Lecture 9

Efficient Computation of Heterogeneous Agent Models with Aggregate Shocks

ECO 521: Advanced Macroeconomics I

Benjamin Moll

Princeton University, Fall 2016

Motivation

- Quantitative DSGE models core of macroeconomic policy analysis

Motivation

- Quantitative DSGE models core of macroeconomic policy analysis
- Rely on representative agent assumptions

Motivation

- Quantitative DSGE models core of macroeconomic policy analysis
- Rely on representative agent assumptions
- Recent work argues micro heterogeneity important for policy analysis
    1. aggregate dynamics depend on distribution
    2. care about distributional implications

Motivation

- Quantitative DSGE models core of macroeconomic policy analysis
- Rely on representative agent assumptions
- Recent work argues micro heterogeneity important for policy analysis
  1. aggregate dynamics depend on distribution
  2. care about distributional implications
- But then distribution is a state variable

Motivation

- Quantitative DSGE models core of macroeconomic policy analysis
- Rely on representative agent assumptions
- Recent work argues micro heterogeneity important for policy analysis
  1. aggregate dynamics depend on distribution
  2. care about distributional implications
- But then distribution is a state variable
- Quantitative DSGE analysis infeasible with current methods
- Today: tell you about project that tries to make progress on this (joint with SeHyoun Ahn, Greg Kaplan, Tom Winberry)

What We Do

- Develop general + efficient method to compute het agent models

What We Do

- Develop general + efficient method to compute het agent models

1. Compute steady state using global approximations

2. Compute aggregate dynamics using local approximations

What We Do

- Develop general + efficient method to compute het agent models

1. Compute steady state using global approximations
    - exploit advantages of continuous time (Achdou et al. 2015)
2. Compute aggregate dynamics using local approximations

What We Do

- Develop general + efficient method to compute het agent models

1. Compute steady state using global approximations
   - exploit advantages of continuous time (Achdou et al. 2015)
2. Compute aggregate dynamics using local approximations
   - linear in entire distribution (Campbell 1998, Reiter 2009)
   - reduce dimensionality using SVDs (Reiter 2009)

What We Do

- Develop general + efficient method to compute het agent models

1. Compute steady state using global approximations
   - exploit advantages of continuous time (Achdou et al. 2015)
2. Compute aggregate dynamics using local approximations
   - linear in entire distribution (Campbell 1998, Reiter 2009)
   - reduce dimensionality using SVDs (Reiter 2009)

- Apply to textbook RBC + income heterogeneity (Krusell-Smith 98)
  - slides, notes, and `Matlab` codes available at `http://www.princeton.edu/~moll/PHACTproject.htm`

What We Do

- Develop general + efficient method to compute het agent models

1. Compute steady state using global approximations
   - exploit advantages of continuous time (Achdou et al. 2015)
2. Compute aggregate dynamics using local approximations
   - linear in entire distribution (Campbell 1998, Reiter 2009)
   - reduce dimensionality using SVDs (Reiter 2009)

- Apply to textbook RBC + income heterogeneity (Krusell-Smith 98)
  - slides, notes, and `Matlab` codes available at `http://www.princeton.edu/~moll/PHACTproject.htm`
- Have also implemented other applications
  - Khan & Thomas (2008), HANK models
  - ultimately: medium-scale DSGE (cost of extra agg states ≈ 0)

Plan For Today

1. Model

2. Solution method

3. Results

Households

$$\max_{\{c_t\}_{t\geq 0}} \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_t) dt \qquad \text{s.t.}$$

$$\dot{a}_t = w_t z_t + r_t a_t - c_t$$

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

$$a_t \geq \underline{a}$$

- c_t: consumption
- u: utility function, $u' > 0$, $u'' < 0$.
- ρ: discount rate
- r_t : interest rate
- $\underline{a} > -\infty$: borrowing limit e.g. if $\underline{a} = 0$, can only save

Firms

- Aggregate production function

$$Y_t = e^{Z_t} K_t^{\alpha} N_t^{1-\alpha}$$

- Perfect competitition in factor markets

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

- Market clearing

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

Equilibrium

- 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

- 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

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

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

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

$$w_t = (1-\alpha)e^{Z_t}K_t^{\alpha},$$

$$r_t = \alpha e^{Z_t}K_t^{\alpha-1} - \delta,$$

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

Solution Method

Extension of Standard Linearization Method

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

Background on linearization methods:

- Deterministic models
  - Chapter 6.3 of Stokey-Lucas-Prescott
  - `http://www.princeton.edu/~moll/ECO503Web/Lecture4_ECO503.pdf`
- Stochastic models
  - Sims (2001) “Solving Linear Expectations Models”
  - these notes `http://www.robertopancrazi.com/LN3_solving_lrem.pdf`

Linearization of Continuous-Time RBC model

- Optimality conditions in RBC model

$$\mathbb{E}_t\left[d\Lambda_t\right] = \Lambda_t\left(\rho + \delta - \alpha e^{Z_t} K_t^{\alpha-1}\right) dt$$

$$dK_t = \left(e^{Z_t} K_t^{\alpha} - \delta K_t - \Lambda_t^{-\frac{1}{\gamma}}\right) dt$$

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

- We have:

$$\begin{aligned} \text{control variable} &= \Lambda_t \\ \text{endog state variables} &= K_t \\ \text{exog state variables} &= Z_t \end{aligned}$$

Linearization of Continuous-Time RBC model

- Can write system as

$$\begin{bmatrix} \mathbb{E}_t[d\Lambda_t] \\ dK_t \\ dZ_t \end{bmatrix} = f(\Lambda_t, K_t, Z_t)dt + \begin{bmatrix} 0 \\ 0 \\ \sigma \end{bmatrix} dW_t$$

with $f : \mathbb{R}^3 \to \mathbb{R}^3$

- Since $\mathbb{E}_t[dW_t] = 0$, above system implies:

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

Linearization of Continuous-Time RBC model

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

Linearization of Continuous-Time RBC model

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

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

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

Linearization of Continuous-Time RBC model

1. Compute non-stochastic steady state $(\Lambda, K, Z = 0)$ by hand
2. Compute first-order Taylor expansion of $f(\Lambda_t, K_t, Z_t)$

$$\mathbb{E}_t \begin{bmatrix} d\hat{\Lambda}_t \\ d\hat{K}_t \\ d\hat{Z}_t \end{bmatrix} = \underbrace{\begin{bmatrix} B_{\Lambda\Lambda} & B_{\Lambda K} & B_{\Lambda Z} \\ B_{K\Lambda} & B_{KK} & B_{KZ} \\ 0 & 0 & -\nu \end{bmatrix}}_{\text{B}} \begin{bmatrix} \hat{\Lambda}_t \\ \hat{K}_t \\ \hat{Z}_t \end{bmatrix} dt$$

Linearization of Continuous-Time RBC model

1. Compute non-stochastic steady state $(\Lambda, K, Z = 0)$ by hand
2. Compute first-order Taylor expansion of $f(\Lambda_t, K_t, Z_t)$

$$\mathbb{E}_t \begin{bmatrix} d\hat{\Lambda}_t \\ d\hat{K}_t \\ d\hat{Z}_t \end{bmatrix} = \underbrace{\begin{bmatrix} B_{\Lambda\Lambda} & B_{\Lambda K} & B_{\Lambda Z} \\ B_{K\Lambda} & B_{KK} & B_{KZ} \\ 0 & 0 & -\nu \end{bmatrix}}_{\text{B}} \begin{bmatrix} \hat{\Lambda}_t \\ \hat{K}_t \\ \hat{Z}_t \end{bmatrix} dt$$

3. Diagonalize matrix **B**, hope same number of stable eigenvalues as state variables (2 in this model)
   - if so, set control variables orthogonal to unstable eigenvectors, get policy function

$$\hat{\Lambda}_t = D_K \hat{K}_t + D_Z \hat{Z}_t$$

Linearization of Heterogeneous Agent Model

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

Linearization of Heterogeneous Agent Model

1. Compute non-linear approximation to non-stochastic steady state
    - finite difference method from Achdou et al. (2015)
    - 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

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

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

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

Step 1: Compute Non-Stochastic Steady State

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

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

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

Step 1: Compute Non-Stochastic Steady State

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

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

$$w = (1-\alpha)K_t^{\alpha}, \quad r = \alpha K^{\alpha-1} - \delta,$$

$$K = \int a g(a,z)\, da dz \qquad \text{(PRICE SS)}$$

Step 1: Compute Non-Stochastic Steady State

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

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

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

Step 1: Compute Non-Stochastic Steady State

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

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

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

Linearization of Heterogeneous Agent Model

1. Compute non-linear approximation to non-stochastic steady state
    - finite difference method from Achdou et al. (2015)
    - 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 of Heterogeneous Agent Model

1. Compute non-linear approximation to non-stochastic steady state
    - finite difference method from Achdou et al. (2015)
    - 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
    - wrote efficient `Matlab` implementation for sparse systems
    - important: different slopes at different point in state space
3. Solve linear stochastic differential equation

Step 2: Linearize Discretized System

- Discretized system with aggregate shocks

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

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

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

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

Step 2: Linearize Discretized System

- Discretized system with aggregate shocks

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

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

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

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

- Key: same general form as RBC model earlier

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

Dimensionality: if 2 income types, 500 wealth grid points, then both $\mathbf{v}_t$ and $\mathbf{g}_t$ are $1000 \times 1 \Rightarrow [\mathbf{v}_t, \mathbf{g}_t, Z_t]'$ is 2001×1

Step 2: Linearize Discretized System

- Discretized system with aggregate shocks

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

- Key: same general form as RBC model earlier

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

Step 2: Linearize Discretized System

- Discretized system with aggregate shocks

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

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

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

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

- 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 of Heterogeneous Agent Model

1. Compute non-linear approximation to non-stochastic steady state
    - finite difference method from Achdou et al. (2015)
    - 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
    - wrote efficient `Matlab` implementation for sparse systems
    - important: different slopes at different point in state space
3. Solve linear stochastic differential equation

Linearization of Heterogeneous Agent Model

1. Compute non-linear approximation to non-stochastic steady state
    - finite difference method from Achdou et al. (2015)
    - 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
    - wrote efficient `Matlab` implementation for sparse systems
    - important: different slopes at different point in state space
3. Solve linear stochastic differential equation
    - moderately-sized systems $\Rightarrow$ standard methods work fine
    - large systems $\Rightarrow$ first reduce dimensionality using SVDs

Step 3: Solve Linear System

- Usual strategy: diagonalize + hope same number of stable eigenvalues as state variables ($I \times J + 1$ in this model)
  - if so, set control variables orthogonal to unstable eigenvectors, get policy function

$$\widehat{\mathbf{v}}_t = \mathbf{D}_g \widehat{\mathbf{g}}_t + \mathbf{D}_Z \widehat{Z}_t$$

Step 3: Solve Linear System

- Usual strategy: diagonalize + hope same number of stable eigenvalues as state variables ($I \times J + 1$ in this model)
  - if so, set control variables orthogonal to unstable eigenvectors, get policy function

$$\widehat{\mathbf{v}}_t = \mathbf{D}_g \widehat{\mathbf{g}}_t + \mathbf{D}_Z \widehat{Z}_t$$

- Works for moderately-sized systems (100-500 wealth grid points)
- Diagonalization prohibitively expensive for large systems (500 - 10,000 wealth grid points)
  - $\Rightarrow$ reduce dimensionality using SVDs (Reiter 2009)
  - background: review article by Antoulas (2005) "An overview of approximation methods for large-scale dynamical systems"
  - also see book by Antoulas (2005) "Approximation of Large Scale Dynamical Systems"

Results

Performance with 200 wealth grid points

(a) Without dim reduction

Command Window

```
Computing steady state...
Time to compute steady state: 0.141 seconds


Taking derivatives of equilibrium conditions...
...Done!
Time to compute derivatives: 0.0726 seconds


Solving linear system...
Existence and Uniqueness?  1 and  1
Time to compute aggregate dynamics: 1.67 seconds
fx >>
```

(b) With dim reduction

Command Window

```
Computing steady state...
Time to compute steady state: 0.113 seconds


Taking derivatives of equilibrium conditions...
...Done!
Time to compute derivatives: 0.0703 seconds


Reducing dimensionality...
...Done!
Time to reduce dimensionality: 0.5387 seconds


Solving linear system...
Existence and Uniqueness?  1 and  1
Time to compute aggregate dynamics: 0.189 seconds
fx >>
```

Performance with 600 wealth grid points

(a) Without dim reduction

Command Window

```
Computing steady state...
Time to compute steady state: 0.264 seconds


Taking derivatives of equilibrium conditions...
...Done!
Time to compute derivatives: 0.2280 seconds


Solving linear system...
Existence and Uniqueness?  1 and  1
Time to compute aggregate dynamics: 42.1 seconds
fx >>
```

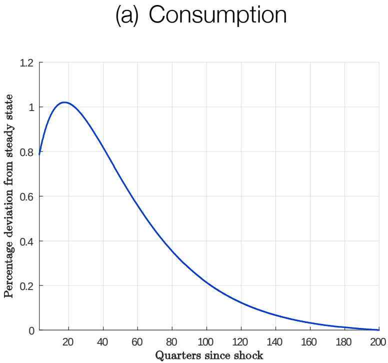
(b) With dim reduction

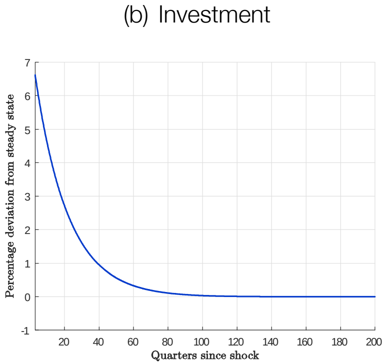
Command Window

```
Computing steady state...
Time to compute steady state: 0.257 seconds


Taking derivatives of equilibrium conditions...
...Done!
Time to compute derivatives: 0.2188 seconds


Reducing dimensionality...
...Done!
Time to reduce dimensionality: 6.8797 seconds


Solving linear system...
Existence and Uniqueness?  1 and  1
Time to compute aggregate dynamics: 3.21 seconds
fx >>
```

Aggregate Dynamics

IRF to 1-quarter TFP shock, entire distribution ($n = 400$ components)

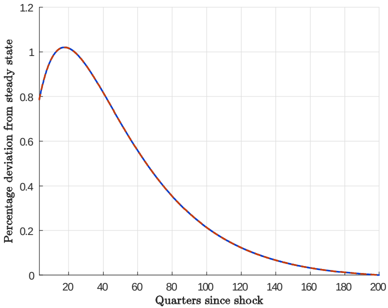




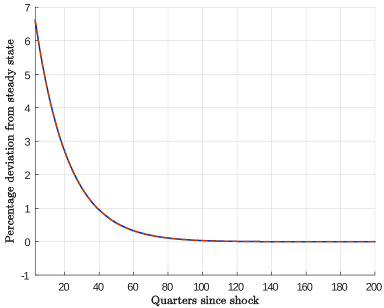
Aggregate Dynamics and Dimensionality Reduction

IRF to 1-quarter TFP shock, reduced distribution ($n = 95$ components)

(a) Consumption



(b) Investment



Micro Heterogeneity and Macro Nonlinearities

- Key motivation for studying het agent models: micro heterogeneity may generate nonlinear dynamics in aggregate variables
  - economy's response to shock may depend on initial distribution of agents
  - economy's response to shock may depend on size of shock

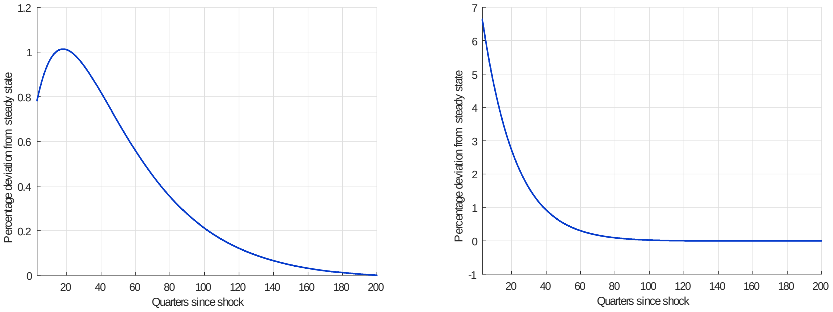
Micro Heterogeneity and Macro Nonlinearities

- Key motivation for studying het agent models: micro heterogeneity may generate nonlinear dynamics in aggregate variables
  - economy's response to shock may depend on initial distribution of agents
  - economy's response to shock may depend on size of shock
- Our methodology preserves such aggregate nonlinearities
  - true even though it relies on linear approximations
  - key: different slopes at different points of state space
  - in contrast to rep agent model: only one slope

Micro Heterogeneity and Macro Nonlinearities

IRF to one quarter TFP shock, starting from steady state

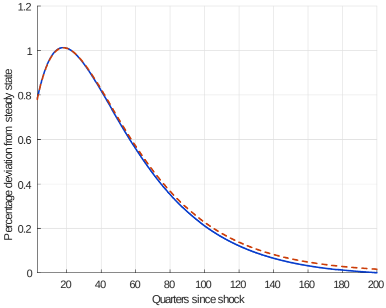
(a) Consumption (b) Investment



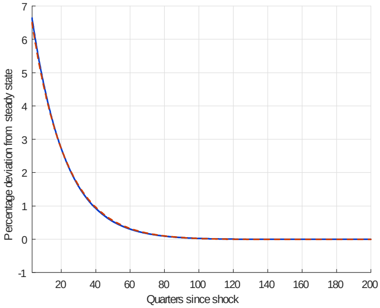
Micro Heterogeneity and Macro Nonlinearities

IRF to one quarter TFP shock, starting from recession

(a) Consumption



(b) Investment



Conclusion

- Developed general + efficient methodology to solve heterogeneous agent macro models

Conclusion

- Developed general + efficient methodology to solve heterogeneous agent macro models
- Extension of standard linearization methods
    - solve for steady state using continuous time
    - compute Taylor expansion using auto diff w/ sparsity
    - reduce state space using SVDs
    - `http://www.princeton.edu/~moll/PHACTproject.htm`

Conclusion

- Developed general + efficient methodology to solve heterogeneous agent macro models
- Extension of standard linearization methods
  - solve for steady state using continuous time
  - compute Taylor expansion using auto diff w/ sparsity
  - reduce state space using SVDs
  - `http://www.princeton.edu/~moll/PHACTproject.htm`
- Next step: medium-scale DSGE model featuring
  - het households w/ leptokurtic shocks + asset choice
  - het firms w/ productivity shocks + fixed costs
  - sticky prices

Instead: Fully Recursive Notation

Back

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

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

$$\begin{aligned}
\rho V(a,z,g,Z) = \max_c \; & u(c) + \partial_a V(a,z,g,Z)[w(g,Z)z + r(g,Z)a - c] \\
& + \lambda_z [V(a,z',g,Z) - V(a,z,g,Z)] \\
& + \partial_Z V(a,z,g,Z)(-\nu Z) + \frac{1}{2}\partial_{ZZ} V(a,z,g,Z)\sigma^2 \\
& + \int \frac{\delta V(a,z,g,Z)}{\delta g(a,z)} T[g,Z](a,z) da dz
\end{aligned} \tag{∞d HJB}$$

$$T[g,Z](a,z) = -\partial_a [s(a,z,g,Z)g(a,z)] - \lambda_z g(a,z) + \lambda_{z'} g(a,z') \tag{KF operator}$$

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

- big problem: distribution g is a state variable (infinite dimensional)
- $\delta V/\delta g(a,z)$: functional derivative of V wrt g at point (a,z)