Heterogeneous Agent Models in Continuous Time

HANK

Felipe Alves

NYU

April 28, 2016

Outline

- Heterogeneous Agent Models in Continuous Time
  - Achdou et. al (2014), Moll's website
- solving model = solving PDEs
  - Hamilton-Jacobi-Bellman equation for individual choices
  - Kolmogorov Forward equation for evolution of distribution
- **Project**: HANK model
  - Household Problem

Textbook Heterog Agent Model

$$\max_{c_t} \quad \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_t) d_t$$

$$\text{S.t} \quad \dot{a}_t = y_t + r_t a_t - c_t$$

y_t : Poisson with intensities λ_1, λ_2

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

Textbook Heterog Agent Model

- The value function of the optimal control problem satisfies the HJB equation

$$\rho v_i(a) = \max_c \ u(c) + v_i'(a)\Big[y_i + ra - c\Big] + \lambda_i\Big(v_j(a) - v_i(a)\Big)$$

- Let $G_i(a,t) \equiv P(a_t \leq a, y_t = y_i)$. Then

$$\partial_t G_i(a,t) = -s_i(a,t)\partial_a G_i(a,t) - \lambda_i G_i(a,t) + \lambda_j G_j(a,t)$$

while differentiating w.r.t. a

$$\partial_t g_i(a,t) = -\frac{d}{da}\big[s_i(a,t)g_i(a,t)\big] - \lambda_i g_i(a,t) + \lambda_j g_j(a,t)$$

Equilibrium

$$\rho v_i(a) = \max_c \; u(c) + v_i'(a)\Big[y_i + ra - c\Big] + \lambda_i\Big(v_j(a) - v_i(a)\Big) \quad \text{(HJB)}$$

$$0 = -\frac{d}{da}\Big[s_i(a)g_i(a)\Big] - \lambda_i g_i(a) + \lambda_j g_j(a) \quad \text{(KF)}$$

$$0 = \int a g_1(a)da + \int a g_2(a)da \quad \text{(equil)}$$

Question: What about state constraint?

state constraint boundary condition: $v_i'(\underline{a}) \geq u'(y_i + r\underline{a})$

How to solve

Theoretical background: Viscosity solutions (??)

Numerical solution : Finite Difference methods

- finite difference scheme converges to unique viscosity solution under some conditions - *Barles and Souganidis (1991)*
    1. Monotonicity
    2. Consistency
    3. Stability

Finite Difference Approximation

HJB equation

- Let's approximate (v_1, v_2) at J discrete points in the space dimension a_j. Putting $v_{i,j} \equiv v_i(a_j)$

$$\rho v_{i,j} = u(c_{i,j}) + v'_{i,j}\left[y_i + r a_j - c_{i,j}\right] + \lambda_i(v_{-i,j} - v_{i,j})$$
$$c_{i,j} = (u')^{-1}(v'_{i,j})$$

- PROBLEMS
  1. forward/backward difference approximation $\rightarrow$ upwind scheme

$$v'_{i,j,F} = \frac{v_{i,j+1} - v_{i,j}}{\Delta a}, \qquad v'_{i,j,B} = \frac{v_{i,j} - v_{i,j-1}}{\Delta a}$$

  2. How to solve for it? $\rightarrow$ explicit $\times$ implicit method

Finite Difference Approximation

HJB equation

Let's forget about idisyncratic income shock for a while

UPWIND

$$\rho v_j = u(c_j) + (v_{j,B})' \Big[y + r a_j - c_j \Big]^- + (v_{j,F})' \Big[y + r a_j - c_j \Big]^+$$

Finite Difference Approximation

HJB equation

Implicit Method

$$\frac{v_j^{n+1}-v_j^n}{\Delta}+\rho v_j^{n+1}=u(c_j^n)+\frac{v_j^{n+1}-v_{j-1}^{n+1}}{\Delta a}\left(s_{j,B}^n\right)^-+\frac{v_{j+1}^{n+1}-v_j^{n+1}}{\Delta a}\left(s_{j,F}^n\right)^+$$
$$c_{j,B}^n=(u')^{-1}({v_{j,B}^n}'),\ c_{j,F}^n=(u')^{-1}({v_{j,F}^n}'),$$

On a matrix form

$$\frac{1}{\Delta}(v^{n+1}-v^n)+\rho v^{n+1}=u^n+\mathbf{A}^n v^{n+1}$$

$\mathbf{A}^n$: Poison transition matrix / intensity matrix

Computational Advantages

- **Maximization Problem**

$$c^{-\gamma} = \partial_a v(a,y) \qquad c^{-\gamma} \geq \beta \int \partial_a v(a',y') dF(y'|y)$$

  - solve by hand
  - tomorrow is today - no need to compute integrals
  - foc always hold with equality - state constraint boundary condition

- **Sparsity**

- **Two birds with one Stone**

Computational Advantages

- **Maximization Problem**
- **Sparsity**
  - solving HJB, KFE = inverting matrix
  - but matrices are *sparse* (Why?)
- **Two birds with one Stone**

Computational Advantages

- **Maximization Problem**
- **Sparsity**
- **Two birds with one Stone**
  - tight link between solving HJB equation and KF equation for distribution
  - matrix in discrete KF is transpose of $\mathbf{A}$ in HJB

Computational Advantages

- Maximization Problem

$$c^{-\gamma} = \partial_a v(a, y) \qquad c^{-\gamma} \geq \beta \int \partial_a v(a', y') dF(y'|y)$$

  - solve by hand
  - tomorrow is today - no need to compute integrals
  - foc always hold with equality

- Sparsity
  - solving HJB, KFE = inverting matrix
  - but matrices are *sparse*
  - Why?

- *Two birds with one Stone*
  - tight link between solving HJB equation and KF equation for distribution
  - matrix in discrete KF is transpose of $\mathbf{A}$ in HJB

HANK

- Framework for quantitative analysis of aggregate shocks and macroeconomic policy
- Building blocks
    1. uninsurable idiosyncratic income risk
    2. assets with different degrees of liquidity
    3. nominal price rigidities
- No aggregate shocks
- Today: only the Household Problem

Households

$$\max_{\{c_t,\ell_t,d_t\}} \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_t, \ell_t) d_t$$

S.t

$$\dot{b}_t = z_t w_t \ell_t - \tilde{T}_t(w_t z_t \ell_t) + r_t^b(b_t) b_t - \Big(d_t + \chi(d_t, a_t)\Big) - c_t$$

$$\dot{a}_t = r_t^a a_t + d_t$$

z_t : Markov Process

$b_t \geq \underline{\text{b}}, \quad a_t \geq 0$

Households

$$\max_{\{c_t,\ell_t,d_t\}} \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_t,\ell_t) d_t$$

$$\text{S.t} \quad \dot{b}_t = z_t w_t \ell_t - \tilde{T}_t(w_t z_t \ell_t) + r_t^b(b_t) b_t - \Big(d_t + \chi(d_t, a_t)\Big) - c_t$$

$$\dot{a}_t = r_t^a a_t + d_t$$

$$z_t : \text{ Markov Process}$$

$$b_t \geq \underline{\mathrm{b}}, \quad a_t \geq 0$$

Households

$$\max_{\{c_t,\ell_t,d_t\}} \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_t,\ell_t) d_t$$

S.t $\quad \dot{b}_t = z_t w_t \ell_t - \tilde{T}_t(w_t z_t \ell_t) + r_t^b(b_t) b_t - \Big(d_t + \chi(d_t, a_t)\Big) - c_t$

$$\dot{a}_t = r_t^a a_t + d_t$$

z_t : Markov Process

$$b_t \geq \underline{\mathsf{b}}, \quad a_t \geq 0$$

Adjustment Cost: $\chi(d,a) = \underbrace{\chi_0 \mathbb{1}\{d \neq 0\}}_{\text{inaction}} + \underbrace{\frac{\chi_1}{2}\left(\frac{d}{a}\right)^2 a}_{\text{finite depo}}$

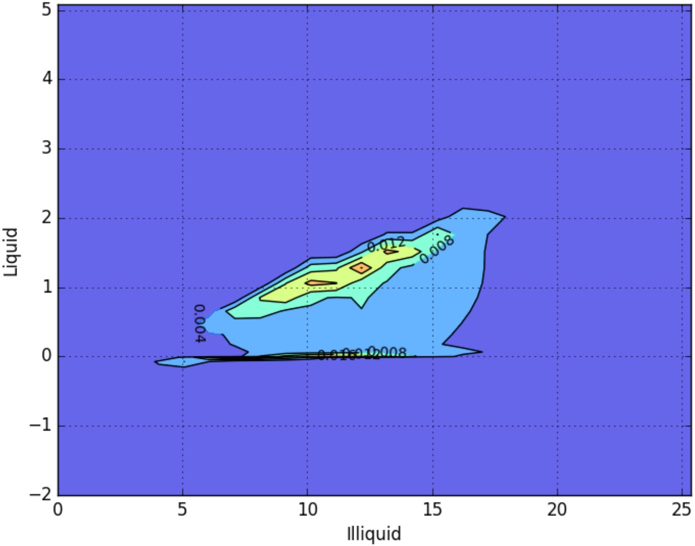
Partial Equil: $\Big\{w, \tilde{T}(\cdot), r^b(b), r^a\Big\}$

Code Structure

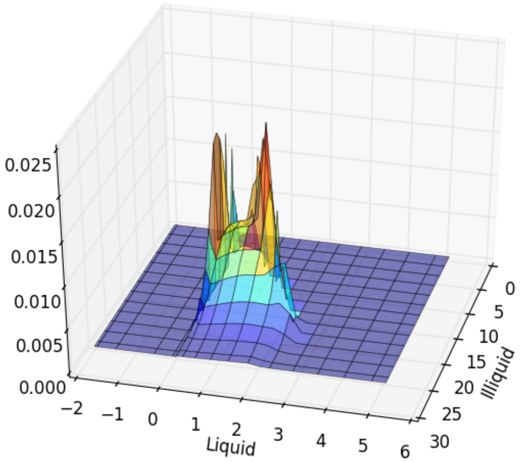
Important types

- `TwoAssetsProblem`
- `Solution`
- `FDSpec`
  - `GridInfo`
  - `Solution`
- `SteadyState`
- `Transition`

Contour plot



3Dplot


0.025
0.020
0.015
0.010
0.005
0.000
−2
−1
0
1
2
3
4
5
6
Liquid
0
5
10
15
20
25
30
Illiquid