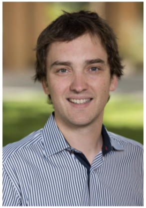
The Intertemporal Keynesian Cross

Adrien Auclert, Matt Rognlie and Ludwig Straub

May 18, 2026

Authors



Adrien Auclert

Stanford

Ludwig Straub

Havard

Matthew Rognlie

Northwestern

Ludwig Straub, Clark Medalist 2026

Ludwig Straub is a macroeconomist who has made fundamental contributions to the incorporation of agent heterogeneity into macroeconomic modeling. The issues he has addressed range from the conceptual to the computational, always with applied questions in mind, such as the impact of non-homothetic savings patterns on interest rates and the efficacy of monetary policy, or the tracing of decentralized Keynesian multipliers through a disaggregated multi-sector economy, or the efficient computation of aggregate responses to economic shocks with endogenously changing income distributions. Straub has also made contributions to spatial economics, public finance, and to the study of information and uncertainty in economic fluctuations, but this citation singles out his work on agent heterogeneity as the contribution that best establishes his research identity.

Several of Straub's most influential papers study the relationship between heterogeneity and aggregate demand. In "Indebted Demand" (*Quarterly Journal of Economics*, 2021, with Atif Mian and Amir Sufi), Straub develops a tractable framework in which non-homothetic preferences and unequal balance sheets link household debt, inequality, and the real interest rate. Over the past 40 years, there has been a rise in debt and decline in the real interest rate in advanced economies, coupled with rising income inequality and financial deregulation. To explain this, Straub and co-authors develop a model with non-homothetic consumption-saving behavior, in which debt is held largely by the relatively poor. In a high-debt regime, funds flow from high-consumption debtors to high-savings creditors, which depresses aggregate demand and, in turn, the rate of interest, pointing to a self-reinforcing low-interest high-debt regime. Low rates of interest encourage borrowing and indebtedness with such debt being mainly held by the relatively poor.

Sequence-Space Jacobian (SSJ)

CI no status

SSJ is a toolkit for analyzing dynamic macroeconomic models with (or without) rich microeconomic heterogeneity.

The conceptual framework is based on our paper Adrien Auclert, Bence Bardóczy, Matthew Rognlie, Ludwig Straub (2021), Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models, Econometrica 89(5), pp. 2375–2408 [ungated copy].

Requirements and installation

SSJ runs on Python 3.7 or newer, and requires Python's core numerical libraries (NumPy, SciPy, Numba). We recommend that you first install the latest Anaconda distribution. This includes all of the packages and tools that you will need to run our code.

To install SSJ, open a terminal and type

```
pip install sequence-jacobian
```

Optional package: There is an optional interface for plotting the directed acyclic graph (DAG) representation of models, which requires Graphviz for Python. With Anaconda, you can install this by typing `conda install -c conda-forge python-graphviz`.

https://github.com/shade-econ/sequence-jacobian

Keynesian Cross

Q: **How does fiscal policy affect aggregate economic activity?**

Textbook answer: the **Keynesian cross**

$$dY_t = dG_t - mpc \cdot dT_t + mpc \cdot dY_t$$

- However: underlying "consumption function" not consistent with theory or data

This paper's answer: **the intertemporal Keynesian cross**

$$d\mathbf{Y} = d\mathbf{G} - \mathbf{M} \cdot d\mathbf{T} + \mathbf{M} \cdot d\mathbf{Y}$$

where $\mathbf{M} \equiv [M_{t,s}]$ is the infinite matrix of partial derivatives $M_{t,s} \equiv \partial \mathcal{C}_t / \partial Y_s$

- consistent with microfoundations and micro consumption data
- sufficient statistics now: **"intertemporal MPCs"** (iMPCs)

Setup

- GE, discrete $t = 0 \cdots \infty$. Small aggregate MIT shocks ($\Leftrightarrow$ 1st order perturbation)
- Mass 1 of households:
  - idiosyncratic shocks to skills e_{it}
  - real pre-tax labor income $y_{it} \equiv W_t/(P_t e_{it} n_{it})$
  - **after tax income** $z_{it} \equiv y_{it} - T_t(y_{it}) \equiv \tau_t y_{it}^{1-\theta}$
- Government sets:
  - **tax revenue** $T_t = \int (y_{it} - z_{it}) di$
  - **government spending** G_t
  - monetary policy: fixed real rate $= r$
- Supply side:
  - linear production function $Y_t = N_t$
  - flexible prices $\Rightarrow P_t = W_t$
  - sticky $W_t \Rightarrow \pi_t^w = \kappa^w \int N_t(v'(n_{it}) - \frac{\partial z_{it}}{\partial n_{it}} u'(c_{it}))di + \beta \pi_{t+1}^w$

Nested models of consumption

Household i solves:

$$\max \ \mathbb{E}\left[\sum_{t\geq 0} \beta^t \left\{u(c_{it}) - v(n_{it})\right\}\right]$$

$$c_{it} + a_{it}^{liq} = z_{it} + (1+r)(1-\zeta)a_{it-1}^{liq} - d_{it} \cdot \mathbf{1}_{\{adj_{it}=1\}}$$

$$a_{it}^{illiq} = (1+r)a_{it-1}^{illiq} + d_{it} \cdot \mathbf{1}_{\{adj_{it}=1\}}$$

- RA: no risk in e (or complete markets)
- TA: share μ of agents with $c_{it} = z_{it}$
- HA-one: one account model, constraint $a_{it} \geq 0$
- HA-two: two account model w. $\Pr(\mathbf{1}_{\{adj_{it}=1\}}) = \nu < 1$, $\zeta > 0$, $a_{it}^{illiq} \geq 0$, $a_{it}^{liq} \geq 0$

Nested models of consumption

Household i solves:

$$\max \ \mathbb{E}\left[\sum_{t\geq 0} \beta^t \left\{u(c_{it}) - v(n_{it}) + \chi(a_{it})\right\}\right]$$

$$c_{it} + a_{it} = z_{it} + (1+r)a_{it-1}$$

- RA: no risk in e (or complete markets)
- TA: share μ of agents with $c_{it} = z_{it}$
- HA-one: one account model, constraint $a_{it} \geq 0$
- HA-two: two account model w. $\Pr(\mathbf{1}_{\{adj_{it}=1\}}) = \nu < 1$, $\zeta > 0$, $a_{it}^{illiq} \geq 0$, $a_{it}^{liq} \geq 0$
- BU/WU: no risk in e, extra additive term $\chi(a_{it})$ in utility

The aggregate consumption function

- Given $\{a_{i,-1}\}$ and r, aggregate consumption function is
$$C_t = \int c_{it} di = \mathcal{C}_t(\{Z_s\}_{s=0}^{\infty})$$
with $Z_t \equiv$ aggregate after-tax labor income
$$Z_t \equiv \int z_{it} di = Y_t - T_t$$
- $\mathcal{C}$ summarizes the heterogeneity and market structure
- Equilibrium defined as usual

Intertemporal MPCs

- Consider $\{G_t, T_t\}$ such that $\sum_{t=0}^{\infty}(1+r)^{-t}(G_t - T_t) = 0$
- An output path $\{Y_t\}_{t=0}^{\infty}$ is part of equilibrium $\Leftrightarrow$

$$Y_t = G_t + \mathcal{C}_t(\{Y_s - T_s\}) \quad \forall t \geq 0$$

- Impulse response to shock $\{dG_t, dT_t\}$ around steady state

$$dY_t = dG_t + \sum_{s=0}^{\infty} \underbrace{\frac{\partial \mathcal{C}_t}{\partial Z_s}}_{\equiv M_{t,s}} \cdot (dY_s - dT_s)$$

- Response $\{dY_t\}$ entirely characterized by $\{M_{ts}\}$: **iMPCs out of income**
  - this is a **sequence-space Jacobian**, made of *partial equilibrium* derivatives
  - how much of income change at date s is spent at date t
  - agent budget constraints $\Rightarrow \sum_{t=0}^{\infty}(1+r)^{s-t}M_{ts} = 1$

The intertemporal Keynesian cross

- Stack objects: $\mathbf{M} \equiv [M_{ts}] = [\frac{\partial \mathcal{C}_t}{\partial Z_s}]$, $d\mathbf{Y} \equiv \{dY_t\}$, $d\mathbf{G} \equiv \{dG_t\}$, $d\mathbf{T} \equiv \{dT_t\}$
  - $d\mathbf{Y}, d\mathbf{G}, d\mathbf{T}$ are sequences in ℓ^∞, $\mathbf{M}$ is bounded linear operator on ℓ^∞
  - $\mathbf{M}$ satisfies $\mathbf{q}'\mathbf{M} = \mathbf{q}'$ where $\mathbf{q}' \equiv \begin{Bmatrix} 1 & \frac{1}{1+r} & (\frac{1}{1+r})^2 \cdots \end{Bmatrix}$ is the PDV sequence

Proposition

Consider a shock $d\mathbf{G}, d\mathbf{T}$ s.t. $\mathbf{q}'d\mathbf{G} = \mathbf{q}'d\mathbf{T}$. Any output response $d\mathbf{Y}$ mustsatisfy

$$d\mathbf{Y} = d\mathbf{G} - \mathbf{M} \cdot d\mathbf{T} + \mathbf{M} \cdot d\mathbf{Y}$$

- This is an **intertemporal Keynesian cross**
  - entire complexity of model is in $\mathbf{M}$
  - with $\mathbf{M}$ from data, could get $d\mathbf{Y}$ without model

Solving the intertemporal Keynesian cross

- Rewrite:
$$(\mathbf{I}-\mathbf{M})d\mathbf{Y} = d\mathbf{G} - \mathbf{M}d\mathbf{T}$$
problem: $\mathbf{I}-\mathbf{M}$ is never invertible
- But, let $\mathbf{K} = -\sum_{t=1}^{\infty}(1+r)^{-t}\mathbf{F}^t$ where $\mathbf{F}$ is the forward operator. Multiplying:
$$\mathbf{K}(\mathbf{I}-\mathbf{M})d\mathbf{Y} = \mathbf{K}(d\mathbf{G} - \mathbf{M}d\mathbf{T})$$
now, $\mathbf{K}(\mathbf{I}-\mathbf{M})$ may be invertible

Proposition

There exists a unique solution to (IKC) for any $d\mathbf{G}, d\mathbf{T}$ *satisfying* $\mathbf{q}'d\mathbf{G} = \mathbf{q}'d\mathbf{T}$, *iff* $\mathbf{K}(\mathbf{I}-\mathbf{M})$ *is invertible. Then, the solution is:*
$$d\mathbf{Y} = \mathcal{M}(d\mathbf{G} - \mathbf{M}d\mathbf{T})$$
where $\mathcal{M} = (\mathbf{K}(\mathbf{I}-\mathbf{M}))^{-1}\mathbf{K}$ is a bounded linear operator ("multiplier")

Baseline model takeaway

- **iMPCs M** $= \left\{ \frac{\partial \mathcal{C}_t}{\partial Z_s} \right\}$ are all that matters for macro response to fiscal policy
- RA, TA, HA, tractable differ in their **M** matrices

Measuring aggregate iMPCs using individual iMPCs

- Object of interest: **(aggregate) iMPCs**

$$M_{ts} = \frac{\partial \mathcal{C}_t}{\partial Z_s}$$

where $C_t = \int c_{it} di$ and $Z_s = \int z_{is} di$

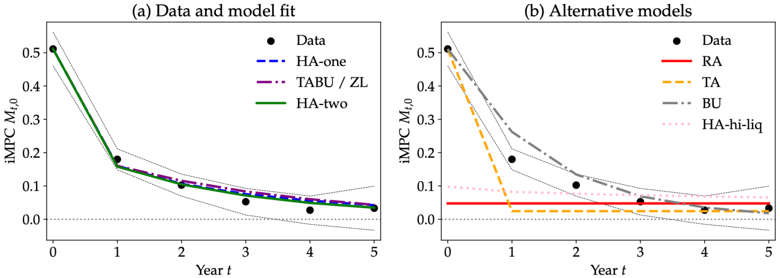
- Direct evidence on M_{ts} is hard to come by for general s
- More work on column $s = 0$ (unanticipated income shock)

$$M_{t0} = \int \underbrace{\frac{z_{i0}}{Z_0}}_{income\ weight} \cdot \underbrace{\frac{\partial \mathbb{E}_0[c_{it}]}{\partial z_{i0}}}_{individual\ iMPC} di$$

$\rightarrow$ aggregate iMPCs are weighted individual iMPCs

iMPCs across models

Figure 2: iMPCs in the Norwegian data and in several models



Notes: All models are calibrated to match $r = 0.05$. RA does not have any other free parameter. The single free parameter in BU (λ), TA (μ), HA-one (A/Z) and HA-two (ν) is calibrated to match $M_{00} = 0.51$. The additional free parameter in TABU and ZL (μ) is calibrated to match $M_{10} = 0.16$ (its value in the HA-one model). The HA-two and HA-hi-liq models are calibrated to an aggregate ratio of assets to post-tax income of $A/Z = 6.29$, its value in the model with capital in section 7.

Simulate model responses

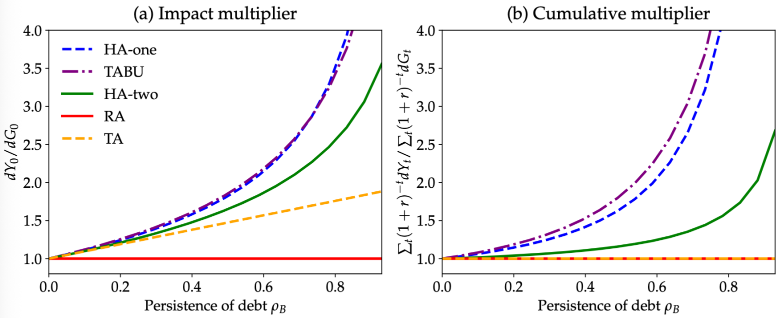
Under balanced-budget policy, $d\mathbf{G} = d\mathbf{T}$, the fiscal multiplier is 1 at all dates:

$$d\mathbf{Y} = d\mathbf{G}$$

iMPC is irrelevant.

When $d\mathbf{G} \neq d\mathbf{T}$ (deficit-financed):

Figure 5: Multipliers according to the IKC



Sequence Space v.s State Space

Definition

A state-space representation describes the economy recursively. At date t, a state variable x_t summarizes all payoff-relevant history, and the model evolves as

$$x_{t+1} = f(x_t, \varepsilon_{t+1}), \qquad y_t = g(x_t).$$

Given x_t, the future is independent of the past.

Definition

A sequence-space representation characterizes equilibrium as a mapping between entire time paths. We seek sequences $\{X_t\}_{t=0}^{\infty}$ such that equilibrium conditions hold at every date:

$$F(\{X_t\}_{t=0}^{\infty}, \{Z_t\}_{t=0}^{\infty}) = \mathbf{0},$$

where $\{X_t\}$ collects endogenous variables and $\{Z_t\}$ collects exogenous shocks or policy paths.

SSJ

- Start from model in nonlinear sequence space: for all t
$$H(\mathbf{U}, \mathbf{Z}) = 0$$
$\mathbf{U} \in \mathbb{R}^{n_u T}$ paths of endog vars, $\mathbf{Z} \in \mathbb{R}^{n_z T}$ paths of exog vars
- **Linearized dynamics** around steady state
$$d\mathbf{U} = \underbrace{-\mathbf{H}_U^{-1}\mathbf{H}_\mathbf{Z}}_{Jacobians} d\mathbf{Z}$$
- **Estimation**: impulse responses = MA(∞) representation
- **Local determinacy**: test singularity of $\mathbf{H}_U$
- **Nonlinear MIT shocks**: Newton's method with $\mathbf{H}_U$

Example: The Krusell-Smith Model

Household Problem

Households: ability e, capital k, inelsatic labor supply $\equiv 1$:

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

- Denote by $c_t^*(e, k_-)$ and $k_t^*(e, k_-)$ the policy functions that solve the Bellman equation.
- Also denote by $D_t(e, K_-)$ the measure of households in state e that own capital in a set K_- at the start of date t.

Example: The Krusell-Smith Model

Household Problem

The distribution D_t has law of motion

$$D_{t+1}(e', K) = D_t(e, k_t^{*-1}(e, K))P(e, e').$$

where $k_t^{*-1}(e, K)$ denotes the inverse of $k_t^*(e, K)$.

- We assume that prior to $t = 0$, the economy is in a steady state.
- We suppose that agents start in this stationary distribution at date 0, so that $D_0 = D_{ss}$.
- The aggregate household capital holdings are characterized by a capital function $\mathcal{K}_t(\{r_s, w_s\}_{s=0}^{\infty})$.

$$\mathcal{K}_t\left(\{r_s, w_s\}_{s=0}^{\infty}\right) = \sum_e \int_{k_-} k_t^*(e, k_-)D_t(e, dk_-)$$

Example: The Krusell-Smith Model

Reduce to $H(\mathbf{K}, \mathbf{Z})$

- Competitive firms, Cobb-Douglas production:
  - $r_t = \alpha Z_t K_{t-1}^{\alpha-1} N_t^{1-\alpha} - \delta$
  - $w_t = (1-\alpha) Z_t K_{t-1}^{\alpha} N_t^{-\alpha}$
- Market clearing (goods redundant):
  - $N_t = 1$
  - $K_t = \mathcal{K}(\{r_s, w_s\})$
- Write
$$H_t(\mathbf{K}, \mathbf{Z}) \equiv \mathcal{K}_t(\{\alpha Z_s K_{s-1}^{\alpha-1} - \delta, (1-\alpha) Z_s K_{s-1}^{\alpha}\}) - K_t = 0$$
- First-order solution is $d\mathbf{K} = -\mathbf{H}_K^{-1} \mathbf{H}_Z d\mathbf{Z}$ where e.g.
$$[\mathbf{H}_K]_{t,s} = \frac{\partial \mathcal{K}_t}{\partial r_{s+1}} \frac{\partial r_{s+1}}{\partial K_s} + \frac{\partial \mathcal{K}_t}{\partial w_{s+1}} \frac{\partial w_{s+1}}{\partial K_s} - \mathbf{1}_{\{s=t\}}$$

Example: The Krusell-Smith Model

Constructing the Jacobian

Let

$$\mathcal{J}^{\mathcal{K},r}_{t,s} = \frac{\partial \mathcal{K}_t}{\partial r_s}, \quad \mathcal{J}^{\mathcal{K},w}_{t,s} = \frac{\partial \mathcal{K}_t}{\partial w_s}$$

be the Jacobians up to some truncation horizon T.

Sufficient statistics

The Jacobians $\mathcal{J}^{\mathcal{K},r}$ and $\mathcal{J}^{\mathcal{K},w}$ are sufficient statistics for the household in sequence space equilibrium:

- Shocks propagate through the dynamics of the household capital distribution, but *we do not need those details if we have the Jacobians.*
- Column s of $\mathcal{J}^{\mathcal{K},r}$ is the impulse response of capital to a news shock dr_s; the Jacobian gives these for all $s = 0, \ldots, T-1$.
- These are rich objects, but *there is a fast new algorithm to compute them.*

Fake News Algorithm

General Model Representation

We represent a heterogeneous-agent problem as a mapping from a time path of aggregate inputs $\{\boldsymbol{X}_t\}$ to a time path of aggregate outputs $\{\boldsymbol{Y}_t\}$.

Assume:

- $\boldsymbol{X}_t \in \mathbb{R}^{n_x}$: aggregate inputs,
- $\boldsymbol{Y}_t \in \mathbb{R}^{n_y}$: aggregate outputs,
- the distribution is discretized on n_g grid points,
- $\boldsymbol{D}_t \in \mathbb{R}^{n_g}$: cross-sectional distribution of agents at time t.

Aggregate outputs are averages of individual outcomes against the distribution:

$$\boldsymbol{Y}_t = \boldsymbol{y}_t' \boldsymbol{D}_t,$$

where $\boldsymbol{y}_t$ is an $n_g \times n_y$ matrix of individual outcomes (for example, consumption, savings, labor supply, or other individual-level objects).

Fake News Algorithm

General formulation of HA blocks

We assume there exist three functions v, Λ, and y such that, for a given initial distribution $\boldsymbol{D}_0$, the aggregate sequence solves

$$\begin{aligned}\boldsymbol{v}_t &= v(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t), \\ \boldsymbol{D}_{t+1} &= \Lambda(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)' \, \boldsymbol{D}_t, \\ \boldsymbol{Y}_t &= y(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)' \boldsymbol{D}_t.\end{aligned}$$

Interpretation:

- $\boldsymbol{v}_t$: backward-looking object, typically the value function;
- $\Lambda(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)$: transition matrix governing the law of motion of the distribution;
- $y(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)$: individual outcomes used to aggregate into $\boldsymbol{Y}_t$.

Fake News Algorithm

Steady State and Transition Paths

For a constant aggregate input $\boldsymbol{X}_{ss}$, the steady state is a fixed point

$$(\boldsymbol{Y}_{ss}, \boldsymbol{v}_{ss}, \boldsymbol{D}_{ss})$$

of the system.

For convenience, define

$$\Lambda_{ss} \equiv \Lambda(\boldsymbol{v}_{ss}, \boldsymbol{X}_{ss}), \qquad y_{ss} \equiv y(\boldsymbol{v}_{ss}, \boldsymbol{X}_{ss}).$$

We consider transition paths of length T that end at the steady state:

$$\boldsymbol{X}_{T-1} = \boldsymbol{X}_{ss}, \qquad \boldsymbol{v}_T = \boldsymbol{v}_{ss}.$$

The initial distribution $\boldsymbol{D}_0$ is given and often set equal to $\boldsymbol{D}_{ss}$.

Hence, this framework is designed to study transitory shocks around a steady state.

Fake News Algorithm

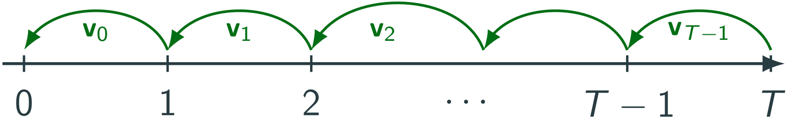
Standard Three-Step Structure

Fake News Algorithm

Standard Three-Step Structure

Step 1: Backward

$$\boldsymbol{v}_t = v(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)$$

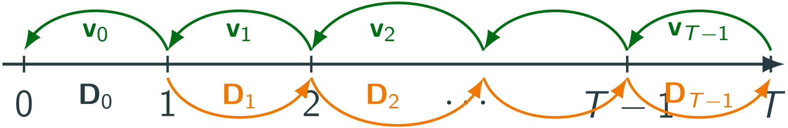


Fake News Algorithm

Standard Three-Step Structure

Step 1: Backward

$$\boldsymbol{v}_t = v(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)$$



Step 2: Forward

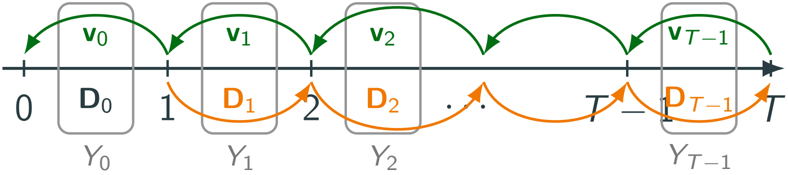
$$\boldsymbol{D}_{t+1} = \Lambda(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)\boldsymbol{D}_t$$

Fake News Algorithm

Standard Three-Step Structure

Step 1: Backward

$$\boldsymbol{v}_t = v(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)$$



Step 2: Forward

$$\boldsymbol{D}_{t+1} = \Lambda(\boldsymbol{v}_{t+1}, \boldsymbol{X}_t)\boldsymbol{D}_t$$

Step 3: Aggregate

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

Fake News Algorithm

Calculating the Jacobian: direct vs. new algorithm

- The direct algorithm requires a separate backward and forward iteration for each $s = 0, \ldots, T-1$; this is costly.
- SSJ method uses a single backward and forward iteration to get all T columns.
- *This improves speed by a factor of roughly T with no loss in accuracy.*
- The key is to exploit time symmetries in the linearized system.

The linearized system:

$$
\begin{aligned}
d\boldsymbol{Y}_t &= d\boldsymbol{y}_t'\boldsymbol{D}_{ss} + \boldsymbol{y}_{ss}'\, d\boldsymbol{D}_t \\
d\boldsymbol{D}_{t+1} &= d\Lambda_t'\boldsymbol{D}_{ss} + \Lambda_{ss}' d\boldsymbol{D}_t \\
\boldsymbol{D}_0 &= \boldsymbol{D}_{ss}
\end{aligned}
$$

Since everything is linear, we can look at terms from backward and forward iterations separately, then combine them.

Fake News Algorithm

Terms from backward iteration

Holding $\boldsymbol{D}_t = \boldsymbol{D}_{ss}$ fixed, what is the effect of $d\boldsymbol{X}_s$?

$$d\boldsymbol{Y}_t = d\boldsymbol{y}_t^0 \boldsymbol{D}_{ss} + \cancel{\boldsymbol{y}_{ss}' d\boldsymbol{D}_t}$$

$$d\boldsymbol{D}_{t+1} = d\Lambda_t^0 \boldsymbol{D}_{ss} + \cancel{\Lambda_{ss}' d\boldsymbol{D}_t}$$

- $d\boldsymbol{y}_t^0$ and $d\Lambda_t^0$ are purely forward-looking from the Bellman equation.
- They depend only on the time until the shock, $s - t$.

Define:

$$\mathcal{Y}_{s-t} \equiv \frac{d\boldsymbol{y}_t^0 \boldsymbol{D}_{ss}}{d\boldsymbol{X}_s} \qquad \mathcal{D}_{s-t} \equiv \frac{d\Lambda_t^0 \boldsymbol{D}_{ss}}{d\boldsymbol{X}_s}$$

Use a single backward iteration with a shock $d\boldsymbol{X}_s$ at $s = T - 1$ to obtain $d\boldsymbol{y}_t^0$ and $d\Lambda_t^0$ for all $t = T - 1, \ldots, 0$.

Fake News Algorithm

Terms from forward iteration

Holding $\boldsymbol{y}_t = \boldsymbol{y}_{ss}$ and $\Lambda_t = \Lambda_{ss}$ fixed, what is the effect of $d\boldsymbol{D}_t$?

$$d\boldsymbol{Y}_t = \cancel{d\boldsymbol{y}_t^0 \boldsymbol{D}_{ss}} + \boldsymbol{y}_{ss}'\, d\boldsymbol{D}_t$$
$$d\boldsymbol{D}_{t+1} = \cancel{d\Lambda_t^0 \boldsymbol{D}_{ss}} + \Lambda_{ss}'\, d\boldsymbol{D}_t$$

Since everything's linear, some linear functional (row vector) maps $d\boldsymbol{D}_t$ to $d\boldsymbol{Y}_t$: what is it?

- Can write for $s \le t$ (zero for $s > t$):
$$d\boldsymbol{Y}_t = \underbrace{\boldsymbol{y}_{ss}'(\Lambda_{ss}')^{t-s}}_{\equiv \mathcal{P}_{t-s}'} d\boldsymbol{D}_s$$
- These $\mathcal{P}_u$ can be calculated recursively as:
$$\mathcal{P}_u = \Lambda_{ss}\mathcal{P}_{u-1}$$
- Thus, a single "transpose" forward iteration gives all $\mathcal{P}_u$.

Fake News Algorithm

Putting it together: the Jacobian

For shock $d\boldsymbol{X}_s$, we want:

$$d\boldsymbol{Y}_t = d\boldsymbol{y}_t'\boldsymbol{D}_{ss} + \boldsymbol{y}_{ss}'\, d\boldsymbol{D}_t$$

Change in policy, holding distribution constant:

$$d\boldsymbol{y}_t'\boldsymbol{D}_{ss} = \mathcal{Y}_{s-t}\, d\boldsymbol{X}_s$$

Change in distribution, holding policy constant:

$$\boldsymbol{y}_{ss}'\, d\boldsymbol{D}_t = (\mathcal{P}_0'\mathcal{D}_{s-t+1} + \ldots + \mathcal{P}_{t-1}'\mathcal{D}_s)\, d\boldsymbol{X}_s$$

sum of shocks $\mathcal{D}$ to distribution from anticipating $d\boldsymbol{X}_s$ before t, propagated forward to effect on $d\boldsymbol{Y}_t$ with $\mathcal{P}$

The full Jacobian is then:

$$\mathcal{J}_{t,s} = \mathcal{Y}_{s-t} + (\mathcal{P}_0'\mathcal{D}_{s-t+1} + \ldots + \mathcal{P}_{t-1}'\mathcal{D}_s)$$

Fake News Algorithm

Fake news matrix

Define the fake news matrix as:

$$\mathcal{F}_{t,s} \equiv \begin{cases} \mathcal{Y}_s & \text{if } t = 0, \\ \mathcal{P}'_{t-1}\mathcal{D}_s & \text{if } t > 0. \end{cases}$$

- **Interpretation:**
  - $t = 0$: agents get news of shock at date $s \geq 0$
  - $t \geq 1$: news of unrealized shock disappears ("fake news")
- Build Jacobian (response to news shocks) recursively from fake news matrix (response to fake news shocks) as:

$$\mathcal{J}_{t,s} \equiv \begin{cases} \mathcal{F}_{t,s} & \text{if } t = 0 \text{ or } s = 0, \\ \mathcal{F}_{t,s} + \mathcal{J}_{t-1,s-1} & \text{otherwise} \end{cases}$$

Fake News Algorithm

Algorithm

To obtain Jacobians $\mathcal{J}^{o,i}$ for many inputs dX^i and outputs dY^o:

1. For each input i, perform backward iteration with T steps to get all $\dagger_u^{o,i}$ and $\mathcal{D}_u^i$.
2. For each output o, perform transpose forward iteration with $T-1$ steps to get all $\mathcal{P}_u^o$.
3. For each pair (o,i), construct the fake news matrix $\mathcal{F}^{o,i}$ from $\dagger^{o,i}$ in the first row and the product $(\mathcal{P}^o)'\mathcal{D}^i$ for other rows.
4. For each pair (o,i), recursively compute $\mathcal{J}^{o,i}$ from $\mathcal{F}^{o,i}$.

Step 1 is almost always the bottleneck.

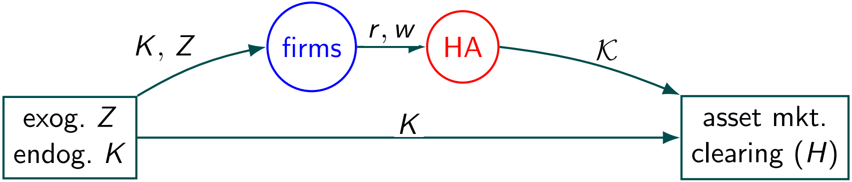
DAG

Krusell-Smith model as a directed acyclic graph (DAG)

Recall that we wrote equilibrium in KS model as system

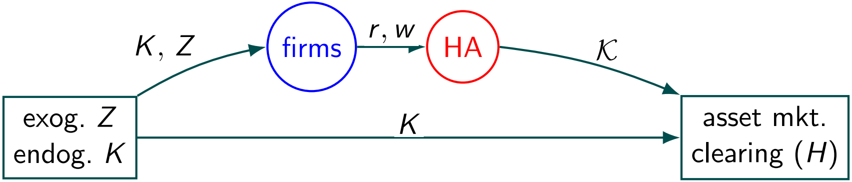
$$H_t(\boldsymbol{K},\boldsymbol{Z}) \equiv \mathcal{K}_t\left(\left\{\underbrace{\alpha Z_s K_{s-1}^{\alpha-1}-\delta}_{r_s}, \underbrace{(1-\alpha)Z_s K_{s-1}^{\alpha}}_{w_s}\right\}\right) - K_t = 0$$

This corresponds to a simple graph:



DAG

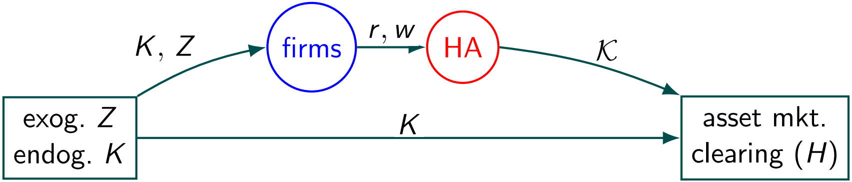
DAGs combining simple and HA blocks



- Each node ("block") takes sequences as inputs and outputs.
  - Inputs of later blocks are outputs of earlier blocks.
  - The number of endogenous variables ("unknowns") equals the number of equations in H ("targets"); solve to get equilibrium.
- Two kinds of blocks in SHADE models:
  - Simple blocks: Analytical equations directly in aggregates, e.g. $r_t = \alpha Z_t K_{t-1}^{\alpha-1} - \delta$; Jacobians are sparse and easy to obtain.
  - HA blocks: As described previously.

DAG

Solving the model with DAG



- Define $\mathbf{J}^{o,i}$ to be the total derivative of o along the DAG with respect to the exogenous or endogenous variable i (distinct from the block Jacobian $\mathcal{J}^{o,i}$).
- Here, for example, $\mathbf{J}^{r,K}$ equals $\mathcal{J}^{r,K}$, etc.: the direct effect is the only effect.
- Then we need the chain rule. For instance:

$$\mathbf{J}^{\mathcal{K},K} = \mathcal{J}^{\mathcal{K},r}\mathbf{J}^{r,K} + \mathcal{J}^{\mathcal{K},w}\mathbf{J}^{w,K}$$

- Finally, we obtain $\mathbf{H}_K$ and $\mathbf{H}_Z$; then $\mathbf{G}^{K,Z} \equiv -\mathbf{H}_K^{-1}\mathbf{H}_Z$ gives the general equilibrium map from Z to K, and all $\mathbf{G}^{o,Z}$ are determined from it.

DAG

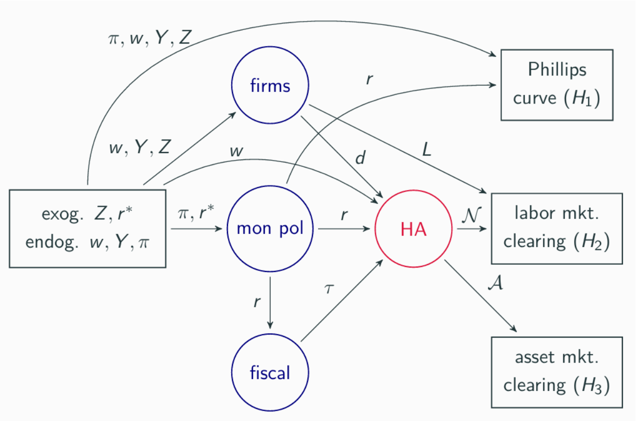
General case: solving with forward accumulation

- Initialize $\mathbf{J}^{i,i}$ as the identity for all exogenous or unknown i.
- Evaluate the chain rule following a topological sort:
$$\mathbf{J}^{o,i} = \sum_{m \in I_b} \mathcal{J}^{o,m} \mathbf{J}^{m,i}$$
on blocks b to get $\mathbf{H}_U = \mathbf{J}^{H,U}$ and $\mathbf{H}_Z = \mathbf{J}^{H,Z}$.
- This is called "forward accumulation" in the algorithmic differentiation literature.
- Compute the general equilibrium mapping: $\mathbf{G}^{U,Z} = -\mathbf{H}_U^{-1}\mathbf{H}_Z$
- Then do forward accumulation again:
$$\mathbf{G}^{o,Z} = \sum_{m \in I_b} \mathcal{J}^{o,m} \mathbf{G}^{m,Z}$$
to get general equilibrium mappings $\mathbf{G}^{o,Z}$ from shocks dZ to all variables o of interest.
- This gives impulse responses to every shock simultaneously.

DAG

One-asset HANK model with endogenous labor as DAG

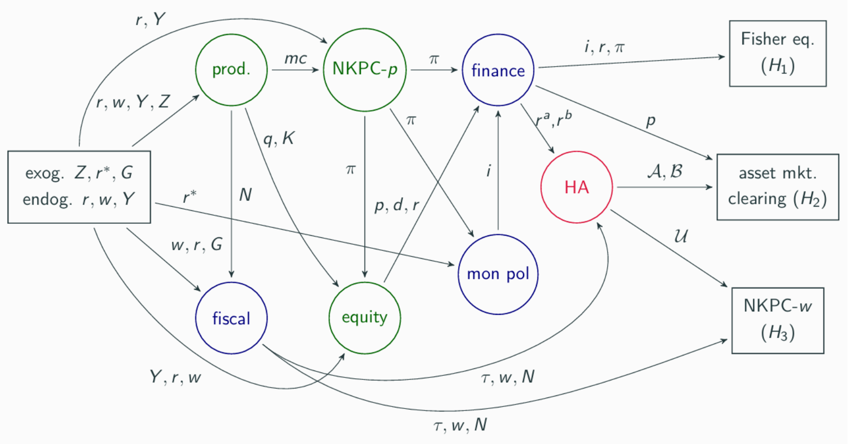
8 endog vars $\to$ 3 unknowns in DAG



DAG

Two-asset HANK model with capital and sticky prices/wages

22 endog vars → 3 unknowns in DAG



Krusell-Smith with SSJ in Python

Set up HA block

```
@het(exogenous='Pi', policy='a', backward='Va', backward_init=household_init)
def household(Va_p, a_grid, e_grid, r, w, beta, eis):
    uc_nextgrid = beta * Va_p
    c_nextgrid = uc_nextgrid ** (-eis)
    coh = (1 + r) * a_grid[np.newaxis, :] + w * e_grid[:, np.newaxis]
    a = interpolate.interpolate_y(c_nextgrid + a_grid, coh, a_grid)
    misc.setmin(a, a_grid[0])
    c = coh - a
    Va = (1 + r) * c ** (-1 / eis)
    return Va, a, c

@simple
def firm(K, L, Z, alpha, delta):
    r = alpha * Z * (K(-1) / L) ** (alpha-1) - delta
    w = (1 - alpha) * Z * (K(-1) / L) ** alpha
    Y = Z * K(-1) ** alpha * L ** (1 - alpha)
    return r, w, Y

@simple
def mkt_clearing(K, A, Y, C, delta):
    asset_mkt = A - K
    goods_mkt = Y - C - delta * K
    return asset_mkt, goods_mkt

ks = create_model([household_ext, firm, mkt_clearing], name="Krusell-Smith")
```

Krusell-Smith with SSJ in Python

Calibrating the steady state & Jacobians

```
# --- Steady State -----
calibration = {'eis': 1, 'delta': 0.025, 'alpha': 0.11, 'rho_e': 0.966, 'sd_e': 0.5, 'L': 1.0,
               'nE': 7, 'nA': 500, 'amin': 0, 'amax': 200}
unknowns_ss = {'beta': 0.98, 'Z': 0.85, 'K': 3.}
targets_ss = {'r': 0.01, 'Y': 1., 'asset_mkt': 0.}

ss = ks.solve_steady_state(calibration, unknowns_ss, targets_ss, solver='hybr')

# --- Jacobians -----
# Simple blocks
J_firm = firm.jacobian(ss, inputs=['K', 'Z'])

# HA blocks
J_ha = household.jacobian(ss, inputs=['r', 'w'], T=5)
```

Krusell-Smith with SSJ in Python

Linearized dynamics using Jacobians

```
# --- Compose Jacobians along the DAG -----
J_curlyK_K = J_ha['A']['r'] @ J_firm['r']['K'] + J_ha['A']['w'] @ J_firm['w']['K']
J_curlyK_Z = J_ha['A']['r'] @ J_firm['r']['Z'] + J_ha['A']['w'] @ J_firm['w']['Z']
J = copy.deepcopy(J_firm)
J.update(JacobianDict({'curlyK': {'K' : J_curlyK_K, 'Z' : J_curlyK_Z}}))

# --- Invert H_K to obtain impulse response -----
H_K = J['curlyK']['K'] - np.eye(T)
H_Z = J['curlyK']['Z']
G = {'K': -np.linalg.solve(H_K, H_Z)}

# --- Get all other impulses -----
G['r'] = J['r']['Z'] + J['r']['K'] @ G['K']
G['w'] = J['w']['Z'] + J['w']['K'] @ G['K']
G['Y'] = J['Y']['Z'] + J['Y']['K'] @ G['K']
G['C'] = J_ha['C']['r'] @ G['r'] + J_ha['C']['w'] @ G['w']
```

Convenient approach:

```
inputs = ['Z']
unknowns = ['K']
targets = ['asset_mkt']
G2 = ks.solve_jacobian(ss, unknowns, targets, inputs, T=T)
```

SSJ in Python

Using SSJ: introductory notebooks

To learn how to use the toolkit, it's best to work through our introductory Jupyter notebooks, which show how SSJ can be used to represent and solve various models. We recommend working through the notebooks in the order listed below. Click here to download all notebooks as a zip.

- RBC
  - represent macro models as collections of blocks (DAG)
  - write SimpleBlocks and CombinedBlocks
  - compute linearized and non-linear (perfect-foresight) impulse responses
- Krusell-Smith
  - write HetBlocks to represent heterogeneous agents
  - construct general-equilibrium Jacobians manually
  - compute the log-likelihood of the model given time-series data
- One-asset HANK
  - adapt an off-the-shelf HetBlock to any macro environment using helper functions
  - see a more advanced example of calibration
- Two-asset HANK
  - write SolvedBlocks to represent implicit aggregate equilibrium conditions
  - re-use saved Jacobians
  - fine tune options of block methods
- Labor search
  - example with multiple exogenous states
  - shocks to transition matrix of exogenous states

Download introductory notebooks: `https://github.com/shade-econ/sequence-jacobian/raw/master/notebooks/notebooks.zip`

Reference I

Achdou, Y., Han, J., Lasry, J.-M., Lions, P.-L., and Moll, B. (2022). Income and wealth distribution in macroeconomics: A continuous-time approach. *The review of economic studies*, 89(1):45–86.

Aiyagari, S. R. (1994). Uninsured idiosyncratic risk and aggregate saving. *The Quarterly Journal of Economics*, 109(3):659–684.

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

Auclert, A., Majic, E., Rognlie, M., and Straub, L. (2025a). Determinacy and large-scale solutions in the sequence space. *Journal of Political Economy: Macroeconomics*.

Auclert, A., Rigato, R., Rognlie, M., and Straub, L. (2025b). Beyond certainty equivalence: Second-order dynamics in the sequence space. Presentation slides.

Auclert, A., Rognlie, M., and Straub, L. (2024). The intertemporal keynesian cross. *Journal of Political Economy*, 132(12):4068–4121.

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

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

Moll, B. (2025). The trouble with rational expectations in heterogeneous agent models: A challenge for macroeconomics. *The Economic Journal*, page ueaf104.