf{H}_{\mathbf{K}}]_{t,s} = \frac{\partial \mathcal{K}_t}{\partial r_{s+1}}\frac{\partial r_{s+1}}{\partial K_s} + \frac{\partial \mathcal{K}_t}{\partial w_{s+1}}\frac{\partial w_{s+1}}{\partial K_s} - 1_{\{s=t\}} \tag{9}$$

A similar expression applies to $\mathbf{H}_{\mathbf{Z}}$. In addition, the derivatives $\frac{\partial r_{s+1}}{\partial K_s}$, $\frac{\partial w_{s+1}}{\partial K_s}$, $\frac{\partial r_{s+1}}{\partial Z_s}$ and $\frac{\partial w_{s+1}}{\partial Z_s}$ at $(\mathbf{K}_{ss}, \mathbf{Z}_{ss})$ can all be computed analytically: for example,

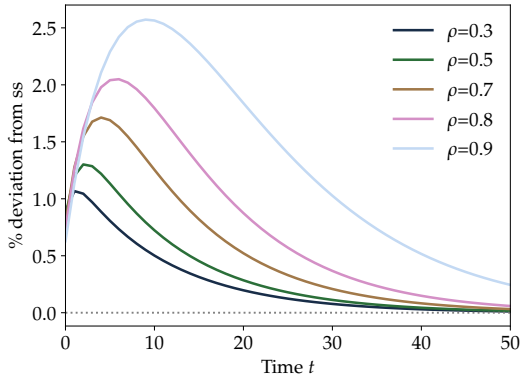
$$\frac{\partial r_{s+1}}{\partial K_s} = \alpha(\alpha-1) Z_{ss}\left(\frac{K_{ss}}{L_{ss}}\right)^{\alpha-2} \cdot \frac{1}{L_{ss}}$$

Hence, to obtain $\mathbf{H}_{\mathbf{K}}^{-1}\mathbf{H}_{\mathbf{Z}}$ in (8), all we need are the Jacobians of the $\mathcal{K}$ function with respect to its two inputs $r$ and $w$. The key remaining challenge is to compute these Jacobians. In the next section, we introduce a fast algorithm for doing so. As table 1 reveals, for a standard calibration of the Krusell-Smith model detailed in appendix B.1, this algorithm takes 90 milliseconds to calculate Jacobians of $\mathcal{K}$, truncated to a horizon of $300 \times 300$.[10] In a "high-dimensional" calibration that increases the dimensionality of the state space from 3,500 to 250,000, it still takes less than 11

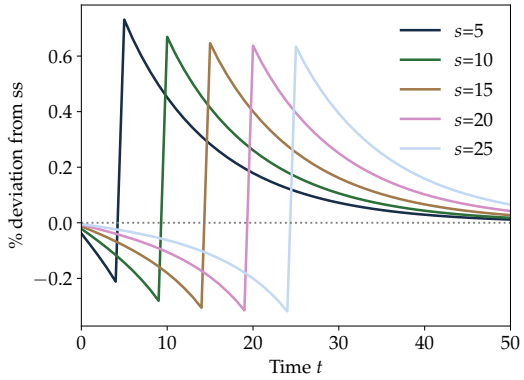
[10] This is long enough to accurately compute the solution given the shocks considered in Figure 1. We discuss how to choose an appropriate truncation horizon in section 4.4.

Figure 1: Impulse responses of capital to 1% TFP shocks in the Krusell-Smith model

(a) AR(1) shock with persistence $\rho$




(b) News shock at time $s$




seconds.

Given these Jacobians, the underlying heterogeneity no longer matters: the Jacobians tell us everything that we need to know, to first order, about the aggregate behavior of the model's heterogeneous agents. This feature of our method is apparent in table 1, where we see that most other computing times are identical between our two calibrations of the Krusell-Smith model, despite the large disparity in the size of their underlying state spaces.

**Impulse responses and news-shock interpretation.** Once we have the Jacobians of $\mathcal{K}$, we can immediately calculate $-\mathbf{H}_{\mathbf{K}}^{-1}\mathbf{H}_{\mathbf{Z}}$. Given (8), applying this matrix to any path for $d\mathbf{Z}$ delivers the impulse response $d\mathbf{K}$ of capital with a single matrix-vector multiplication. Panel (a) of figure 1 does this for a variety of $d\mathbf{Z}$, representing 1% AR(1) shocks to TFP with different persistences $\rho$ in our high-dimensional Krusell-Smith model. Note that the same matrix is applied to all these $d\mathbf{Z}$ vectors: once we have computed an impulse response, it is almost costless to compute others.

It is, in particular, immediate to obtain the effect of the "news" at date 0 that TFP will be higher by 1% at time $s$, as in panel (b) of figure 1. By definition, the impulse responses to $s$-period ahead news are equal to the $s^{th}$ column of the matrix $-\mathbf{H}_{\mathbf{K}}^{-1}\mathbf{H}_{\mathbf{Z}}$. This "news shock" interpretation of the columns provides a useful way of understanding their role in the computation of generic impulse responses. For example, the impulse responses to AR(1) TFP paths of persistence $\rho$ in panel (a) can be reinterpreted as the effect of the simultaneous news, at date 0, of an increase of $\rho^s$ in TFP at times $s = 0, 1, \ldots$

## 3 Computing Jacobians for heterogeneous-agent problems

In the previous section we established the usefulness of knowing the Jacobians $\partial\mathcal{K}/\partial r$ and $\partial\mathcal{K}/\partial w$ for computing the impulse responses of the Krusell-Smith model. In this section, we generalize the $\mathcal{K}$ function, to encompass the mapping from inputs to outputs in a broad class of heterogeneous-

agent problems. In this general case, *inputs* are the aggregates relevant to the decision-making of individual agents, such as interest rates or wages, while *outputs* can describe aggregate savings, consumption, investment, or other decisions by heterogeneous households or firms. We introduce a fast algorithm, which we call the *fake news algorithm*, for computing the Jacobian of any output with respect to any input.

### 3.1 General model representation

We begin by introducing a generic representation of a heterogeneous-agent problem as a mapping between a time path of aggregate inputs $\mathbf{X}_t$ and a time path of aggregate outputs $\mathbf{Y}_t$. Assume that there are $n_x$ inputs and $n_y$ outputs, and that the distribution is discretized on $n_g$ grid points. Let $\mathbf{D}_t$ be the $n_g \times 1$ vector representing the distribution of agents at time $t$, and suppose that the aggregate outputs of interest are the averages of individual "outcomes" against the distribution, that is, $\mathbf{Y}_t = \mathbf{y}_t'\mathbf{D}_t$, with $\mathbf{y}_t$ denoting the $n_g \times n_y$ matrix of individual outcomes (an outcome can be an agent's policy, e.g. consumption, or any other variable of interest defined at the individual level).[11] We assume that there exists three functions $v$, $\Lambda$ and $y$ such that, for a given initial distribution $\mathbf{D}_0$, $\mathbf{Y}_t$ is the solution to the system of equations:

$$\mathbf{v}_t = v(\mathbf{v}_{t+1}, \mathbf{X}_t) \tag{10}$$

$$\mathbf{D}_{t+1} = \Lambda(\mathbf{v}_{t+1}, \mathbf{X}_t)'\mathbf{D}_t \tag{11}$$

$$\mathbf{Y}_t = y(\mathbf{v}_{t+1}, \mathbf{X}_t)'\mathbf{D}_t \tag{12}$$

Here, $\mathbf{X}_t$ is the $n_x \times 1$ vector of aggregate inputs. Equation (10) expresses how the vector representing the value function, $\mathbf{v}_t$, relates to $\mathbf{X}_t$ and to its own value in the next period. Equation (11) updates the distribution, with $\Lambda(\mathbf{v}_{t+1}, \mathbf{X}_t)$ an $n_g \times n_g$ transition matrix representing the discretized law of motion for this distribution. Finally, equation (12) defines the $n_y \times 1$ vector of aggregate outputs $\mathbf{Y}_t$, with the $n_g \times n_y$ matrix of individual outcomes $\mathbf{y}_t$ defined by $y(\mathbf{v}_{t+1}, \mathbf{X}_t)$. Later, we will argue that many heterogeneous-agent models indeed take this form.

For given $\mathbf{X}_{ss}$, the *steady state* of the model is the fixed point $(\mathbf{Y}_{ss}, \mathbf{v}_{ss}, \mathbf{D}_{ss})$ of (10)–(12) that obtains when $\mathbf{X}_t = \mathbf{X}_{ss}$ at all times. For convenience, we write $\Lambda_{ss} \equiv \Lambda(\mathbf{v}_{ss}, \mathbf{X}_{ss})$ and $\mathbf{y}_{ss} \equiv y(\mathbf{v}_{ss}, \mathbf{X}_{ss})$. We consider transitions of length $T$ that end at this steady state, so that the terminal values are $\mathbf{X}_{T-1} = \mathbf{X}_{ss}$, and $\mathbf{v}_T = \mathbf{v}_{ss}$. The initial distribution $\mathbf{D}_0$ is given, and our main result assumes that it is also equal to $\mathbf{D}_{ss}$. Hence, this setting allows us to study transitory shocks around a steady state.[12]

Given this setup, (10)–(12) define a mapping from the $T \times n_x$ stacked vector of inputs $\mathbf{X}$, to the

[11] As we show in appendix A, it is straightforward to extend our method to include higher order moments, such as the variance of consumption, among the outputs of interest.

[12] Section 6 discusses how to solve for nonlinear transition dynamics with arbitrary initial distributions $\mathbf{D}_0$, including the effects of permanent shocks that change the steady state.

$T \times n_y$ stacked vector of outputs $\mathbf{Y}$, which we write

$$\mathbf{Y} = h(\mathbf{X}) \tag{13}$$

We assume that the functions $v$, $\Lambda$ and $y$ are differentiable around $(\mathbf{v}_{ss}, \mathbf{X}_{ss})$, so that the function $h$ is also differentiable around $\mathbf{X}_{ss}$. Our goal is to characterize the Jacobian $\mathcal{J}$ of $h$ evaluated at $\mathbf{X} = \mathbf{X}_{ss}$. $\mathcal{J}$ represents the aggregate response of heterogeneous agents to perturbations to their environment at different dates. This Jacobian can be of interest in its own right. But, critically, it is the key object required to compute the general equilibrium solution.

**Example: Krusell and Smith.** In the model of section 2, the inputs are $\mathbf{X}_t = (r_t, w_t)$, and one natural choice for outputs is $\mathbf{Y}_t = (K_t, C_t)$. The model can be solved with value function iteration. In this case, $\mathbf{v}_t$ is the value function $V_t$ in equation (2) at each point on the grid for states $(e, k_-)$, and $\mathbf{D}_t$ is the fraction of agents at time $t$ at each point on this grid. Given $\mathbf{v}_{t+1}$ and $\mathbf{X}_t$, the solution to (2) involves a maximized value function $\mathbf{v}_t$—equation (10)—and a policy function $\mathbf{k}_t$. We use the Young (2010) lottery method to convert this policy into a transition matrix on the grid, and compose this with the process for $e$ to obtain the full transition matrix $\Lambda_t'$ from current states $(e, k_-)$ to next-period states $(e', k)$—equation (11). Finally, aggregate capital and consumption are obtained by taking the dot product of the policies $\mathbf{k}_t$ and $\mathbf{c}_t$ with the distribution $\mathbf{D}_t$: this is equation (12), with $\mathbf{y}_t \equiv (\mathbf{k}_t, \mathbf{c}_t)$.

An alternative approach, which is typically faster and more accurate in practice, is to use the Euler equation, as in Carroll (2006). In this approach, $\mathbf{v}_t$ is the derivative of the value function $\frac{\partial V_t}{\partial k_-}$ at each point on the grid for $(e, k_-)$. The Euler equation maps $\mathbf{v}_{t+1}$ and $\mathbf{X}_t$ to optimal capital and consumption policies $\mathbf{k}_t$ and $\mathbf{c}_t$, and the envelope theorem implies $\mathbf{v}_t = (1 + r_t) u'(\mathbf{c}_t)$. Combining these, we obtain equation (10). Again, combining $\mathbf{k}_t$ with the exogenous law of motion for the state $e$ delivers $\Lambda$ in equation (11), and $\mathbf{y}_t \equiv (\mathbf{k}_t, \mathbf{c}_t)$ aggregates individual policies into $K_t$ and $C_t$ in equation (12).

Beyond this example, many other heterogeneous-agent problems can also be cast into the framework of equations (10)–(12). The scope and limitations of our framework will become clearer after we have presented our algorithm, so we postpone this discussion to the end of the next section.

### 3.2 Fake news algorithm

In this section, we provide a fast algorithm for computing $\mathcal{J}$, which we call the "fake news" algorithm. We start with two preliminaries: notational conventions, and a direct method for computing $\mathcal{J}$ that will serve as a benchmark for our algorithm.

**Notation.** To present our algorithm in an intuitive manner, we start by assuming that there is only one input and one output, $n_x = n_y = 1$, and later generalize to any $n_x$ and $n_y$. Define the

$T \times 1$ vector $\mathbf{e}^s$ to have 0's everywhere except at the $s$th entry, where it has a 1. For a given $dx$, we say there is a "shock at time $s$" when the $T \times 1$ input vector is given by $\mathbf{X}^s \equiv \mathbf{X}_{ss} + \mathbf{e}^s dx$. Let $\mathbf{v}_t^s$, $\mathbf{D}_t^s$, and $Y_t^s$ denote the solution to equations (10)–(12) given $\mathbf{X}^s$. Also, let $\Lambda_t^s \equiv \Lambda(\mathbf{v}_{t+1}^s, X_t^s)$ denote the transition matrix between states at time $t$, and $\mathbf{y}_t^s \equiv y(\mathbf{v}_{t+1}^s, X_t^s)$ denote the outcome function at time $t$, in response to the shock at time $s$. Finally, denote with a $d$ the difference of all objects relative their steady state level, so that $dY_t^s \equiv Y_t^s - Y_{ss}$, $d\mathbf{y}_t^s \equiv \mathbf{y}_t^s - \mathbf{y}_{ss}$, $d\Lambda_t^s \equiv \Lambda_t^s - \Lambda_{ss}$, and $d\mathbf{D}_t^s \equiv \mathbf{D}_t^s - \mathbf{D}_{ss}$. The $s^{th}$ column of the Jacobian $\mathcal{J}$ is then the limit of $\frac{d\mathbf{Y}^s}{dx}$ as $dx \to 0$.

**Direct method.** A direct method for computing the $s^{th}$ column of the Jacobian using one-sided numerical differentiation is as follows. First, starting with a small shock $dx$ at time $s$, iterate (10) backward, starting with $\mathbf{v}_T = \mathbf{v}_{ss}$, and compute the value function $\mathbf{v}_t^s$, the transition matrix $\Lambda_t^s$, and individual policies $\mathbf{y}_t^s$, for $t = T-1, \ldots, 0$. Second, iterate (11) forward, starting with $\mathbf{D}_0 = \mathbf{D}_{ss}$, to solve recursively for the distributions $\mathbf{D}_t^s$ for $t = 1, \ldots, T-1$, by applying the transition matrices $\Lambda_t^s$. Next, for each $t$, take the distribution-weighted sum $(\mathbf{y}_t^s)' \mathbf{D}_t^s$ of individual policies to obtain $Y_t^s$ in (12). Finally, form $\mathcal{J}_{t,s} = dY_t^s/dx = (Y_t^s - Y_{ss})/dx$. To obtain the entire Jacobian $\mathcal{J}$, repeat this process $T$ times, once for each $s$. This is costly in practical applications, since $T$ is typically at least equal to 300.

**Structure of Jacobian $\mathcal{J}$.** We now turn to our algorithm, which relies on several results about the structure of the Jacobian $\mathcal{J}$. The key is to recognize that the columns of $\mathcal{J}$ are closely related. Using our $s$ superscript notation, equation (12) defines the output at time $t$ in response to the shock at time $s$ as

$$Y_t^s = (\mathbf{y}_t^s)' \mathbf{D}_t^s \tag{14}$$

and (11) defines the distribution at time $t+1$ given the shock at time $s$ as

$$\mathbf{D}_{t+1}^s = (\Lambda_t^s)' \mathbf{D}_t^s \tag{15}$$

We first show how to efficiently obtain the policy functions $\mathbf{y}_t^s$ and transition matrices $\Lambda_t^s$. This makes use of the following implication of dynamic programming.

**Lemma 1.** *For any $s \geq 1$, $t \geq 1$, we have:*

$$\mathbf{y}_t^s = \begin{cases} \mathbf{y}_{ss} & s < t \\ \mathbf{y}_{T-1-(s-t)}^{T-1} & s \geq t \end{cases} \quad \text{and} \quad \Lambda_t^s = \begin{cases} \Lambda_{ss} & s < t \\ \Lambda_{T-1-(s-t)}^{T-1} & s \geq t \end{cases} \tag{16}$$

Lemma 1 follows immediately from the recursive structure of equation (10) and the definition of $\mathbf{y}_t^s$ and $\Lambda_t^s$. The intuition is that agents only care about the distance to the shock $s-t$, rather than calendar time $t$ and $s$ separately, when deciding on their own behavior. For instance, the response of their consumption policy at time $t$ to any shock at time $t+1$ is the same as the response of their consumption policy at time 0 to the same shock at time 1.

By implication, we can compute *all* policies $\mathbf{y}_t^s$ from a single perturbation of the input at date $s = T-1$. The same argument applies to the transition matrices $\Lambda_t^s$. Lemma 1 therefore suggests an immediate improvement to the direct algorithm for computing the Jacobian: replace the $T$ backward iterations by a single backward iteration starting from a shock at date $T-1$. This is enough to deliver all policy functions $\mathbf{y}_t^s$ and $\Lambda_t^s$ for all shock dates $s$ and all $t$. Observe that this result is true nonlinearly, i.e. irrespective of the size of $dx$.

Our next result speeds up the algorithm even further, for the case in which the transition begins and ends at the same steady state ($\mathbf{D}_0 = \mathbf{D}_{ss}$) and the shock $dx$ is infinitesimal. The result concerns aggregate outcomes $Y_t^s$. For any $s \geq 1$, $t \geq 1$, we define

$$\mathcal{F}_{t,s} \cdot dx \equiv dY_t^s - dY_{t-1}^{s-1} \tag{17}$$

as the difference between the aggregate response of the output at $t$ to a shock at date $s$, and its response at $t-1$ to a shock at date $s-1$. Since equation (16) implies that $d\mathbf{y}_t^s = d\mathbf{y}_{t-1}^{s-1}$ for all $s, t \geq 1$, one might conjecture that $\mathcal{F}_{t,s}$ is identically zero. But this conjecture is not quite right, since the distribution in equation (14) is also changing over time. The next lemma characterizes $\mathcal{F}_{t,s}$.

**Lemma 2.** *Assume that $\mathbf{D}_0 = \mathbf{D}_{ss}$. For infinitesimal $dx$, and any $s \geq 1$, $t \geq 1$, we have:*

$$\mathcal{F}_{t,s} \cdot dx = \mathbf{y}'_{ss} \left(\Lambda'_{ss}\right)^{t-1} d\mathbf{D}_1^s \tag{18}$$

*Proof.* First, since $\mathbf{D}_0 = \mathbf{D}_{ss}$, in the absence of any shock ($dx = 0$) we have $Y_t = Y_{ss}$, $\mathbf{y}_t^s = \mathbf{y}_{ss}$ and $\mathbf{D}_t^s = \mathbf{D}_{ss}$ for all $t$. Since $y$ and $\Lambda$ are differentiable, in the limit as $dx \to 0$, equation (14) implies

$$dY_t^s = \mathbf{y}'_{ss} d\mathbf{D}_t^s + \left(d\mathbf{y}_t^s\right)' \mathbf{D}_{ss} \tag{19}$$

Subtracting $dY_t^s$ and $dY_{t-1}^{s-1}$ and using the fact that equation (16) implies $d\mathbf{y}_t^s = d\mathbf{y}_{t-1}^{s-1}$, we obtain

$$\mathcal{F}_{t,s} \cdot dx = \mathbf{y}'_{ss} \left(d\mathbf{D}_t^s - d\mathbf{D}_{t-1}^{s-1}\right) \tag{20}$$

Next, in the limit as $dx \to 0$, equation (15) implies both

$$d\mathbf{D}_t^s = \Lambda'_{ss} d\mathbf{D}_{t-1}^s + \left(d\Lambda_{t-1}^s\right)' \mathbf{D}_{ss} \tag{21}$$

and

$$d\mathbf{D}_{t-1}^{s-1} = \Lambda'_{ss} d\mathbf{D}_{t-2}^{s-1} + \left(d\Lambda_{t-2}^{s-1}\right)' \mathbf{D}_{ss}$$

Subtracting and using the fact that equation (16) implies $d\Lambda_{t-1}^s = d\Lambda_{t-2}^{s-1}$, we therefore finally have

simply

$$\begin{aligned} d\mathbf{D}_t^s - d\mathbf{D}_{t-1}^{s-1} &= \Lambda_{ss}' \left( d\mathbf{D}_{t-1}^s - d\mathbf{D}_{t-2}^{s-1} \right) \\ &= \left( \Lambda_{ss}' \right)^2 \left( d\mathbf{D}_{t-2}^s - d\mathbf{D}_{t-3}^{s-1} \right) \\ &\vdots \\ &= \left( \Lambda_{ss}' \right)^{t-1} d\mathbf{D}_1^s \end{aligned} \tag{22}$$

where the last line follows because, given that $\mathbf{D}_0 = \mathbf{D}_{ss}$, we have $d\mathbf{D}_0^{s-1} = 0$ for all $s \geq 1$. □

The intuition for equation (18) is as follows. Suppose that we know the path of the aggregate output $Y_t$ at all dates $t = 0 \dots T-1$ in response to a shock at date $s-1$. How does this compare to the path of $Y_t$ in response to a shock at date $s$, from date $t = 1$ onwards? From lemma 1, the behavior of agents at all dates is identical in both cases. Therefore, the only difference is that the initial distribution in the second case is $\mathbf{D}_1^s$ rather than $\mathbf{D}_{ss}$. To first order, this difference in initial distribution affects aggregates at all dates as if agents followed their steady state behavior, which is what equation (18) expresses.

For a given $s$, $\mathcal{F}_{t,s}$ can be interpreted as the impulse response to a "date-$s$ fake news shock": a shock to date $s$ announced at date 0, and retracted at date 1.[13] At date 0, agents react to the announcement, which leads to the distribution $\mathbf{D}_1^s$. After the announcement is retracted, they revert to steady-state policies, so the effect on output at all dates $t \geq 1$ is $\mathbf{y}_{ss}' \cdot \left( \Lambda_{ss}' \right)^{t-1} d\mathbf{D}_1^s$. This expression can usefully be rewritten with the help of the following definition.

**Definition 1.** The *expectation vector* for outcome $Y$ at time $t$ is defined by

$$\mathcal{E}_t \equiv \left( \Lambda_{ss} \right)^t \mathbf{y}_{ss} \tag{23}$$

For each grid point, the time path of $\mathcal{E}_t$ represents the expected time path of outcome $Y$, in the steady state, for an agent starting at that grid point.[14] Equation (18) then reads $\mathcal{F}_{t,s} \cdot dx = \mathcal{E}_{t-1}' d\mathbf{D}_1^s$.

We can now use lemmas 1 and 2 to arrive at the following proposition.

**Proposition 1.** *Assume that* $\mathbf{D}_0 = \mathbf{D}_{ss}$. *For infinitesimal* $dx$, *define the* $(t,s)$*-th element of the* fake news matrix $\mathcal{F}$ *as*

$$\mathcal{F}_{t,s} \cdot dx \equiv \begin{cases} dY_0^s & t = 0 \\ \mathcal{E}_{t-1}' d\mathbf{D}_1^s & t \geq 1 \end{cases} \tag{24}$$

[13] This information structure is the same as that used by Christiano, Ilut, Motto and Rostagno (2010) to generate a boom-bust episode in response a shock to productivity that later turns out not to happen. In our case, the "fake news" shock for date $s$ is unlearned at date 1. It is also related, though not formally equivalent, to the "noise shocks" considered in the belief-driven business cycle literature literature (e.g. Lorenzoni 2009.)

[14] In the literature on control theory, the matrix with rows $\mathcal{E}_0', \mathcal{E}_1', \dots$ is sometimes called the *observability matrix*. This concept is also used by Reiter (2010) and Ahn et al. (2018b).

*where* $dY_0^s = (d\mathbf{y}_0^s)' \mathbf{D}_{ss}$ *and* $d\mathbf{D}_1^s = (d\Lambda_0^s)' \mathbf{D}_{ss}$. *Then, the Jacobian* $\mathcal{J}$ *of* $h$ *satisfies the recursion* $\mathcal{J}_{t,s} = \mathcal{J}_{t-1,s-1} + \mathcal{F}_{t,s}$ *for* $t, s \geq 1$, *with* $\mathcal{J}_{t,s} = \mathcal{F}_{t,s}$ *for* $t = 0$ *or* $s = 0$, *and is therefore given by*

$$\mathcal{J}_{t,s} = \sum_{k=0}^{\min\{s,t\}} \mathcal{F}_{t-k,s-k} \tag{25}$$

*Proof.* When $t, s \geq 1$, the recursion immediately follows from lemma 1. When $t = 0$, $\mathcal{J}_{t,s} \cdot dx = \mathcal{F}_{t,s} \cdot dx = dY_0^s$ by definition.

Finally, when $s = 0$ and $t \geq 1$, lemma 1 implies that $d\mathbf{y}_t^0 = 0$, and by equation (19), we have $\mathcal{J}_{t,0} \cdot dx = dY_t^0 = \mathbf{y}_{ss}' d\mathbf{D}_t^0$. Since $d\Lambda_t^0 = 0$ for all $t \geq 1$, using (21) we can write $\mathcal{J}_{t,0} \cdot dx = \mathbf{y}_{ss}' d\mathbf{D}_t^0 = \mathbf{y}_{ss}' \Lambda_{ss}' d\mathbf{D}_{t-1}^0 = \cdots = \mathbf{y}_{ss}' (\Lambda_{ss}')^{t-1} d\mathbf{D}_1^0 = \mathcal{E}_{t-1}' d\mathbf{D}_1^0$, which is $\mathcal{F}_{t,0} \cdot dx$ in (24). □

Proposition 1 characterizes the first-order aggregate response of heterogeneous agents to changes in their environment at any date $s$, as a function of *only* $dY_0^s$ (the aggregate initial response to shocks at date $s$), $d\mathbf{D}_1^s$ (the response of the distribution at date 1 to shocks at date $s$), and the expectation vectors $\mathcal{E}_t$ which can be obtained from the stationary solution. The intuition goes back to lemmas 1 and 2: since policy functions only depend on the distance to the shock, and since the steady state expectation vectors $\mathcal{E}_t$ give information about the behavior of aggregates after shocks to the initial distribution, it is possible to project the effect at any date $t$ from knowledge of the effects of future shocks on aggregates and distributions at date 0. The expectation vectors, in turn, are easy to compute thanks to the following observation.

**Lemma 3.** *The expectation vectors defined in (23) solve the recursion* $\mathcal{E}_t = \Lambda_{ss} \mathcal{E}_{t-1}$, *with* $\mathcal{E}_0 = \mathbf{y}_{ss}$.

**Algorithm for a single input and output.** Proposition 1 and lemma 3 inspire our fast "fake news" algorithm. When implemented with one-sided numerical differentiation, given a small $dx > 0$, the algorithm consists of four steps:

1. Calculate $\mathbf{y}_0^s$ and $\Lambda_0^s$ for each $s$ using a single backward iteration from time $T - 1$. Combining these with the initial steady state distribution, form two key objects: the $T$ scalars $\mathcal{Y}_s$ defined by $\mathcal{Y}_s dx \equiv dY_0^s = (d\mathbf{y}_0^s)' \mathbf{D}_{ss}$, representing the effect on the output at date 0 from the shock to the input at date $s$; and the $T$ $n_g \times 1$-vectors $\mathcal{D}_s dx \equiv d\mathbf{D}_1^s = (d\Lambda_0^s)' \mathbf{D}_{ss}$, giving the change in the distribution at date 1 from the shock at date $s$.[15]

2. Calculate the $T - 1$ $n_g \times 1$ expectation vectors $\mathcal{E}_t \equiv (\Lambda_{ss})^t \mathbf{y}_{ss}$, using the recursion from lemma 3.

3. Combine results from the previous two steps to form the fake news matrix $\mathcal{F}$ from proposition 1. The first row ($t = 0$) of this matrix contains the $\mathcal{Y}'$ s from step 1, and other rows

[15] In practice, it is usually more accurate to compute the differences $d\mathbf{y}_0^s$ and $d\Lambda_0^s$ by subtracting "ghost runs" rather than steady states. That is, compute $\mathbf{y}_0^s$ as described for some small $dx > 0$. Repeat the same procedure with $dx = 0$ to get $\tilde{\mathbf{y}}_0^s$. Set $d\mathbf{y}_0^s = \mathbf{y}_0^s - \tilde{\mathbf{y}}_0^s$. This procedure is more accurate than subtracting steady state values whenever those have not fully converged, i.e. whenever $\tilde{\mathbf{y}}_0^s \neq \mathbf{y}_{ss}$. Do the same for $d\Lambda_0^s$. See appendix C.1 for more details on this and other ways to manage numerical error.

($t \geq 1$) contain the product $\mathcal{E}'_{t-1}\mathcal{D}_s$ from steps 1 and 2:

$$\mathcal{F} = \begin{bmatrix} \mathcal{Y}_0 & \mathcal{Y}_1 & \mathcal{Y}_2 & \cdots & \mathcal{Y}_{T-1} \\ \mathcal{E}'_0\mathcal{D}_0 & \mathcal{E}'_0\mathcal{D}_1 & \mathcal{E}'_0\mathcal{D}_2 & & \mathcal{E}'_0\mathcal{D}_{T-1} \\ \mathcal{E}'_1\mathcal{D}_0 & \mathcal{E}'_1\mathcal{D}_1 & \mathcal{E}'_1\mathcal{D}_2 & & \mathcal{E}'_1\mathcal{D}_{T-1} \\ \vdots & \vdots & \vdots & & \vdots \\ \mathcal{E}'_{T-2}\mathcal{D}_0 & \mathcal{E}'_{T-2}\mathcal{D}_1 & \mathcal{E}'_{T-2}\mathcal{D}_2 & \cdots & \mathcal{E}'_{T-2}\mathcal{D}_{T-1} \end{bmatrix} \tag{26}$$

4. Using proposition 1, build up the Jacobian $\mathcal{J}_{t,s} = \mathcal{J}_{t-1,s-1} + \mathcal{F}_{t,s}$ recursively from its first row and first column. By equation (25), the element $(t, s)$ of the Jacobian $\mathcal{J}$ is the sum of the $(t, s)$ element of the $\mathcal{F}$ matrix and of all the elements on the diagonal to its immediate upper left in (26). For instance, we have $\mathcal{J}_{2,2} = \mathcal{E}'_1\mathcal{D}_2 + \mathcal{E}'_0\mathcal{D}_1 + \mathcal{Y}_0$.

At this stage it is clear why this algorithm achieves significantly higher speed than the direct method for computing the Jacobian: it requires only the computation of the primitive objects $\mathcal{Y}_t$ and $\mathcal{D}_t$, which can be obtained with one backward iteration starting from a shock at $T - 1$, and of $\mathcal{E}_t$, which can be obtained by recursive application of the steady-state transition matrix, starting with the vector $\mathbf{y}_{ss}$ of steady-state outcomes.

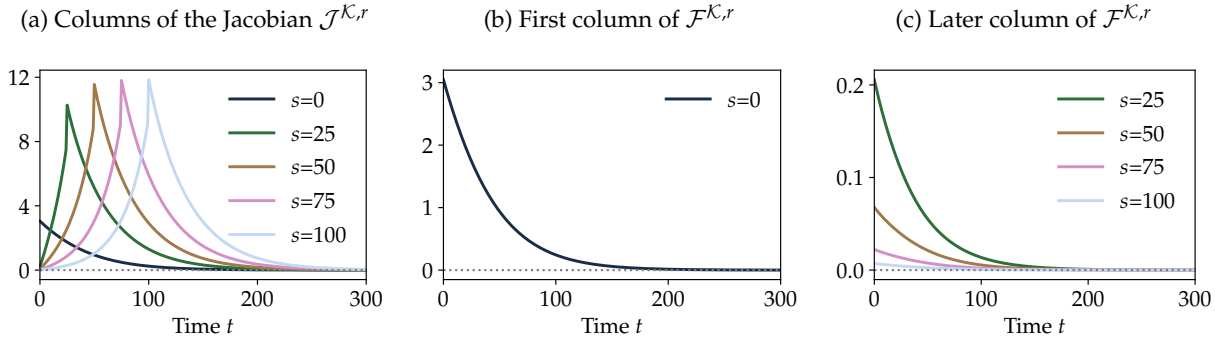
**Example: Krusell and Smith.** Panel (a) of figure 2 displays several columns of the Jacobian $\mathcal{J}^{\mathcal{K},r}$ for the Krusell-Smith model of section 2. By the news shock interpretation, these columns represent the time path of aggregate capital accumulation $\{\mathcal{K}_t\}$ when households learn at date 0 about an increase in the rental rate $r_s$ at various dates $s$.

When the shock takes place at date 0, households are surprised by a higher return on existing assets. They save some of this additional return (a standard wealth effect), accumulating assets that they later progressively decumulate. When the shock takes place at later dates $s > 0$, households also have an intertemporal substitution response, which leads them to save in anticipation of the increase in $r$. This generates a "tent" pattern in the Jacobian $\mathcal{J}$.

Proposition 1 shows that the columns of $\mathcal{J}$ reflect the accumulation of terms from the fake news matrix $\mathcal{F}$. The columns of that matrix are depicted in panels (b) and (c). The first column of $\mathcal{J}$ is the same as that of $\mathcal{F}$. By contrast, the other columns of $\mathcal{J}$ are a combination of a shifted-down version of the first column of $\mathcal{F}$ and of its other columns $\mathcal{F}_{\cdot s}$ for $s > 0$. By the "fake news" interpretation, these columns represent the behavior of aggregate assets when households first save at date 0 in anticipation of an increase in $r$ at date $s$, and then dissave after the announcement is retracted at date 1.

One striking feature of the columns of the Jacobian $\mathcal{J}$ is that they converge to a regular pattern around the main diagonal: the $s = 50$ impulse response around $t = 50$ is almost the same as the $s = 75$ and $s = 100$ impulse responses around $t = 75$ and $t = 100$. In other words, if the shock is anticipated far enough in advance, all impulse responses are just shifted versions of each other. This reflects the fact that $\mathcal{F}_{t,s}$ goes to zero both for high $t$ (the effect of date-0 behavior through the

Figure 2: Jacobian $\mathcal{J}^{\mathcal{K},r}$ and fake news matrix $\mathcal{F}^{\mathcal{K},r}$ in the Krusell-Smith model.




distribution dies away) and for high $s$ (the effect of far-out shocks on date-0 behavior dies away), so that $\mathcal{J}_{t,s} \approx \mathcal{J}_{t-1,s-1}$ for high $t$ and $s$.[16]

**Generalization to many inputs and outputs.** In the general case in which the $h$ function has multiple inputs $i$ and outputs $o$, the algorithm above is straightforward to apply separately for each $i$ and $o$. However, some further speed gain can be achieved by observing that certain objects can be reused several times. Specifically, the $\mathcal{D}_s$ depend only on the input shock $dX^i$, so they only need to be computed once per input and can be written as $\mathcal{D}^i_s$. Moreover, the $\mathcal{E}_t$ defined in step 2 depend only on the output of interest $dY^o$, so they only need to be computed once per output and can be written as $\mathcal{E}^o_t$. By contrast, the $\mathcal{Y}_s$ defined in step 1 depend on both the input shock $dX^i$ and the output of interest $dY^o$. They are computed by doing a backward iteration in response to each input shock $i$, and then taking the aggregate response of each $o$ for each $s$. This delivers a $\mathcal{Y}^{o,i}_s$ for each $o$ and $i$. The $\mathcal{F}^{o,i}$ matrix can then be computed as in equation (26), but with $\mathcal{Y}^{o,i}$ in the first row, and the products $(\mathcal{E}^o_t)'\mathcal{D}^i_s$ in the other rows $t, s$.

**Implementation and accuracy.** The result in Proposition 1 shows that the direct method to compute $\mathcal{J}$ described at top of this section, and the fast algorithm building up $\mathcal{J}$ from the $\mathcal{F}$ matrix, should deliver exactly the same solution. We suggested implementing these algorithms with one-sided numerical differentiation. While this is simple to do in practice, it also introduces small errors from the differentiation procedure. Here, we evaluate these errors in computing the Jacobian, and discuss how to mitigate them.

We first discuss alternatives to one-sided numerical differentiation. A simple alternative is two-sided (or symmetric) numerical differentiation. For example, to obtain the $s$th column $\mathcal{J}_{\cdot,s}$ via the direct method with this procedure, compute for a given $dx$ the response $\mathbf{Y}^{s+}$ to $\mathbf{X}^{s+} \equiv \mathbf{X}_{ss} + \mathbf{e}^s dx$, $\mathbf{Y}^{s-}$ to $\mathbf{X}^{s-} \equiv \mathbf{X}_{ss} - \mathbf{e}^s dx$, and then form $\mathcal{J}_{\cdot,s} = \frac{\mathbf{Y}^{s+} - \mathbf{Y}^{s-}}{2dx}$. This method is known to provide a better approximation to the derivative, but it requires twice the computational cost. Another route is to use automatic differentiation. This is typically more difficult to implement in practice,

[16] This "asymptotic time invariance" property is a general feature of the Jacobians of heterogeneous-agent problems. A previous version of this paper (Auclert, Bardóczy, Rognlie and Straub 2019) provided a formal proof.

Table 2: Direct and fake news algorithms to compute $300 \times 300$ Jacobians $\mathcal{J}$.

| | Krusell-Smith | HD Krusell-Smith | one-asset HANK | two-asset HANK |
|---|---|---|---|---|
| **Direct** | **21 s** | **2102 s** | **156 s** | **956 s** |
| step 1 (backward) | 13 s | 1302 s | 132 s | 846 s |
| step 2 (forward) | 8 s | 800 s | 24 s | 111 s |
| **Fake news** | **0.086 s** | **10.467 s** | **0.317 s** | **3.498 s** |
| step 1 | 0.060 s | 8.654 s | 0.236 s | 3.159 s |
| step 2 | 0.011 s | 1.061 s | 0.022 s | 0.119 s |
| step 3 | 0.011 s | 0.758 s | 0.045 s | 0.201 s |
| step 4 | 0.003 s | 0.003 s | 0.014 s | 0.018 s |
| Grid points $n_g$ | 3,500 | 250,000 | 3,500 | 10,500 |
| Inputs $n_x$ | 2 | 2 | 4 | 5 |
| Outputs $n_y$ | 2 | 2 | 4 | 4 |
| Jacobians $n_x \times n_y$ | 4 | 4 | 16 | 20 |

since it requires writing all routines in a way that can be accepted by automatic differentiation packages,[17] but has the benefit of computing the exact derivatives without approximation error. Appendix C.1 discusses all three approaches to differentiation in more detail.

In appendix D.1, we evaluate the errors that result from various types of differentiation when computing the Jacobian $\mathcal{J}^{\mathcal{K},r}$ in figure 2. We use as a benchmark the Jacobian that results from the direct method with automatic differentiation. We then compare this Jacobian to the one that results from one or two-sided numerical differentiation, in either the direct or the fake news method.

The conclusions from this exercise are as follows: first, under automatic differentiation, the direct and the fake news method deliver exactly the same Jacobian, to near-machine precision. This verifies Proposition 1. Second, two-sided numerical differentiation is always more accurate than one-sided numerical differentiation, closing the gap with the automatic differentiation solution by one to two orders of magnitude. Third, when implemented with numerical differentiation, the fake news method is typically more accurate than the direct method. In all cases, the errors are small, less than 0.01% of the peak response.

**Efficiency.** Table 2 displays the time it takes to compute $\mathcal{J}$s for the heterogeneous-agent block of each of our three benchmark models: the Krusell-Smith model already introduced, a one-asset HANK model with endogenous labor described in appendix B.2, and a two-asset HANK model described in appendix B.3. We report the times from one-sided numerical differentiation, which is the easiest and fastest to implement in practice. The speed-up from using the fake news rather than the direct algorithm is very large in all cases: a factor of over 200 for for all models.

What is the source of the large efficiency gain? When there are $n_x$ inputs and $n_y$ outputs, the direct algorithm discussed at the top of this section requires $n_x T^2$ backward "steps" and $n_x T^2$

[17] Our implementation uses Python's jax automatic differentiation package.

forward "steps". By contrast, the fake news algorithm requires $n_x T$ backward steps and $n_y(T-1)$ applications of the matrix $\Lambda_{ss}$ to construct the expectation vectors $\mathcal{E}_t$, reducing computational effort in steps 1 and 2 by a factor of around $T$, which in our application is $T = 300$.[18,19]

**Jacobians as sufficient statistics for the heterogeneous-agent problem.** Since the Jacobians $\mathcal{J}$ locally describe the mapping $\mathbf{Y} = h(\mathbf{X})$, they are all that is needed to capture the local behavior of the heterogeneous-agent problem. This observation implies that all of the complexity introduced by heterogeneity in any given model boils down entirely to the Jacobian of the resulting heterogeneous-agent problem. This facilitates the analysis of the importance of heterogeneity for general equilibrium, and the connection of models to the data. For example, in simple general equilibrium models, the Jacobian of aggregate consumption with respect to income $\mathcal{J}^{C,y}$ is all that is needed for general equilibrium (Auclert, Rognlie and Straub 2018).

**Scope and limitations.** In addition to the Krusell-Smith model, the two other models we consider in this paper fit into the framework of equations (10)–(12), so that the fake news algorithm applies to them directly. Our one-asset HANK model features a multidimensional choice over both labor and asset policies. Our two-asset HANK model features two endogenous states, a liquid and an illiquid asset, with the price of the illiquid asset varying in response to shocks.

A number of other models can also be directly cast into the framework of equations (10)–(12). This includes models with search and matching where the inputs $\mathbf{X}$ matter directly for the transition between employment states (as in Gornemann, Kuester and Nakajima 2016), and models where higher-order moments of the distribution of agents are relevant as an output $\mathbf{Y}$ (such as the variance of consumption or a CES price index). It also includes models where some non-grid-based representation of the value function, such as Chebyshev polynomials, is used. Appendix A.1 covers these direct applications.

Other models require a slightly more general framework than (10)–(12). This includes models where the distribution of agent features entry and exit (e.g. Hopenhayn 1992 and simple overlapping generations models), or models where a nonlinear function of the distribution, such as the $u$th quantile function, is relevant. It also includes models where the distribution is represented parametrically, as in Algan, Allais and Den Haan (2010). Appendix A.2 covers these more complex applications, which require a modification of Proposition 1, after which the fake news algorithm continues to apply.

[18] The computation of expectation vectors in step 2 takes far less time than the backward iteration in step 1, especially for the more complex models, because it only requires repeatedly multiplying by $\Lambda_{ss}$—which can be split into multiplication by a small transition matrix for the exogenous state, and multiplication by a highly sparse matrix with policies for endogenous states, both of which we implement efficiently.

[19] There are two additional steps required for the fast algorithm, steps 3 and 4. Step 3 involves the multiplication of $T \times n_g$ and $n_g \times T$ matrices, which has a cost proportional to $n_g T^2$ for each input-output pair—but since matrix multiplication is implemented extremely efficiently by standard numerical libraries, this is less of a bottleneck overall than the backward iteration in step 1, especially for models like the two-asset HANK where backward iteration is especially complex. Step 4 is even faster, since it is a simple recursion on $T \times T$ matrices.

Appendix A also covers how to approach models featuring discrete choice: for example, over the extensive margin of labor supply (e.g. Chang and Kim 2007), or over resetting a price or investing in the presence of fixed costs (Golosov and Lucas 2007, Khan and Thomas 2008). These fit into our original framework when taste shocks smooth out the discrete choice (appendix A.1), and into our extended framework in other cases (appendix A.2). Since these decision problems are often naturally posed in multiple stages, in appendix A.3 we further extend our framework to accommodate multiple stages within each period.

Our extended framework in appendix A allows for very general equations governing the distribution $\mathbf{D}_t$ and aggregate outputs $\mathbf{Y}_t$. An important limitation, however, is that it does not change the structure of equation (10): in particular, the value function $\mathbf{v}_t$ is not allowed to depend on $\mathbf{D}_t$. This prevents us from applying the fake news algorithm when the behavior of heterogeneous agents depends on the anticipated future distribution through the value function, in a way that cannot be intermediated via aggregates $\mathbf{X}_t$ in general equilibrium. This includes, for instance, OLG models with an endogenous distribution of bequests that are received in mid-life (e.g. de Nardi 2004, Straub 2017), and money-search models where the anticipated distribution of cash balances matters directly for agent decisions (e.g. Molico 2006).

# 4 Obtaining general equilibrium impulse responses

A typical general equilibrium heterogeneous-agent model consists of one or more heterogeneous-agent problems of the type described above, as well as additional sets of equations that govern production, market clearing, and so on. In this section, we explain how to solve for general equilibrium impulse responses once the Jacobians of the underlying heterogeneous-agent problem(s) are known.

## 4.1 General equilibrium in the sequence space

A general equilibrium model in the sequence space is always characterized by a certain system of nonlinear equations

$$\mathbf{H}(\mathbf{U}, \mathbf{Z}) = 0 \tag{27}$$

where $\mathbf{Z}$ denotes the path of exogenous "shocks", with $\mathbf{Z}_t$ any $n_z \times 1$ vector at each $t$, and $\mathbf{U}$ denotes the path of "unknown" vectors, with $\mathbf{U}_t$ an $n_u \times 1$ vector at each $t$. We assume that the model has as many equations (or "targets") as unknowns, and that it is *locally determinate*, i.e. that $\mathbf{H}$ is invertible near the steady state $(\mathbf{U}^{ss}, \mathbf{Z}^{ss})$. Then, equation (27) truncated to a horizon of $T$ is a nonlinear system of $n_u \times T$ equations in $n_u \times T$ unknowns, which delivers the general equilibrium impulse response to any change $d\mathbf{Z}$ in the path of $\mathbf{Z}$ relative to $\mathbf{Z}^{ss}$.

We solve for the impulse responses of the model to first order around the steady state. By the implicit function theorem applied to equation (27), the response of unknowns $d\mathbf{U}$ to the shock $d\mathbf{Z}$

is given by:

$$d\mathbf{U} = -\mathbf{H}_{\mathbf{U}}^{-1}\mathbf{H}_{\mathbf{Z}}d\mathbf{Z} \tag{28}$$

where the Jacobians $\mathbf{H}_{\mathbf{U}}$ and $\mathbf{H}_{\mathbf{Z}}$ are evaluated at $(\mathbf{U}^{ss}, \mathbf{Z}^{ss})$. Hence, to obtain $d\mathbf{U}$, we need a method to construct $\mathbf{H}_{\mathbf{U}}$ and $\mathbf{H}_{\mathbf{Z}}$ from the Jacobians of the heterogeneous-agent problem that we obtained in section 3.

**Dealing with dimensionality: reducing $n_u$ with variable substitution.** There are typically many options for putting a given model into the form of equation (27). Some of these options use more endogenous variables than others. For instance, a simple direct approach consists of including all endogenous variables at time $t$ in the set of unknowns $\mathbf{U}_t$.[20] Under this direct approach, the Jacobians obtained from section 3 enter directly as blocks of the $\mathbf{H}_{\mathbf{U}}$ or $\mathbf{H}_{\mathbf{Z}}$ matrices. However, this direct approach is computationally inefficient, because $n_u$ is then typically quite large in realistic applications—typical DSGE models have dozens of endogenous variables—so that inverting the $n_u T \times n_u T$ matrix $\mathbf{H}_{\mathbf{U}}$ in equation (28) becomes impractical.

This dimensionality problem is well-recognized in the literature solving equation (27) nonlinearly. The typical approach is to alleviate this problem by substituting some endogenous variables for others. For example, in section 2, we used this approach to substitute $r_t$ and $w_t$ and solve only for $K_t$ as an unknown variable. This brought down $n_u$ from 3 to 1, reducing the size of the matrix $\mathbf{H}_{\mathbf{U}}$ by a factor of 9. After any such variable substitution, the $\mathbf{H}$ function becomes a composite of subfunctions from different parts of the model, which we must now differentiate to obtain $\mathbf{H}_{\mathbf{U}}$ and $\mathbf{H}_{\mathbf{Z}}$.

**Obtaining $\mathbf{H}_{\mathbf{U}}$ and $\mathbf{H}_{\mathbf{Z}}$ with automatic differentiation.** One way to differentiate $\mathbf{H}$ is to use an automatic differentiation package. Directly applying such a package on the full set of equations governing $\mathbf{H}(\mathbf{U}, \mathbf{Z})$, including the subset of these equations that involves the heterogeneous-agent problem, would be computationally very costly.[21] This is where the computational advances of section 3.2 are critical: one can compute the Jacobians of the heterogeneous-agent problem using the fake news algorithm, and then supply these Jacobians to the automatic differentiation package.[22] The package then delivers $\mathbf{H}_{\mathbf{U}}$ and $\mathbf{H}_{\mathbf{Z}}$ much faster.

In the following two sections, we describe an explicit procedure to obtain $\mathbf{H}_{\mathbf{U}}$ and $\mathbf{H}_{\mathbf{Z}}$ that does not require an automatic differentiation package. This procedure is closely related to what

[20] In the Krusell-Smith model, if aggregate labor efficiency is normalized to $\sum \pi(e)\, el = 1$, this direct approach sets $\mathbf{U}_t \equiv (K_t, r_t, w_t)$, and includes in $\mathbf{H}_t$ the asset market clearing equation, $H_{1t} \equiv \mathcal{K}_t(\{r_s, w_s\}) - K_t$, as well as the firms' first-order conditions $H_{2t} \equiv r_t + \delta - \alpha Z_t K_{t-1}^{\alpha-1}$ and $H_{3t} \equiv w_t - (1-\alpha) Z_t K_{t-1}^{\alpha}$.

[21] See Ahn, Moll and Schaab (2018a) for an example of such an application of automatic differentiation to the $\mathbf{H}$ function. Effectively, the computer then follows an approach resembling the direct method discussed in section 3.1, and does not take advantage of the speedups afforded by the structure of the Jacobian in Proposition 1. See also Childers (2018) for an application of automatic differentiation to heterogeneous-agent models.

[22] Typically, automatic differentiation packages allow the user to provide custom evaluations of derivatives. A simple alternative is to replace the heterogeneous-agent equations with their linearized counterpart in the $\mathbf{H}$ function. For instance, for the Krusell-Smith model, one would write $\mathcal{K} - K_{ss} = \mathcal{J}^{K,w} \cdot (\mathbf{w} - w_{ss}) + \mathcal{J}^{K,r} \cdot (\mathbf{r} - r_{ss})$ using the Jacobians obtained from section 3.

such packages do, since it also computes derivatives by systematically applying the chain rule, but it has a number of computational and conceptual benefits. Computationally, it affords greater control over how derivatives are computed and accumulated, which we use to increase efficiency and accuracy.[23] Conceptually, by being explicit about the variable substitution pattern embedded in $\mathbf{H}$, it delivers a concise representation of models that can be used to understand how individual "blocks" of the models interact, inspect the transmission mechanism of shocks, and understand the implications of changing blocks one at a time. We also use this representation to recover the state-space law of motion in section 4.3.

### 4.2 Equilibrium computation as a directed acyclic graph

In this section, we introduce a formal way of representing the variable substitutions that underlie the $\mathbf{H}$ function. The idea is to organize the model as a set of blocks arranged along a directed acyclic graph, or DAG. While the technical definition requires some formalism, we note at the outset that all of the main complexities of defining and computing a model in this way can be automated, as we have done in our online code repository.

The general type of model whose Jacobians we can compute consists of any combination of heterogeneous-agent problems (or *heterogeneous-agent blocks*), characterized by the mapping (13), and *simple blocks*, which capture typical aggregate relationships in dynamic macro models. Formally, we define simple blocks as mappings between inputs $\mathbf{X}$ and outputs $\mathbf{Y}$ for which there exist $k, l \in \mathbb{N}$ and a time-invariant function $h$ such that $\mathbf{Y}_t$ is only a function of neighboring $\mathbf{X}_t$'s, that is,

$$\mathbf{Y}_t = h(\mathbf{X}_{t-k}, \ldots, \mathbf{X}_{t+l})$$

For instance, a neoclassical firm sector can be represented as a simple block mapping $\mathbf{X}_t = (K_t, Z_t)$ to $\mathbf{Y}_t = (Y_t, r_t, w_t)$. Combining such a sector with a heterogeneous-agent block mapping $\mathbf{X}_t = (r_t, w_t)$ to $\mathbf{Y}_t = \mathcal{K}_t(\{r_s, w_s\})$, as well as a simple block mapping $\mathbf{X}_t = (\mathcal{K}_t, K_t)$ to market clearing $\mathbf{Y}_t = \mathcal{K}_t - K_t$, we obtain the Krusell-Smith model of section 2. Jacobians of simple blocks are straightforward to compute explicitly.

We call "sequence-space model" any combination of these blocks that maps *shocks* (like $Z_t$) and *unknowns* (like $K_t$) to *targets* (like asset market clearing) along a directed acyclic graph.

**Definition 2.** A *sequence-space model* is defined by:

1. A set of sequence indices $\mathcal{N} = \mathcal{Z} \cup \mathcal{U} \cup \mathcal{O}$, where $\mathcal{Z}$ are *exogenous shocks*, $\mathcal{U}$ are *unknowns*, $\mathcal{O}$ are *outputs*, and $\mathcal{H} \subset \mathcal{O}$ are *targets*,

2. A set of *blocks*, each either simple or heterogeneous-agent blocks, indexed by $\mathcal{B}$, where each block $b \in \mathcal{B}$ has *inputs* $\mathcal{I}_b \subset \mathcal{N}$ and *outputs* $\mathcal{O}_b \subset \mathcal{O}$, such that each output $o \in \mathcal{O}$ belongs

[23] For example, we get efficiency gains by reusing Jacobians for blocks whose parameters do not change, as in the reestimation in section 5.4, and we increase accuracy by avoiding the truncation error of simple blocks, as described in appendix C.4.

to exactly one block, and for each output $o \in \mathcal{O}_b$, block $b$ provides a function $h^o(\{\mathbf{X}^i\}_{i\in\mathcal{I}_b})$ mapping the block's input sequences to this output sequence,

such that a) the number of unknowns and targets are equal, that is, $n_u = n_h$, and b) the *directed graph* of blocks, formed by drawing an edge from $b$ to $b'$ whenever some output $o \in \mathcal{O}_b$ is used as an input $o \in \mathcal{I}_{b'}$, is *acyclic*.

**Definition 3.** An *equilibrium* of a sequence-space model, given sequences $\{\mathbf{X}^i\}_{i\in\mathcal{Z}}$ for the exogenous shocks, is a set of sequences $\{\mathbf{X}^i\}_{i\in\mathcal{U}\cup\mathcal{O}}$ such that: a) $\mathbf{X}^o = h^o(\{\mathbf{X}^i\}_{i\in\mathcal{I}_b})$ for any output $o \in \mathcal{O}$, and b) $\mathbf{X}^o = 0$ for any target $o \in \mathcal{H}$.

**Definition 4.** A *steady state equilibrium* is an equilibrium in which all sequences are constant over time, $\mathbf{X}^i_t = \mathbf{X}^i_{ss}$ for all $i \in \mathcal{N}$.

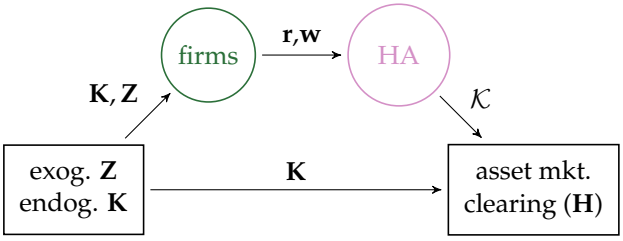
A sequence-space model thus consists of a combination of blocks that are linked along a directed graph. Each individual block covers a different aspect of the economy and computes either equilibrium conditions themselves (i.e. outputs that are also "targets") , or variables that are useful for other blocks (i.e. outputs that are also inputs). Any such variable can be viewed as having been "substituted out" and need not be carried around as an unknown. Thus, the directed graph represents the (possibly complex) pattern of variable substitution that the modeler uses to reduce the number of unknowns.

An important property we require is that the directed graph that connects blocks be *acyclic*, that is, it does not feature circular dependencies across blocks. Instead, there is always an ordering $b^1, \ldots, b^{n_b}$ of the blocks (formally, a "topological sort") such that all input variables used in later blocks, e.g. $b^3$ or $b^4$, are either output variables from earlier blocks, e.g. $b^1$ or $b^2$ (in which case those variables were substituted out) or they are shocks or unknowns. Recursively, this implies that starting with shocks and unknowns $\{\mathbf{X}^i, i \in \mathcal{Z}\cup\mathcal{U}\}$ we can follow along this ordering, computing each block's output, one-by-one, starting with block $b^1$ (which only uses inputs that are either shocks or unknowns), then moving to block $b^2$ (whose inputs can also be outputs of $b^1$), and so on. When we are done, we will have calculated all outputs, including the target sequences $\mathcal{H}$.

Thus, this procedure gives us a mapping from shocks and unknowns $\{\mathbf{X}^i, i \in \mathcal{Z}\cup\mathcal{U}\}$ to targets $\{\mathbf{X}^o\}_{o\in\mathcal{H}}$, with possibly lots of intermediate variables that are computed as outputs and used as inputs in between. We can write this mapping in condensed form as $\mathbf{H}(\mathbf{U}, \mathbf{Z})$, where $\mathbf{U}$ is defined as the stacked vector of unknown sequences $\{\mathbf{X}^i\}_{i\in\mathcal{U}}$, $\mathbf{Z}$ is defined as the stacked vector of exogenous sequences $\{\mathbf{X}^i\}_{i\in\mathcal{Z}}$, and $\mathbf{H}(\mathbf{U}, \mathbf{Z})$ itself is the implied stacked vector of targets $\{\mathbf{X}^i\}_{i\in\mathcal{H}}$. When we combine these sequences with all intermediate variables $\mathbf{X}^o = h^o(\{\mathbf{X}^i\}_{i\in\mathcal{I}_b})$ that are computed along the DAG when evaluating $\mathbf{H}$, the only equilibrium condition that is left is that targets are equal to zero, $\mathbf{H}(\mathbf{U}, \mathbf{Z}) = 0$.

**Example: Krusell-Smith model.** Figure 3 visualizes the DAG for the Krusell-Smith model that corresponds to the variable substitutions we made in section 2. It has three blocks (neoclassical

Figure 3: DAG representation of Krusell-Smith economy




firms, heterogeneous households "HA", and a block to compute the asset market clearing condition $H$), one exogenous shock (productivity $\mathcal{Z} = \{Z\}$), one unknown (capital $\mathcal{U} = \{K\}$), four outputs (capital return, wage, household savings, asset market clearing, so $\mathcal{O} = \{r, w, \mathcal{K}, H\}$), and one target (asset market clearing $\mathcal{H} = \{H\}$).[24] The DAG is best read from left to right. The "firms" block maps the unknown sequence of capital stocks $K$ and the exogenous shocks $Z$ into the interest rate and wage sequences $r, w$. Those are then used to substitute out $r, w$ in the "HA" block, before asset market clearing $H$ is computed.

**One and two-asset HANK models.** The Krusell-Smith model allows for a relatively straightforward DAG that reduces the number of unknowns to $n_u = 1$. Figure 4 gives a DAG for a more complex case: a one-asset HANK model similar to McKay, Nakamura and Steinsson (2016). This model combines standard NK elements—sticky prices, flexible wages, and a Taylor rule for monetary policy, but no capital—with a one-asset incomplete market HA household sector where labor supply is endogenous. It is introduced formally in appendix B.2.

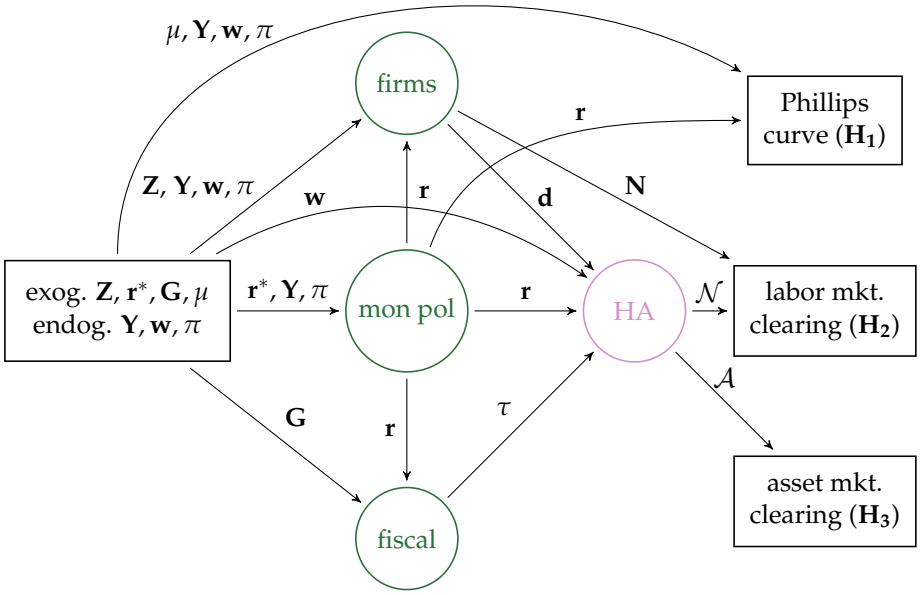
As figure 4 shows, the DAG for this model features three unknowns (wages $w$, output $Y$, and inflation $\pi$), which are used to compute six intermediate outputs, ultimately yielding three targets (a Phillips curve condition $H_1$, labor market clearing $H_2$, and asset market clearing $H_3$). In other words, the DAG substitutes out six variables that would otherwise have to be included as unknowns. We introduce four exogenous shocks (productivity $Z$, Taylor rule intercept $r^*$, government spending $G$, and markups $\mu$).

The DAG makes it easy to visualize the dependencies between macroeconomic aggregates that are embedded in the model: for instance, the dividends from firms are distributed to households (according to a certain rule), so the output $d$ of the firm block is an input to the HA block. Similarly, the real interest rate $r$ affects the taxes required for the government to achieve its balanced-budget target, so $r$ is an input to the fiscal block, which has an output $\tau$ that is an input to the HA block.

Even as models grow in complexity beyond this one, they often still admit DAGs with small numbers of unknowns and targets. In appendix B.3, we introduce our third model example, a two-asset HANK model, and we show that still only three unknowns and targets are required to write down its DAG. This DAG has 18 intermediate variables being computed by its blocks, thus

[24] Not visualized are firm production $Y$ or household consumption $C$, which could be additional outputs of the firm and HA blocks, respectively, but are not strictly necessary since we are using asset rather than goods market clearing to define equilibrium.

Figure 4: DAG representation of one-asset HANK economy




corresponding to a complicated web of 18 variable substitutions. The model is similar to Kaplan, Moll and Violante (2018), with households saving in liquid and illiquid assets subject to convex adjustment costs of portfolio adjustment. On the supply side, it features wage as well as price rigidities, as well as capital with quadratic adjustment costs.

### 4.3 Jacobians and impulse responses

We now show how to use the DAG representation of the model to automatically evaluate the Jacobians $\mathbf{H_U}$ and $\mathbf{H_Z}$. To do so, we systematically apply the chain rule along the model's DAG, implementing a technique known as forward accumulation in the automatic differentiation literature (Griewank and Walther 2008). This technique combines the Jacobians of individual blocks to build up $\mathbf{H_U}$ and $\mathbf{H_Z}$. The key idea is to apply the chain rule in the same order that we would evaluate a function itself.[25]

**Total Jacobians J.** To start, we need a new concept. For any exogenous shock or unknown $i \in \mathcal{Z} \cup \mathcal{U}$ and any output $o$, let the $\mathbf{J}^{o,i}$ denote the *total Jacobian* of $o$ with respect to $i$ when $o$ is evaluated along the DAG. For instance, in the one-asset HANK model in figure 4, the total Jacobian $\mathbf{J}^{\mathcal{N},w}$ of household labor supply with respect to wages combines two forces: the direct effect of $w$ on household decisions, and the indirect effect working through the influence of $w$ on firm profits and therefore the dividends $d$ received by households. This is in contrast to $\mathcal{J}^{\mathcal{N},w}$, which is a

[25] In actual computations, the methods in this section will be applied on Jacobians that are truncated to some horizon $T \times T$. For simple blocks we use a simple sparse representation of the Jacobian, described in appendix C.4, and do not need to truncate.

partial Jacobian that captures only the direct effect.

To obtain $\mathbf{J}^{o,i}$ through forward accumulation, we first initialize $\mathbf{J}^{i,i}$ to the identity for each $i \in \mathcal{Z} \cup \mathcal{U}$. We then go through blocks following the ordering (topological sort) $b^1, \ldots, b^{n_b}$ one-by-one beginning with $b^1$. For each block $b$, we evaluate the total Jacobian of all its outputs $o \in \mathcal{O}_b$ with respect to shocks and unknowns $i \in \mathcal{Z} \cup \mathcal{U}$:

$$\mathbf{J}^{o,i} = \sum_{m \in \mathcal{I}_b} \mathcal{J}^{o,m} \mathbf{J}^{m,i} \tag{29}$$

This systematically applies the chain rule: for each input $m$, (29) takes the product of the partial Jacobian $\mathcal{J}^{o,m}$ with the already-calculated total derivative $\mathbf{J}^{m,i}$ of $m$ with respect to $i$. (When $m = i$, then the latter is the identity and the term is just the partial Jacobian $\mathcal{J}^{o,i}$.) The benefit of building up the $\mathbf{J}^{o,i}$ progressively via forward accumulation is that the chain rule is applied in an efficient way, without redundant computations.

**General equilibrium Jacobians G.** Using the $\mathbf{J}$ matrices, we can compute the total Jacobians of all targets with respect to all unknowns and shocks, $\mathbf{H_U} = \mathbf{J}^{\mathcal{H},\mathcal{U}}$ and $\mathbf{H_Z} = \mathbf{J}^{\mathcal{H},\mathcal{Z}}$. By equation (28), these are the objects needed to solve the equilibrium response of unknowns, $d\mathbf{U} = -\mathbf{H_U}^{-1}\mathbf{H_Z} d\mathbf{Z}$. We can compute this response for any arbitrary shock vector $d\mathbf{Z}$ by simple multiplication with the matrix $-\mathbf{H_U}^{-1}\mathbf{H_Z}$. We refer to this matrix as *general equilibrium Jacobian* of unknowns to shocks, and denote it by $\mathbf{G}^{\mathcal{U},\mathcal{Z}}$.

It is often important to also compute the general equilibrium responses of variables that are not unknowns. To do this, we generalize the idea of the general equilibrium Jacobian to all outputs. We denote by $\mathbf{G}^{o,z}$ the matrix that maps an arbitrary sequence of a shock $z \in \mathcal{Z}$ into the corresponding impulse response of any output $o \in \mathcal{O}$. We adopt the convention that $\mathbf{G}^{o,\mathcal{Z}}$ is the matrix one arrives at by stacking $\mathbf{G}^{o,z}$ for all $z \in \mathcal{Z}$. Given $\mathbf{G}^{o,\mathcal{Z}}$, one computes the impulse response $d\mathbf{X}^o$ with a simple matrix multiplication:

$$d\mathbf{X}^o = \mathbf{G}^{o,\mathcal{Z}} d\mathbf{Z} \tag{30}$$

To compute $\mathbf{G}^{o,\mathcal{Z}}$ for any output $o$, we trace the same forward accumulation steps as before, and build $\mathbf{G}^{o,\mathcal{Z}}$ recursively, using[26]

$$\mathbf{G}^{o,\mathcal{Z}} = \sum_{m \in \mathcal{I}_b} \mathcal{J}^{o,m} \mathbf{G}^{m,\mathcal{Z}} \tag{31}$$

Each $\mathbf{G}^{o,\mathcal{Z}}$ has $n_z T$ columns, each of which can be interpreted as the impulse response of $o$ to some news shock. One way to think about this approach, therefore, is that we are simultaneously calculating $n_z T$ general equilibrium impulse responses. For our Krusell-Smith, one-asset HANK, and two-asset HANK models, $n_z T$ is $1 \times 300 = 300$, $3 \times 300 = 900$, and $3 \times 300 = 900$, respectively.[27]

[26] An alternative is to re-use total Jacobians and write $\mathbf{G}^{o,\mathcal{Z}} = \mathbf{J}^{o,\mathcal{U}}\mathbf{G}^{\mathcal{U},\mathcal{Z}} + \mathbf{J}^{o,\mathcal{Z}}$. In our experience, the recursive approach tended to be more efficient.

[27] If one only needs to compute one impulse response, it is possible to obtain this impulse response faster using an alternative method described in appendix C.6. Interestingly, it is not too much more expensive to calculate the full set of impulse responses in $\mathbf{G}$: in our one-asset HANK example, obtaining 900 rather than one impulse response only takes

Table 3: Computing times for $\mathbf{G}$.

| | Krusell-Smith | one-asset HANK | two-asset HANK |
|---|---|---|---|
| **Total** | **3.3 ms** | **50.6 ms** | **173.5 ms** |
| step 1 (forward accumulate $\mathbf{H_U}$ and $\mathbf{H_Z}$) | 0.6 ms | 7.5 ms | 27.0 ms |
| step 2 (compute $\mathbf{G}^{\mathcal{U},\mathcal{Z}} = -\mathbf{H_U}^{-1}\mathbf{H_Z}$) | 1.2 ms | 25.9 ms | 51.6 ms |
| step 3 (forward accumulate for all $\mathbf{G}^{o,\mathcal{Z}}$) | 1.5 ms | 17.2 ms | 95.0 ms |
| No. of unknowns | 1 | 3 | 3 |
| No. of exogenous shocks | 1 | 3 | 7 |

Comparing table 3 to table 2, we see that computing $\mathbf{G}$s is, in each of our cases, significantly cheaper than applying our fast algorithm to obtain $\mathcal{J}$s for the heterogeneous-agent block: for instance, it takes about 50 milliseconds for the one-asset HANK model, while the fake news algorithm took 320 milliseconds. This shows the power of $\mathcal{J}$s as sufficient statistics: once we have them, it is just a matter of linear algebra to obtain a full characterization of equilibrium.[28]

**Recovering the state-space law of motion.** From the linearized solution in the sequence space, given a particular shock process expressed in state-space form, it is possible to recover the equivalent state-space law of motion. The general idea is to determine the effect that a perturbation to any state, and any innovation to the shock process, has on all states in the following period. This can be done in three steps: one first finds (a) the effect of the perturbation on the targets $d\mathbf{H}$, then (b) the response of unknowns to targets $d\mathbf{U}$, and finally (c) the response of next-period states to unknowns. Since the distribution of agents is a state, this process requires two pieces of information about the distribution, which are both computed by the fake news algorithm in section 3.2: the expectation vectors $\mathcal{E}_t^o$ for step (a), and the distribution perturbation vectors $\mathcal{D}_1^i$ for step (c).[29] One can then use the state-space law of motion for any standard application, such as simulation or estimation using state-space methods.

### 4.4 Numerical accuracy: equivalence to the Reiter method

We now establish that our implementation of the sequence-space Jacobian ("SSJ") method is accurate by comparing the solution it produces to that obtained using the Reiter method.

In principle, the two methods should give the same solution, i.e. the first-order solution of the model in aggregates. Indeed, they are both intended to solve to first order the same system of equations, which includes (10)-(12) and simple blocks from $t = 0, \ldots, \infty$. There are, however, two potential concerns to rule out. First, both our method and Reiter—like all first-order perturbation

about 5 times as long. This is possible because we only need to calculate $\mathbf{H_U}$ once, independent of shocks.

[28] Appendix C.7 also shows that using the direct approach to computation described in section 4.1, rather than a DAG representation with a small number of unknowns, comes at a significant computational cost.

[29] Appendix C.8 provides additional details about this procedure for the case of the Krusell-Smith model with AR(1) TFP shocks.

methods—are subject to error when computing derivatives. Second, rather than solving the true infinite-dimensional system in sequence space, our method truncates sequence-space Jacobians at some high $T$.[30]

To verify accuracy, we choose as our benchmark the Reiter method with automatic differentiation. Automatic differentiation, unlike numerical differentiation, ensures that there is no approximation error when computing derivatives. We then show that the SSJ method—with the same parameterization on the same grid and $T = 300$, and the fake news algorithm implemented using automatic differentiation—delivers identical impulse responses to Reiter, and that its state-space law of motion is also identical.[31] We find that the SSJ method with numerical differentiation delivers larger but still relatively minor errors. We finally discuss practical considerations in choosing $T$.

As is well known, the main bottleneck of the Reiter method is that it cannot be used directly with large idiosyncratic state spaces without model reduction (the method scales in the cube of the size of the idiosyncratic state space). For this reason, we restrict our comparison to the Krusell-Smith model and the one-asset HANK model, each computed on a small grid of $n_g = 300$ points (100 asset grid points and 3 income states). We describe our implementation of the Reiter method in appendix C.2.

**Preliminaries: benchmark representative-agent models.** Before proceeding, we first check that our implementation of the SSJ method is accurate for models without heterogeneity. Appendix D.2 verifies that the linear impulse responses of output to all shocks in the Smets and Wouters (2007) model and in the Herbst and Schorfheide (2015) model—two benchmark models used extensively for estimation in the literature—are identical to those obtained using the first-order solution from Dynare, which uses standard state-space methods for computation.

**Accuracy of impulse responses.** Figure 5 compares the impulse responses from the Reiter method with automatic differentiation, and the impulse responses produced by our SSJ method using the three types of differentiation discussed in section 3.2. All panels compute impulse responses to a 1% TFP shock with persistence $\rho = 0.9$, and the SSJ truncation horizon is chosen at $T = 300$. The top panels show the impulse response of capital in the Krusell-Smith model. The bottom panels show the impulse response of output in the one-asset HANK model. Panels (a) and (c) plot the impulse responses in levels using all four methods. The impulse responses look visually identical. To get a sense of the differences, panels (b) and (d) plot the absolute value of the differences between the lines in the left panel and the Reiter solution at each $t$. When solved with automatic

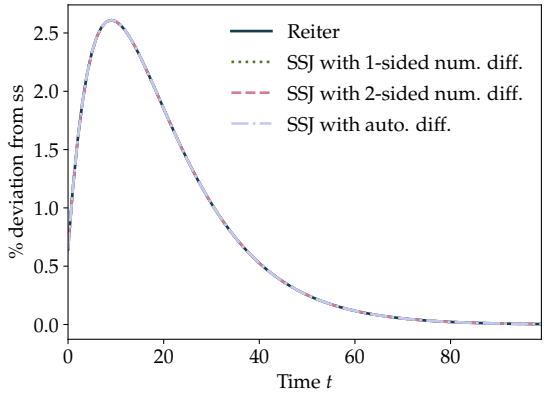
[30] Recall from equations (10)–(12) that we take as given a discretized model, which we take to be the "true" model. We compare the solution to the Reiter method applied to the same discretized model. If the "true" model is continuous instead, getting to the discretized model in the first place involves some error, but this does not affect the comparison here.

[31] Note that this usage of automatic differentiation in the backward iteration of the fake news algorithm, which is discussed in section 3.2, is distinct from the usage of automatic differentiation discussed in section 4.1: the former is to obtain error-free derivatives of $v$, $\Lambda$, and $y$ in (10)–(12), while the latter is to obtain derivatives of $H$.

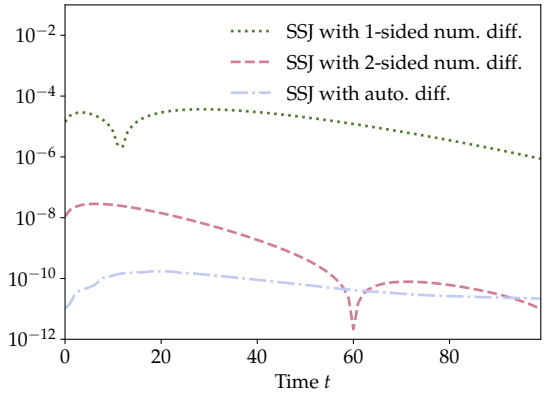
Figure 5: Equivalence between the SSJ and the Reiter (2009) methods.

Krusell-Smith model

(a) Impulse responses of capital across methods

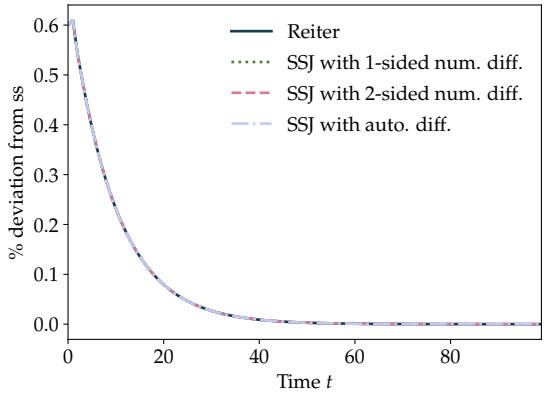



(b) Differences to Reiter (2009)

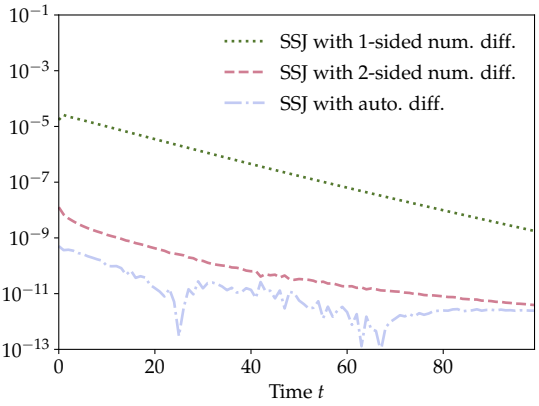



One-asset HANK model

(c) Impulse responses of output across methods



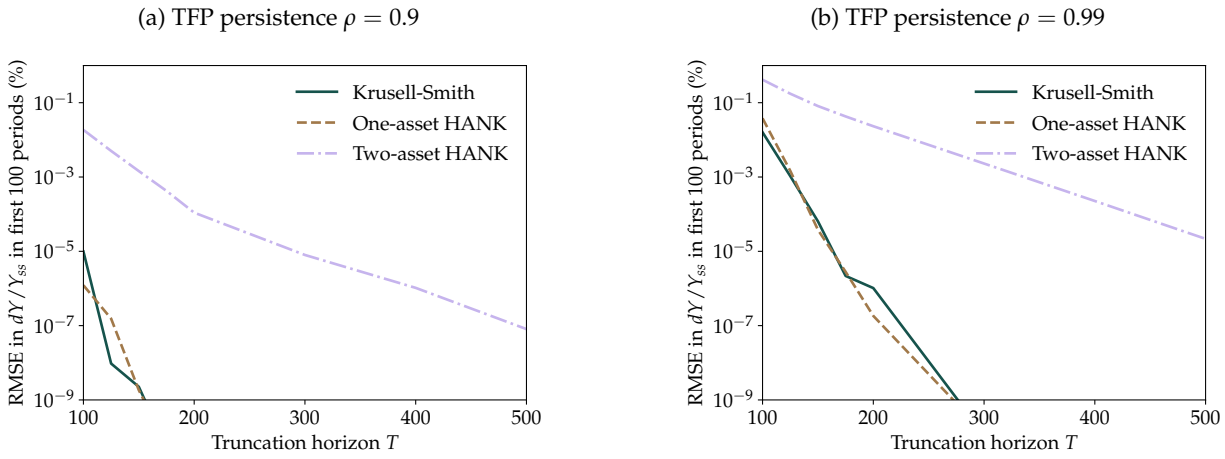

(d) Differences to Reiter (2009)




differentiation, our method yields essentially the same result as the Reiter solution with automatic differentiation, up to ten digits of accuracy. Applying our method with numerical differentiation predictably introduces a small error, comparable to the errors typically found when implementing the Reiter method with numerical differentiation. Despite this error already being small, one can reduce it significantly at just twice the computational cost by using two-sided numerical differentiation. This is important because two-sided numerical differentiation is often easier to implement than automatic differentiation in practice.

**Accuracy of the state-space law of motion.** While comparison of individual impulse responses is a useful proof of accuracy, a more definitive test is to verify that the state-space laws of motion are identical. To do this, we compare the matrices describing the state-space law of motion in

Figure 6: Impulse response error (relative to $T = 1000$) as a function of the truncation horizon $T$




the Krusell-Smith model and the one-asset HANK model, both computed via automatic differentiation so as not to introduce numerical differentiation error. We find that the sup norm of the difference between these matrices is below $10^{-8}$ for both models.

**Choice of truncation horizon $T$.** We have established that our method, for large enough $T$, delivers the same solution as the Reiter method. In practice, however, a separate computation via the Reiter method is usually not available as a benchmark. In these cases, how can one ensure that $T$ is, in fact, high enough? To answer this question, we now examine the sensitivity of the sequence-space solution to $T$.

We take as our benchmark the sequence space solution with a very large $T$ ($T = 1000$), well above the horizon required to achieve agreement with the Reiter method in our previous exercise. We then see how the impulse response in the first 100 periods differs from this benchmark as we reduce $T$. Throughout, we compute Jacobians with one-sided numerical differentiation, since it is the easiest and most commonly used in practice.

Figure 6 performs this exercise for our three models, each in response to a 1% shock to TFP with persistence $\rho$. In the left panel, $\rho = 0.9$. We see that even for truncation horizons $T$ far shorter than the $T = 300$ used in this paper, the RMSE of the output impulse response $dY/Y_{ss}$ is near zero in both the Krusell-Smith and one-asset HANK models. The two-asset HANK model, on the other hand, only achieves five digits of accuracy by $T = 300$. This is because that model has much greater internal persistence: the magnitude of the underlying output impulse response at $t = 300$ is above $10^{-4}$%, while it is below $10^{-9}$% for the two other models. (By that point, the shock itself is below $10^{-13}$%.)

The right panel looks at a more extreme case, with $\rho = 0.99$. Here, the shock at $t = 300$ remains at 5% of its level on impact. Now the two-asset HANK model achieves less than 3 digits of accuracy even with $T = 300$, and about 5 digits of accuracy with $T = 500$. This indicates the

importance of a high $T$ when dealing with highly persistent shocks, especially in models with high internal persistence.[32]

Overall, to be sure that the truncation horizon is long enough, it is most important to ensure that both the shock itself and the endogenous impulse response—which may feature some internal persistence—are near zero by $T$. An additional check is to make sure that changing $T$ to a higher value does not change the impulse response: as figure 6 indicates, when $T$ is not yet high enough to deliver accuracy, the results are sensitive to changes in $T$. These two simple checks ensure that one has minimized truncation error, leaving numerical differentiation as the only possible remaining source of error relative to the true first-order solution of the model in aggregates.

## 5 Application to estimation

We now discuss how to use the sequence-space Jacobian method to estimate models on time-series data. This application uses the equivalence of impulse responses with the moving-average ($MA$) representation of the model with aggregate shocks. This equivalence, in turn, follows immediately from the certainty equivalence property of first-order perturbation methods such as ours (see e.g. Simon 1956, Theil 1957, Judd and Guu 1993, and Boppart, Krusell and Mitman 2018).

We use the convention that variables with tildes denote observed time paths, while variables without tildes denote impulse responses. We assume that each shock $z \in \mathcal{Z}$ follows the moving-average process

$$d\tilde{Z}_t^z = \sum_{s=0}^{\infty} dZ_s^z \epsilon_{t-s}^z \tag{32}$$

where $\{\epsilon_t^z\}$ are mutually iid standard normally distributed innovations. It follows from (32) that a unit innovation to $\epsilon_t^z$ has an impulse response of $dZ_0^z, dZ_1^z, \ldots$, starting at $t$.

In the previous section, we showed how to find the impulse responses $dX_0^o, dX_1^o, \ldots$ for any output $o \in \mathcal{O}$ in response to simultaneous impulses to all shocks $z \in \mathcal{Z}$. Now, we are considering the impulse of $o$ driven by an innovation $\epsilon_t^z$ for a specific $z$, and we denote this impulse by $dX_0^{o,z}, dX_1^{o,z}, \ldots$. Truncating all impulses to horizon $T$, and writing $\{dX_s^{o,z}\}_s \equiv d\mathbf{X}^{o,z}$ and $\{dZ_s^z\}_s \equiv d\mathbf{Z}^z$, we can apply equation (30) to obtain

$$d\mathbf{X}^{o,z} = \mathbf{G}^{o,z} d\mathbf{Z}^z \tag{33}$$

Certainty equivalence then implies that each output $d\tilde{X}_t^o$ follows the moving-average process

$$d\tilde{X}_t^o = \sum_{s=0}^{T-1} \sum_{z \in \mathcal{Z}} dX_s^{o,z} \epsilon_{t-s}^z. \tag{34}$$

[32] Interestingly, the error in the first 100 periods in the other two models is below $10^{-9}$% with $T = 300$, in spite of the shock's persistence. Of course, there is still error near $t = 300$, as is inevitable when the shock has not died out by the truncation horizon, but this error does not propagate backward in these models to nearly the same extent. This lack of backward propagation is related to the weaker internal persistence in these models relative to the two-asset HANK model.

In summary, an $MA\,(T-1)$ representation of $d\widetilde{\mathbf{X}}$ can be obtained by simple matrix multiplications of the general equilibrium Jacobians $\mathbf{G}$ from section 4.3 with the shock impulse-response vectors $d\mathbf{Z}$.[33] The rapid computation of this $MA$ representation is the foundation of our applications in this section, which will build up to likelihood-based estimation of our three models.

### 5.1 Simulation

As pointed out by Boppart, Krusell and Mitman (2018), the formulation in equation (34) is useful to simulate sample paths for aggregate variables generated by any model, including a heterogeneous-agent model. Assume impulse responses $d\mathbf{X}^{o,z}$ have been computed for a given outcome $o$ for all shocks $z$, with truncation horizon $T$. Then, a procedure to simulate a random sample path of $o$ is as follows: first, draw paths for the shock innovations, that is, a sequence $\{\epsilon_t^z\}$ for each shock $z$, up to a large horizon $\mathbb{T}$. Second, evaluate (34) for each $t$. Finally, discard the first $T$ elements of the path $d\tilde{X}_t^o$. This procedure generates a random sample path of outcome $o$ of length $\mathbb{T}-T$.[34] Panel (a) of Figure 7 presents an example of such simulations for the Krusell-Smith model with AR(1) TFP shocks.[35]

Table 1 shows that, given the $\mathbf{G}$ matrices, this simulation procedure is extremely fast in practice: to draw sample paths of length 100,000 for the observables used in the estimation of our four main models at their posterior modes in section 5.4, we only need 5 ms for the Krusell-Smith model (one observable, one shock), 22 ms for the one-asset HANK model (three observables, three shocks), and 105 ms for the two-asset HANK model (seven observables, seven shocks).

One use of these simulated sample paths, as Boppart, Krusell and Mitman (2018) show, is to compute the second moments of outcomes—their variances and covariances, at various lags and leads. In the next section, however, we will instead discuss a more efficient, analytical way of computing these moments, based directly on the impulse responses $d\mathbf{X}^o$.

### 5.2 Analytical second moments

Let $d\tilde{\mathbf{X}}_t = (d\tilde{X}_t^o)_{o\in\mathcal{O}}$ be the vector-valued stochastic process of all $n_o$ outputs, and let $d\mathbf{X}_s$ be the stacked $n_o \times n_z$ matrix of $MA$ coefficients $dX_s^{o,z}$. The autocovariances of $d\tilde{\mathbf{X}}_t$ are given by the standard expression (see, for instance, Box and Jenkins 1970 and Hamilton 1994):

$$\operatorname{Cov}(d\tilde{\mathbf{X}}_t, d\tilde{\mathbf{X}}_{t'}) = \sum_{s=0}^{T-1-(t'-t)} (d\mathbf{X}_s)\,(d\mathbf{X}_{s+t'-t})' \tag{35}$$

The covariance in (35) only depends on the distance $t'-t$, not on $t$ and $t'$ separately.

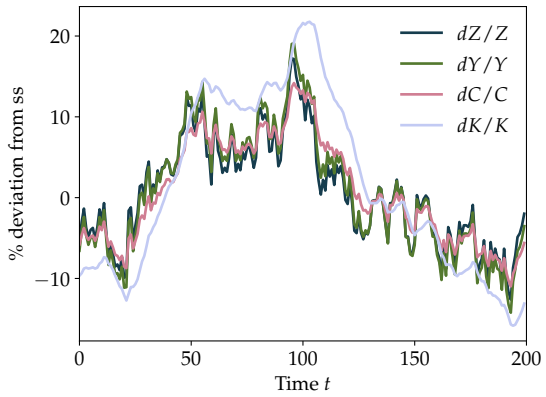
[33] In appendices A.1 and A.2, we describe how to calculate Jacobians when $o$ is a distributional statistic, like the coefficient of variation or quantiles of the asset distribution. Given the resulting $\mathbf{G}^{o,\mathcal{Z}}$, we can apply (33) to get the $MA$ representation, just like with any other $o$.

[34] In appendix C.9, we explain how to extend this procedure to simulate panels of individuals.

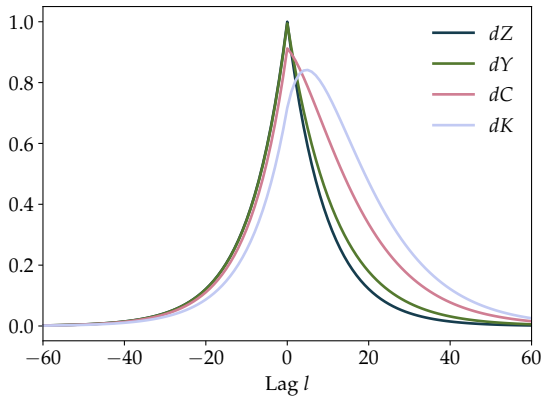
[35] For this simulation, we assume an AR(1) process for TFP with persistence $\rho = 0.9$, and innovations with standard deviation of $\sigma = 0.02$. We set $T = 300$ and $\mathbb{T} = 500$, so there are 200 periods of observation.

Figure 7: Simulations and second moments of the Krusell and Smith (1998) model for AR(1) TFP, $\rho = 0.9$

(a) Simulation of various series $X$




(b) Corr$(dX_{t+l}, dZ_t)$ for various series $X$




In panel (b) of figure 7, we provide an illustration of these second moments for the parameterization of the stochastic Krusell-Smith model simulated in panel (a). The figure shows the correlations of productivity, output, consumption, and capital with the underlying productivity process, at various lags. The figure shows that capital and consumption—and to a much lesser extent, output—tend to lag productivity. This reflects the typical transmission mechanism of TFP shocks in RBC models.

As table 4 reveals, it is very fast to calculate autocovariances in this way: for our estimation exercises of section 5.4, moving from the $MA(T-1)$ representation to a full set of autocovariances, which are stacked in constructing the matrix $\mathbf{V}$, only takes between 0.4 and 0.7 milliseconds.[36] These autocovariances can be used directly to calibrate or estimate a model—as in the simulated method of moments, but without the need for explicit simulation.[37] Alternatively, they can be used to evaluate the likelihood function, as an input to likelihood-based estimation on time series data. This is where we turn next.

### 5.3 Evaluating the likelihood function

The typical approach to likelihood-based estimation in the DSGE literature is to compute the likelihood by applying the Kalman filter to the model's state-space representation (see e.g. Smets and Wouters 2007, An and Schorfheide 2007, Herbst and Schorfheide 2015, and Fernández-Villaverde, Rubio-Ramírez and Schorfheide 2016). This approach is appropriate for models with small state spaces. With the large state spaces that characterize heterogeneous-agent models, however, evaluating the likelihood in this fashion can be prohibitively slow.

[36] This is facilitated by using the fast Fourier transform to calculate (35) in a highly efficient way, a process that we describe in appendix C.10.

[37] For recent examples of this approach to estimation, see Auclert and Mitman (2020) and Bardóczy (2020). For a previous instance of an analytical approach to calculating second moments from a heterogeneous-agent model, see Harmenberger and Sievertsen (2017).

We now suggest an alternative approach: to use the $MA$ representation provided by the sequence-space Jacobian method to rapidly compute (and recompute) the likelihood. The idea of using the $MA$ representation of a DSGE model directly to calculate the likelihood goes back to at least Hansen and Sargent (1981). There are multiple ways to perform this calculation. The approach we employ in our application builds directly on the analytical moments from the previous section.[38] Let

$$d\tilde{\mathbf{X}}_t^{obs} = B\, d\tilde{\mathbf{X}}_t + \mathbf{u}_t \tag{36}$$

denote the vector of $n_{obs}$ observables whose likelihood we would like to determine.[39] Here $\{\mathbf{u}_t\}$ is iid normal with mean 0 and covariance matrix $\Sigma_u$, and $B$ is a $n_{obs} \times n_o$ matrix. Since $d\tilde{\mathbf{X}}_t^{obs}$ is a linear combination of the $\epsilon_t^z$ and $\mathbf{u}_t$ terms, it has a multivariate normal distribution. Moreover, its second moments are a simple linear transformation of those of $d\tilde{\mathbf{X}}_t$:

$$\mathrm{Cov}(d\tilde{\mathbf{X}}_t^{obs}, d\tilde{\mathbf{X}}_{t'}^{obs}) = 1_{t=t'} \cdot \Sigma_{\mathbf{u}} + B\,\mathrm{Cov}(d\tilde{\mathbf{X}}_t, d\tilde{\mathbf{X}}_{t'})\, B' \tag{37}$$

We stack these covariances into a large symmetric $n_{obs}T_{obs} \times n_{obs}T_{obs}$ matrix $\mathbf{V}$, where $T_{obs}$ is the number of time periods in our data.[40] The log-likelihood function is then the conventional log multivariate density. Dropping the constant term, it can be expressed as a function of the observed data $d\tilde{\mathbf{X}}^{obs} = (d\tilde{\mathbf{X}}_t^{obs})$ (stacked as an $n_{obs}T_{obs}$-dimensional vector) as

$$\mathcal{L} = -\frac{1}{2}\log\det\mathbf{V} - \frac{1}{2}\left[d\tilde{\mathbf{X}}^{obs}\right]'\mathbf{V}^{-1}\left[d\tilde{\mathbf{X}}^{obs}\right] \tag{38}$$

We evaluate this expression by performing a Cholesky decomposition of $\mathbf{V}$, from which we can quickly calculate both the log determinant $\log\det\mathbf{V}$ and the quadratic form $\left[d\tilde{\mathbf{X}}^{obs}\right]'\mathbf{V}^{-1}\left[d\tilde{\mathbf{X}}^{obs}\right]$.[41] Table 4 reveals that this is quite efficient in our applications: calculating $\mathcal{L}$ takes about 2 milliseconds or less in all except the two-asset HANK, where it takes about 12 milliseconds.

A weakness of this approach is that the Cholesky decomposition of $\mathbf{V}$ requires time proportional to $n_{obs}^3 T_{obs}^3$. As the number of time-series observations $T_{obs}$ grows, this can become quite costly.[42] An alternative that scales better with $T_{obs}$ is to use the Whittle approximation to the likelihood, as in Hansen and Sargent (1981) and Plagborg-Møller (2019), which can be efficiently

[38] Recent papers in the DSGE literature that use the same approach include Mankiw and Reis (2007) and Schmitt-Grohé and Uribe (2010).

[39] These may include moments of micro data (if we interpret $\mathbf{u}_t$ as sampling error), since we can construct model impulse responses for such moments using our extended methods from appendices A.1 and A.2. More comprehensively integrating micro data into time series estimation requires other methods. For promising work along these lines, see Chang, Chen and Schorfheide (2018) and Plagborg-Møller and Liu (2019).

[40] Missing or mixed-frequency data can be easily accommodated by replacing the $B$ in (36) with a time-specific $B_t$, which can have a time-varying number of rows. The second term on the right in (37) then becomes $B_t\,\mathrm{Cov}(d\tilde{\mathbf{X}}_t, d\tilde{\mathbf{X}}_{t'})\, B_{t'}'$.

[41] We provide an accuracy check of our implementation of (38) by using it to find the posterior modes of the Herbst and Schorfheide (2015) and the Smets and Wouters (2007) models, on their original datasets. Table D.1 in appendix D.3 shows that the modes are numerically identical to those obtained in Dynare given the state-space formulations of these models.

[42] One relatively minor modification, which exploits the block Toeplitz structure of $\mathbf{V}$, is to use Levinson recursion instead of the Cholesky decomposition (e.g. Meyer-Gohde 2010). Asymptotically, this scales with $T_{obs}^2$ instead, although for our applications it did not deliver a major improvement.

calculated using the Fast Fourier Transform.

Another approach is to construct a state-space system from the $MA$ representation, including the most recent $T$ realizations of each innovation $\epsilon_t^z$, and then apply the Kalman filter. This system is distinct from the usual state-space one, and does not scale with the underlying heterogeneity—but since its size is proportional to the truncation horizon $T$, it is often large enough in our applications that applying the Kalman filter is costly. Still, this approach has a number of advantages: for instance, its cost only scales linearly in $T_{obs}$, and if desired we can apply the Kalman smoother to do inference on shocks.

### 5.4 Bayesian estimation

In this section, we perform a Bayesian estimation of macro parameters for our three example economies. Our primary objective is to illustrate that, by reusing Jacobians, this can be done very efficiently with the SSJ method. We first estimate the posterior mode, and then use a standard Markov chain Monte Carlo method (Random Walk Metropolis Hastings, RWMH) to trace out the shape of the posterior distribution, as described in Herbst and Schorfheide (2015). We leave a detailed understanding of the economics behind the estimation results to future research.

**Reusing Jacobians.** Likelihood-based estimation involves computing the likelihood function many times for different parameters. In our case, given equation (38), this requires computing the covariance matrix of model observations $\mathbf{V}$ for each parameter draw, which in turn requires computing the impulse responses $d\mathbf{X}$. Our key innovation is to make the repeated computation of $d\mathbf{X}$ very efficient by reusing Jacobians. The benefits of this procedure, however, depend on which parameters we are estimating.

Consider first the estimation of the parameters of shock processes. In this case, the matrices $\mathbf{G}^{o,z}$ in equation (33) is unchanged across parameter draws, since these parameters only change the vector $d\mathbf{Z}^z$. Hence, it is sufficient to compute all $\mathbf{G}$ matrices once, and then for each parameter draw form $d\mathbf{X}$, $\mathbf{V}$ and $\mathcal{L}$ without re-solving the model. This separation constitutes a clear advantage of our sequence-space approach relative to a state-space approach to estimation.

Next, consider the estimation of parameters that do not change the steady state of the heterogeneous-agent problem. This includes parameters that govern price stickiness, capital adjustment costs, or monetary policy rules. These parameters also do not affect the Jacobian of the heterogeneous-agent problem. Hence, in the construction of the total model Jacobian in equation (29), these Jacobians can be held fixed at their initial value.[43] Since heterogeneous-agent Jacobians are by far the most time-consuming step in obtaining $\mathbf{G}$ (see Table 1), this is still very fast.

Finally, consider estimating parameters that do change the steady state of the heterogeneous agent problem. There, the Jacobian of that problem needs to be recomputed for each new draw of parameters, together with the steady state. While the fake news algorithm speeds up Jacobian

[43] This statement is true, more generally, of the Jacobian of any block in the DAG whose parameters do not change.

computation considerably, the additional time cost of re-evaluating the steady state on each draw remains substantial, and we do not pursue this type of estimation here.[44]

**Priors, data, and estimation.** We now proceed to our main estimation exercise. Across all models, we assume the following prior distributions. We assume that the priors for the standard deviations of all shocks are Inverse-Gamma distributed with mean 0.4 and standard deviation 4. We assume that the priors for persistence parameters are Beta distributed with mean 0.5 and standard deviation 0.2. We also assume no measurement error, $\Sigma_{\mathbf{u}} = 0$. We describe the priors of model-specific parameters below. We search for the posterior mode using a standard optimization routine, starting with the prior mode as our initial guess.[45] We run a standard RWMH in which the proposal distribution is a multivariate normal with variance equal to the inverse Hessian at the posterior mode, scaled by a factor $c$ that is adjusted to get an acceptance rate around 25%.[46] We simulate the Markov chain for 100,000 draws, discarding the first 50,000.

For each model, we use the same U.S. data series as those used in Smets and Wouters (2007), over the same sample period (1966:1–2004:4). We linearly de-trend the logs of all growing variables (output, consumption, investment, wages, hours) and take out the sample means of inflation and nominal interest rates. The individual models are then estimated as follows.

**Krusell-Smith model.** We estimate our Krusell and Smith (1998) model with a single shock, TFP, and a single time series $\{dX_t^{obs}\}$, output. We assume that TFP shocks follow an $ARMA(1,1)$ process, $(1-\rho L)d\tilde{Z}_t = (1-\theta L)\sigma\epsilon_t$, where $L$ denotes the lag operator. We estimate the roots $\rho, \theta$ as well as the standard deviation $\sigma$. Table E.1 in appendix E shows the estimates, finding a persistent AR root $\rho \approx 0.9$, as well as a relatively small $MA$ component $\theta \approx 0.03$. Recursive means and univariate plots of the posterior distribution sampled with the RWMH algorithm suggest that the Markov chain has converged and the posterior distribution is well behaved.

**One-asset HANK model.** We estimate our one-asset HANK model both with only shock parameters, and with shock and model parameters together. In both cases, we use three shocks (monetary policy shocks, government spending shocks, and price markup shocks) and three time series (output, inflation, and nominal interest rates). Each shock is modeled as an $AR(1)$ with its own standard deviation and persistence. Thus, there are six shock parameters for this model. The first four posterior columns in table E.2 in appendix E show our estimates when only estimating

[44] In the literature, Winberry (2018), Auclert, Rognlie and Straub (2020), and Bayer, Born and Luetticke (2020) all calibrate the steady-state micro parameters governing the heterogeneous-agent problem, and use time series data only to estimate macro parameters, as we do here. In recent work, Acharya, Cai, Del Negro, Dogra, Matlin and Sarfati (2020) use time series data to also estimate micro parameters, with sequential Monte Carlo methods to speed up estimation. Also see Plagborg-Møller and Liu (2019), who estimate micro parameters using a mix of micro and macro data.

[45] Specifically, we use the SciPy implementation of L-BFGS-B, imposing some non-binding bounds to guide the routine away from poorly-behaved regions of the parameter space.

[46] One simple improvement, which we do not attempt, might be to use a proposal distribution where a positive probability of draws only change the parameters of shock processes, not other parameters. Since we can re-use $\mathbf{G}$, calculating the likelihood for these draws would be much faster.

those shock parameters; we find persistent government spending shock and price markup shocks, while monetary policy shocks are less so. The last four posterior columns in this table report the estimated shock and model parameters in the joint estimation. We find a Taylor coefficient $\phi$ of around 1.3, a modest responsiveness of the Taylor rule to output $\phi_y \approx 0.13$, and a Phillips curve slope parameter $\kappa$ around 0.14. These are standard values in the literature. Again, recursive means and posterior distribution plots suggest good convergence properties for the RWMH algorithm.

**Two-asset HANK model.** We add all seven shocks from Smets and Wouters (2007) to the two-asset model: shocks to TFP, government spending, monetary policy, price and wage markups. The two exceptions are that we use discount factor shocks rather than "risk premium" shocks (both shock the Euler equation and are thus very similar), and we shock firms' first-order conditions for capital instead of using investment-specific technology shocks.[47] We estimate the parameters of these seven shock processes using time series data on output, consumption, investment, hours, wages, nominal interest rates and price inflation. As with the one-asset model, we estimate two versions of the model, one with only shock parameters and one with shock and model parameters (table E.3 in appendix E). Compared to the one-asset model, we find here somewhat less responsive coefficients of the Taylor rule on inflation and output at the mode, but also a much wider 90% credible interval. We also find smaller Phillips curve slope parameters $\kappa^p, \kappa^w$. We also estimate the degree of capital adjustment costs $\epsilon_I$ and find it to be in line with standard estimates from the literature. The evolution of the recursive means across the 50,000 non-discarded draws, as well as the estimated posterior distributions, suggest good convergence properties when we only estimate shocks, but less stability when estimating both shocks and parameters. This could be due to the fact that the model is not designed explicitly to fit the hump shapes in the time series; see Auclert, Rognlie and Straub (2020) for a model that addresses this shortcoming.

**Estimation times.** Table 4 lists computing times for each of our five estimation exercises, including times for each likelihood evaluation and their breakdown into the three steps described in section 5.3.

Once the $\mathbf{G}^{o,z}$ matrices are computed (table 3), the Krusell-Smith model's likelihood can be evaluated in less than one millisecond, and the posterior mode can be found in around 60 milliseconds. The entire RWMH estimation takes just over a minute. We attain similar speeds estimating the shock processes for the one-asset HANK model (just under 5 minutes for RWMH). Since we allow for seven shocks when estimating the two-asset HANK model, estimating the parameters of these shock processes is somewhat slower than in the other two models; this has nothing to do with the complexity or micro heterogeneity of the two-asset model. Still, a single likelihood evaluation only takes a few milliseconds, the posterior mode is found in a few seconds, and RWMH estimation takes about 24 minutes.

[47] Investment-specific technology shocks are known to have counterfactual implications for the relative price of investment—see, for instance, Justiniano, Primiceri and Tambalotti (2011).

Table 4: Estimation times.

| | Krusell-Smith | one-asset HANK | | two-asset HANK | |
|---|---|---|---|---|---|
| | shocks | shocks | model + shocks | shocks | model + shocks |
| **Single likelihood evaluation** | **0.639 ms** | **2.353 ms** | **56.001 ms** | **13.992 ms** | **227.245 ms** |
| step 1 ($MA$) | 0.015 ms | 0.091 ms | 53.686 ms | 1.139 ms | 214.396 ms |
| step 2 (autocovariances) | 0.041 ms | 0.144 ms | 0.146 ms | 0.706 ms | 0.712 ms |
| step 3 (log-likelihood) | 0.582 ms | 2.117 ms | 2.169 ms | 12.148 ms | 12.137 ms |
| **Posterior mode optimization** | **0.06 s** | **0.66 s** | **13.95 s** | **16.22 s** | **522.03 s** |
| no. of evaluations | 81 | 237 | 580 | 1094 | 6560 |
| **Random Walk Metropolis Hastings** | **66.20 s** | **284.21 s** | **5608.50 s** | **1449.79 s** | **21281.95 s** |
| no. of evaluations | 100,000 | 100,000 | 100,000 | 100,000 | 100,000 |
| acceptance rate | 0.248 | 0.246 | 0.251 | 0.255 | 0.245 |
| scaling factor $c$ | 2.50 | 1.10 | 0.65 | 0.40 | 0.10 |
| No. of shocks | 1 | 3 | 3 | 7 | 7 |
| No. of estimated shock parameters | 3 | 6 | 6 | 14 | 14 |
| No. of estimated model parameters | 0 | 0 | 3 | 0 | 5 |
| Total no. of estimated parameters | 3 | 6 | 9 | 14 | 19 |

When model parameters are also estimated, the likelihood takes a bit longer to be re-evaluated. This is entirely due to step 1—the computation of impulse responses. The single likelihood evaluation for the two-asset model, for instance, takes 227 ms rather than 14 ms when model parameters change, and finding the posterior mode takes less than 9 minutes. Running RWMH with 100,000 evaluations on the two-asset model when estimating 19 shock processes and model parameters takes less than 6 hours. To the best of our knowledge, these are much faster speeds for estimation of such models than what any other method has been able to achieve at a comparable level of accuracy.

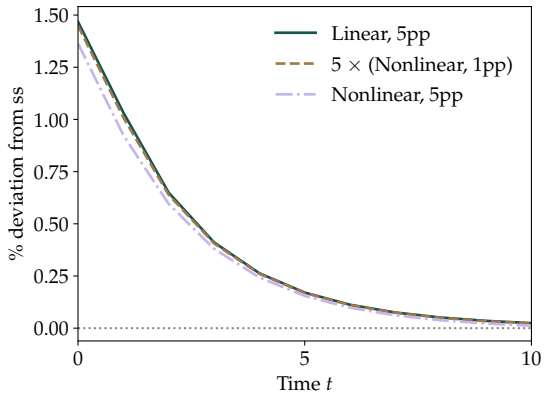
Our main contribution in this section, the idea of reusing Jacobians, is essential to achieving these speeds. To illustrate this, observe that re-evaluating the heterogeneous-agent Jacobian 100,000 times for the two-asset model would take about 4 full days with our fake news algorithm (approximately 3.5 s per evaluation), and about 3 years with the direct algorithm (approximately 950 s per evaluation).

## 6 Application to nonlinear perfect foresight transitions

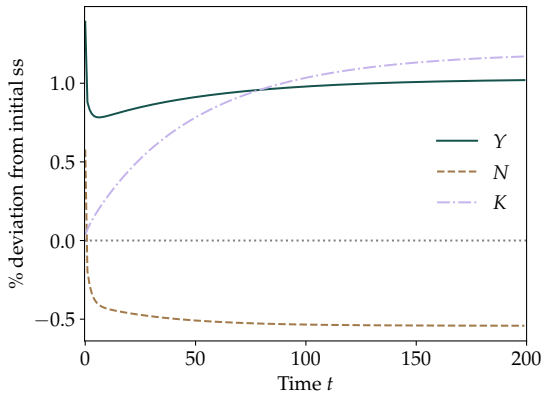
We now discuss how to use sequence-space Jacobians to obtain nonlinear solutions to equation (27). These solutions are the nonlinear perfect foresight impulse responses to unexpected shocks perturbing the aggregate steady state at date 0 (sometimes called "MIT shocks"). In the literature, these are used by researchers to explore size dependence and sign asymmetries (see e.g. Kaplan

Figure 8: Nonlinear impulse responses and transitional dynamics for the two-asset HANK model

(a) Consumption after shocks to Taylor rule




(b) Transition after a 1% permanent TFP shock




and Violante 2018 for fiscal policy and Berger, Guerrieri, Lorenzoni and Vavra 2018 for house price changes), and to simulate transitions between two steady states in applications that involve long-term changes (see e.g. Heathcote, Storesletten and Violante 2010 for rising inequality, and Krueger and Ludwig 2007 for demographic change). Recently, Boppart, Krusell and Mitman (2018) have also suggested examining the extent of size dependence in these shocks as a test of how closely the first-order aggregate perturbation matches the nonlinear solution with aggregate risk.

To find the $\mathbf{U}$ that solves $\mathbf{H}(\mathbf{U}, \mathbf{Z}) = 0$ for a given $\mathbf{Z}$, truncated to some $T$, we use the following iterative procedure. First, starting from $j = 0$, guess a path $\mathbf{U}^0$ (typically, $\mathbf{U}^0 = \mathbf{U}_{ss}$). Second, calculate $\mathbf{H}\left(\mathbf{U}^j, \mathbf{Z}\right)$. Third, form the $j+1$ guess using

$$\mathbf{U}^{j+1} = \mathbf{U}^j - \left[\mathbf{H}_{\mathbf{U}}\left(\mathbf{U}_{ss}, \mathbf{Z}_{ss}\right)\right]^{-1} \mathbf{H}\left(\mathbf{U}^j, \mathbf{Z}\right) \tag{39}$$

This algorithm falls in the class of quasi-Newton methods,[48] since the steady-state sequence space Jacobian $\mathbf{H}_{\mathbf{U}}\left(\mathbf{U}_{ss}, \mathbf{Z}_{ss}\right)$ is used instead of the actual Jacobian $\mathbf{H}_{\mathbf{U}}\left(\mathbf{U}^j, \mathbf{Z}\right)$.[49] We illustrate this method using our two-asset HANK model in two ways.

**Nonlinear impulse responses.** Panel (a) of figure 8 shows three impulse responses of consumption in the two-asset HANK model in response to monetary policy shock with quarterly persistence $\rho = 0.6$. Two are the linear and nonlinear impulse responses to a -5pp shock to the Taylor rule, and the other is the nonlinear impulse response to a -1pp shock, scaled up by a factor of 5.

[48] This idea of Newton's method to compute nonlinear impulse responses dates back to Laffargue (1990), Boucekkine (1995), and Juillard (1996). For heterogeneous-agent models, previous versions of the method in (39) were implemented by approximating the $\mathbf{H}_{\mathbf{U}}\left(\mathbf{U}_{ss}, \mathbf{Z}_{ss}\right)$ matrix: see, among others, Auclert and Rognlie (2018), Straub (2017), and Koby and Wolf (2020). For an example using automatic differentiation to obtain $\mathbf{H}_{\mathbf{U}}\left(\mathbf{U}_{ss}, \mathbf{Z}_{ss}\right)$, see Ahn, Moll and Schaab (2018a).

[49] One alternative is to build an approximation to $\mathbf{H}_{\mathbf{U}}\left(\mathbf{U}^j, \mathbf{Z}\right)$ for each new guess $\mathbf{U}^j$, holding heterogeneous-agent Jacobians constant at their steady state values but using exact Jacobians elsewhere. This is useful when there are substantial nonlinearities originating outside the heterogeneous-agent block.

The linear and scaled-up nonlinear impulse responses are almost identical, indicating that linearity is an accurate assumption for -1pp shocks. The nonlinear impulse response to the -5pp shock is visibly a bit smaller than the other two, but still similar: 1.36% on impact, rather than 1.46%. This similar response, despite the extreme size of the monetary shock, suggests that nonlinearities in the household model—such as large shocks moving households away from their borrowing constraints—do not play an important role for plausibly-sized shocks.

The algorithm above converges to $|\mathbf{H}| < 10^{-8}$ in 13 iterations for the -5pp shock, and 5 iterations for the -1pp shock. By contrast, methods that rely on ad-hoc adjustment criteria sometimes require hundreds of iterations before convergence of equilibrium sequences.

In Table 1, we use the algorithm to compute nonlinear impulse responses for all four of our models, and report the time this requires, which ranges from 0.32 s for Krusell-Smith to 15 s for two-asset HANK. (For comparability, these numbers are for a 1% shock to TFP, which is available in every model.)

**Transition to a new steady state.** We can use this algorithm to compute the response to a permanent shock. Here, it important to use the Jacobian $\mathbf{H}_{\mathbf{U}}(\mathbf{U}_{ss}, \mathbf{Z}_{ss})$ around the terminal steady state. For example, panel (b) of figure 8 reports the nonlinear transition, starting from the initial steady state of our two-asset HANK model, to a one-time permanent shock of 1% to TFP. In this example, it takes 7 iterations to reach $|\mathbf{H}| < 10^{-8}$.

## 7 Conclusion

This paper presents a highly efficient method for computing heterogeneous-agent models. The core idea is that *sequence-space Jacobians* are sufficient statistics that summarize all we need to know about the heterogeneity in order to determine general equilibrium dynamics, to first order with respect to aggregate shocks. Our main contribution is a fast algorithm for computing the Jacobians of heterogeneous-agent problems. We combine this algorithm with a systematic method for computing model Jacobians and linearized impulse responses. We apply these objects to estimate models with high-dimensional state spaces, and compute their nonlinear transitional dynamics.

Our methods allow us to find the posterior mode of a two-asset HANK model in under ten minutes and trace its with posterior distribution with MCMC in under six hours, estimation times that had so far been out of reach for the literature. We hope that they will prove useful to solve and estimate other heterogeneous-agent models and facilitate new developments in the field.

## References

**Acharya, Sushant, Michael D Cai, Marco Del Negro, Keshav Dogra, Ethan Matlin, and Reca Sarfati**, "Estimating HANK: Macro Time Series and Micro Moments," *Manuscript*, March 2020.

**Achdou, Yves, Jiequn Han, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll**, "Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach," *Review of Economic Studies*, October 2020, *Forthcoming*.

**Ahn, SeHyoun, Benjamin Moll, and Andreas Schaab**, "Huggett Model: Transition Dynamics, Newton method using @myAD. Available at https://benjaminmoll.com/wp-content/uploads/2020/06/huggett_newton_myAD.m," April 2018.

**_ , Greg Kaplan, Benjamin Moll, Thomas Winberry, and Christian Wolf**, "When Inequality Matters for Macro and Macro Matters for Inequality," *NBER Macroeconomics Annual*, April 2018, *32* (1), 1–75.

**Algan, Yann, Olivier Allais, and Wouter J. Den Haan**, "Solving the Incomplete Markets Model with Aggregate Uncertainty Using Parameterized Cross-Sectional Distributions," *Journal of Economic Dynamics and Control*, January 2010, *34* (1), 59–68.

**_ , _ , _ , and Pontus Rendahl**, "Chapter 6 - Solving and Simulating Models with Heterogeneous Agents and Aggregate Uncertainty," in Karl Schmedders and Kenneth L. Judd, eds., *Handbook of Computational Economics*, Vol. 3, Elsevier, January 2014, pp. 277–324.

**An, Sungbae and Frank Schorfheide**, "Bayesian Analysis of DSGE Models," *Econometric Reviews*, April 2007, *26* (2-4), 113–172.

**Auclert, Adrien and Kurt Mitman**, "Consumer Bankruptcy as Aggregate Demand Management," *Manuscript*, August 2020.

**_ and Matthew Rognlie**, "Inequality and Aggregate Demand," Working Paper 24280, National Bureau of Economic Research, February 2018.

**_ , Bence Bardóczy, Matthew Rognlie, and Ludwig Straub**, "Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models," Working Paper 26123, National Bureau of Economic Research, July 2019.

**_ , Matthew Rognlie, and Ludwig Straub**, "The Intertemporal Keynesian Cross," Working Paper 25020, National Bureau of Economic Research, September 2018.

**_ , _ , and _** , "Micro Jumps, Macro Humps: Monetary Policy and Business Cycles in an Estimated HANK Model," Working Paper 26647, National Bureau of Economic Research, January 2020.

**Bardóczy, Bence**, "Spousal Insurance and the Amplification of Business Cycles," Job Market Paper, Northwestern University 2020.

**Bayer, Christian and Ralph Luetticke**, "Solving Discrete Time Heterogeneous Agent Models with Aggregate Risk and Many Idiosyncratic States by Perturbation," *Quantitative Economics*, November 2020, *11* (4), 1253–1288–1288.

_ , **Benjamin Born, and Ralph Luetticke**, "Shocks, Frictions, and Inequality in US Business Cycles," *Manuscript*, January 2020.

**Berger, David, Veronica Guerrieri, Guido Lorenzoni, and Joseph Vavra**, "House Prices and Consumer Spending," *Review of Economic Studies*, July 2018, *85* (3), 1502–1542.

**Boppart, Timo, Per Krusell, and Kurt Mitman**, "Exploiting MIT Shocks in Heterogeneous-Agent Economies: The Impulse Response as a Numerical Derivative," *Journal of Economic Dynamics and Control*, April 2018, *89*, 68–92.

**Boucekkine, Raouf**, "An Alternative Methodology for Solving Nonlinear Forward-Looking Models," *Journal of Economic Dynamics and Control*, May 1995, *19* (4), 711–734.

**Box, George E. P. and Gwilym M. Jenkins**, *Time Series Analysis: Forecasting and Control*, San Francisco: Holden-Day, 1970.

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

**Carroll, Christopher D.**, "The Method of Endogenous Gridpoints for Solving Dynamic Stochastic Optimization Problems," *Economics Letters*, 2006, *91* (3), 312–320.

**Chang, Minsu, Xiaohong Chen, and Frank Schorfheide**, "Heterogeneity and Aggregate Fluctuations," *Manuscript*, October 2018.

**Chang, Yongsung and Sun-Bin Kim**, "Heterogeneity and Aggregation: Implications for Labor-Market Fluctuations," *American Economic Review*, December 2007, *97* (5), 1939–1956.

**Childers, David**, "Automated Solution of Heterogeneous Agent Models," *Manuscript*, December 2018.

**Christiano, Lawrence, Cosmin L. Ilut, Roberto Motto, and Massimo Rostagno**, "Monetary Policy and Stock Market Booms," *Jackson Hole Symposium Proceedings*, August 2010.

**de Nardi, Mariacristina**, "Wealth Inequality and Intergenerational Links," *Review of Economic Studies*, 2004, *71* (3), 743–768.

**den Haan, Wouter J.**, "Solving Dynamic Models with Aggregate Shocks and Heterogeneous Agents," *Macroeconomic Dynamics*, June 1997, *1* (2), 355–386.

**Fernández-Villaverde, Jesús, Juan F. Rubio-Ramírez, and Frank Schorfheide**, "Chapter 9 - Solution and Estimation Methods for DSGE Models," in John B. Taylor and Harald Uhlig, eds., *Handbook of Macroeconomics*, Vol. 2, Elsevier, January 2016, pp. 527–724.

_ , **Samuel Hurtado, and Galo Nuño**, "Financial Frictions and the Wealth Distribution," Working Paper 26302, National Bureau of Economic Research, September 2019.

**Golosov, Mikhail and Robert E. Lucas**, "Menu Costs and Phillips Curves," *Journal of Political Economy*, April 2007, *115* (2), 171–199.

**Gornemann, Nils, Keith Kuester, and Makoto Nakajima**, "Doves for the Rich, Hawks for the Poor? Distributional Consequences of Monetary Policy," *Manuscript*, April 2016.

**Griewank, Andreas and Andrea Walther**, *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation, Second Edition*, SIAM, November 2008.

**Hamilton, James D.**, *Time Series Analysis*, 1 ed., Princeton University Press, January 1994.

**Hansen, Lars Peter and Thomas J. Sargent**, "Exact Linear Rational Expectations Models: Specification and Estimation," *Federal Reserve Bank of Minneapolis Staff Report*, 1981.

**Harmenberger, Karl and Hans Henrik Sievertsen**, "The Labor-Market Origins of Cyclical Income Risk," *Manuscript*, November 2017.

**Heathcote, Jonathan, Kjetil Storesletten, and Giovanni L. Violante**, "The Macroeconomic Implications of Rising Wage Inequality in the United States," *Journal of Political Economy*, August 2010, *118* (4), 681–722.

**Herbst, Edward P. and Frank Schorfheide**, *Bayesian Estimation of DSGE Models*, Princeton University Press, December 2015.

**Hopenhayn, Hugo A.**, "Entry, Exit, and firm Dynamics in Long Run Equilibrium," *Econometrica*, 1992, *60* (5), 1127–1150.

**Iskhakov, Fedor, Thomas H. Jørgensen, John Rust, and Bertel Schjerning**, "The Endogenous Grid Method for Discrete-Continuous Dynamic Choice Models with (or Without) Taste Shocks," *Quantitative Economics*, 2017, *8* (2), 317–365.

**Judd, Kenneth L. and Sy-Ming Guu**, "Perturbation Solution Methods for Economic Growth Models," in Hal R. Varian, ed., *Economic and Financial Modeling with Mathematica®*, New York, NY: Springer New York, 1993, pp. 80–103.

**Juillard, Michel**, "Dynare: A Program for the Resolution and Simulation of Dynamic Models with Forward Variables Through the Use of a Relaxation Algorithm," *CEPREMAP Working Paper no 9602*, 1996.

**Justiniano, Alejandro, Giorgio E. Primiceri, and Andrea Tambalotti**, "Investment Shocks and the Relative Price of Investment," *Review of Economic Dynamics*, 2011.

**Kaplan, Greg and Giovanni L. Violante**, "Microeconomic Heterogeneity and Macroeconomic Shocks," *Journal of Economic Perspectives*, August 2018, *32* (3), 167–194.

**_ , Benjamin Moll, and Giovanni L. Violante**, "Monetary Policy According to HANK," *American Economic Review*, March 2018, *108* (3), 697–743.

**Khan, Aubhik and Julia K. Thomas**, "Idiosyncratic Shocks and the Role of Nonconvexities in Plant and Aggregate Investment Dynamics," *Econometrica*, March 2008, *76* (2), 395–436.

**Koby, Yann and Christian Wolf**, "Aggregation in Heterogeneous-Firm Models: Theory and Measurement," *Manuscript*, July 2020.

**Krueger, D., K. Mitman, and F. Perri**, "Chapter 11 - Macroeconomics and Household Heterogeneity," in John B. Taylor and Harald Uhlig, eds., *Handbook of Macroeconomics*, Vol. 2, Elsevier, January 2016, pp. 843–921.

**Krueger, Dirk and Alexander Ludwig**, "On the Consequences of Demographic Change for Rates of Returns to Capital, and the Distribution of Wealth and Welfare," *Journal of Monetary Economics*, January 2007, *54* (1), 49–87.

**Krusell, Per and Anthony A. Smith**, "Income and Wealth Heterogeneity in the Macroeconomy," *Journal of Political Economy*, October 1998, *106* (5), 867–896.

**Laffargue, Jean-Pierre**, "Résolution D'un Modèle Macroéconomique Avec Anticipations Rationnelles," *Annales d'Économie et de Statistique*, 1990, (17), 97–119.

**Lorenzoni, Guido**, "A Theory of Demand Shocks," *American Economic Review*, December 2009, *99* (5), 2050–2084.

**Mankiw, N. Gregory and Ricardo Reis**, "Sticky Information in General Equilibrium," *Journal of the European Economic Association*, May 2007, *5* (2-3), 603–613.

**McKay, Alisdair, Emi Nakamura, and Jón Steinsson**, "The Power of Forward Guidance Revisited," *American Economic Review*, October 2016, *106* (10), 3133–3158.

**Mertens, Thomas M. and Kenneth L. Judd**, "Solving an Incomplete Markets Model with a Large Cross-Section of Agents," *Journal of Economic Dynamics and Control*, June 2018, *91*, 349–368.

**Meyer-Gohde, Alexander**, "Linear Rational-Expectations Models with Lagged Expectations: A Synthetic Method," *Journal of Economic Dynamics and Control*, May 2010, *34* (5), 984–1002.

**Molico, Miguel**, "The Distribution of Money and Prices in Search Equilibrium," *International Economic Review*, August 2006, *47* (3), 701–722.

**Plagborg-Møller, Mikkel**, "Bayesian Inference on Structural Impulse Response Functions," *Quantitative Economics*, 2019, *10* (1), 145–184.

_ **and Laura Liu**, "Full-Information Estimation of Heterogeneous Agent Models Using Macro and Micro Data," *Manuscript*, June 2019.

**Proehl, Elisabeth**, "Approximating Equilibria with Ex-Post Heterogeneity and Aggregate Risk," SSRN Scholarly Paper ID 2620937, Social Science Research Network, Rochester, NY February 2019.

**Reiter, Michael**, "Solving Heterogeneous-Agent Models by Projection and Perturbation," *Journal of Economic Dynamics and Control*, March 2009, *33* (3), 649–665.

_ , "Approximate and Almost-Exact Aggregation in Dynamic Stochastic Heterogeneous-Agent Models," *IHS Working Paper 258*, 2010.

**Schmitt-Grohé, Stephanie and Martín Uribe**, "Evaluating the Sample Likelihood of Linearized DSGE Models Without the Use of the Kalman Filter," *Economics Letters*, December 2010, *109* (3), 142–143.

**Simon, Herbert A.**, "Dynamic Programming Under Uncertainty with a Quadratic Criterion Function," *Econometrica*, 1956, *24* (1), 74–81.

**Smets, Frank and Rafael Wouters**, "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, June 2007, *97* (3), 586–606.

**Straub, Ludwig**, "Consumption, Savings, and the Distribution of Permanent Income," *Manuscript*, November 2017.

**Theil, Henry**, "A Note on Certainty Equivalence in Dynamic Planning," *Econometrica*, 1957, *25* (2), 346–349.

**Whiteman, Charles**, *Linear Rational Expectations Models: A User's Guide*, Univ Of Minnesota Press, April 1983.

**Winberry, Thomas**, "A Method for Solving and Estimating Heterogeneous Agent Macro Models," *Quantitative Economics*, November 2018, *9* (3), 1123–1151–1151.

**Young, Eric R**, "Solving the Incomplete Markets Model with Aggregate Uncertainty Using the Krusell–Smith Algorithm and Non-Stochastic Simulations," *Journal of Economic Dynamics and Control*, 2010, *34* (1), 36–41.

# Online Appendix to "Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models"

## A Generalizing the fake news algorithm

### A.1 Direct applications of the existing framework

First, we identify several ways in which the existing framework can be adapted to include model elements differing from our examples, with either no change or limited changes to the algorithm.

**Non-grid representations of value function.** (10)-(12) assume that the distribution is discretized as a finite grid, and that $y$ and $\Lambda$ give the value of the output $Y$ at each point and the transition probabilities between points. None of this places any restriction, however, on how the value function is discretized as $\mathbf{v}$. Our algorithm therefore accommodates a variety of discrete representations of $\mathbf{v}$ (splines, Chebyshev polynomials, parametric, etc.) without any modification.

**Higher moments.** At first glance, (12) seems to require that we are taking the mean $\mathbf{y}_t'\mathbf{D}_t$ of some individual outcome $\mathbf{y}_t$. But if we redefine the individual outcome as $\mathbf{y}_t^k$, then we can the calculate $k$th (non-centered) power moment $\left(\mathbf{y}_t^k\right)'\mathbf{D}_t$ as well. Applying this strategy as necessary for different $k$ and combining the results using a simple block, we can obtain the Jacobian for any transformation of these moments, such as the variance, the coefficient of variation, or a CES price index.

This allows us to calculate many moments of interest, though not all; for instance, for some distributional moments like the Gini coefficient, we need the general framework of the next section.[50]

**Leads and lags.** The equations (10)-(12) include only contemporaneous $\mathbf{X}_t$, without any leads or lags. What if, instead, a lagged or future variable appears, such as $\mathbf{X}_{t-1}$ or $\mathbf{X}_{t+1}$? In the case of leads like $\mathbf{X}_{t+1}$, the algorithm works without any change: lemma 1 goes through without modification, so that iterating backward from a shock at $T-1$ still gives the $d\mathbf{y}_0^s$ and $d\Lambda_0^s$ needed in proposition 1. Intuitively, this is because our backward iteration already incorporates the effects of a future shock working through the value function, and nothing more is needed to handle the case where future $X$ also appears directly in (10)-(12).

If, on the other hand, a lag like $\mathbf{X}_{t-1}$ appears in (10)-(12), then it is no longer true that $\mathbf{y}_t^s = \mathbf{y}_{ss}$ and $\Lambda_t^s = \Lambda_{ss}$ for $t = s+1$ in (16), because both are affected by the lagged shock. Lemma 1 fails,

[50] Note, however, that this is only necessary if we need Jacobians for these moments. If, instead, we only need impulse responses for these moments (and the moments themselves are not needed to solve for general equilibrium), we can apply the linearized (10)-(12) to the equilibrium impulse responses for $\mathbf{X}_t$ and recover impulse responses $\mathbf{y}_t$ and $\mathbf{D}_t$, then directly compute any desired moments from these.

and our method—which does not account for the possibility that "past" shocks affect current individual outcomes at a particular point in the state space—no longer works.

The simplest solution is to transform variables outside the heterogeneous-agent block, e.g. define a new variable $\tilde{\mathbf{X}}_t \equiv \mathbf{X}_{t-1}$ (which can be the output of a simple block taking in $\mathbf{X}$), so that within the algorithm, only a contemporaneous variable $\tilde{\mathbf{X}}_t$ appears, matching the exact form of (10)-(12).[51]

**Discrete choice with taste shocks.** The models we simulate in this paper all have the feature that policy functions are continuous in the underlying idiosyncratic state variables. This is no longer generally the case for models that feature discrete choices, such as lumpy adjustment of durables, price setting with menu costs, or a discrete labor-leisure choice (see e.g. Bardóczy 2020). For such models, if the problem is discretized using a grid, linearization can give extremely misleading results: if none of the grid points at a point where the discrete choice changes, then the first-order response of the discrete choice to any shock is zero.

This problem is common to all perturbation methods. One standard solution is to assume continuously-distributed iid taste shocks affecting the value of each discrete choice. The probability of each discrete choice then varies continuously with the (pre-taste shock) state.[52] To write the model in the form (10)-(12), $\mathbf{D}_t$ should then be the discretized pre-taste shock distribution, and $\mathbf{v}_t, \mathbf{y}_t, \Lambda_t$ should be the expected values at each state in this distribution.

An alternative to taste shocks, which we discuss in the next section, is to use a continuous representation of the distribution rather than a discrete grid.

**Endogenous distribution.** The distribution $\mathbf{D}_t$ in equation (11) is assumed to be unaffected by the current shock $\mathbf{X}_t$ and the value function $\mathbf{v}_{t+1}$. In short, it is predetermined at date $t$. What if we want events at date $t$ to affect the distribution—for instance, if shocks at date $t$ can affect capital gains on wealth at date $t$, or can affect the probability of unemployment at date $t$?

Within the framework (10)-(12), the solution is to keep $\mathbf{D}_t$ predetermined at date $t$, and incorporate these shocks into the functions $v, \Lambda, y$ instead. For instance, in our two-asset HANK example, the date-0 return on the illiquid asset includes an endogenous capital gain. The distribution $\mathbf{D}_0$ gives the state prior to this capital gain, and then the ex-post return on illiquid assets, $r_0^a$ is included as part of $\mathbf{X}_0$ as an input to $v, \Lambda, y$.

Similarly, if the probability of unemployment is endogenous at date $t$, $\mathbf{D}_t$ should still be the state *prior* to the realization of the idiosyncratic unemployment shock, and then $v, \Lambda, y$ should take *expectations* over the realizations of this shock.

[51] To implement the fake news algorithm directly with lags, we would need to calculate $\mathbf{y}_0^s$ and $\Lambda_0^s$ for all $s$ from $-u$ to $T-1$, where $u$ is the maximum lag length, use these to build a fake news matrix $\mathcal{F}$ with columns $s = -u, \ldots, T-1$, then apply the recursion $\mathcal{J}_{t,s} = \mathcal{J}_{t-1,s-1} + \mathcal{F}_{t,s}$ in step 4 starting from this new leftmost column $-u$. In our experience, this is more difficult and error-prone than the $\tilde{\mathbf{X}}_t$ solution above.

[52] One particularly convenient approach is to use extreme value taste shocks as in Iskhakov, Jørgensen, Rust and Schjerning (2017), which are smooth and lead to logit choice probabilities. Bardóczy (2020) implements the fake news algorithm using this approach.

Although this procedure can virtually always be used to put a model into the framework of (10)-(12), it becomes unwieldy in complex cases. In appendix A.3, we describe how to apply the fake news algorithm to a model where the distribution evolves over multiple subperiods within each period. This provides a more formal, structured approach.

### A.2 Nonlinear $\mathcal{Y}$ or $D$

We now generalize our algorithm to the case of nonlinear functions for $\mathbf{D}_{t+1}$ and $\mathbf{Y}_t$ in (11)-(12). The key is the following generalization of proposition 1.

**Proposition 2.** *Assume that equations (11) and (12) are replaced, respectively, by*

$$\mathbf{D}_{t+1} = D\left(\mathbf{v}_{t+1}, \mathbf{X}_t, \mathbf{D}_t\right) \tag{40}$$

$$\mathbf{Y}_t = \mathcal{Y}\left(\mathbf{v}_{t+1}, \mathbf{X}_t, \mathbf{D}_t\right) \tag{41}$$

*for some functions $D\left(\mathbf{v}, \mathbf{X}, \mathbf{D}\right)$ and $\mathcal{Y}\left(\mathbf{v}, \mathbf{X}, \mathbf{D}\right)$. Then proposition 1 still holds, provided that definition 1 is changed to*

$$\mathcal{E}_t \equiv (D'_{\mathbf{D}})^t \mathcal{Y}'_{\mathbf{D}} \tag{42}$$

*where $D_{\mathbf{D}} \equiv \frac{\partial D}{\partial \mathbf{D}}(\mathbf{v}_{ss}, \mathbf{X}_{ss}, \mathbf{D}_{ss})$ and $\mathcal{Y}_{\mathbf{D}} \equiv \frac{\partial \mathcal{Y}}{\partial D}(\mathbf{v}_{ss}, \mathbf{X}_{ss}, \mathbf{D}_{ss})$ are the $n_D \times n_D$ Jacobian and $1 \times n_D$ gradient of $D$ and $\mathcal{Y}$ with respect to $\mathbf{D}$, respectively.*

*Proof.* In the proof of lemma 2, we replace (19) by $d\mathcal{Y}_t^s = \mathcal{Y}_{\mathbf{D}} d\mathbf{D}_t^s + \mathcal{Y}_{\mathbf{v}} d\mathbf{v}_{t+1}^s + \mathcal{Y}_{\mathbf{X}} d\mathbf{X}_t^s$. Subtracting $d\mathcal{Y}_t^s$ and $d\mathcal{Y}_{t-1}^{s-1}$ and using $d\mathbf{v}_{t+1}^s = d\mathbf{v}_t^{s-1}$ from (16) and $d\mathbf{X}_t^s = d\mathbf{X}_{t-1}^{s-1}$ by construction, we get $\mathcal{F}_{t,s} \cdot dx = \mathcal{Y}_{\mathbf{D}}(d\mathbf{D}_t^s - d\mathbf{D}_{t-1}^{s-1})$, which is identical to (20) except with $\mathbf{y}'_{ss}$ replaced by $\mathcal{Y}_{\mathbf{D}}$.

Similarly, replacing (21) with $d\mathbf{D}_t^s = D_{\mathbf{D}} \cdot d\mathbf{D}_{t-1}^s + D'_{\mathbf{v}} d\mathbf{v}_t^s + D'_{\mathbf{X}} d\mathbf{X}_{t-1}^s$, we follow the same steps to show that $d\mathbf{D}_t^s - d\mathbf{D}_{t-1}^{s-1} = (D_{\mathbf{D}})^{t-1} d\mathbf{D}_1^s$, which is identical to (22) except with $\Lambda'_{ss}$ replaced by $D_{\mathbf{D}}$. The modified lemma 2 follows, with $\mathbf{y}'_{ss}, \Lambda'_{ss}$ replaced by $\mathcal{Y}_{\mathbf{D}}, D_{\mathbf{D}}$. Replacing these in the definition of $\mathcal{E}_t$, the proof of proposition 1 goes through. □

Remarkably, the only change needed in proposition 2, relative to proposition 1, is to redefine $\mathcal{E}_t$ as $(D'_{\mathbf{D}})^t \mathcal{Y}'_{\mathbf{D}}$ rather than $(\Lambda_{ss})^t \mathbf{y}_{ss}$. This redefinition is natural: the Jacobian $D_{\mathbf{D}}$, which gives the first-order effect of yesterday's distribution on today's, is the generalized counterpart of the forward iteration matrix $\Lambda'_{ss}$, and the gradient $\mathcal{Y}_{\mathbf{D}}$, which gives the first-order effect of today's distribution on the aggregate output, is the generalized counterpart of $\mathbf{y}'_{ss}$.

Given this redefined $\mathcal{E}_t$, which can be calculated recursively via $\mathcal{E}_t = (D'_{\mathbf{D}})\mathcal{E}_{t-1}$ and $\mathcal{E}_0 = \mathcal{Y}'_{\mathbf{D}}$, the fake news algorithm is otherwise unchanged. We now discuss some applications.

**Entry and exit.** In general, if we modify our original framework to allow for entry and exit, we have an equation (40) of the more specific form

$$\mathbf{D}_{t+1} = \Lambda(\mathbf{v}_{t+1}, \mathbf{X}_t)\mathbf{D}_t + D^{entry}(\mathbf{v}_{t+1}, \mathbf{X}_t) \tag{43}$$

where $\Lambda$ is a Markov matrix with rows that may sum to less than one (because of exit, which may be endogenous) and $D^{entry}$ accounts for the possibly-endogenous entry of agents. If, additionally, new entrants show up in the aggregate output, then we also have an equation (41) of the form

$$\mathbf{Y}_t = y(\mathbf{v}_{t+1}, \mathbf{X}_t)'\mathbf{D}_t + Y^{entry}(\mathbf{v}_{t+1}, \mathbf{X}_t) \tag{44}$$

where $Y^{entry}$ accounts for the effect of the new entrants.

Note that from (43) and (44), we have $D_{\mathbf{D}} = \Lambda'_{ss}$ and $Y_{\mathbf{D}} = \mathbf{y}'_{ss}$. Hence the expectation vector (42) is the same as our original definition from section 3, and proposition 1 and the fake news algorithm apply in their original form.

**Alternative representations of the distribution.** In our original equations (11)-(12), we assumed that the distribution vector $\mathbf{D}_t$ consisted of probability masses at discrete grid points. Now, in (40)-(41), $\mathbf{D}_t$ can be an arbitrary vector describing the distribution. For instance, suppose that the state is one-dimensional and continuous. Then if $\mathbf{D}_t$ is a vector of parameters[53] encoding a density $f(\theta; \mathbf{D}_t)$ for $\theta \in (-\infty, \infty)$, we can write a function $D(\mathbf{v}_{t+1}, \mathbf{X}_t, \mathbf{D}_t)$ that specifies how these parameters evolve over time in our problem. We can also define the aggregate output $Y$ as the average of some idiosyncratic outcome $y(\theta; \mathbf{v}_{t+1}, \mathbf{X}_t)$ of interest:

$$Y(\mathbf{v}_{t+1}, \mathbf{X}_t, \mathbf{D}_t) \equiv \int_{-\infty}^{\infty} y(\theta; \mathbf{v}_{t+1}, \mathbf{X}_t) \cdot f(\theta; \mathbf{D}_t) d\theta \tag{45}$$

Assuming that we already have a way to calculate $D$ and $Y$, all we need to implement the fake news algorithm is $D_{\mathbf{D}}$ and $Y_{\mathbf{D}}$. If $\mathbf{D}$ is not too high-dimensional, then numerical differentiation is usually a simple strategy to calculate these, although automatic differentiation or (in special cases) analytical differentiation may also be useful.

**Moments of the distribution.** Suppose that we want the Jacobian for some moment that can not be represented as a transformation of power moments as in the previous section. For instance—to take a simple example—suppose that $\mathbf{D}$ is a vector of parameters describing the distribution of assets, and we want the $u$th quantile of this asset distribution. This is a nonlinear function $Y(\mathbf{D}_t)$, and to apply the fake news algorithm we only need to calculate the gradient $Y_{\mathbf{D}}$, which (as above) can be done using either numerical or automatic differentiation.

If $\mathbf{D}$ is instead a simple discretized distribution, then the $u$th quantile function is discontinuous, consisting of many steps, and its Jacobian is therefore essentially meaningless (wherever it can be calculated, it is identically zero). We could obtain a more interesting object, however, by converting this function to be piecewise linear, interpolating between the discrete mass points. With many grid points, numerical differentiation might be impractical in this case, but thanks to

[53] For an example of a parametric family of distributions often used with heterogeneous-agent models, see Algan et al. (2014). In some cases, another possibility is to represent the distribution with a more flexible set of basis functions, such as Chebyshev polynomials.

the simplicity of the linearly interpolated quantile function, one can write the gradient $Y_{\mathbf{D}}$ analytically instead.

**Discrete choice without taste shocks.** As discussed in appendix A.1, first-order methods are misleading for endogenous discrete choices on a state space that has been discretized to a grid, since locally these choices will not respond to shocks unless the grid points happen to be at the discontinuities (in which case there is instead a singularity). The suggestion of appendix A.1 was to assume iid taste shocks, so that the probabilities of each discrete choice vary continuously.

If $\mathbf{D}$ is a vector of parameters parametrizing a smooth density, however, then the integral (45) aggregating a discrete choice $y$ will generally vary smoothly in $\mathbf{X}_t$ even if $y$ itself is discontinuous. Similarly, the law of motion (40) should also vary smoothly. At this point, there is no particular computational problem posed by discrete choice, and we can apply proposition 2 as long as we can calculate $D_{\mathbf{D}}$ and $Y_{\mathbf{D}}$, just like above.[54]

### A.3 Multi-stage problems

In appendix A.1, we observe that in cases where the "distribution" at time $t$ seems endogenous to events at time $t$ (e.g. unemployment risk), our basic framework (10)-(12) can be applied if we interpret $\mathbf{D}_t$ as being the distribution prior to any time-$t$ events, and $v, \Lambda, y$ as taking expectations over these events. But as models become more complex, with more within-period structure, implementing this approach manually can become difficult.

We therefore further generalize our framework to account for multiple "stages" $j \in \{0, \ldots, J-1\}$ within a given period $t$. We now assume that the three equations (10)-(12) are replaced by

$$\mathbf{v}_{t,j} = v_j(\mathbf{v}_{t,j+1}, \mathbf{X}_{t,j}) \tag{46}$$

$$\mathbf{D}_{t,j+1} = D_j\left(\mathbf{v}_{t,j+1}, \mathbf{D}_{t,j}, \mathbf{X}_{t,j}\right) \tag{47}$$

$$\mathbf{Y}_{t,j} = Y_j\left(\mathbf{v}_{t,j+1}, \mathbf{D}_{t,j}, \mathbf{X}_{t,j}\right) \tag{48}$$

where we adopt the convention that $\cdot_{t,J} = \cdot_{t+1,0}$, and assume that the initial distribution $\mathbf{D}_{0,0}$ is given exogenously.

At each stage $j$, we allow for stage-specific inputs $\mathbf{X}_{t,j}$ and outputs $\mathbf{Y}_{t,j}$.[55] Given a path for $\{\mathbf{X}_{t,j}\}$, one solves (46)-(48) to obtain $\{\mathbf{Y}_{t,j}\}$ in the standard way, except that all iterations go through both

[54] One caveat, however, is that a smooth density is not always realistic in models with discrete choice. For instance, if agents always reset to the same ideal state, then there will be a mass point at that state; further, if uncertainty in the model is discrete, then there will be mass points corresponding to the each finite sequence of shocks that might be realized after that state. To avoid having the distribution consist entirely of mass points in this way, it is useful to introduce some stochastic variables that are drawn from a continuous distribution (e.g. in household models, iid lognormal income risk in each period, in addition to whatever other income risk is present). This is also true if we want to avoid mass points in a model with occasionally binding constraints (e.g. in household models, a borrowing constraint).

[55] If some stages have either no outputs or no inputs, we can simply disregard the relevant terms. If multiple stages include the same input, then we can use the algorithm to calculate Jacobians with respect to the input at each stage individually, and then sum the Jacobians.

$t$ and, within each $t$, through each stage $j$. For instance, iterating backward over (46), starting from some steady-state $\mathbf{v}_{T,0} = \mathbf{v}_{ss,0}$, would involve iterating through

$$\mathbf{v}_{T-1,J-1}, \mathbf{v}_{T-1,J-2}, \ldots, \mathbf{v}_{T-1,0}, \mathbf{v}_{T-2,J-1}, \mathbf{v}_{T-2,J-2}, \ldots,$$

and so on.

Write $D^j_{\mathbf{D}} \equiv \frac{\partial D_j}{\partial \mathbf{D}}$ and $Y^k_{\mathbf{D}} \equiv \frac{\partial Y_k}{\partial D}$, and use these to define $\mathcal{E}^k_{t,j}$ recursively, iterating backward over $(t, j)$ starting with the initial condition $\mathcal{E}^k_{0,k} = (Y^k_{\mathbf{D}})'$ and then writing

$$\mathcal{E}^k_{t,j} \equiv (D^j_{\mathbf{D}})' \mathcal{E}^k_{t,j+1} \tag{49}$$

for all $t > 0$ or $t = 0$ and $j < k$. For any $s$, the vector $\mathcal{E}^k_{t,j}$ gives the first-order impact of the distribution $\mathbf{D}_{s,j}$ at time $s$, stage $j$ on the output $\mathbf{Y}_{s+t,k}$ at the time $s + t$, stage $k$.

Next, assuming a shock $dx$ to $\mathbf{X}^k_s$, we define

$$\mathcal{F}^{j,k}_{t,s} \cdot dx \equiv \begin{cases} dY^{s,k}_{0,j} & t = 0 \\ \mathcal{E}^j_{t-1,0} d\mathbf{D}^{s,k}_{1,0} & t \geq 1 \end{cases} \tag{50}$$

and have the following further refinement of proposition 1.

**Proposition 3.** *Assume $\mathbf{D}_{0,0} = \mathbf{D}_{ss,0}$. The Jacobian $\mathcal{J}$ of $h$ satisfies the recursion $\mathcal{J}^{j,k}_{t,s} = \mathcal{J}^{j,k}_{t-1,s-1} + \mathcal{F}^{j,k}_{t,s}$ for $t, s \geq 1$, with $\mathcal{J}^{j,k}_{t,s} = \mathcal{F}^{j,k}_{t,s}$ for $t = 0$ or $s = 0$, and is therefore given by*

$$\mathcal{J}^{j,k}_{t,s} = \sum_{u=0}^{\min\{s,t\}} \mathcal{F}^{j,k}_{t-u,s-u} \tag{51}$$

*where $\mathcal{F}^{j,k}_{t,s}$ is defined in (50).*

Obtaining entries of the Jacobian is therefore no more complicated than in our original case, conditional on being able to evaluate (50) to obtain $\mathcal{F}^{j,k}_{t,s}$.

For each $k$, we can still obtain $dY^{s,k}_{0,j}$ and $d\mathbf{D}^{s,k}_{1,0}$ for all $s = 0, \ldots, T-1$ and $j$ by iterating backward from a shock at date $T-1$. This is slightly more involved than before, however. One must first obtain $d\mathbf{v}^{s,k}_{0,j}$ for each $s$ and $j$ through backward iteration, then for each $s$, combine this with (47) and (48), iterating forward through the $j$s, to obtain $dY^{s,k}_{0,j}$ for all $j$s and finally $d\mathbf{D}^{s,k}_{1,0}$. (This is in contrast to the original algorithm, where obtaining the $dY^s_0$ involved no forward iteration at all.)

Table B.1: Calibration of our Krusell-Smith economy

| Parameter | | Value |
|---|---|---|
| $r$ | Real interest rate | 0.01 |
| $\sigma$ | Risk aversion | 1 |
| $\alpha$ | Capital share | 0.11 |
| $\delta$ | Depreciation rate | 0.025 |
| $\rho$ | Skill mean reversion | 0.966 |
| $\sigma/\sqrt{1-\rho^2}$ | Cross-sectional std of log earnings | 0.5 |
| $n_e$ | Points in Markov chain for $e$ | 7 |
| $n_k$ | Points on asset grid | 500 |

# B Model descriptions

## B.1 Calibration of the Krusell-Smith economy

We follow a standard calibration. We assume that $P(e, e')$ discretizes a log AR(1) process,

$$\log e_t = \rho \log e_{t-1} + \sigma \epsilon_t$$

with normal innovations $\epsilon_t \sim \mathcal{N}(0,1)$. We use the Rouwenhorst method for discretization. Table B.1 summarizes the rest of our calibration. The number of grid points refers to our baseline calibration with 3500 idiosyncratic states. We also consider two variations on the number of grid points. In the high-dimensional ("HD") version, $n_e = 50$ and $n_k = 5000$, so there are a total of 250,000 idiosyncratic states. In the lower-dimension version that we consider for comparison to the Reiter method, $n_e = 3$ and $n_k = 100$.

## B.2 One-asset HANK model

This is an economy without capital, but with nominal rigidities.

### Households.

Households now get to choose labor supply in addition to consumption and savings to maximize a standard separable utility function. A household in state $e$ working $n_t$ hours earns labor income $w_t n_t e$, pays a lump-sum tax $\overline{\tau}(e)\,\tau_t$, and receives dividend $\overline{d}(e)\,d_t$ per period from his ownership of a nontradable firm share. The only asset households can trade is one-period nominal bond that pays $r_t$ in real terms. All in all, the Bellman equation can be written as

$$V_t(e, a_-) = \max_{c,n,a} \left\{ \frac{c^{1-\sigma}}{1-\sigma} - \varphi \frac{n^{1+\nu}}{1+\nu} + \beta \sum_{e'} V_{t+1}(e', a) \mathcal{P}(e, e') \right\}$$

$$c + a = (1 + r_t) a_- + w_t e n - \tau_t \bar{\tau}(e) + d_t \bar{d}(e)$$

$$a \geq 0$$

The solution is a collection of policy functions $c_t(e, a_-)$, $n_t(e, a_-)$ and $a_t(e, a_-)$ that depend on the paths $\{r_s, w_s, \tau_s, d_s\}_{s \geq t}$ that households take as given. Analogously to section 2, we can summarize the household block by aggregate consumption, labor supply and asset demand functions

$$\mathcal{C}_t\left(\{r_s, w_s, \tau_s, d_s\}_{s \geq 0}\right) \equiv \int c_t(a, e)\ dD_t(e, a_-) \tag{52}$$

$$\mathcal{N}_t\left(\{r_s, w_s, \tau_s, d_s\}_{s \geq 0}\right) \equiv \int e n_t(a, e)\ dD_t(e, a_-) \tag{53}$$

$$\mathcal{A}_t\left(\{r_s, w_s, \tau_s, d_s\}_{s \geq 0}\right) \equiv \int a\, dD_t(e, a_-) \tag{54}$$

**Firms.**

There is a continuum of identical firms that produce differentiated goods using labor only. To preserve symmetry, we assume that firm employs a representative workforce. They engage in monopolistic competition and set the price of their product subject to the usual iso-elastic demand curve and quadratic adjustment costs. The Bellman equation of firm $j$ is

$$J_t(p_{jt-1}) = \max_{y_{jt}, p_{jt}, n_{jt}} \left\{ \frac{p_{jt}}{p_t} y_{jt} - w_t n_{jt} - \frac{\mu_t}{\mu_t - 1} \frac{1}{2\kappa} \Big[ \log(1 + \pi_{jt}) \Big]^2 Y_t + \frac{J_{t+1}(p_{jt})}{1 + r_{t+1}} \right\}$$

$$\text{s.t. } y_{jt} = Z n_{jt}$$

$$y_{jt} = \left( \frac{p_{jt}}{p_t} \right)^{-\frac{\mu_t}{\mu_t - 1}} Y_t$$

This is a standard problem that yields the following equilibrium conditions:

- Phillips curve:

$$\log(1 + \pi_t) = \kappa \left( \frac{w_t}{Z} - \frac{1}{\mu_t} \right) + \frac{1}{1 + r_{t+1}} \frac{Y_{t+1}}{Y_t} \log(1 + \pi_{t+1}). \tag{55}$$

- Production:

$$Y_t = Z N_t \tag{56}$$

- Price adjustment cost:

$$\psi_t = \frac{\mu_t}{\mu_t - 1} \frac{1}{2\kappa} \Big[ \log(1 + \pi_t) \Big]^2 Y_t. \tag{57}$$

- Dividends:

$$d_t = Y_t - w_t N_t - \psi_t \tag{58}$$

**Policy and market clearing**

The government runs a balanced budget, spends $G_t$ each period, and maintains a constant level of bonds $B$ by adjusting taxes. Monetary policy sets the nominal rate according to a Taylor rule that

is subject to a shock $r_t^*$

$$\tau_t = r_t B + G_t, \tag{59}$$

$$i_t = r_t^* + \phi \pi_t + \phi_y \left(Y_t - Y_{ss}\right) \tag{60}$$

$$1 + r_t = \frac{1 + i_{t-1}}{1 + \pi_t} \tag{61}$$

Market clearing requires that $B = \mathcal{A}_t, N_t = \mathcal{N}_t$ and $Y_t = \mathcal{C}_t + G_t + \psi_t$.

**Shocks.**

There are three shocks to the model: monetary policy shocks $r_t^*$, government spending shocks $G_t$, and markup shocks $\mu_t$. Hence, these enter as "exogenous" variables in the DAG in figure 4. Given the time path of the three unknowns variables $(Y_t, w_t, r_t)$, one can solve for the value of the three targets: the residual in the Phillips curve (55), labor market clearing $N_t = \mathcal{N}_t$ and asset market clearing $B = \mathcal{A}_t$.

**Calibration.**

The calibration mostly follows McKay, Nakamura and Steinsson (2016) and is summarized in table B.2. The number of grid points refers to our baseline calibration with 3500 idiosyncratic states. In the lower-dimension version that we consider for comparison to the Reiter method, $n_e = 3$ and $n_a = 100$.

### B.3 Two-asset HANK model

This embeds a two-asset household block, described in more detail in appendix C.3, into a New Keynesian model.

**Households.**

Income $z_{it}$ is determined as

$$z_{it} = (1 - \tau_t) w_t N_t e_{it} \tag{62}$$

where $e_{it}$ is individual productivity following a Markov process as in the other models. All in all, the household block takes as inputs the sequences of interest rates for illiquid and liquid assets $\{r_s^a, r_s^b\}$, wage per efficiency units $\{w_s\}$, labor tax rate $\{\tau_s\}$ and labor demand $\{N_s\}$ as inputs. The relevant outputs are illiquid asset demand, liquid asset demand, productivity-weighted marginal

Table B.2: Calibration of our one-asset HANK economy

| Parameter | | Value | Target |
|---|---|---|---|
| *Households* | | | |
| $\beta$ | Discount factor | 0.982 | $r = 0.005$ |
| $\varphi$ | Disutility of labor | 0.786 | $\mathcal{N} = 1$ |
| $\sigma$ | Inverse IES | 2 | |
| $\nu$ | Inverse Frisch | 2 | |
| $\underline{b}$ | Borrowing constraint | 0 | |
| $\rho_e$ | Autocorrelation of earnings | 0.966 | |
| $\sigma_e$ | Cross-sectional std of log earnings | 0.5 | |
| *Firms* | | | |
| $\mu$ | Steady-state markup | 1.2 | |
| $\kappa$ | Slope of Phillips curve | 0.1 | |
| *Policy* | | | |
| $B$ | Bond supply | 5.6 | |
| $G$ | Government spending | 0 | |
| $\phi$ | Taylor rule coefficient on inflation | 1.5 | |
| $\phi_y$ | Taylor rule coefficient on output | 0 | |
| *Discretization* | | | |
| $n_e$ | Points in Markov chain for $e$ | 7 | |
| $n_a$ | Points on asset grid | 500 | |

utility, consumption, and portfolio adjustment costs:

$$\mathcal{A}_t\left(\{r_s^a, r_s^b, w_s, \tau_s, N_s\}\right) = \int a\, dD_t(e,b,a), \tag{63}$$

$$\mathcal{B}_t\left(\{r_s^a, r_s^b, w_s, \tau_s, N_s\}\right) = \int b_t(e,b,a)\, dD_t(e,b,a), \tag{64}$$

$$\mathcal{U}_t\left(\{r_s^a, r_s^b, w_s, \tau_s, N_s\}\right) = \int e \cdot c_t(e,b,a)^{-\sigma}\, dD_t(e,b,a), \tag{65}$$

$$\mathcal{C}_t\left(\{r_s^a, r_s^b, w_s, \tau_s, N_s\}\right) = \int c_t(e,b,a)\, dD_t(e,b,a), \tag{66}$$

$$\mathcal{P}_t\left(\{r_s^a, r_s^b, w_s, \tau_s, N_s\}\right) = \int \Phi(a_t(e,b,a),a)\, dD_t(e,b,a). \tag{67}$$

The last two are required only for checking the omitted goods market clearing condition.

**Labor unions.**

We assume that every household provides a continuum of differentiated labor services, each of which is represented by a labor union. Unions set hours and wages to maximize the average utility of members, taking as given their consumption-savings decisions as well as the decisions of other (identical) unions. Changing the nominal wage incurs quadratic adjustment costs. The

Bellman equation of union $k$ is

$$U_t(w_{kt-1}) = \max_{n_{kt}, w_{kt}} \int u(c_{it}) - v(n_{kt})\, dD_t - \frac{\mu_w}{1-\mu_w}\frac{1}{2\kappa_w}\left[\log(1+\pi^w_{kt})\right]^2 N_t + \beta U_{t+1}(w_{kt})$$
$$\text{s.t. } n_{kt} = \left(\frac{w_{kt}}{w_t}\right)^{-\frac{\mu_w}{\mu_w - 1}} N_t$$

where $\pi^w_{kt}$ is wage inflation

$$\pi^w_{kt} = (1+\pi_t)\frac{w_{kt}}{w_{kt-1}} - 1. \tag{68}$$

This setup is almost identical to that in Auclert, Rognlie and Straub (2018) and, as shown there, leads to a wage Phillips curve of the form

$$\log(1+\pi^w_t) = \kappa_w\left[\varphi N_t^{1+\nu} - \mu_w(1-\tau_t)w_t N_t \mathcal{U}_t\right] + \beta\log(1+\pi^w_{t+1}). \tag{69}$$

For convenience later on, let's introduce the following shorthand for wage adjustment costs

$$\psi^w_t = \frac{\mu_w}{1-\mu_w}\frac{1}{2\kappa_w}\left[\log(1+\pi^w_t)\right]^2 N_t. \tag{70}$$

**Firms.**

Let there be a continuum of identical firms that produce differentiated goods using labor and capital. They own capital, and hire a representative workforce from the labor union. They engage in monopolistic competition and set the price of their product subject to the usual iso-elastic demand curve and quadratic adjustment costs. The Bellman equation of firm $j$ is

$$J_t(k_{jt-1}, p_{jt-1}) = \max_{y_{jt}, p_{jt}, k_{jt}, i_{jt}, n_{jt}} \left\{ \frac{p_{jt}}{p_t} y_{jt} - w_t n_{jt} - i_{jt} - \frac{\mu_p}{\mu_p - 1}\frac{1}{2\kappa_p}\left[\log(1+\pi_{jt})\right]^2 Y_t \right.$$
$$\left. - \frac{1}{2\delta\epsilon_I}\left(\frac{k_{jt}-k_{jt-1}}{k_{jt-1}}\right)^2 k_{jt-1} + \frac{J_{t+1}(k_{jt}, p_{jt})}{1+r_{t+1}} \right\}$$
$$\text{s.t. } y_{jt} = Z_t k^\alpha_{jt-1} n^{1-\alpha}_{jt}$$
$$y_{jt} = \left(\frac{p_{jt}}{p_t}\right)^{-\frac{\mu_p}{\mu_p - 1}} Y_t$$
$$k_{jt} = (1-\delta)\, k_{jt-1} + i_{jt}$$

This is a standard problem that yields the following equilibrium conditions

- Phillips curve:

$$\log(1+\pi_t) = \kappa_p\left(mc_t - \frac{1}{\mu_p}\right) + \frac{1}{1+r_{t+1}}\frac{Y_{t+1}}{Y_t}\log(1+\pi_{t+1}). \tag{71}$$

- Labor demand:

$$w_t = (1-\alpha)\frac{Y_t}{N_t}mc_t. \tag{72}$$

- Valuation:

$$(1+r_{t+1})Q_t = \alpha\frac{Y_{t+1}}{K_t}mc_{t+1} - \left[\frac{K_{t+1}}{K_t} - (1-\delta) + \frac{1}{2\delta\epsilon_I}\left(\frac{K_{t+1}-K_t}{K_t}\right)^2\right] + \frac{K_{t+1}}{K_t}Q_{t+1}. \tag{73}$$

- Investment:

$$Q_t = 1 + \frac{1}{\delta\epsilon_I}\frac{K_t - K_{t-1}}{K_{t-1}}. \tag{74}$$

- Capital law of motion:

$$I_t = K_t + (1-\delta)K_{t-1} \tag{75}$$

- Production:

$$Y_t = Z_t K_{t-1}^{\alpha} N_t^{1-\alpha} \tag{76}$$

- Price adjustment cost:

$$\psi_t^p = \frac{\mu_p}{\mu_p - 1}\frac{1}{2\kappa_p}\Big[\log(1+\pi_t)\Big]^2 Y_t \tag{77}$$

- Dividends:

$$d_t = Y_t - w_t N_t - I_t - \psi_t \tag{78}$$

**Policy.**

Fiscal policy follows a balanced-budget policy

$$\tau_t w_t N_t = r_t B^g + G_t, \tag{79}$$

Monetary policy follows an interest rate rule

$$i_t = r_t^* + \phi\pi_t + \phi_y\left(Y_t - Y_{ss}\right) \tag{80}$$

$$1 + r_t = \frac{1+i_{t-1}}{1+\pi_t} \tag{81}$$

$$r_t^* \text{ is exogenous.} \tag{82}$$

**Assets.**

The economy has two assets, nominal government bonds and firm equity, both illiquid by default. The ex-post real return on government bonds is simply $r_t$. Let $p_t$ denote the ex-dividend price of equity. The real return on equity is

$$\frac{d_{t+1} + p_{t+1}}{p_t}.$$

In the absence of aggregate uncertainty, these assets have to earn the same return and so we get the following no arbitrage condition

$$p_t = \frac{d_{t+1} + p_{t+1}}{1 + r_{t+1}}. \tag{83}$$

If bonds and equity are both illiquid, where do liquid assets come from? We assume that there is a representative financial intermediary endowed with the technology to turn illiquid nominal assets into liquid nominal assets. This process has a proportional cost $\omega$. This simple setup implies that

$$r_t^b = r_t - \omega. \tag{84}$$

**Market clearing.**

The only market clearing conditions not implied implicitly by notation are that of goods and assets

$$Y_t = \mathcal{C}_t + G_t + I_t + \mathcal{P}_t + \omega \mathcal{B}_t + \psi_t^p + \psi_t^w, \tag{85}$$

$$\mathcal{A}_t + \mathcal{B}_t = p_t + B^g. \tag{86}$$

**Accounting for surprise inflation and capital gains.**

When an unanticipated shock hits, the no arbitrage condition (83) will fail for one period due to surprise inflation and capital gains. Accounting for these effects requires further assumptions on the illiquid portfolio of households. For simplicity, we assume that all households hold the same ratio of bonds and equity. This means that the ex-post illiquid return can be written as

$$1 + r_t^a = \frac{p_{t-1}}{\mathcal{A}_{t-1}} \cdot \frac{d_t + p_t}{p_{t-1}} + \frac{B^g - \mathcal{B}_{t-1}}{\mathcal{A}_{t-1}} \cdot (1 + r_t). \tag{87}$$

This equation holds both in periods with and without unexpected shocks. We could stop here, but then we would have to add $r_t^a$ to the unknowns of the DAG. The reason is that it is an input of the household block that also depends on outputs of the household block. We can avoid this by replacing (87) with

$$1 + r_t^a = \frac{p}{\mathcal{A}} \cdot \frac{d_t + p_t}{p_{t-1}} + \frac{B^g - \mathcal{B}}{\mathcal{A}} \cdot (1 + r_t). \tag{88}$$

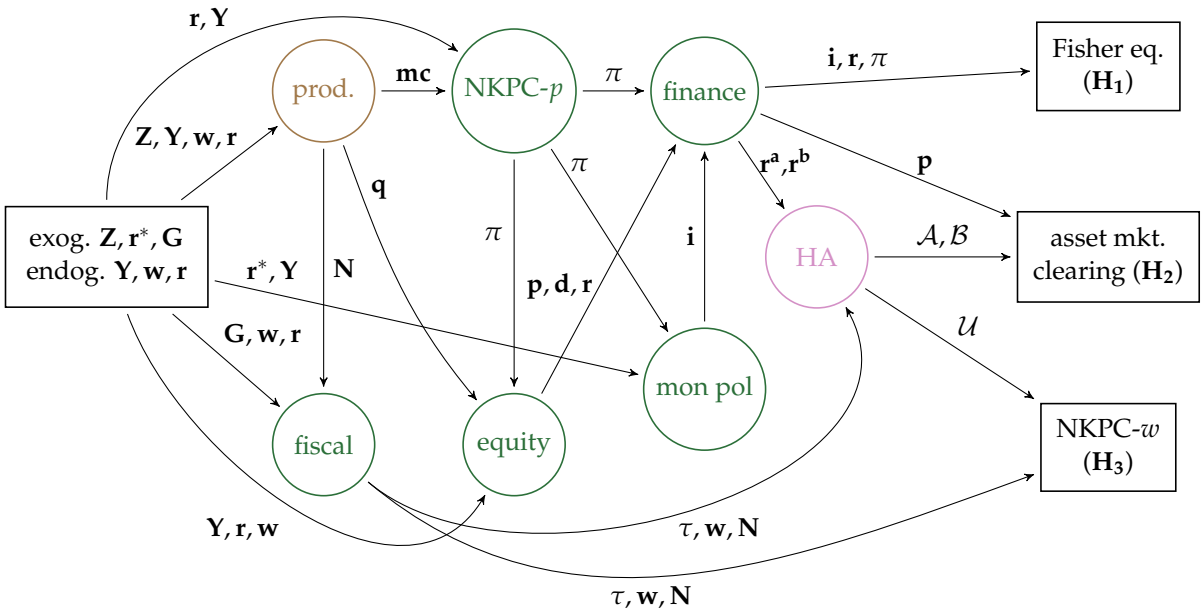
The difference is the use of steady-state portfolio shares. These are targeted in the calibration, and thus effectively exogenous. Importantly, they are the correct shares in period 0, when the unexpected shock hits. Although they are incorrect in periods $t > 0$, this does not matter, because by then the no arbitrage condition (83) is restored.

**Calibration.** Table B.3 summarizes the calibration of our two-asset model.

Table B.3: Calibration of our two-asset HANK economy

| Parameter | | Value | Target |
|---|---|---|---|
| *Households* | | | |
| $\beta$ | Discount factor | 0.976 | $r = 0.0125$ |
| $\sigma$ | Inverse IES | 2 | |
| $\chi_0$ | Portfolio adj. cost pivot | 0.25 | |
| $\chi_1$ | Portfolio adj. cost scale | 6.416 | $\mathcal{B} = 1.04Y$ |
| $\chi_2$ | Portfolio adj. cost curvature | 2 | |
| $\underline{b}$ | Borrowing constraint | 0 | |
| $\rho_e$ | Autocorrelation of earnings | 0.966 | |
| $\sigma_e$ | Cross-sectional std of log earnings | 0.92 | |
| *Labor unions* | | | |
| $\varphi$ | Disutility of labor | 2.073 | $N = 1$ |
| $\nu$ | Inverse Frisch elasticity | 1 | |
| $\mu_w$ | Steady state wage markup | 1.1 | |
| $\kappa_w$ | Slope of wage Phillips curve | 0.1 | |
| *Firms* | | | |
| $Z$ | TFP | 0.468 | $Y = 1$ |
| $\alpha$ | Capital share | 0.33 | $K = 10Y$ |
| $\mu_p$ | Steady-state markup | 1.015 | $\mathcal{A} + \mathcal{B} = 14Y$ |
| $\delta$ | Depreciation | 0.02 | |
| $\kappa_p$ | Slope of price Phillips curve | 0.1 | |
| *Financial intermediary* | | | |
| $\omega$ | Liquidity premium | 0.005 | |
| *Policy* | | | |
| $\tau$ | Labor tax | 0.356 | budget balance |
| $G$ | Government spending | 0.2 | |
| $B^g$ | Bond supply | 2.8 | |
| $\phi$ | Taylor rule coefficient | 1.5 | |
| $\phi_y$ | Taylor rule coefficient on output | 0 | |
| *Discretization* | | | |
| $n_e$ | Points in Markov chain for $e$ | 3 | |
| $n_b$ | Points on liquid asset grid | 50 | |
| $n_a$ | Points on illiquid asset grid | 70 | |

Figure B.1: DAG representation of two-asset HANK economy




**DAG and shocks in the two-asset HANK model.** Figure B.1 illustrates our preferred DAG for the two-asset HANK model. For simplicity, the DAG is drawn with only three shocks: to TFP $Z_t$, monetary policy $r_t^*$ and government spending $G_t$. In our estimation, we add to these four additional shocks: we let the price markup $\mu_p$ vary over time (a price markup shock), the wage markup $\mu_w$ vary over time (a wage markup shock), the discount rate of households $\beta$ vary over time (a preference shock), and we add a spread $r_{It}$ to the interest rate $r_t$ that enters the firm valuation equation (73), and let that spread vary over time (an investment shock).

# C Additional computational details

## C.1 Numerical and automatic differentiation details

A key implementation question is how to obtain the two objects in (26), $dY_0^s$ and $d\mathbf{D}_1^s$. As discussed in the main text, given some $dx$, one starts a backward iteration from $T-1$, obtaining $\mathbf{y}_0^s = \mathbf{y}_{T-1-s}^{T-1}$ and $\Lambda_0^s = \Lambda_{T-1-s}^{T-1}$ for all $s = 0, \ldots, T-1$, and then $dY_0^s = (d\mathbf{y}_0^s)'\mathbf{D}_{ss}$ and $d\mathbf{D}_1^s = (d\Lambda_0^s)'\mathbf{D}_{ss}$. There are two important practical complications:

1. We only get the correct derivative when $dx$ is infinitesimal.

2. In typical applications, we have not solved for the steady state solving (10) exactly (i.e. such that $\mathbf{v}_{ss} = v(\mathbf{v}_{ss}, X_{ss})$ exactly), but instead for a steady state such that (10) holds up to some numerical tolerance (i.e. such that $\|\mathbf{v}_{ss} - v(\mathbf{v}_{ss}, X_{ss})\| < 10^{-9}$). As a result, iterating backward will generally give $d\mathbf{y}_0^s \neq 0$ and $d\Lambda_0^s \neq 0$, even if $dx = 0$.

The first issue is about how to do differentiation, and is common to all perturbation methods. The second issue is more specific to our approach. We now describe three ways to perform differentiation—addressing the first issue—and, within each, discuss how to deal with the second issue.

**One-sided numerical differentiation.** Here, we simply choose some small but non-zero $dx$ and then iterate backward as described above. As is standard, $dx$ should be chosen to trade off error from second-order effects (which grow with $dx$) and from numerical issues like rounding error (which shrink with $dx$). In our application, one potential source of the latter error is, as discussed above, that the steady state is not exact. This will often be worse than the typical rounding error from floating-point numbers, since we usually pick a numerical tolerance for value function convergence (e.g. $10^{-9}$) that is larger than machine precision ($\sim 10^{-16}$).

There are two ways to address this issue:

(a) Do an additional full backward iteration from $T-1$ to 0 (a "ghost run") starting from $dx = 0$, and denote the results as $\tilde{\mathbf{y}}_0^s$ and $\tilde{\Lambda}_0^s$. Set $d\mathbf{y}_0^s = \mathbf{y}_0^s - \tilde{\mathbf{y}}_0^s$ and $d\Lambda_0^s = \Lambda_0^s - \tilde{\Lambda}_0^s$.

(b) At each step, subtract off $v(\mathbf{v}_{ss}, X_{ss})$ and recenter around the steady state. Starting with the shock $dx$, calculate $d\mathbf{v}_0^0 = v(\mathbf{v}_{ss}, X_{ss} + dx) - v(\mathbf{v}_{ss}, X_{ss})$. Then, for each $s$ calculate

$$d\mathbf{v}_0^s = v(\mathbf{v}_{ss} + d\mathbf{v}_0^{s-1}, X_{ss}) - v(\mathbf{v}_{ss}, X_{ss}) \tag{89}$$

Do the same for the functions $\Lambda$ and $y$ as well (e.g. at each step $s \geq 1$ calculate $d\mathbf{y}_0^s = y(\mathbf{v}_{ss} + d\mathbf{v}_0^{s-1}, X_{ss}) - y(\mathbf{v}_{ss}, X_{ss})$).

We can think of approach (a) as follows: if $\mathbf{y}_0^s$ and $\Lambda_0^s$ are functions of the shock $dx$, we are using one-sided numerical differentiation around $dx = 0$ to calculate $d\mathbf{y}_0^s/dx$ and $d\Lambda_0^s/dx$ for each $s$. Approach (b) is related, but instead effectively uses one-sided numerical differentiation at *each* step of the backward iteration.[56]

Approach (b) is usually more efficient than (a), since it does not require a full backward iteration from $T-1$ to 0 with $dx = 0$, and instead only requires $v(\mathbf{v}_{ss}, X_{ss})$, $y(\mathbf{v}_{ss}, X_{ss})$, and $\Lambda(\mathbf{v}_{ss}, X_{ss})$, which take a single step to compute. Approach (b) is also more accurate, since it corrects for error in the steady state at each step and does not allow these errors to compound. We therefore use (b) as our default for one-sided calculations in this paper (with $dx = 10^{-4}$).

One advantage of (a) is that it may be easier to implement with minimal changes to existing code, since it involves two complete backward iterations from $T-1$ to 0, and does not require changing the steps themselves as in (89).

**Two-sided numerical differentiation.** Here we have the following two analogues of approaches (a) and (b) above.

[56] To make this interpretation clearer, we could divide the right side of (89) by $dx$ to get $d\mathbf{v}_0^s/dx$, and then use $dx \cdot (d\mathbf{v}_0^{s-1}/dx)$ as an input. Practically, however, this involves unnecessary offsetting divisions and multiplications by $dx$.

(a) Iterate backward from $T-1$ to 0 for shocks at $T-1$ of $dx$ and $-dx$, denoting the results by $(\mathbf{y}_0^{s+}, \Lambda_0^{s+})$ and $(\mathbf{y}_0^{s-}, \Lambda_0^{s-})$, respectively, and set $d\mathbf{y}_0^s = (\mathbf{y}_0^{s+} - \mathbf{y}_0^{s-})/2$ and $\Lambda_0^s = (\Lambda_0^{s+} - \Lambda_0^{s-})/2$.

(b) Iterate backward from $T-1$ to 0, recentering around the steady state in each step. Specifically, starting with the shock $dx$, calculate $d\mathbf{v}_0^0 = (v(\mathbf{v}_{ss}, X_{ss} + dx) - v(\mathbf{v}_{ss}, X_{ss} - dx))/2$. Then, for $s$ calculate

$$d\mathbf{v}_0^s = \frac{v(\mathbf{v}_{ss} + d\mathbf{v}_0^{s-1}, X_{ss}) - v(\mathbf{v}_{ss} - d\mathbf{v}_0^{s-1}, X_{ss})}{2} \tag{90}$$

Do the same for the functions $\Lambda$ and $y$ as well (e.g. at each step $s \geq 1$ calculate $d\mathbf{y}_0^s = (y(\mathbf{v}_{ss} + d\mathbf{v}_0^{s-1}, X_{ss}) - y(\mathbf{v}_{ss} - d\mathbf{v}_0^{s-1}, X_{ss}))/2$).

Analogous to above, approach (a) effectively does two-sided numerical differentiation on the entire backward iteration process, while approach (b) does two-sided numerical differentiation at each step of the process. Note that it is no longer necessary to calculate any responses with $dx = 0$.

Both approaches (a) and (b) have similar efficiency and accuracy. (b) is in principle more accurate for the same reason as above, since it immediately recenters rather than allowing errors in the steady state to build up, but in practice this accuracy advantage seems minor. For consistency with the above, we use approach (b) as our default for two-sided calculations (also with $dx = 10^{-4}$).

**Automatic differentiation.** Automatic differentiation allows us to calculate the two objects in (26), $dY_0^s$ and $d\mathbf{D}_1^s$, for infinitesimal $dx$, getting exact derivatives $dY_0^s/dx$ and $d\mathbf{D}_1^s/dx$. One option is to take whatever code iterates backward from $T-1$ to 0, starting with some shock $dx$, and simply feed it into an automatic differentiation package, telling it to differentiate with respect to $dx$.

Though this approach works, it also suffers from error in the steady state: since $\mathbf{v}_{ss}$ is not exactly the same as $v(\mathbf{v}_{ss}, X_{ss})$, the package will be differentiating around a slightly different "steady state" at each step, which may be inefficient. It is therefore beneficial to apply automatic differentiation to a backward iteration routine that recenters around the steady state at each step, as in (89), so that differentiation will be done around the same steady state at each step.

In our implementation, we go slightly further, pre-calculating all derivatives $\partial v/\partial \mathbf{v}$, $\partial v/\partial X$, $\partial y/\partial \mathbf{v}$, and so on around the steady-state $(\mathbf{v}_{ss}, X_{ss})$, and then using these derivatives to iterate backward starting with infinitesimal $dx$. Specifically, we first use the Python automatic differentiation package "jax" to calculate all derivatives.[57] Then, we do backward iterations, starting

[57] This required extensive modifications to our code to make it compatible with jax. For instance, jax requires a more functional style—it does not allow operations that overwrite existing arrays—and it cannot immediately differentiate our routines written to be compiled by Numba (a Python just-in-time compiler). Further, one major source of inefficiency is that Jacobians like $\partial v/\partial \mathbf{v}$ tend to be highly sparse, but jax (like most automatic differentiation packages) cannot internally use sparse array operations. Although we convert the derivatives provided by jax into SciPy's sparse matrix representation before doing additional computations with them, jax's internal computations are still slow in high-dimensional cases because of this limitation. Implementation difficulties of this kind are why we chose numerical

with $d\mathbf{v}_0^0/dx = \partial v/\partial X$, and then iterating backward $d\mathbf{v}_0^s/dx = (\partial v/\partial \mathbf{v}) \cdot (d\mathbf{v}_0^0/dx)$. We similarly calculate each $d\mathbf{y}_0^s/dx$ and $d\Lambda_0^s/dx$.

**Applying to direct method.** In appendix D.1, we apply one-sided, two-sided, and automatic differentiation to the direct method of computing columns $s$ of the Jacobian. For one-sided differentiation, we apply the direct method exactly as described at the beginning of section 3.2, calculating the impulse response to an small shock $dx = 10^{-4}$ at date $s$. However, in the spirit of (a) above, we subtract off the results from a "ghost run" with a shock $dx = 0$ to eliminate inaccuracy from an imperfect steady state. For two-sided differentiation, we calculate the impulse responses to small shocks $dx$ and $-dx$ at date $s$ and then take half the difference between the two. Finally, for automatic differentiation, we use automatic differentiation to precalculate all derivatives as above, and use them to evaluate the linearized equations (10)-(12) and obtain a linear impulse response to a shock at each date $s$.

## C.2 Reiter method implementation

We now briefly describe our implementation of the "Reiter method". The idea is to arrange the equations governing equilibrium into a system of nonlinear equations with at most a single lead and a single lag, at which point we can use standard linear rational expectations methods to obtain the first-order solution.

**Krusell-Smith model.** Here we construct a stacked vector $\mathbf{X}_t$ of length $2n_g + 1$, where $n_g$ is the number of points in our grid. This includes:

- the entire vector $\mathbf{v}_t$ representing the value function at time $t$ (in our implementation, this is the length $n_g$ vector giving the derivative of the value function at each point)
- the distribution $\mathbf{D}_{t+1}$ excluding the last entry, which equals one minus the other elements and is therefore redundant (length $n_g - 1$ )
- capital $K_t$ (scalar)
- productivity $Z_t$ (scalar)

Note that since in our model, the distribution $\mathbf{D}_{t+1}$ is determined by information available at time $t$, we include it in $\mathbf{X}_t$ in line with the usual timing convention for these models.

We now build a function $\mathbf{F}(\mathbf{X}_{t-1}, \mathbf{X}_t, \mathbf{X}_{t+1}, \epsilon_t)$ with a $2n_g + 1$-dimensional output, which includes all equilibrium conditions:

- $n_g$ entries for the equation (10) that determines $\mathbf{v}_t$ given $\mathbf{v}_{t+1}$ and the inputs $K_{t-1}$ and $Z_t$ (which together determine $r_t$ and $w_t$, entering into the household's problem)

differentiation to be our primary approach. (See Ahn et al. (2018b) for an example of a paper employing a custom-built automatic differentiation toolkit that makes use of sparsity internally.)

- $n_g - 1$ entries for the equation (11) that determines $\mathbf{D}_{t+1}$ given $\mathbf{v}_{t+1}$, $\mathbf{D}_t$, and the inputs $K_{t-1}$ and $Z_t$ (note that again, we drop the last entry, which is redundant since the distribution sums to one)
- 1 entry for the equation (12) that expresses aggregate $K_t$ as the total of individual holdings given by $\mathbf{D}_{t+1}$
- 1 entry for the assumed AR(1) law of motion $\log(Z_t/Z_{ss}) = \rho \log(Z_{t-1}/Z_{ss}) + \epsilon_t$ for productivity.

Recursive stochastic equilibrium corresponds to the condition $\mathbb{E}_t F(\mathbf{X}_{t-1}, \mathbf{X}_t, \mathbf{X}_{t+1}, \epsilon_t) = 0$. We use the automatic differentiation package jax to linearize this as

$$A\mathbb{E}_t d\mathbf{X}_{t+1} + Bd\mathbf{X}_t + Cd\mathbf{X}_{t-1} + E\epsilon_t = 0 \tag{91}$$

where $A$, $B$, and $C$ are $(2n_g + 1) \times (2n_g + 1)$ matrices and $E$ is a $(2n_g + 1) \times 1$ vector. (91) is a standard form for a linear rational expectations model, and can be solved using a variety of standard techniques. We use Alisdair McKay's Python toolkit, which implements a version of Sims's gensys algorithm to solve (91).[58] This gives us a solution, expressed as the recursive law of motion

$$d\mathbf{X}_t = Pd\mathbf{X}_{t-1} + Q\epsilon_t \tag{92}$$

where $P$ is a $(2n_g + 1) \times (2n_g + 1)$ matrix and $Q$ is a $(2n_g + 1) \times 1$ vector. Note that the first $n_g$ columns of $P$ are all zeros, since $\mathbf{v}_{t-1}$ is not a state and has no direct impact on $\mathbf{X}_t$. We can then unstack (92) to obtain the linear law of motion relating the individual components of $\mathbf{X}_t$ ($\mathbf{v}_t$, $\mathbf{D}_{t+1}$, $K_t$, and $Z_t$) to $\mathbf{D}_t$, $K_{t-1}$, $Z_{t-1}$, and $\epsilon_t$.

To obtain the impulse response to a unit shock to $\epsilon_0$ (i.e. a unit productivity shock) for comparison to our sequence-space solution, we plug $d\mathbf{X}_{t-1} = 0$ and $\epsilon_0 = 1$ into (92) and iterate forward to get $d\mathbf{X}_0, d\mathbf{X}_1, \ldots$

**One-asset HANK model.** Since our implementation here is mostly the same as above, we will only describe the differences.

The stacked vector $\mathbf{X}_t$ now has length $2n_g + 4$, including $n_g$ entries $\mathbf{v}_t$ giving the derivative of the value function at each grid point, the $n_g - 1$ first entries of $\mathbf{D}_{t+1}$, and then $w_t$, $Y_t$, $\pi_t$, $r_t$, and $Z_t$.

The function $F(\mathbf{X}_{t-1}, \mathbf{X}_t, \mathbf{X}_{t+1}, \epsilon_t)$ now also has output of length $2n_g + 4$, including $n_g$ entries for (10), $n_g - 1$ entries for (11), 2 entries for (12) corresponding to the aggregation of assets and labor (and clearing in the respective markets), 1 entry for the Phillips curve, 1 entry for the AR(1) law of motion $\log(Z_t/Z_{ss}) = \rho \log(Z_{t-1}/Z_{ss}) + \epsilon_t$ for productivity, and 1 entry for the combined Taylor rule and Fisher equation giving the ex-post real rate, $r_t = (1 + r^*_{t-1} + \phi\pi_{t-1})/(1 + \pi_t) - 1$.

Note that in addition to the equations for $\mathbf{v}_t$, $\mathbf{D}_{t+1}$, and $Z_t$, we have one equation in $F$ for each of the targets in figure 3 (asset market clearing, labor market clearing, and the Phillips curve), but

[58] See https://alisdairmckay.com/Notes/HetAgents/index.html.

also an additional equation for $r_t$. Similarly, we have one entry in $\mathbf{X}_t$ for each of the unknowns in figure 3 ($Y$, $w$, and $\pi$), but also $r_t$ as an unknown. We calculate each target in $F$ by starting with the unknowns and progressively calculating the date-$t$ output of each block in figure 3. Since we need $r_{t+1}$ as an input to the calculation of the Phillips curve[59], however, we include $r$ as part of $\mathbf{X}$ and add the equation from the "monetary" block explicitly to $F$.

### C.3 Two-asset household model algorithm

In this section we describe a generic two-asset household model with convex adjustment costs in an illiquid asset whose return is superior to that of a liquid asset, in the spirit of Kaplan, Moll and Violante (2018). We then describe an efficient algorithm, based on the endogenous grid points approach of Carroll (2006), to solve this model.

**Generic setup.** Households' individual state variables are (exogenous) income $z \in \{z_1, \ldots, z_m\}$, liquid assets $b \in [\underline{b}, \infty)$, and illiquid assets $a \in [0, \infty)$. Adjusting the illiquid account incurs a convex cost $\Phi(a_t, a_{t-1})$. The Bellman equation is

$$\begin{aligned} V_t(z_t, b_{t-1}, a_{t-1}) &= \max_{c_t, b_t, a_t} u(c_t) + \beta \mathbb{E}_t V_{t+1}(z_{t+1}, b_t, a_t) \\ \text{s.t. } c_t + a_t + b_t &= z_t + (1 + r_t^a)a_{t-1} + (1 + r_t^b)b_{t-1} - \Phi(a_t, a_{t-1}) \\ a_t &\geq 0, \quad b_t \geq \underline{b}. \end{aligned}$$

Let the adjustment cost function be specified as

$$\Phi(a_t, a_{t-1}) = \frac{\chi_1}{\chi_2} \left| \frac{a_t - (1 + r_t^a)a_{t-1}}{(1 + r_t^a)a_{t-1} + \chi_0} \right|^{\chi_2} [(1 + r_t^a)a_{t-1} + \chi_0] \tag{93}$$

with $\chi_0, \chi_1 > 0$ and $\chi_2 > 1$. Note that $\Phi(a_t, a_{t-1})$ is bounded, differentiable, and convex in the choice $a_t$.[60]

**First-order and envelope conditions.** The Bellman equation can be rewritten more compactly as

$$\begin{aligned} V_t(z_t, b_{t-1}, a_{t-1}) = \max_{b_t, a_t} u \Big( z_t + (1 + r_t^a)a_{t-1} + (1 + r_t^b)b_{t-1} - \Phi(a_t, a_{t-1}) - a_t - b_t \Big) \\ + \lambda_t(b_t - \underline{b}) + \mu_t a_t + \beta \mathbb{E} V_{t+1}(z_{t+1}, b_t, a_t) \end{aligned}$$

[59] This appears in the nonlinear $F$ but actually falls out to first order, so is irrelevant to the calculation we will perform.

[60] The functional form (93) is chosen for concreteness; more generally, we could have a mix of terms with different $\chi_2 > 1$.

The first-order conditions with respect to $b_t$ and $a_t$ are

$$u'(c_t) = \lambda_t + \beta \mathbb{E} \partial_b V_{t+1}(z_{t+1}, b_t, a_t), \tag{94}$$

$$u'(c_t)\Big[1 + \Phi_1(a_t, a_{t-1})\Big] = \mu_t + \beta \mathbb{E} \partial_a V_{t+1}(z_{t+1}, b_t, a_t), \tag{95}$$

and the envelope conditions are

$$\partial_b V_t(z_t, b_{t-1}, a_{t-1}) = (1 + r_t^b) u'(c_t), \tag{96}$$

$$\partial_a V_t(z_t, b_{t-1}, a_{t-1}) = \big[1 + r_t^a - \Phi_2(a_t, a_{t-1})\big] u'(c_t). \tag{97}$$

It's convenient to define the *post-decision value function* $W_t(z_t, b_t, a_t) \equiv \beta \mathbb{E}_t V_{t+1}(z_t, b_t, a_t)$. Note that the partials of this are just what we have on the right-hand side of the Euler equations (94) and (95).

**Algorithm.** The algorithm is a variant of the endogenous grid point method of Carroll (2006) that we developed for this two-asset problem. The key trick is, whenever the household is partially constrained, to include Lagrange multipliers in the backward iteration. We also exploit the fact that, endogenously, the constraint on the illiquid asset will never be binding unless the constraint on the liquid asset is also binding (otherwise, a simple variation will improve utility)—and if both are binding, then the policy is trivial.

Overall, we start from a guess for the (discretized) partials of the value function and iterate backward until convergence. Throughout, we will use $(z', b', a')$ to refer to *tomorrow's grid* and $(z, b, a)$ *today's grid*.

1. **Initial guess.** Guess $V_a(z', b', a')$ and $V_b(z', b', a')$.

2. **Common** $z' \to z$. By definition

$$W_b(z, b', a') = \beta \Pi V_b(z', b', a') \tag{98}$$

$$W_a(z, b', a') = \beta \Pi V_a(z', b', a') \tag{99}$$

3. **Unconstrained** $\underline{a' \to a}$. Assuming that no constraints bind, $\lambda_t = \mu_t = 0$, and (94) and (95) become

$$u'(c) = W_b(z, b', a'), \tag{100}$$

$$u'(c)\Big[1 + \Phi_1(a', a)\Big] = W_a(z, b', a'). \tag{101}$$

Combine these to get

$$0 = F(z, b', a, a') \equiv \frac{W_a(z, b', a')}{W_b(z, b', a')} - 1 - \Phi_1(a', a) \tag{102}$$

which characterizes $a'(z, b', a)$. Use this to map $W_b(z, b', a')$ into $W_b(z, b', a)$ by interpolation, then compute consumption as

$$c(z, b', a) = W_b(z, b', a)^{-\frac{1}{\sigma}}. \tag{103}$$

4. **Unconstrained** $\underline{b' \to b}$. Now using $a'(z, b', a)$ and $c(z, b', a)$ from the previous step, use the budget constraint to obtain

$$b(z, b', a) = \frac{c(z, b', a) + a'(z, b', a) + b' - (1 + r^a)a + \Phi(a'(z, b', a), a) - z}{1 + r^b}.$$

We invert this function via interpolation to get $b'(z, b, a)$. The same interpolation weights can be used to do $a'(z, b', a) \to a'(z, b, a)$.

5. **Liquidity constrained** $\underline{a' \to a}$. This branch is analogous to the unconstrained case. Assuming that the liquidity constraint is binding, $\lambda_t > 0$, and (94) and (95) become

$$\begin{aligned} u'(c) &= \lambda + W_b(z, 0, a'), \\ u'(c)\Big[1 + \Phi_1(a', a)\Big] &= W_a(z, 0, a'). \end{aligned}$$

To help with scaling, let us define $\kappa \equiv \lambda / W_b(z, 0, a')$ and rewrite the first equation as

$$u'(c) = (1 + \kappa) W_b(z, 0, a').$$

Divide and rearrange to get

$$0 = F(z, \kappa, a, a') \equiv \frac{1}{1 + \kappa} \frac{W_a(z, 0, a')}{W_b(z, 0, a')} - 1 - \Phi_1(a', a). \tag{104}$$

We solve this for $a'(z, \kappa, a)$, and compute consumption as

$$c(z, \kappa, a) = \big[(1 + \kappa) W_b(z, \kappa, a)\big]^{-\frac{1}{\sigma}}. \tag{105}$$

6. **Liquidity constrained** $\underline{\kappa \to b}$. Now using $a'(z, \kappa, a)$ and $c(z, \kappa, a)$ from the previous step, use the budget constraint to obtain

$$b(z, \kappa, a) = \frac{c(z, \kappa, a) + a'(z, \kappa, a) + \underline{b} - (1 + r^a)a + \Phi(a'(z, \kappa, a), a) - z}{1 + r^b}.$$

We invert this function via interpolation to get $\kappa(z, b, a)$. The same interpolation weights can be used to map $a'(z, \kappa, a)$ into $a'(z, b, a)$. We already know that $b'(z, b, a) = \underline{b}$.

7. **Update guesses.** The final $b'(z, b, a)$ is the element-wise maximum of its unconstrained and liquidity-constrained counterparts. Replace the unconstrained $a'(z, b, a)$ with constrained

one at the exact same points. Compute consumption from the budget constraint as

$$c(z,b,a) = z + (1+r^a)a + (1+r^b)b - \Phi(a'(z,b,a),a) - a'(z,b,a) - b'(z,b,a). \tag{106}$$

Finally use the envelope conditions (96) and (97) to update the guesses

$$V_b(z,b,a) = (1+r^b)c(z,b,a)^{-\sigma}, \tag{107}$$

$$V_a(z,b,a) = \left[1 + r^a - \Phi_2\Big(a'(z,b,a),a\Big)\right]c(z,b,a)^{-\sigma}. \tag{108}$$

Go back to step 2, repeat until convergence.

### C.4 Efficient multiplication of simple Jacobians

One important detail underlying the speeds in table 3 is a set of special routines that efficiently handle the Jacobians of simple blocks. These simple blocks comprise the majority of our DAGs. Their Jacobians are easy to obtain to high accuracy (for instance, with symmetric numerical differentiation), and have a special sparse structure: they can be expressed as linear combinations of a few *shift* operators $S_i$ on sequences.

For positive $i$, $S_i$ maps $(x_0, x_1, \ldots) \to (0, \ldots, 0, x_0, x_1, \ldots)$, with $i$ zeros inserted at the beginning, and for negative $-i$, $S_{-i}$ maps $(x_0, x_1, \ldots) \to (x_i, x_{i+1}, \ldots)$. The former takes an $i$-period *lag* in sequence space, while the latter takes an $i$-period *lead* in sequence space.[61] For instance, in the one-asset HANK economy depicted in figure 4, the Jacobian $\mathcal{J}^{H_1,\pi}$ of the Phillips curve condition with respect to price inflation $\pi$ is $S_0 - \frac{1}{1+r}S_{-1}$.[62]

For the most part, these operators obey simple rules: if $i$ and $j$ are both positive, $S_iS_j = S_{i+j}$, and so on. However, as is well known from an older literature that works with the lag algebra (e.g. Whiteman 1983), the $S$ are not quite closed under multiplication. To take the simplest example, $S_1S_{-1}$, a one-period lag of a one-period lead, maps $(x_0, x_1, x_2, \ldots) \to (0, x_1, x_2, \ldots)$, zeroing out the first entry of a sequence and leaving everything else unchanged. Fortunately, we have found a more general set of operators that includes the $S$ and is closed under multiplication following an easy-to-compute rule, as we derive in the following proposition.

**Proposition 4.** *Let $S_i$ be the shift operator on sequences, and $Z_m$ be the "zero" operator that replaces the first $m$ entries of a sequence with zeros. If we define*

$$Q_{i,m} \equiv \begin{cases} S_iZ_m & i > 0 \\ Z_mS_i & i < 0 \end{cases} \tag{109}$$

*then $Q_{i,m}Q_{j,n} = Q_{k,l}$, where*

$$k = i + j \tag{110}$$

[61] In matrix form, $S_i$ has zeros everywhere, except for ones on the $i$th diagonal below the main diagonal.

[62] This corresponds to a linearized curve of the form $\pi_t = \ldots + \frac{1}{1+r}\mathbb{E}_t\pi_{t+1}$.

*and*

$$l = \begin{cases} \max(m-j, n) & i, j \geq 0 \\ \max(m, n) + \min(i, -j) & i \geq 0,\ j \leq 0 \\ \max(m-i-j, n) & i \leq 0,\ j \geq 0,\ i+j \geq 0 \\ \max(n+i+j, m) & i \leq 0,\ j \geq 0,\ i+j \leq 0 \\ \max(m, n+i) & i, j \leq 0 \end{cases} \tag{111}$$

This proposition nests the shift operators $S_i$ in a more general class of operators $Q_{i,m}$.[63] This has two advantages. First, it makes multiplying the Jacobians of simple blocks vastly more efficient: rather than doing matrix multiplication with large $T \times T$ matrices, we just need to apply rules (110) and (111) a few times. Second, it is computationally easy to multiply $Q_{i,m}$ and an ordinary matrix Jacobian (or vector), since this is a combination of shifting and zeroing elements. Together, these features make forward accumulation on the DAG, which consists mostly of simple blocks, vastly more efficient.

In our online code, we implement this by simply overriding the matrix multiplication operator, so that sparse linear combinations of $Q_{i,m}$ and ordinary matrices can be used interchangeably. With this in place, the methods of section 4.3 can be applied without any outwardly visible modification.

Exploiting sparsity has played a prominent role in both the heterogeneous-agent literature (e.g. Achdou, Han, Lasry, Lions and Moll 2020) and the literature on solving for perfect-foresight paths using Newton's method (e.g. Juillard 1996). Our approach builds on the latter, but our much more compact representation of Jacobians offers additional efficiencies. For instance, to store $0.5 \cdot Q_{1,1}$, we only need a few numbers, while a conventional $T \times T$ sparse matrix representation not taking advantage of this structure would need $T - 2$ separate entries, and still create some truncation error.

**Proof of proposition 4.** Here, we derive the rules for multiplication of the operator (109), where $S_i$ is the shift operator on sequences by $i$ and $Z_m$ zeros out the first $m$ elements of sequences, by doing case-by-case analysis on the product $Q_{i,m}Q_{j,n}$. In our derivation, we will exploit the following fact about multiplication of $S_i$:

$$S_i S_j = \begin{cases} S_{i+j} Z_{-j} & i > 0,\ j < 0,\ i+j > 0 \\ Z_i S_{i+j} & i > 0,\ j < 0,\ i+j < 0 \\ S_{i+j} & \text{otherwise} \end{cases}$$

and the rules $S_{-i}Z_j = Z_{\max(j-i,0)}S_{-i}$ and $Z_j S_i = S_i Z_{\max(j-i,0)}$ for multiplication of $S$ and $Z$.

[63] The matrix representation of $Q_{i,m}$ is the same as that of $S_i$, except that the first $m$ entries on the diagonal are zeros.

*Case 1: positive i, positive j.* Here we have

$$\begin{aligned} Q_{i,m}Q_{j,n} &= S_i Z_m S_j Z_n \\ &= S_i S_j Z_{\max(m-j,0)} Z_n \\ &= S_{i+j} Z_{\max(m-j,n)} \\ &= Q_{i+j,\max(m-j,n)} \end{aligned} \tag{112}$$

*Case 2: positive i, negative j.* Here we have

$$\begin{aligned} Q_{i,m}Q_{j,n} &= S_i Z_m Z_n S_j \\ &= S_i Z_{\max(m,n)} S_j \end{aligned}$$

If $i + j > 0$, then we write

$$Z_{\max(m,n)} S_j = S_j Z_{\max(m,n)-j}$$

and then

$$\begin{aligned} S_i Z_{\max(m,n)} S_j &= S_i S_j Z_{\max(m,n)-j} \\ &= S_{i+j} Z_{-j} Z_{\max(m,n)-j} \\ &= S_{i+j} Z_{\max(m,n)-j} \\ &= Q_{i+j,\max(m,n)-j} \end{aligned}$$

If $i + j < 0$, then we write

$$S_i Z_{\max(m,n)} = Z_{\max(m,n)+i} S_i$$

and then

$$\begin{aligned} S_i Z_{\max(m,n)} S_j &= Z_{\max(m,n)+i} S_i S_j \\ &= Z_{\max(m,n)+i} Z_i S_{i+j} \\ &= Z_{\max(m,n)+i} S_{i+j} \\ &= Q_{i+j,\max(m,n)+i} \end{aligned}$$

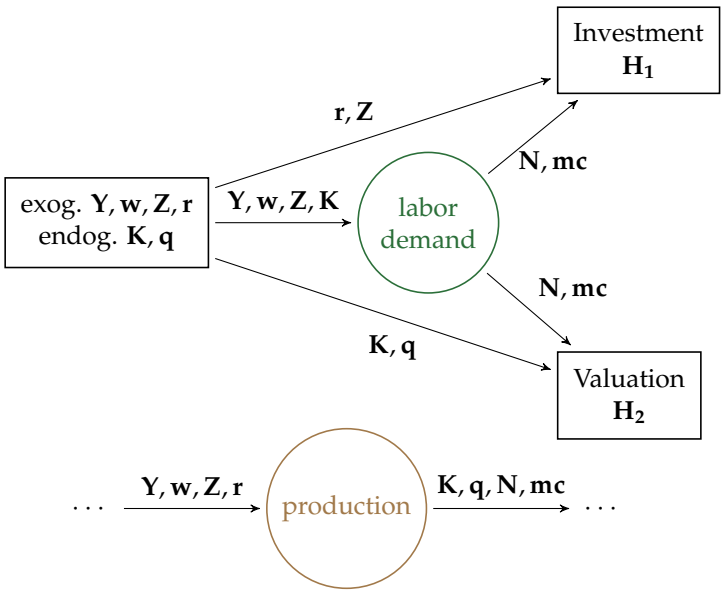
Both these cases boil down to the simpler form

$$Q_{i,m}Q_{j,n} = Q_{i+j,\max(m,n)+\min(i,-j)} \tag{113}$$

*Case 3: negative i, positive j.* Then we have

$$\begin{aligned} Q_{i,m}Q_{j,n} &= Z_m S_i S_j Z_n \\ &= Z_m S_{i+j} Z_n \end{aligned}$$

Figure C.1: The concept of a solved block, applied to the production block of our two-asset HANK model




If $i + j > 0$, then we write $Z_m S_{i+j} = S_{i+j} Z_{\max(m-i-j,0)}$ and get

$$Q_{i,m} Q_{j,n} = Q_{i+j,\max(m-i-j,n)} \tag{114}$$

If $i + j < 0$, then we write $S_{i+j} Z_n = Z_{\max(n+i+j,0)} S_{i+j}$ and get

$$Q_{i,m} Q_{j,n} = Q_{i+j,\max(n+i+j,m)} \tag{115}$$

*Case 4: negative $i$, negative $j$.* Then we have

$$\begin{aligned} Q_{i,m} Q_{j,n} &= Z_m S_i Z_n S_j \\ &= Z_m Z_{\max(n+i,0)} S_i S_j \\ &= Z_{\max(m,n+i)} S_{i+j} \\ &= Q_{i+j,\max(m,n+i)} \end{aligned} \tag{116}$$

Combined, (112)-(116) give (110) and (111) in proposition 4.

## C.5 Solved blocks

The two-asset model introduced via the DAG in figure B.1 included a green "production" block. Production with adjustment costs is well-known to involve the joint determination of investment and $q$, and it is natural to solve for these two jointly inside a block. This leads us to introduce a "solved block" concept, as follows:

**Definition 5.** A *solved block $b$* has an underlying sequence-space model with shocks $\tilde{\mathcal{Z}}$, unknowns

$\tilde{\mathcal{U}}$, outputs $\tilde{\mathcal{O}}$, and targets $\tilde{\mathcal{T}}$, and an equilibrium that is locally unique around the steady state, where we define:

1. The inputs of the solved block to be the shocks of the underlying sequence-space model: $\mathcal{I}_b \equiv \tilde{\mathcal{Z}}$.
2. The outputs of the solved block to be the unknowns and outputs, minus targets, of the underlying sequence-space model: $\mathcal{O}_b \equiv \tilde{\mathcal{U}} \cup (\tilde{\mathcal{O}} \setminus \tilde{\mathcal{T}})$.
3. For each output $o \in \mathcal{O}_b$, the function $h^o(\{\mathbf{x}^i\}_{i\in\mathcal{I}_b})$ is the locally unique equilibrium path of $o$ in the underlying sequence-space model given sequences $\{\mathbf{x}^i\}_{i\in\tilde{\mathcal{Z}}}$ for the exogenous shocks in that model (recalling that $\mathcal{I}_b = \tilde{\mathcal{Z}}$).

Informally, a solved block is a sequence-space model, turned into a block. Figure C.1 illustrates how this concept works in the case of the production block of our two-asset HANK model. Given the exogenous inputs $Y, w, Z, r$, the solved block solves for the endogenous paths for $K$ and $Q$ that jointly satisfy the $q$ theory equations, so that its outputs are $K, Q$ as well as labor demand $N$ and marginal costs $mc$.

### C.6 Fast solution for individual impulse responses

In the case where we are only interested in a single impulse response, we only need to do full forward accumulation (29) for $i \in \mathcal{U}$ to obtain $o \in \mathcal{H}$, which gives $\mathbf{H_U} = \mathbf{J}^{\mathcal{H},\mathcal{U}}$. Then, to deal with shocks, we do forward accumulation on *vectors* rather than matrices, writing

$$\mathbf{J}^{o,\mathcal{Z}} d\mathbf{Z} = \sum_{m\in\mathcal{I}_b} \mathcal{J}^{o,m} \mathbf{J}^{m,\mathcal{Z}} d\mathbf{Z} \tag{117}$$

This gives $\mathbf{H_Z} d\mathbf{Z} = \mathbf{J}^{\mathcal{H},\mathcal{Z}} d\mathbf{Z}$. We then solve the linear system $\mathbf{H_U} d\mathbf{U} = -\mathbf{H_Z} d\mathbf{Z}$ to obtain $d\mathbf{U}$. Finally, to obtain equilibrium impulse responses $d\mathbf{X}^o$ for $o \notin \mathcal{Z} \cup \mathcal{U}$, we need to calculate

$$d\mathbf{X}^o = \mathbf{J}^{o,\mathcal{Z}} d\mathbf{Z} + \mathbf{J}^{o,\mathcal{U}} d\mathbf{U} \tag{118}$$

The first term, $\mathbf{J}^{o,\mathcal{Z}} d\mathbf{Z}$, has already been calculated in (117). For the second term, we do forward accumulation on vectors as in (117), just solving for $\mathbf{J}^{o,\mathcal{U}} d\mathbf{U}$ rather than $\mathbf{J}^{o,\mathcal{Z}} d\mathbf{Z}$.[64]

Table C.1 shows the time each step of this process takes for our three models, starting from the Jacobians $\mathcal{J}$ for each model block. In general, this process is very cheap, with the only costly parts being the steps that involve matrices rather than vectors: forward accumulation in step 1 to get $\mathbf{H_U} = \mathbf{J}^{\mathcal{H},\mathcal{U}}$, and second, solving the linear system $\mathbf{H_U} d\mathbf{U} = -\mathbf{H_Z} d\mathbf{Z}$ for $d\mathbf{U}$ in step 3.

[64] Another approach is to use the $\mathbf{J}^{o,\mathcal{U}}$ that we already calculated as part of the initial forward accumulation to obtain $\mathbf{H_U} = \mathbf{J}^{\mathcal{H},\mathcal{U}}$, and directly apply these to $d\mathbf{U}$. This approach has similar (and low) cost, but is less useful in general because it does give $o$ that were not necessary in calculating $\mathbf{H_U}$.

Table C.1: Computing times for impulse responses.

| | Krusell-Smith | one-asset HANK | two-asset HANK |
|---|---|---|---|
| **Total** | **0.9 ms** | **15.5 ms** | **30.1 ms** |
| step 1 (forward accumulate $\mathbf{H_U}$) | 0.4 ms | 6.2 ms | 19.7 ms |
| step 2 (forward accumulate $\mathbf{J}^{o,Z}d\mathbf{Z}$) | 0.1 ms | 0.1 ms | 0.4 ms |
| step 3 (solve linear system for $d\mathbf{U}$) | 0.4 ms | 9.0 ms | 8.9 ms |
| step 4 (forward accumulate $\mathbf{J}^{o,\mathcal{U}}d\mathbf{U}$, get $d\mathbf{X}^o$) | 0.1 ms | 0.3 ms | 1.2 ms |
| No. of unknowns | 1 | 3 | 3 |
| No. of exogenous shocks | 1 | 3 | 7 |

Table C.2: Computing times for $\mathbf{G}$, efficient vs. flat DAG.

| | Krusell-Smith | one-asset HANK | two-asset HANK |
|---|---|---|---|
| Total with efficient DAG | 4.7 ms | 57.8 ms | 198.0 ms |
| Total with flat DAG | 33.0 ms | 167.5 ms | 1452.7 ms |
| No. of unknowns (efficient DAG) | 1 | 3 | 3 |
| No. of unknowns (flat DAG) | 3 | 7 | 18 |
| No. of exogenous shocks | 1 | 3 | 7 |

Since these steps are costly because they involve the shock-independent matrix $\mathbf{H_U}$, there are clear economies of scale from computing the impulse response to multiple shocks. We can calculate $\mathbf{H_U}$ a single time, and then also calculate $\mathbf{H_U}^{-1}$ (or, better, an LU factorization of $\mathbf{H_U}$) a single time, at which point the marginal cost of computing additional impulse responses is very low. This is the approach we use in section 5.3 to evaluate the likelihood when redrawing model parameters, since this involves finding impulse responses to each shock simultaneously. Taking this idea to its fullest extent, we can calculate the impulse responses to all shocks simultaneously, which is the "$\mathbf{G}$ matrix" approach in section 4.3.

## C.7 Comparison of computing times with efficient and flat DAG

How important is the pattern of variable substitution along the DAG to efficiently solving heterogeneous-agent models? Table C.2 compares the times needed to compute the $\mathbf{G}$ matrices using our preferred DAG ("efficient DAG") and using no substitution of endogenous variables ("flat DAG"). The latter approach results in a greater number of unknowns and is substantially slower, by a factor that ranges from 2 to 10. The reason is that the dimensionality of the linear system becomes so high that solving it is quite costly.

## C.8 Recovering the state-space law of motion

Here we explain how to recover the state-space law of motion in the Krusell-Smith model with AR(1) TFP shocks of persistence $\rho$. The state then consists of the $n_g$ points of the distribution and

the two aggregate states $K_{t-1}$ and $Z_{t-1}$, so the law of motion reads

$$\begin{pmatrix} \mathbf{D}_{t+1} \\ K_t \\ Z_t \end{pmatrix} = \mathbf{A} \begin{pmatrix} \mathbf{D}_t \\ K_{t-1} \\ Z_{t-1} \end{pmatrix} + \mathbf{B}\epsilon_t \tag{119}$$

with $\mathbf{A}$ an $(n_g + 2) \times (n_g + 2)$ matrix indicating the dependence of states on past states, and $\mathbf{B}$ an $(n_g + 2) \times 1$ vector indicating how states respond to innovations $\epsilon_t$ to the TFP process. To elicit $\mathbf{A}$ and $\mathbf{B}$ from a sequence-space model, we treat the initial states ($\epsilon_0$, $\mathbf{D}_0$, $K_{-1}$ and $Z_{-1}$) as exogenous "shocks" and then look for their effects on the state the following period, $(\mathbf{D}_1, K_0, Z_0)$.

Using the sequence-space model, we can compute the effect of shocks to ($\epsilon_0$, $\mathbf{D}_0$, $K_{-1}$ and $Z_{-1}$) on on the time paths of targets $d\mathbf{H}$. This is straightforward to do for aggregates, while for the distribution $\mathbf{D}_0$, these perturbations are directly given by the expectation vectors $\mathcal{E}_t^{\mathcal{K}}$ from definition 1. Next, we solve for the equilibrium path of capital $d\mathbf{K} = -\mathbf{H}_{\mathbf{K}}^{-1} d\mathbf{H}$ that results from the perturbation. The date-0 element of this vector delivers the rows of $\mathbf{A}$ and $\mathbf{B}$ corresponding to $K$. Finally, using the general equilibrium Jacobian matrices $J^{w,K}$ and $J^{r,K}$, we can recover the effect on the time paths of wages and interest rates $d\mathbf{w}$ and $d\mathbf{r}$ for all shocks ($\epsilon_0$, $\mathbf{D}_0$, $K_{-1}$ and $Z_{-1}$). Then, using the equation $d\mathbf{D}_1 = \mathcal{D}_1^r d\mathbf{r} + \mathcal{D}_1^w d\mathbf{w}$, where the $\mathcal{D}_t$ vectors are discussed in section 3.2, we obtain the effect of all shocks ($\epsilon_0$, $\mathbf{D}_0$, $K_{-1}$ and $Z_{-1}$) on the distribution at date 1. This gives us the first $n_g$ rows of $\mathbf{A}$ and $\mathbf{B}$. Finally, the last row is just the exogenous law of motion of the shock process. Altogether, this procedure allows us to construct a state-space law of motion for the Krusell-Smith model. The procedure can easily be generalized to any alternative sequence-space model with a known state space.

### C.9 Simulating panels of individuals

Here, we briefly describe how to simulate a panel of individuals. The first option is to recover the state-space law of motion, as described at the end of section (4.3), augmented with policies. Then, one can simulate using the state space.

The second option is to recover the MA for policies. For example, in the Krusell-Smith model, the MA for the capital policy, truncated to $T$, in response to innovations $\epsilon_t$ to TFP, takes the form

$$d\mathbf{k}_t = \sum_{s=0}^{T} \mathbf{K}_s \epsilon_{t-s} \tag{120}$$

where the $i^{th}$ entry in the vector $d\mathbf{k}_t$ corresponds to the change in the capital policy of households in state $i$. Hence, in order to simulate a panel of individuals, we need to recover the $N \times 1$ vectors $\mathbf{K}_s$. This can be done as follows. Using the policy function symmetry property from Lemma 1, we

know that

$$\begin{pmatrix} d\mathbf{k}_0 \\ d\mathbf{k}_1 \\ d\mathbf{k}_2 \\ \vdots \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathbf{k}_0}{\partial w_0} & \frac{\partial \mathbf{k}_0}{\partial w_1} & \frac{\partial \mathbf{k}_0}{\partial w_2} \\ 0 & \frac{\partial \mathbf{k}_0}{\partial w_0} & \frac{\partial \mathbf{k}_0}{\partial w_1} \\ 0 & 0 & \frac{\partial \mathbf{k}_0}{\partial w_0} \end{pmatrix} \begin{pmatrix} dw_0 \\ dw_1 \\ dw_2 \\ \vdots \end{pmatrix} + \begin{pmatrix} \frac{\partial \mathbf{k}_0}{\partial r_0} & \frac{\partial \mathbf{k}_0}{\partial r_1} & \frac{\partial \mathbf{k}_0}{\partial r_2} \\ 0 & \frac{\partial \mathbf{k}_0}{\partial r_0} & \frac{\partial \mathbf{k}_0}{\partial r_1} \\ 0 & 0 & \frac{\partial \mathbf{k}_0}{\partial r_0} \end{pmatrix} \begin{pmatrix} dr_0 \\ dr_1 \\ dr_2 \\ \vdots \end{pmatrix}$$

$$= \begin{pmatrix} \frac{\partial \mathbf{k}_0}{\partial w_0} & \frac{\partial \mathbf{k}_0}{\partial w_1} & \frac{\partial \mathbf{k}_0}{\partial w_2} & \cdots \end{pmatrix} \begin{pmatrix} dw_0 & 0 & 0 \\ dw_1 & dw_0 & \ddots \\ dw_2 & dw_1 & \ddots \\ dw_3 & dw_2 & \ddots \\ \vdots & \vdots & \ddots \end{pmatrix} + \begin{pmatrix} \frac{\partial \mathbf{k}_0}{\partial r_0} & \frac{\partial \mathbf{k}_0}{\partial r_1} & \frac{\partial \mathbf{k}_0}{\partial r_2} & \cdots \end{pmatrix} \begin{pmatrix} dr_0 & 0 & 0 \\ dr_1 & dr_0 & \ddots \\ dr_2 & dr_1 & \ddots \\ dr_3 & dr_2 & \ddots \\ \vdots & \vdots & \ddots \end{pmatrix} \tag{121}$$

The derivatives of the first-period policy $\mathbf{k}_0$ with respect to $w_t$ and $r_t$ are byproducts of step 1 of the fake news algorithm. Further, from the procedure to recover the MA representation described in section 5.1, we obtain the matrices $M^{w,\epsilon}$ and $M^{r,\epsilon}$ satisfying

$$\begin{pmatrix} dw_0 \\ dw_1 \\ dw_2 \\ \vdots \end{pmatrix} = M^{w,\epsilon} \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \vdots \end{pmatrix} \quad \begin{pmatrix} dr_0 \\ dr_1 \\ dr_2 \\ \vdots \end{pmatrix} = M^{r,\epsilon} \begin{pmatrix} \epsilon_0 \\ \epsilon_1 \\ \epsilon_2 \\ \vdots \end{pmatrix} \tag{122}$$

Combining (121) and (122) delivers the coefficients $\mathbf{K}_t$ in (120). Note that equation (121) is fast to implement, since it involves a multiplication of $N \times T$ matrices by $T \times T$ matrices that can be formed efficiently from (122).

Finally, to simulate a panel of individuals from the MA for policies, one can then take these perturbed policy functions and apply them to simulated individuals.

### C.10 Fast Fourier transform to compute analytical second moments.

Consider any sequences $a_0, \ldots, a_{T-1}$ and $b_0, \ldots, b_{T-1}$ of real scalars. If we define the sequences

$$\begin{aligned} (\hat{a}_0, \ldots, \hat{a}_{2T-2}) &= (a_0, \ldots, a_{T-1}, 0, \ldots, 0) \\ (\hat{b}_0, \ldots, \hat{b}_{2T-2}) &= (b_0, \ldots, b_{T-1}, 0, \ldots, 0) \end{aligned}$$

to be $a$ and $b$ each padded by $T-1$ zeros, then

$$a_0 b_u + a_1 b_{u+1} + \ldots + a_{T-1-u} b_{T-1} = \sum_{\ell=0}^{2T-2} \hat{a}_\ell \hat{b}_{u+\ell} \tag{123}$$

where $\hat{b}_{u+\ell} \equiv \hat{b}_{u+\ell-(2T-2)}$ when $u+\ell \geq 2T-2$. It then follows from the standard properties of the discrete Fourier transform $\mathcal{F}$ that for any $u \in 0, \ldots, T-1$

$$\sum_{\ell=0}^{2T-2} \hat{a}_\ell \hat{b}_{u+\ell} = \left(\mathcal{F}^{-1}\left(\mathcal{F}(\hat{a})^* \cdot \mathcal{F}(\hat{b})\right)\right)_s \tag{124}$$

where $*$ denotes complex conjugation.[65]

Since the discrete Fourier transform is a linear operator, we can extend this method to apply to the matrices $d\mathbf{X}_0, \ldots, d\mathbf{X}_{T-1}$ in (35), where we interpret $\mathcal{F}$ as applying element-by-element to a sequence of matrices. Letting $d\hat{\mathbf{X}}_0, \ldots, d\hat{\mathbf{X}}_{2T-2}$ denote the sequence padded with zeros like above, we have from (123) and (124), substituting $u = t' - t$, that

$$\begin{aligned} \sum_{s=0}^{T-1-(t'-t)} (d\mathbf{X}_s)(d\mathbf{X}_{s+t'-t})' &= [d\mathbf{X}_0][d\mathbf{X}_{t'-t}]' + \ldots + [d\mathbf{X}_{T-1-(t'-t)}][d\mathbf{X}_{T-1}]' \\ &= \left(\mathcal{F}^{-1}\left(\mathcal{F}(d\hat{\mathbf{X}})^* \mathcal{F}(d\hat{\mathbf{X}}')\right)\right)_{t'-t} \end{aligned} \tag{125}$$

where $d\hat{\mathbf{X}}$ is the stacked sequence $d\hat{\mathbf{X}}_0, \ldots, d\hat{\mathbf{X}}_{2T-2}$, the transpose $d\hat{\mathbf{X}}'$ is applied individually to each matrix in the sequence, and $\mathcal{F}(d\hat{\mathbf{X}})^* \mathcal{F}(d\hat{\mathbf{X}}')$ is the product of each pair of matrices in the frequency-by-frequency sequence.[66]

We simply apply (125), using the fast Fourier transform for $\mathcal{F}$, to calculate the covariances in (35) for each $t' - t$. Since the two key operations—the FFT and matrix multiplication—have extremely efficient implementations widely available, this can be done very quickly, taking only a few milliseconds in table 4 for the examples in this paper. It is far faster than a naive calculation of the sum in (35).

This procedure is closely related to the standard FFT approach to calculating the empirical autocovariance function (although many implementations only apply to 1-dimensional series, missing the efficiencies from exploiting linearity in equation (125)).[67] It is also similar to the standard formulas for the spectral density of an $MA$, and for inverting this spectrum (see e.g. Hansen and Sargent 1981).

# D Evaluation of accuracy

## D.1 Accuracy of alternative methods to compute $\mathcal{J}$

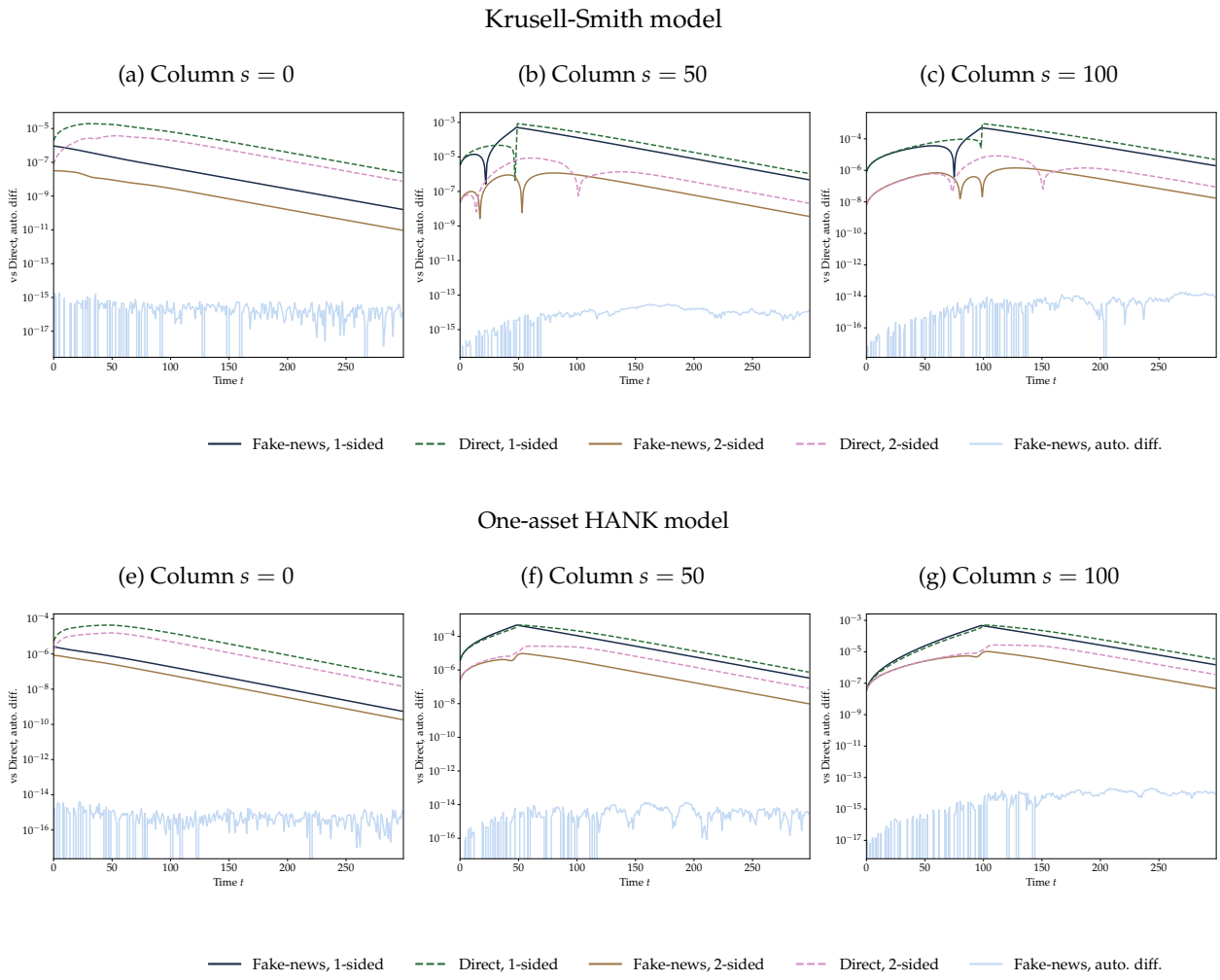
Figure D.1 verifies the relative accuracy of various methods for computing the Jacobian of aggregate assets with respect to the interest rate ($\mathcal{J}^{\mathcal{K},r}$ or $\mathcal{J}^{\mathcal{A},r}$) in our Krusell-Smith and one-Asset

[65] The padding with zeros to create $\hat{a}$ and $\hat{b}$ is necessary so that the wraparound $\hat{b}_{s+\ell}$ terms for large $s+\ell$ do not affect the sum.

[66] Since the inputs are real, the full transform is redundant and we can deal only with the first $T$ entries; the final $T-2$ are complex conjugates of entries 1 through $T-1$. This economizes on the time for $\mathcal{F}$ and also for matrix multiplication.

[67] For instance, in the Python "statsmodels" package, the now-default "fft=True" option uses the FFT to calculate the autocovariances of a one-dimensional time series.

Figure D.1: Accuracy of methods for computing $\mathcal{J}$: distance of columns from direct, auto.diff. method




HANK models. These are the two models for which it is feasible for us to compute the model with automatic differentiation. Our benchmark is the model computed using the direct method under automatic differentiation, which we will refer as the "true" impulse response. The impulse response in levels for the Krusell-Smith model are then those displayed in figure 2, panel (a). For the one-asset HANK model, the levels are very similar. Here we focus on the differences between various methods, for columns $s = 0, 50, 100$.[68]

We first compare impulse response obtained using our fake news algorithm under automatic differentiation to the "truth". The purple line on all graphs shows errors of the order of $10^{-14}$. In other words, the direct and the fake news method yield the same answer to machine precision: this verifies Proposition 1.

Next, we compare the impulse response obtained with one-sided numerical differentiation (blue and orange lines). There, the errors can get as large as $10^{-3}$, or 0.01% of the peak of the level response (which is around 10). Two-sided numerical differentiation (green and red lines) mitigate

[68] Using the sup norm over the entire Jacobian, as well as other Jacobians obtained via this method, yields the same findings.

this error by one to two digits of accuracy. In practice, two sided numerical differentiation is just as simple to implement as one-sided numerical differentiation and only twice as costly in terms of computation time, so this may provide a useful alternative when very good accuracy is required.

Finally, we note that, unless automatic differentiation is used, the fake news method actually generally has better performance than the direct method. This is because it imposes some of the linearity implications of the true first-order derivative. Hence, the fake news impulse response is not only around $T$ times faster to compute than the direct impulse response, it also tends to be more accurate.

### D.2 Equivalence between SSJ and Dynare for representative-agent models

In order to illustrate the accuracy of our routines to calculate impulse responses for representative-agent models, here we perform two tests of the accuracy of impulse responses on two classic representative agent models. Specifically, we simulate the Smets and Wouters (2007) model and the benchmark model described in Herbst and Schorfheide (2015), with the parameters at the estimated mode presented in these sources, using both our method and Dynare. Figure D.2 compares the impulse responses of output to all shocks for the Smets-Wouters exercise: the left panel shows the level of the impulse responses, and the right panel displays the difference to Dynare. Figure D.3 repeats this exercise with the Herbst-Schorfheide model. As can be seen, our method delivers the same impulse responses for these benchmark representative-agent models with very high accuracy.

### D.3 Accuracy of likelihood computation for representative-agent models

Here, we continue the exercise of section D.2 by showing that, for both the Smets and Wouters (2007) model and Herbst and Schorfheide (2015) model, estimating the posterior mode on the original dataset yields the same answer with our method as it does with the routines offered in Dynare (which uses the Kalman filter on a state-space representation of the model). Table D.1 shows that the posterior mode is identical to three digit accuracy. As discussed in the main text, this verifies that our method to compute the likelihood for these benchmark models is accurate.

Figure D.2: Equivalence between SSJ and Dynare for the Smets and Wouters (2007) model.

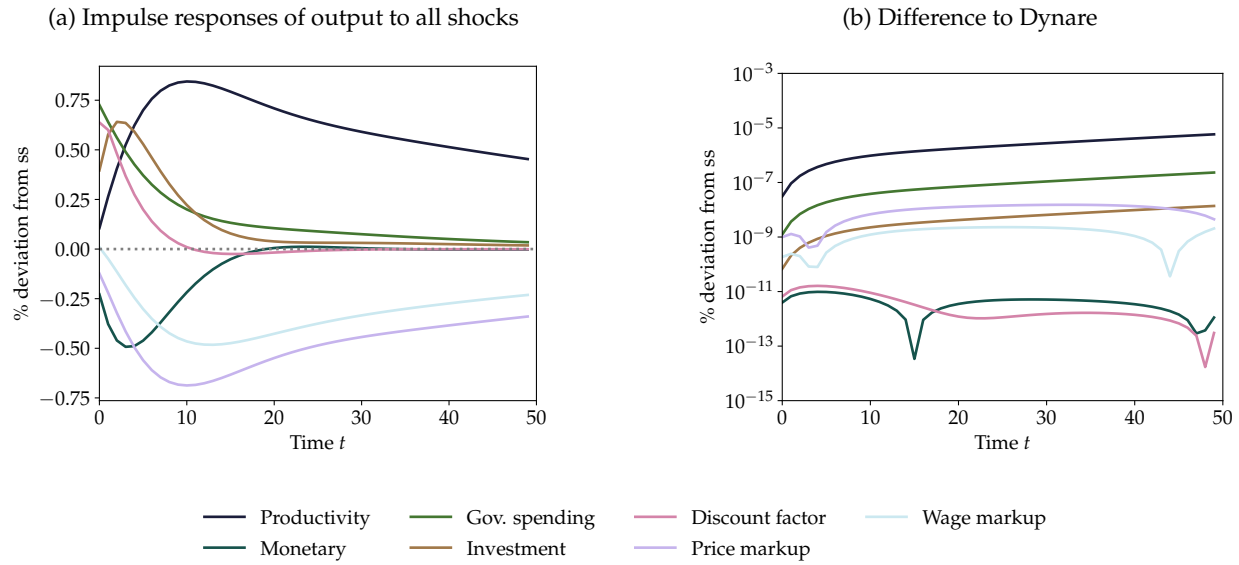



Figure D.3: Equivalence between SSJ and Dynare for the Herbst and Schorfheide (2015) model.

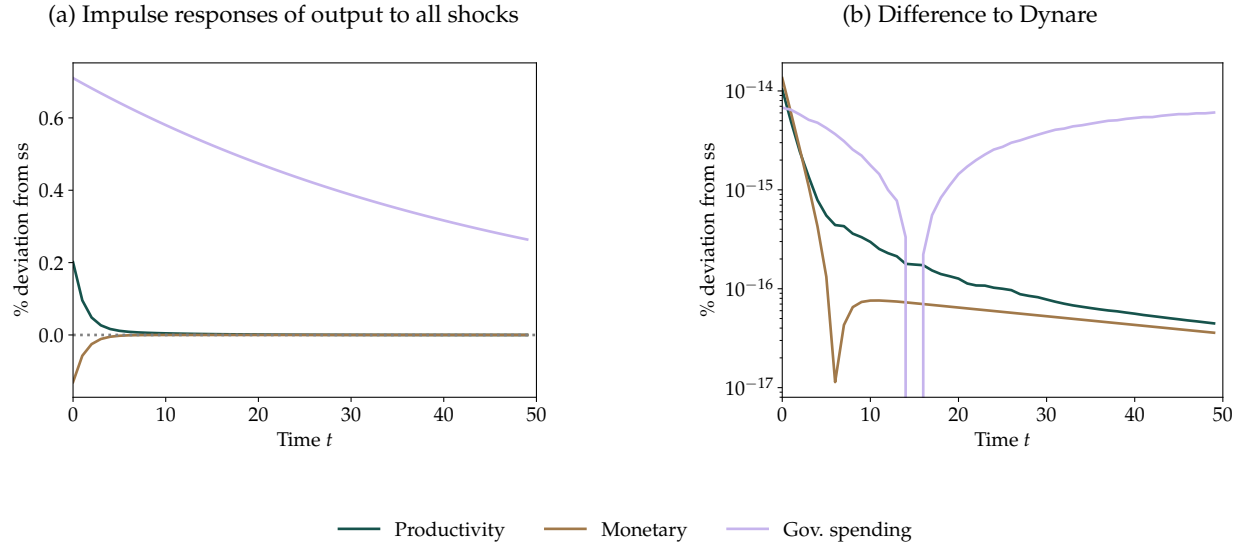


Table D.1: Estimated parameters for the Smets-Wouters and Herbst-Schorfheide economies

| Model | Method | Parameters | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $\sigma_a$ | $\rho_a$ | $\sigma_b$ | $\rho_b$ | $\sigma_g$ | $\rho_g$ | $\sigma_I$ | $\rho_I$ |
| Smets-Wouters | SSJ | 0.446 | 0.978 | 0.246 | 0.250 | 0.589 | 0.971 | 0.461 | 0.662 |
| | Dynare | 0.446 | 0.978 | 0.245 | 0.252 | 0.589 | 0.970 | 0.460 | 0.663 |
| | | $\sigma_i$ | $\rho_i$ | $\sigma_p$ | $\rho_p$ | $\rho_p^{ma}$ | $\sigma_w$ | $\rho_w$ | $\rho_w^{ma}$ |
| | SSJ | 0.229 | 0.086 | 0.135 | 0.975 | 0.740 | 0.256 | 0.976 | 0.925 |
| | Dynare | 0.229 | 0.086 | 0.133 | 0.975 | 0.735 | 0.256 | 0.975 | 0.924 |
| Herbst-Schorfheide | | $\tau$ | $\kappa$ | $\psi_1$ | $\psi_2$ | $\overline{r}$ | $\overline{\pi}$ | $\overline{y}$ | |
| | SSJ | 2.3162 | 1.0000 | 1.9684 | 0.4754 | 0.3043 | 3.4468 | 0.6214 | |
| | Dynare | 2.3164 | 1.0000 | 1.9684 | 0.4753 | 0.3051 | 3.4472 | 0.6213 | |
| | | $\sigma_r$ | $\rho_r$ | $\sigma_g$ | $\rho_g$ | $\sigma_z$ | $\rho_z$ | | |
| | SSJ | 0.1905 | 0.7978 | 0.6531 | 0.9908 | 0.1855 | 0.9252 | | |
| | Dynare | 0.1905 | 0.7978 | 0.6530 | 0.9903 | 0.1855 | 0.9252 | | |

# E Bayesian estimation results

Table E.1 shows the results from the estimation of the Krusell-Smith economy.Table E.2 shows the results from the estimation of the one-asset HANK economy. Table E.3 shows the results from the estimation of the two-asset HANK economy. Figures E.1-E.10 show recursive means and posterior modes for all our estimated models.

Table E.1: Estimated parameters for our Krusell-Smith economy

| | | | Posterior | | |
|---|---|---|---|---|---|
| **Shock** | | **Prior distribution** | **Mode** | **Mean** | **[0.05, 0.95] CI** |
| TFP shock | s.d. | Invgamma(0.4, 4) | 0.179 | 0.182 | [0.165, 0.201] |
| | AR-1 | Beta(0.5, 0.2) | 0.908 | 0.908 | [0.864, 0.951] |
| | MA-1 | Beta(0.5, 0.2) | 0.032 | 0.047 | [0.015, 0.097] |

Table E.2: Estimated parameters for our one-asset HANK economy

| | | | Posterior (shocks) | | | Posterior (shocks + model) | | |
|---|---|---|---|---|---|---|---|---|
| **Parameter / shock** | | **Prior distribution** | **Mode** | **Mean** | **[0.05, 0.95] CI** | **Mode** | **Mean** | **[0.05, 0.95] CI** |
| Monetary policy shock | s.d. | Invgamma(0.4, 4) | 0.429 | 0.433 | [0.392, 0.477] | 0.419 | 0.429 | [0.384, 0.479] |
| | AR-1 | Beta(0.5, 0.2) | 0.529 | 0.524 | [0.475, 0.568] | 0.463 | 0.462 | [0.396, 0.527] |
| $G$ shock | s.d. | Invgamma(0.4, 4) | 0.580 | 0.583 | [0.514, 0.660] | 0.569 | 0.582 | [0.507, 0.666] |
| | AR-1 | Beta(0.5, 0.2) | 0.872 | 0.871 | [0.839, 0.900] | 0.833 | 0.821 | [0.770, 0.867] |
| $P$ markup shock | s.d. | Invgamma(0.4, 4) | 0.099 | 0.101 | [0.091, 0.112] | 0.092 | 0.096 | [0.067, 0.129] |
| | AR-1 | Beta(0.5, 0.2) | 0.881 | 0.878 | [0.849, 0.905] | 0.913 | 0.909 | [0.875, 0.942] |
| $\phi$ | | Gamma(1.5, 0.25) | | | | 1.319 | 1.352 | [1.229, 1.495] |
| $\phi_y$ | | Gamma(0.5, 0.25) | | | | 0.125 | 0.145 | [0.059, 0.255] |
| $\kappa$ | | Gamma(0.1, 0.1) | | | | 0.140 | 0.143 | [0.104, 0.186] |

Table E.3: Estimated parameters for our two-asset HANK economy

| | | | Posterior (shocks) | | | Posterior (shocks + model) | | |
|---|---|---|---|---|---|---|---|---|
| **Parameter / shock** | | **Prior distribution** | **Mode** | **Mean** | **[0.05, 0.95] CI** | **Mode** | **Mean** | **[0.05, 0.95] CI** |
| TFP shock | s.d. | Invgamma(0.4, 4) | 0.072 | 0.073 | [0.066, 0.080] | 0.071 | 0.073 | [0.066, 0.079] |
| | AR-1 | Beta(0.5, 0.2) | 0.994 | 0.941 | [0.912, 0.967] | 0.970 | 0.949 | [0.918, 0.976] |
| $G$ shock | s.d. | Invgamma(0.4, 4) | 0.433 | 0.437 | [0.395, 0.482] | 0.467 | 0.600 | [0.518, 0.687] |
| | AR-1 | Beta(0.5, 0.2) | 0.515 | 0.513 | [0.464, 0.558] | 0.292 | 0.948 | [0.913, 0.978] |
| $\beta$ shock | s.d. | Invgamma(0.4, 4) | 0.093 | 0.094 | [0.085, 0.103] | 0.093 | 0.096 | [0.087, 1.105] |
| | AR-1 | Beta(0.5, 0.2) | 0.941 | 0.938 | [0.907, 0.966] | 0.970 | 0.944 | [0.911, 0.972] |
| $r_I$ (investment) shock | s.d. | Invgamma(0.4, 4) | 0.179 | 0.184 | [0.151, 0.222] | 0.089 | 0.441 | [0.390, 0.501] |
| | AR-1 | Beta(0.5, 0.2) | 0.770 | 0.766 | [0.720, 0.809] | 0.867 | 0.6473 | [0.576, 0.708] |
| Monetary policy shock | s.d. | Invgamma(0.4, 4) | 0.144 | 0.149 | [0.125, 0.176] | 0.656 | 0.625 | [0.450, 0.849] |
| | AR-1 | Beta(0.5, 0.2) | 0.825 | 0.823 | [0.792, 0.851] | 0.844 | 0.747 | [0.687, 0.802] |
| $P$ markup shock | s.d. | Invgamma(0.4, 4) | 0.092 | 0.093 | [0.084, 0.102] | 0.059 | 0.051 | [0.034, 0.070] |
| | AR-1 | Beta(0.5, 0.2) | 0.902 | 0.900 | [0.878, 0.920] | 0.888 | 0.895 | [0.859, 0.925] |
| $W$ markup shock | s.d. | Invgamma(0.4, 4) | 0.434 | 0.438 | [0.400, 0.480] | 0.142 | 0.157 | [0.122, 0.196] |
| | AR-1 | Beta(0.5, 0.2) | 0.887 | 0.885 | [0.857, 0.910] | 0.6498 | 0.574 | [0.446, 0.707] |
| $\phi$ | | Gamma(1.5, 0.25) | | | | 1.203 | 1.229 | [1.020, 1.564] |
| $\phi_y$ | | Gamma(0.5, 0.25) | | | | 0.086 | 2.714 | [2.416, 3.077] |
| $\kappa^p$ | | Gamma(0.1, 0.1) | | | | 0.035 | 0.034 | [0.012, 0.065] |
| $\kappa^w$ | | Gamma(0.1, 0.1) | | | | 0.007 | 0.009 | [0.006, 0.014] |
| $\epsilon_I$ | | Gamma(4, 2) | | | | 0.267 | 0.534 | [0.358, 0.763] |

Figure E.1: Recursive means for the RWMH estimation of the Krusell-Smith model

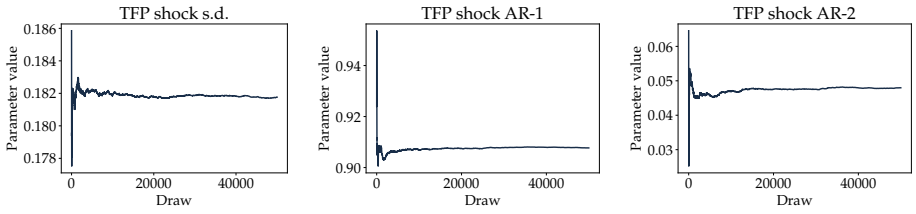



Figure E.2: Posterior distributions for the RWMH estimation of the Krusell-Smith model

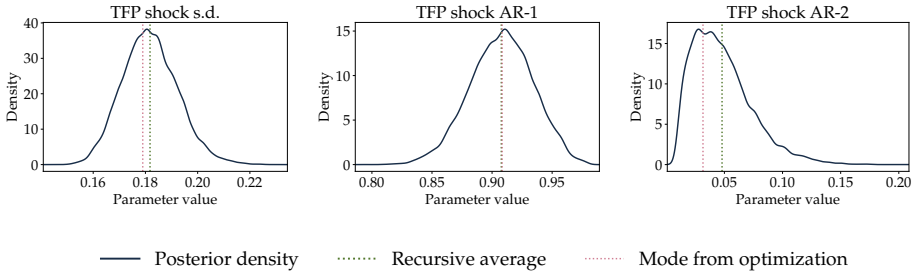


Figure E.3: Recursive means for the RWMH estimation of the one-asset HANK model with shocks

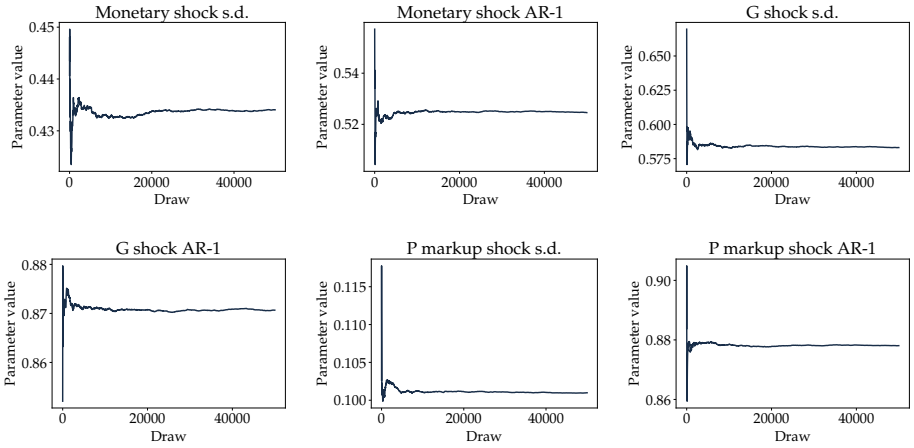



Figure E.4: Posterior distributions for the RWMH estimation of the one-asset HANK model with shocks

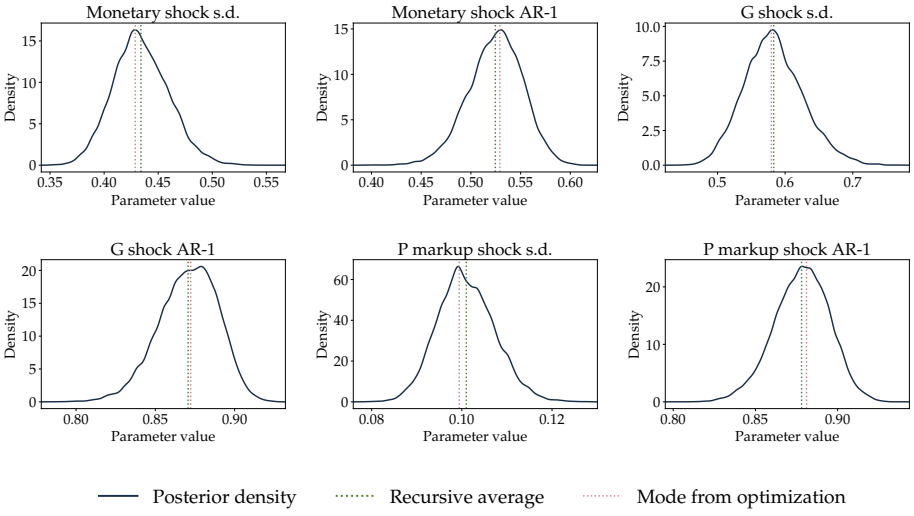


Figure E.5: Recursive means for the RWMH estimation of the one-asset HANK model with shocks and parameters

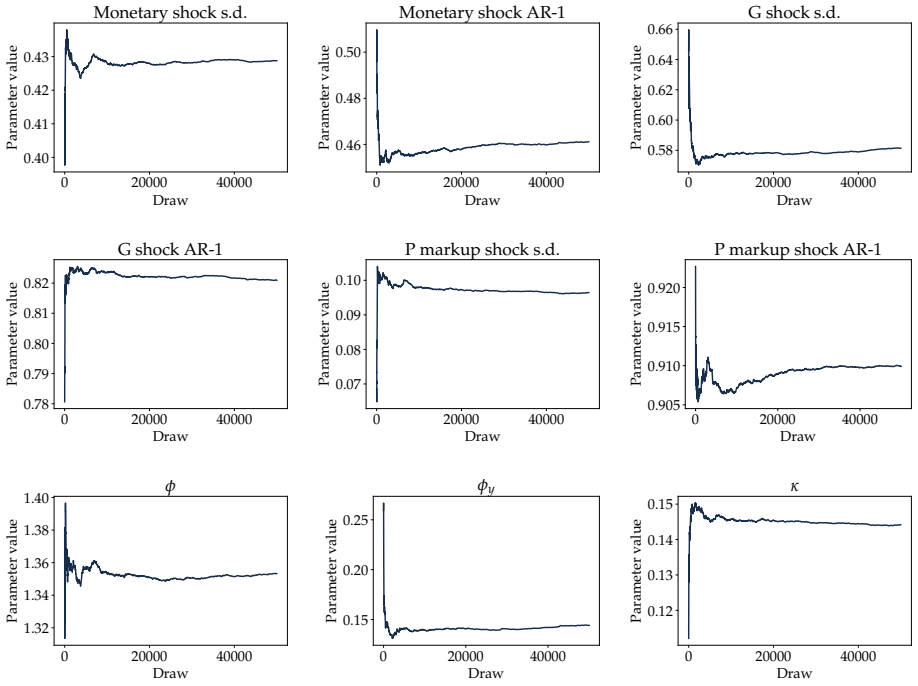



Figure E.6: Posterior distributions for the RWMH estimation of the one-asset HANK model with shocks and parameters

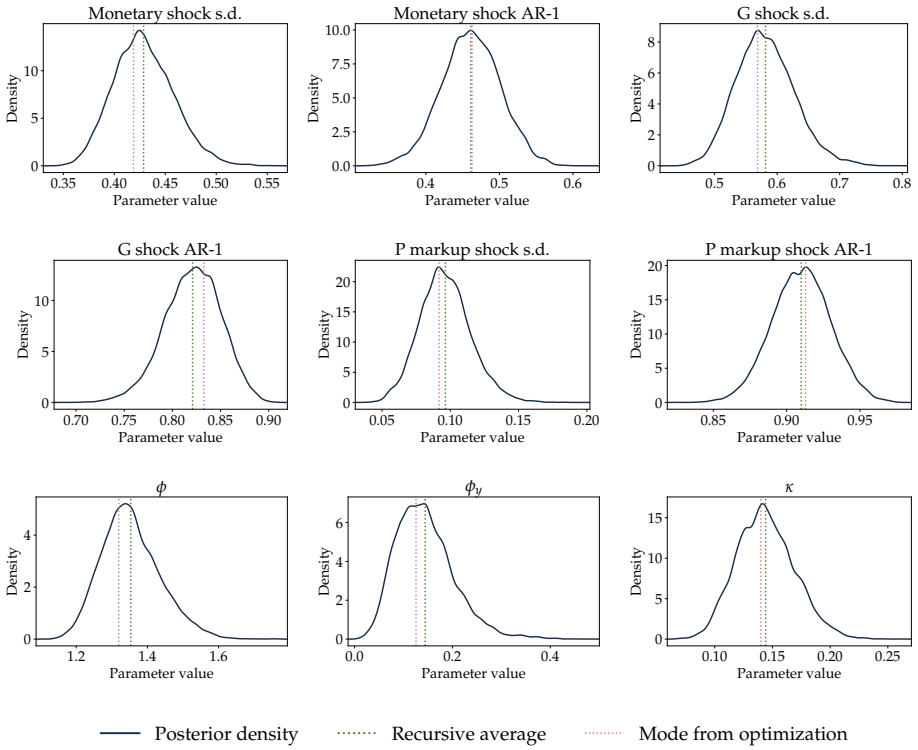


Figure E.7: Recursive means for the RWMH estimation of the two-asset HANK model with shocks

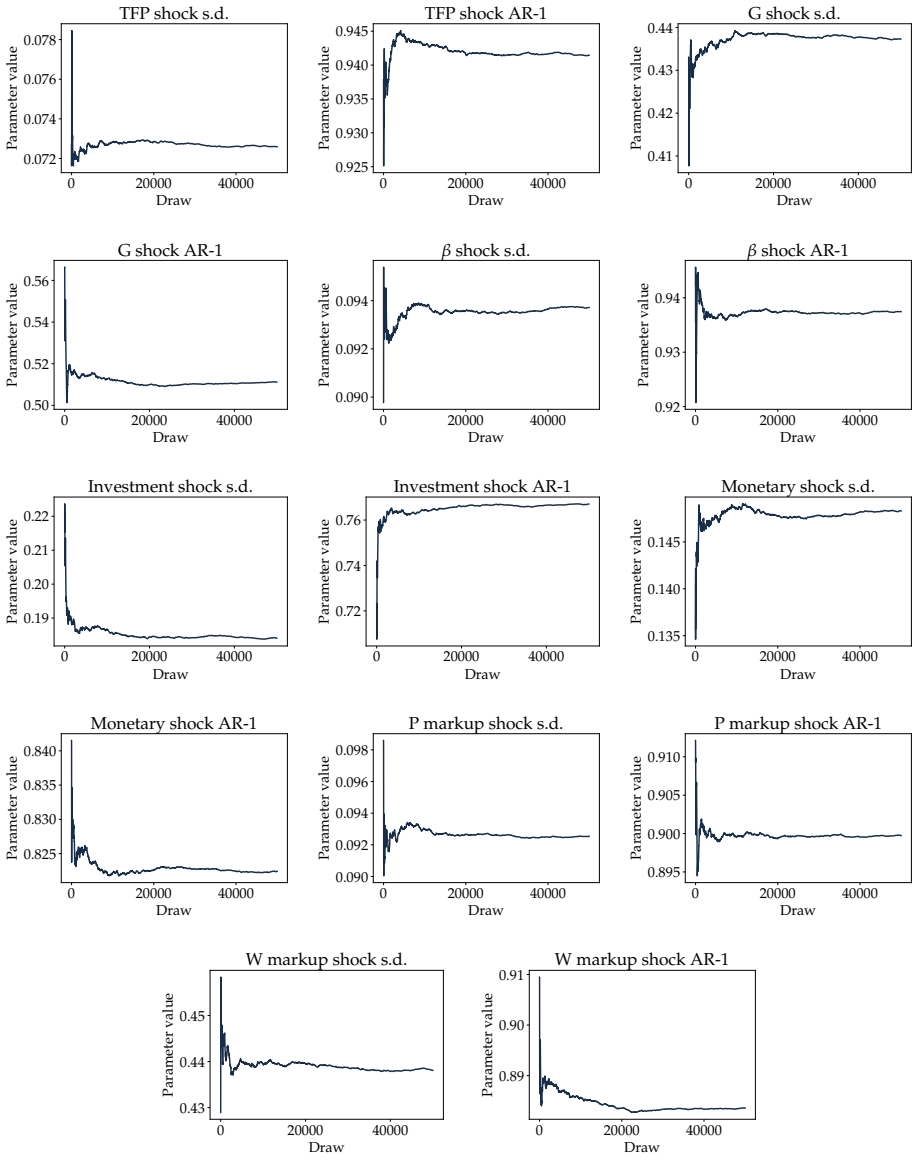


Figure E.8: Posterior distributions for the RWMH estimation of the two-asset HANK model with shocks

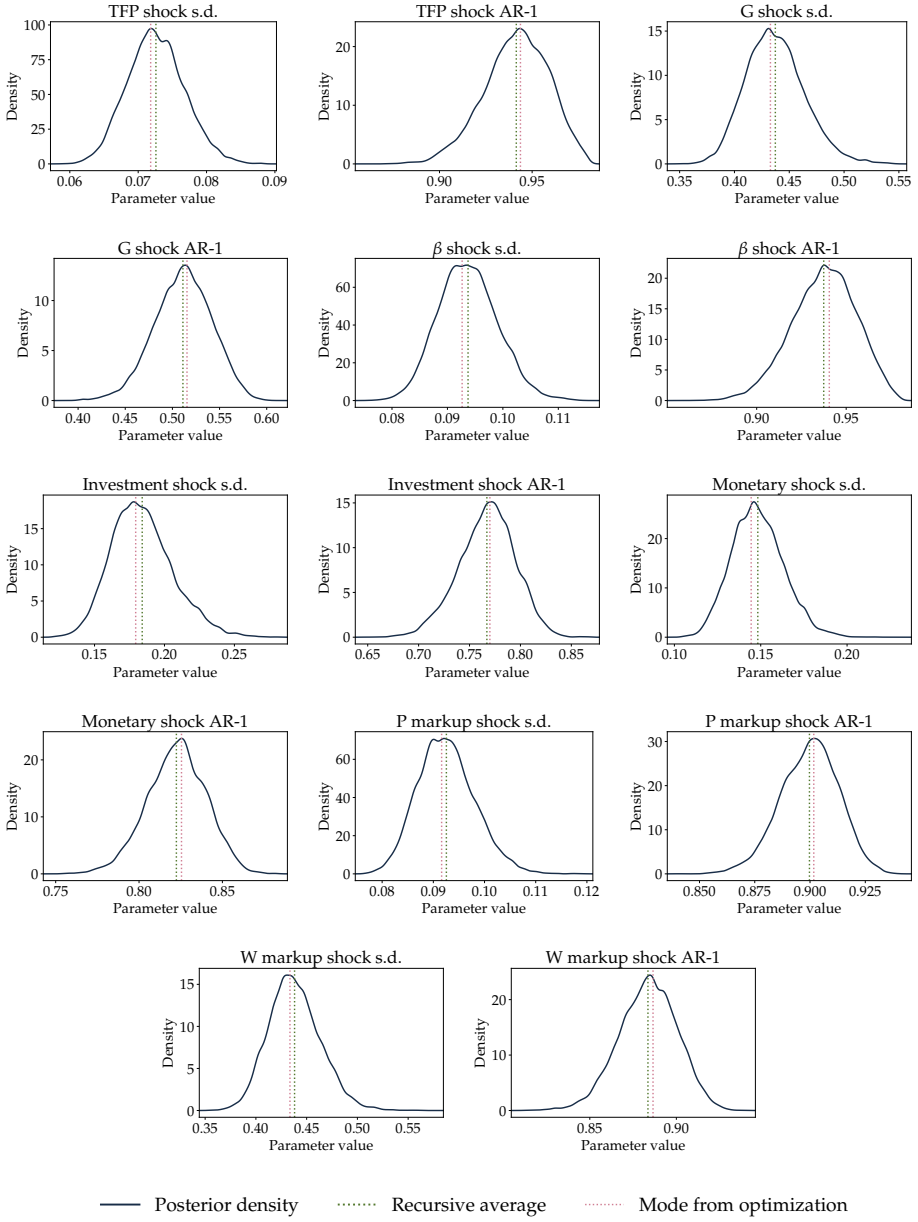


Figure E.9: Recursive means for the RWMH estimation of the two-asset HANK model with shocks and parameters

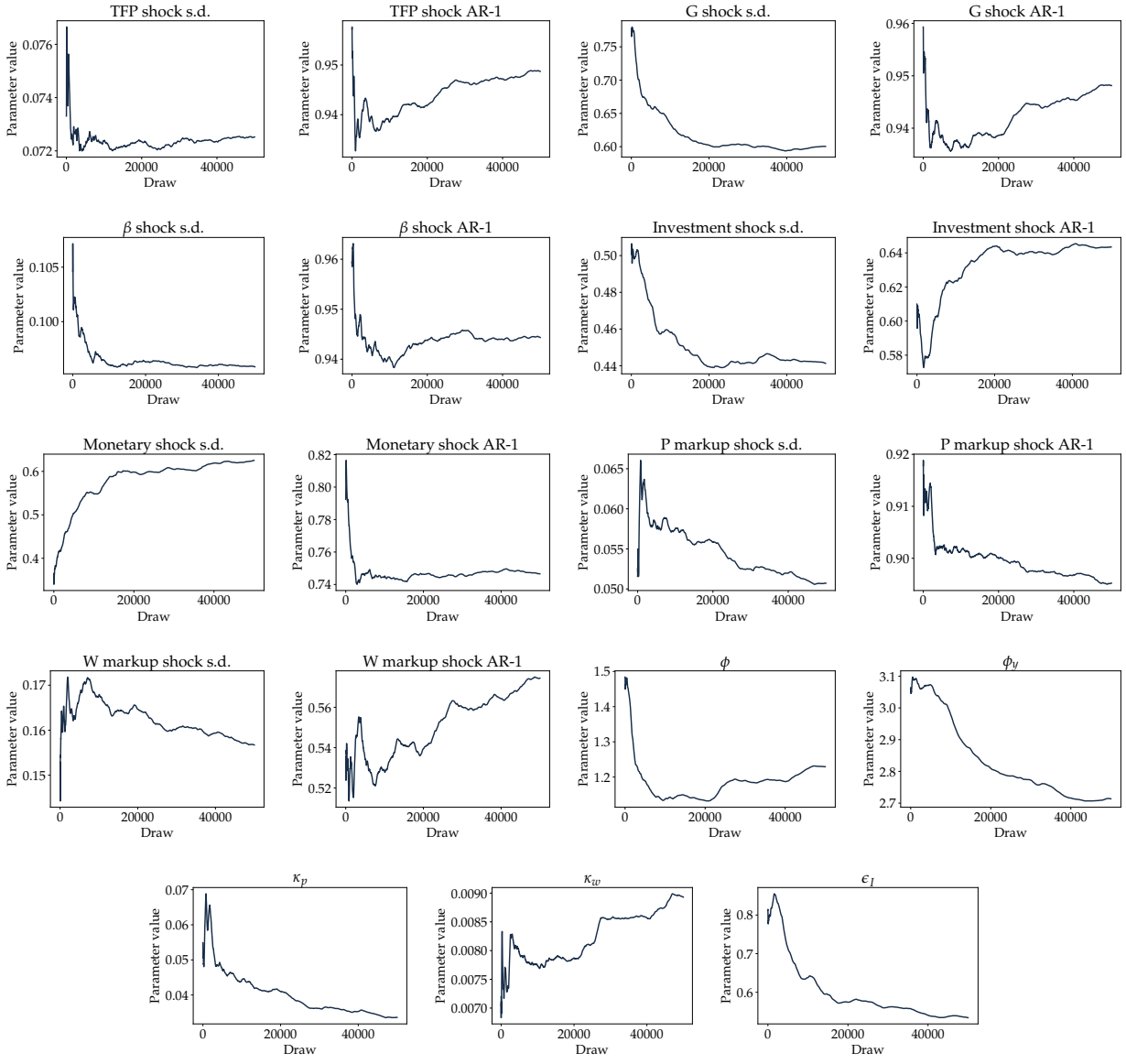


Figure E.10: Posterior distributions for the RWMH estimation of the two-asset HANK model with shocks and parameters