When Inequality Matters for Macro and Macro Matters for Inequality

SeHyoun Ahn
Princeton

Greg Kaplan
Chicago

Benjamin Moll
Princeton

Tom Winberry
Chicago

Christian Wolf
Princeton

New York Fed, 29 November 2017
(slides are on my website)

Motivation

- Last 30 years: a lot of progress developing macro models with rich heterogeneity in income, wealth, consumption in micro data
- Promising tool for macroeconomics
  - implications for policies differ: inequality matters for macro
  - distributional implications: macro matters for inequality

Motivation

- Last 30 years: a lot of progress developing macro models with rich heterogeneity in income, wealth, consumption in micro data
- Promising tool for macroeconomics
  - implications for policies differ: inequality matters for macro
  - distributional implications: macro matters for inequality
- Not yet part of policymakers' toolbox. Two excuses:
  1. computational difficulties because distribution endogenous
  2. perception that aggregate dynamics similar to rep agent

Motivation

- Last 30 years: a lot of progress developing macro models with rich heterogeneity in income, wealth, consumption in micro data
- Promising tool for macroeconomics
  - implications for policies differ: inequality matters for macro
  - distributional implications: macro matters for inequality
- Not yet part of policymakers' toolbox. Two excuses:
  1. computational difficulties because distribution endogenous
  2. perception that aggregate dynamics similar to rep agent
- Our paper: these excuses less valid than you thought

These excuses are less valid than you thought

1. Efficient and easy-to-use computational method
   - open source `Matlab` toolbox online now
   - extension of linearization (Campbell 1998, Reiter 2009)
   - different slopes at each point in state space
   - exploit advantages of continuous time (Achdou et al. 2017)
2. Use methodology to illustrate interaction of macro + inequality
   - match micro behavior ⇒ realistic aggregate C + Y dynamics
   - aggregate shocks generate inequality dynamics...
   - ... and IRFs in HA model can differ dramatically from RA case

Outline

1. Explain methods in one-asset (Krusell-Smith) model
    - model description
    - linearization
    - dimensionality reduction
    - illustrative results
    - `https://sehyoun.com/EXAMPLE_PHACT_KS.html`
2. Two applications to illustrate macro + inequality interactions
    - richer two-asset (Kaplan-Moll-Violante) model
3. (Not in paper) a simple one-asset HANK model
    - `https://sehyoun.com/EXAMPLE_one_asset_HANK_web.html`

One-Asset Heterogeneous Agent Model with Aggregate Shocks (Krusell-Smith)

Households

$$\max_{\{c_{jt}\}_{t\geq 0}} \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_{jt}) dt \qquad \text{such that}$$

$$\dot{a}_{jt} = w_t z_{jt} + r_t a_{jt} - c_{jt}$$

$$z_{jt} \in \{z_\ell, z_h\} \text{ Poisson with intensities } \lambda_\ell, \lambda_h$$

$$a_{jt} \geq 0$$

- c_{jt}: consumption
- u: utility function, $u' > 0$, $u'' < 0$.
- ρ: discount rate
- r_t : interest rate

Production

- Aggregate production function

$$Y_t = e^{Z_t} K_t^{\alpha} N_t^{1-\alpha} \text{ with } dZ_t = -\nu Z_t dt + \sigma dW_t$$

- Perfect competition in factor markets

$$w_t = (1-\alpha)\frac{Y_t}{N_t}, \qquad r_t = \alpha \frac{Y_t}{K_t} - \delta$$

- Market clearing

$$K_t = \int a g_t(a,z) da dz,$$
$$N_t = \int z g_t(a,z) da dz \equiv 1$$

Warm-Up: Stationary Eq without Aggregate Shocks

- This slide only: turn off aggregate shocks $Z_t \equiv 0$

$$\begin{aligned} \rho v(a,z) = &\max_c \; u(c) + \partial_a v(a,z)(wz + ra - c) \\ &+ \lambda_z (v(a,z') - v(a,z)) \end{aligned} \tag{HJB SS}$$

$$0 = -\partial_a[s(a,z)g(a,z)] - \lambda_z g(a,z) + \lambda_{z'} g(a,z') \tag{KF SS}$$

$$\begin{aligned} w &= (1-\alpha)K^\alpha, \quad r = \alpha K^{\alpha-1} - \delta, \\ K &= \int a g(a,z) da dz \end{aligned} \tag{PRICE SS}$$

Equilibrium with Aggregate Shocks

Aggregate state: $(g_t, Z_t) \Rightarrow$ absorb into time subscript t

- Recursive notation w.r.t. individual states only
- $\mathbb{E}_t$ is expectation w.r.t. aggregate states only

▸ fully recursive

Equilibrium with Aggregate Shocks

Aggregate state: $(g_t, Z_t) \Rightarrow$ absorb into time subscript t

- Recursive notation w.r.t. individual states only
- $\mathbb{E}_t$ is expectation w.r.t. aggregate states only

fully recursive

$$\begin{aligned}\rho v_t(a,z) = \max_c \; & u(c) + \partial_a v_t(a,z)(w_t z + r_t a - c) \\ & + \lambda_z(v_t(a,z') - v_t(a,z)) + \frac{1}{dt}\mathbb{E}_t\left[dv_t(a,z)\right],\end{aligned} \tag{HJB}$$

$$\partial_t g_t(a,z) = -\partial_a[s_t(a,z)g_t(a,z)] - \lambda_z g_t(a,z) + \lambda_{z'} g_t(a,z'), \tag{KF}$$

$$w_t = (1-\alpha)e^{Z_t}K_t^{\alpha}, \quad r_t = \alpha e^{Z_t}K_t^{\alpha-1} - \delta \tag{P}$$

$$K_t = \int a g_t(a,z)\,da\,dz$$

$$dZ_t = -\nu Z_t dt + \sigma dW_t$$

Note: $\frac{1}{dt}\mathbb{E}_t\left[dv_t\right]$ means $\lim_{s\downarrow 0}\mathbb{E}_t\left[v_{t+s} - v_t\right]/s$

Linearization

Extension of standard linearization

1. Compute non-linear approximation to non-stochastic steady state

2. Compute first-order Taylor expansion around steady state

3. Solve linear stochastic differential equation

Warm Up: Linearizing a Representative Agent Model

- Optimality conditions in RBC model

$$\mathbb{E}_t \begin{bmatrix} dC_t \\ dK_t \\ dZ_t \end{bmatrix} = f(C_t, K_t, Z_t)dt, \qquad f : \mathbb{R}^3 \to \mathbb{R}^3$$

- C_t = consumption
- K_t = capital
- Z_t = productivity
- f_1 = Euler equation
- f_2 = resource constraint
- f_3 = productivity process

Warm Up: Linearizing a Representative Agent Model

- Optimality conditions in RBC model

$$\mathbb{E}_t \begin{bmatrix} dC_t \\ dK_t \\ dZ_t \end{bmatrix} = f(C_t, K_t, Z_t)dt, \qquad f : \mathbb{R}^3 \to \mathbb{R}^3$$

- C_t = consumption = control variable
- K_t = capital = endogenous state variable
- Z_t = productivity = exogenous state variable
- f_1 = Euler equation
- f_2 = resource constraint
- f_3 = productivity process

Warm Up: Linearizing a Representative Agent Model

1. Compute non-stochastic steady state $(C, K, Z = 0)$: by hand

Warm Up: Linearizing a Representative Agent Model

1. Compute non-stochastic steady state $(C, K, Z = 0)$: by hand

2. Compute first-order Taylor expansion of $f(C_t, K_t, Z_t)$

$$\mathbb{E}_t \begin{bmatrix} dC_t \\ dK_t \\ dZ_t \end{bmatrix} = f(C_t, K_t, Z_t)dt$$

Warm Up: Linearizing a Representative Agent Model

1. Compute non-stochastic steady state $(C, K, Z = 0)$: by hand

2. Compute first-order Taylor expansion of $f(C_t, K_t, Z_t)$

$$\mathbb{E}_t \begin{bmatrix} d\widehat{C}_t \\ d\widehat{K}_t \\ d\widehat{Z}_t \end{bmatrix} = \underbrace{\begin{bmatrix} B_{CC} & B_{CK} & B_{\Lambda Z} \\ B_{KC} & B_{KK} & B_{KZ} \\ 0 & 0 & B_{ZZ} \end{bmatrix}}_{\mathrm{B}} \begin{bmatrix} \widehat{C}_t \\ \widehat{K}_t \\ \widehat{Z}_t \end{bmatrix} dt$$

Warm Up: Linearizing a Representative Agent Model

1. Compute non-stochastic steady state $(C, K, Z = 0)$: by hand

2. Compute first-order Taylor expansion of $f(C_t, K_t, Z_t)$

$$\mathbb{E}_t \begin{bmatrix} d\widehat{C}_t \\ d\widehat{K}_t \\ d\widehat{Z}_t \end{bmatrix} = \underbrace{\begin{bmatrix} B_{CC} & B_{CK} & B_{\Lambda Z} \\ B_{KC} & B_{KK} & B_{KZ} \\ 0 & 0 & B_{ZZ} \end{bmatrix}}_{\mathrm{B}} \begin{bmatrix} \widehat{C}_t \\ \widehat{K}_t \\ \widehat{Z}_t \end{bmatrix} dt$$

3. Diagonalize matrix **B**, hope same number of stable eigenvalues as state variables (2 in this model)

   Set control variables $\perp$ to unstable eigenvectors $\Rightarrow$ policy function

$$\widehat{C}_t = D_K \widehat{K}_t + D_Z \widehat{Z}_t$$

Main Event: Linearizing a Heterogeneous Agent Model

1. **Compute non-linear approx. of non-stochastic steady state**

2. Compute first-order Taylor expansion around steady state

3. Solve linear stochastic differential equation

Main Event: Linearizing a Heterogeneous Agent Model

1. **Compute non-linear approx. of non-stochastic steady state**
   - Finite difference method from Achdou et al. (2017)
   - Steady state reduces to sparse matrix equations
   - Borrowing constraint absorbed into boundary conditions

2. Compute first-order Taylor expansion around steady state

3. Solve linear stochastic differential equation

Step 1: Compute non-stochastic steady state

$$\rho v(a,z) = \max_c \; u(c) + \partial_a v(a,z)(wz + ra - c) + \lambda_z (v(a,z') - v(a,z)) \qquad \text{(HJB SS)}$$

$$0 = -\partial_a[s(a,z)g(a,z)] - \lambda_z g(a,z) + \lambda_{z'} g(a,z') \qquad \text{(KF SS)}$$

$$w = (1-\alpha)K^{\alpha}, \quad r = \alpha K^{\alpha-1} - \delta, \qquad K = \int a g(a,z) da dz \qquad \text{(PRICE SS)}$$

Step 1: Compute non-stochastic steady state

$$\begin{aligned}\rho v_{i,j} =& u(c_{i,j}) + \partial_a v_{i,j}(wz_j + ra_i - c_{i,j}) \\ &+ \lambda_j(v_{i,-j} - v_{i,j}), \text{ with } c_{i,j} = u'^{-1}(\partial_a v_{i,j})\end{aligned} \qquad \text{(HJB SS)}$$

$$0 = -\partial_a[s(a,z)g(a,z)] - \lambda_z g(a,z) + \lambda_{z'} g(a,z') \qquad \text{(KF SS)}$$

$$\begin{aligned}& w = (1-\alpha)K^{\alpha}, \quad r = \alpha K^{\alpha-1} - \delta, \\ & K = \int a g(a,z)\, da\, dz\end{aligned} \qquad \text{(PRICE SS)}$$

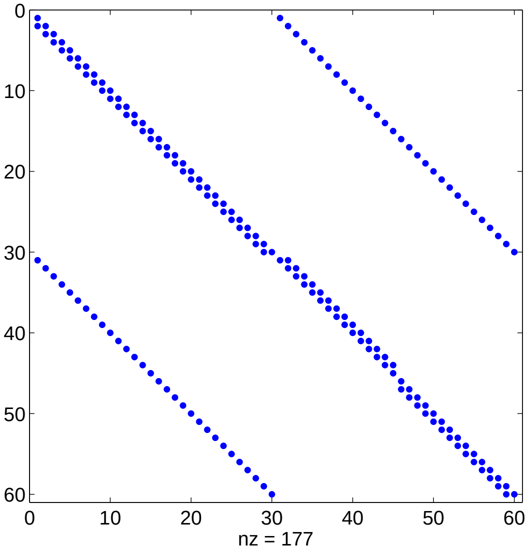
Step 1: Compute non-stochastic steady state

$$\rho \mathbf{v} = \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v}; \mathbf{p})\,\mathbf{v} \qquad \text{(HJB SS)}$$

$$0 = -\partial_a[s(a,z)g(a,z)] - \lambda_z g(a,z) + \lambda_{z'} g(a,z') \qquad \text{(KF SS)}$$

$$\begin{aligned} w &= (1-\alpha)K^{\alpha}, \quad r = \alpha K^{\alpha-1} - \delta, \\ K &= \int a g(a,z)\,da\,dz \end{aligned} \qquad \text{(PRICE SS)}$$

Visualization of **A** (output of `spy(A)` in Matlab)



Step 1: Compute non-stochastic steady state

$$\rho \mathbf{v} = \mathbf{u}\left(\mathbf{v}\right) + \mathbf{A}\left(\mathbf{v}; \mathbf{p}\right)\mathbf{v} \qquad \text{(HJB SS)}$$

$$0 = -\partial_a[s(a,z)g(a,z)] - \lambda_z g(a,z) + \lambda_{z'} g(a,z') \qquad \text{(KF SS)}$$

$$\begin{aligned} w &= (1-\alpha)K^{\alpha}, \quad r = \alpha K^{\alpha-1} - \delta, \\ K &= \int a g(a,z) da dz \end{aligned} \qquad \text{(PRICE SS)}$$

Step 1: Compute non-stochastic steady state

$$\rho \mathbf{v} = \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v};\mathbf{p})\,\mathbf{v} \tag{HJB SS}$$

$$\mathbf{0} = \mathbf{A}(\mathbf{v};\mathbf{p})^{\mathsf{T}}\,\mathbf{g} \tag{KF SS}$$

$$\begin{aligned} w &= (1-\alpha)K^{\alpha}, \quad r = \alpha K^{\alpha-1} - \delta, \\ K &= \int a g(a,z)\,da\,dz \end{aligned} \tag{PRICE SS}$$

Step 1: Compute non-stochastic steady state

$$\rho \mathbf{v} = \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v};\mathbf{p})\,\mathbf{v} \qquad \text{(HJB SS)}$$

$$\mathbf{0} = \mathbf{A}(\mathbf{v};\mathbf{p})^{\mathsf{T}}\,\mathbf{g} \qquad \text{(KF SS)}$$

$$\mathbf{p} = \mathbf{F}(\mathbf{g}) \qquad \text{(PRICE SS)}$$

- More generals models: (PRICE SS) becomes

$$0 = \mathbf{F}(\mathbf{g}, \mathbf{p})$$

Linearization: three steps

1. Compute non-linear approximation to non-stochastic steady state
    - Finite difference method from Achdou et al. (2017)
    - Steady state reduces to sparse matrix equations
    - Borrowing constraint absorbed into boundary conditions
2. **Compute first-order Taylor expansion around steady state**
3. Solve linear stochastic differential equation

Linearization: three steps

1. Compute non-linear approximation to non-stochastic steady state
   - Finite difference method from Achdou et al. (2017)
   - Steady state reduces to sparse matrix equations
   - Borrowing constraint absorbed into boundary conditions
2. **Compute first-order Taylor expansion around steady state**
   - Automatic differentiation: exact numerical derivatives
   - Efficient `Matlab` implementation for sparse systems
   - Different slopes at each point in state space
3. Solve linear stochastic differential equation

Step 2: Linearize discretized system

- Discretized system with aggregate shocks

$$
\begin{aligned}
\rho \mathbf{v}_t &= \mathbf{u}(\mathbf{v}_t) + \mathbf{A}(\mathbf{v}_t; \mathbf{p}_t)\, \mathbf{v}_t + \frac{1}{dt}\mathbb{E}_t[d\mathbf{v}_t] \\
\frac{d\mathbf{g}_t}{dt} &= \mathbf{A}(\mathbf{v}_t; \mathbf{p}_t)^{\top} \mathbf{g}_t \\
\mathbf{p}_t &= \mathbf{F}(\mathbf{g}_{\mathrm{t}}; Z_t) \\
dZ_t &= -\nu Z_t dt + \sigma dW_t
\end{aligned}
$$

Step 2: Linearize discretized system

- Discretized system with aggregate shocks

$$\rho \mathbf{v}_t = \mathbf{u}(\mathbf{v}_t) + \mathbf{A}(\mathbf{v}_t; \mathbf{p}_t)\, \mathbf{v}_t + \frac{1}{dt}\mathbb{E}_t[d\mathbf{v}_t]$$

$$\frac{d\mathbf{g}_t}{dt} = \mathbf{A}(\mathbf{v}_t; \mathbf{p}_t)^{\mathrm{T}}\, \mathbf{g}_t$$

$$\mathbf{p}_t = \mathbf{F}(\mathbf{g}_{\mathrm{t}}; Z_t)$$

$$dZ_t = -\nu Z_t dt + \sigma dW_t$$

- Write in general form

$$\mathbb{E}_t \begin{bmatrix} d\mathbf{v}_t \\ d\mathbf{g}_t \\ \mathbf{0} \\ dZ_t \end{bmatrix} = f(\mathbf{v}_t, \mathbf{g}_t, \mathbf{p}_t, Z_t)dt, \qquad \begin{bmatrix} \mathbf{v}_t \\ \mathbf{g}_t \\ \mathbf{p}_t \\ Z_t \end{bmatrix} = \begin{bmatrix} \text{control} \\ \text{endog state} \\ \text{prices} \\ \text{exog state} \end{bmatrix}$$

Step 2: Linearize discretized system

- Discretized system with aggregate shocks

$$
\begin{aligned}
\rho \mathbf{v}_t &= \mathbf{u}(\mathbf{v}_t) + \mathbf{A}(\mathbf{v}_t;\mathbf{p}_t)\,\mathbf{v}_t + \frac{1}{dt}\mathbb{E}_t[d\mathbf{v}_t] \\
\frac{d\mathbf{g}_t}{dt} &= \mathbf{A}(\mathbf{v}_t;\mathbf{p}_t)^{\mathrm{T}}\,\mathbf{g}_t \\
\mathbf{p}_t &= \mathbf{F}(\mathbf{g}_{\mathrm{t}};Z_t) \\
dZ_t &= -\nu Z_t dt + \sigma dW_t
\end{aligned}
$$

- Linearize using automatic differentiation (code: `@myAD`)

$$
\mathbb{E}_t \begin{bmatrix} d\widehat{\mathbf{v}}_t \\ d\widehat{\mathbf{g}}_t \\ \mathbf{0} \\ dZ_t \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{B}_{vv} & \mathbf{0} & \mathbf{B}_{vp} & \mathbf{0} \\ \mathbf{B}_{gv} & \mathbf{B}_{gg} & \mathbf{B}_{gp} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{pg} & -\mathbf{I} & \mathbf{B}_{pZ} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\nu \end{bmatrix}}_{\mathrm{B}} \begin{bmatrix} \widehat{\mathbf{v}}_t \\ \widehat{\mathbf{g}}_t \\ \widehat{\mathbf{p}}_t \\ Z_t \end{bmatrix} dt
$$

Linearization: three steps

1. Compute non-linear approximation to non-stochastic steady state
   - Finite difference method from Achdou et al. (2017)
   - Steady state reduces to sparse matrix equations
   - Borrowing constraint absorbed into boundary conditions
2. Compute first-order Taylor expansion around steady state
   - Automatic Differentiation: exact numerical derivatives
   - Efficient `Matlab` implementation for sparse systems
   - Different slopes at each point in state space
3. **Solve linear stochastic differential equation**

Linearization: three steps

1. Compute non-linear approximation to non-stochastic steady state
   - Finite difference method from Achdou et al. (2017)
   - Steady state reduces to sparse matrix equations
   - Borrowing constraint absorbed into boundary conditions
2. Compute first-order Taylor expansion around steady state
   - Automatic Differentiation: exact numerical derivatives
   - Efficient `Matlab` implementation for sparse systems
   - Different slopes at each point in state space
3. **Solve linear stochastic differential equation**
   - Moderately-sized systems $\Longrightarrow$ standard methods OK
   - Large systems $\Longrightarrow$ dimensionality reduction

Model-Free Reduction Method

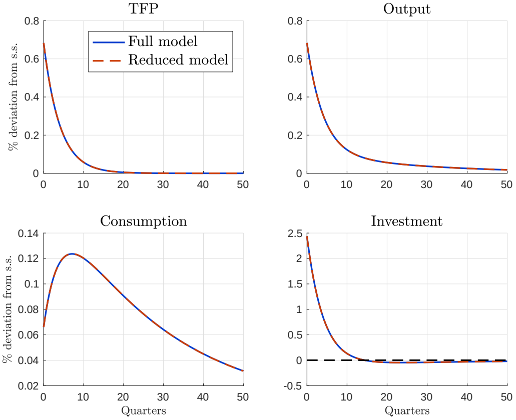
- Key insight: only need distribution $\mathbf{g}_t$ to forecast prices
  1. Krusell & Smith: guess moments ex-ante, check accuracy ex-post
  2. Our approach: computer chooses "moments", guarantee accuracy
- Approximate N-dimensional distribution with k-dimensional basis

$$\mathbf{g}_t \approx \gamma_{1t}\mathbf{x}_1 + \ldots + \gamma_{kt}\mathbf{x}_k$$

$\Rightarrow$ how to choose the basis $\mathbf{x}_1, \ldots, \mathbf{x}_k$?

- State-space reduction tools from engineering literature (Reiter 2010)
  - use "observability" criterion $\equiv$ matching impulse responses
  - adapt to problems with forward-looking decisions

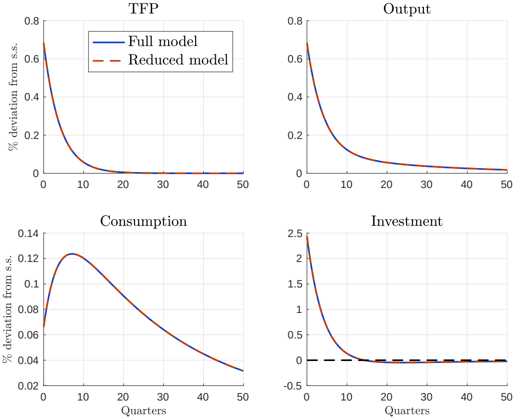
Approximate Aggregation in Krusell & Smith Model



- Comparison of full distribution vs. $k = 1$ approximation

  $\Longrightarrow$ recovers Krusell & Smith's "approximate aggregation"

Approximate Aggregation in Krusell & Smith Model



- Large-scale models in applications require $k = 300$

  $\Longrightarrow$ no approximate aggregation

Our Method Is Fast, Accurate in Krusell & Smith Model

Our method is fast

	w/o Reduction	**w/ Reduction**
Steady State	0.082 sec	0.082 sec
Linearize	0.021 sec	0.021 sec
Reduction	$\times$	0.007 sec
Solve	0.14 sec	0.002 sec
Total	0.243 sec	0.112 sec

- JEDC comparison project (2010): fastest alternative $\approx$ 7 minutes

Our Method Is Fast, Accurate in Krusell & Smith Model

Our method is fast

	w/o Reduction	**w/ Reduction**
Steady State	0.082 sec	0.082 sec
Linearize	0.021 sec	0.021 sec
Reduction	$\times$	0.007 sec
Solve	0.14 sec	0.002 sec
Total	0.243 sec	0.112 sec

- JEDC comparison project (2010): fastest alternative $\approx$ 7 minutes

Our method is accurate

Agg Shock σ	0.01%	0.1%	0.7%	1%	5%
Den Haan Error	0.000%	0.002%	0.053%	0.135%	3.347%

- JEDC comparison project: most accurate alternative $\approx$ 0.16%

Applications

A Model of Distribution of Income, Wealth, and MPCs

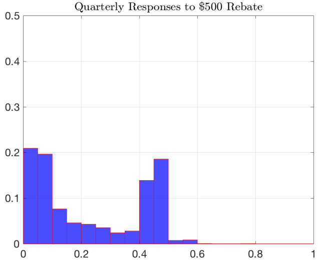
- Households: two-asset incomplete markets (Kaplan-Moll-Violante)
  - liquid asset
  - illiquid assets subject to transaction cost
- Aggregate production function with growth rate shocks

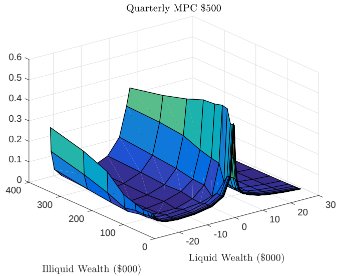
$$
\begin{aligned}
Y_t &= Q_t K_t^{\alpha} N_t^{1-\alpha} \\
\mathrm{d} \log Q_t &= Z_t \mathrm{d}t \\
\mathrm{d}Z_t &= -\eta Z_t \mathrm{d}t + \sigma \mathrm{d}W_t
\end{aligned}
$$

- Market clearing:
  - K_t = illiquid assets
  - B = liquid assets (fixed supply)

Application 1: Inequality Matters for C + Y Dynamics

- Campbell-Mankiw (1989): how match aggregate C + Y dynamics?
- Calibrate model to match
  1. Household side: distribution of income, wealth, and MPCs
  2. Firm side: dynamics of $\Delta \log Y_t$





Application 1: Inequality Matters for C + Y Dynamics

- Campbell-Mankiw (1989): how match aggregate C + Y dynamics?
- Calibrate model to match
  1. Household side: distribution of income, wealth, and MPCs
  2. Firm side: dynamics of $\Delta \log Y_t$

	Data	**Models**		
		Rep agent	Two-Asset	CM
Sensitivity to Income				
IV($\Delta \log C_t$ on $\Delta \log Y_t$ using $\Delta \log Y_{t-1}$)	0.503	0.247	0.656	
Smoothness				
$\frac{\sigma(\Delta \log C_t)}{\sigma(\Delta \log Y_t)}$	0.518	0.709	0.514	

Application 1: Inequality Matters for C + Y Dynamics

- Campbell-Mankiw (1989): how match aggregate C + Y dynamics?
- Calibrate model to match
  1. Household side: distribution of income, wealth, and MPCs
  2. Firm side: dynamics of $\Delta \log Y_t$

	Data	**Models**		
		Rep agent	Two-Asset	CM
Sensitivity to Income				
IV($\Delta \log C_t$ on $\Delta \log Y_t$ using $\Delta \log Y_{t-1}$)	0.503	0.247	0.656	0.505
Smoothness				
$\frac{\sigma(\Delta \log C_t)}{\sigma(\Delta \log Y_t)}$	0.518	0.709	0.514	0.676

Application 2: Agg Shocks Matter for Inequality

- With Cobb-Douglas prod'n, labor income inequality exogenous

$$\text{labor income } = w_t \times z_{jt}$$

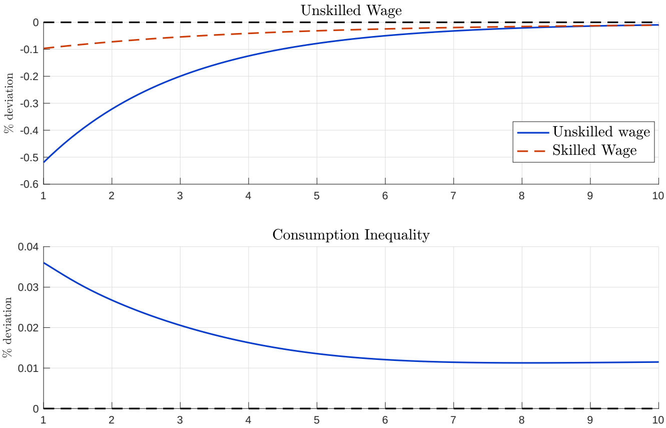
- Modify production function to generate endogenous inequality

$$Y_t = \left[\mu(Z_t^U N_t^U)^\sigma + (1-\mu)\left(\lambda K_t^\rho + (1-\lambda)(N_t^S)^\rho\right)^{\frac{\sigma}{\rho}}\right]^{\frac{1}{\sigma}}$$

  - N_t^U: unskilled labor w/ low persistent productivity z_{jt}
  - N_t^S: skilled labor w/ high persistent productivity z_{jt}
  - Z_t^U: unskilled-specific productivity shock

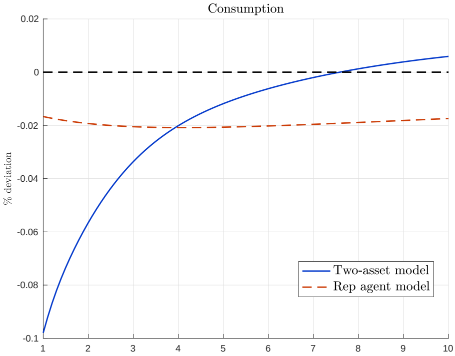
- Calibrate σ and ρ to generate capital-skill complementarity

Unskilled-Specific Shock Increases Inequality...



- Fluctuations in income inequality ≈ aggregate income

... And Generates Sharp Consumption Bust



- Many low-skill households hand-to-mouth

  $\Longrightarrow$ larger consumption drop than in rep agent model

One-Asset HANK Model

One-Asset HANK – Model Outline

- For details see `https://github.com/gregkaplan/phact/blob/master/examples/one_asset_HANK/docs/one_asset_hank_no_capital.pdf`
- Households:
  - as in Krusell-Smith model + endogenous labor supply
  - policy functions $c_t(a, z), \ell_t(a, z)$, distribution $g_t(a, z)$
- Firms:
  - monopolistic intermediate-good producers, labor demand L_t
  - quadratic price adjustment costs à la Rotemberg (1982)
  - $\Rightarrow$ New Keynesian Phillips curve
- Government: issues liquid debt B_t^g, spends, taxes/transfers
- Monetary authority: sets nominal rate based on a Taylor rule
- Equilibrium:

$$B_t^g = \int a g_t(a, z) da dz, \qquad L_t = \int \ell_t(a, z) g_t(a, z) da dz$$

Walking you through the code

- `https://sehyoun.com/EXAMPLE_one_asset_HANK_web.html`

Macro With Inequality: No More Excuses!

1. Efficient and easy-to-use computational method
   - open source `Matlab` toolbox online now
2. Use methodology to illustrate interaction of macro + inequality
   - match micro behavior ⇒ realistic aggregate C + Y dynamics
   - aggregate shocks generate inequality dynamics...
   - ... and IRFs in HA model can differ dramatically from RA case
3. Check out one-asset HANK model at
   `https://sehyoun.com/EXAMPLE_one_asset_HANK_web.html`

- Estimating models w/ micro data on distributions within reach
- Lots of cool applications: come talk to us!

Fully recursive notation ▸ Back

$$w(g,Z) = (1-\alpha)e^Z K(g)^\alpha, \quad r(g,Z) = \alpha e^Z K(g)^{\alpha-1} - \delta \tag{P}$$

$$K(g) = \int a g(a,z) da dz \tag{K}$$

$$\begin{aligned}\rho V(a,z,g,Z) = \max_c \ & u(c) + \partial_a V(a,z,g,Z)(w(g,Z)z + r(g,Z)a - c) \\ & + \lambda_z (V(a,z',g,Z) - V(a,z,g,Z)) \\ & + \partial_Z V(a,z,g,Z)(-\nu Z) + \frac{1}{2}\partial_{ZZ} V(a,z,g,Z)\sigma^2 \\ & + \int \frac{\delta V(a,z,g,Z)}{\delta g(\tilde a,\tilde z)} (\mathcal{K}_Z g)(\tilde a,\tilde z) d\tilde a d\tilde z \end{aligned} \tag{∞d HJB}$$

$$(\mathcal{K}_Z g)(a,z) = -\partial_a[s(a,z,g,Z)g(a,z)] - \lambda_z g(a,z) + \lambda_{z'} g(a,z') \tag{KF operator}$$

$$s(a,z,g,Z) = w(g,Z)z + r(g,Z)a - c^*(a,z,g,Z)$$

- $\delta V/\delta g(a,z)$: functional derivative of V wrt g at point (a,z)