Heterogeneous Agent Models in Continuous Time
Part I

Benjamin Moll
Princeton

Rochester, 1 March 2017

What this lecture is about

- Many interesting questions require thinking about distributions
  - Why are income and wealth so unequally distributed?
  - Is there a trade-off between inequality and economic growth?
  - What are the forces that lead to the concentration of economic activity in a few very large firms?
- Modeling distributions is hard
  - closed-form solutions are rare
  - computations are challenging
- Goal: teach you some new methods that make progress on this
  - solving heterogeneous agent model = solving PDEs
  - main difference to existing continuos-time literature: handle models for which closed-form solutions do not exist
  - based on joint work with Yves Achdou, SeHyoun Ahn, Jiequn Han, Greg Kaplan, Pierre-Louis Lions, Jean-Michel Lasry, Gianluca Violante, Tom Winberry, Christian

Solving het. agent model = solving PDEs

- More precisely: a system of two PDEs
  1. Hamilton-Jacobi-Bellman equation for individual choices
  2. Kolmogorov Forward equation for evolution of distribution
- Many well-developed methods for analyzing and solving these
  - codes: `http://www.princeton.edu/~moll/HACTproject.htm`
- Apparatus is very general: applies to any heterogeneous agent model with continuum of atomistic agents
  1. heterogeneous households (Aiyagari, Bewley, Huggett,...)
  2. heterogeneous producers (Hopenhayn,...)
- can be extended to handle aggregate shocks (Krusell-Smith,...)

Outline

Lecture 1

1. Refresher: HJB equations
2. Textbook heterogeneous agent model
3. Numerical solution of HJB equations
4. Models with non-convexities (Skiba)

Lecture 2

1. Analysis and numerical solution of heterogeneous agent model
2. Transition dynamics/MIT shocks
3. Stopping time problems
4. Models with multiple assets (HANK)

"When Inequality Matters for Macro and Macro Matters for Inequality"

1. Aggregate shocks via perturbation (Reiter)
2. Application to consumption dynamics

Computational Advantages relative to Discrete Time

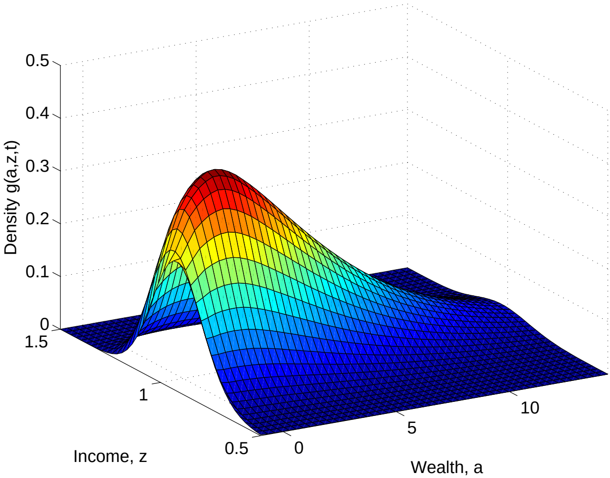
1. Borrowing constraints only show up in boundary conditions
    - FOCs always hold with "="
2. "Tomorrow is today"
    - FOCs are "static", compute by hand: $c^{-\gamma} = v_a(a, y)$
3. Sparsity
    - solving Bellman, distribution = inverting matrix
    - but matrices very sparse ("tridiagonal")
    - reason: continuous time $\Rightarrow$ one step left or one step right
4. Two birds with one stone
    - tight link between solving (HJB) and (KF) for distribution
    - matrix in discrete (KF) is transpose of matrix in discrete (HJB)
    - reason: diff. operator in (KF) is adjoint of operator in (HJB)

Real Payoff: extends to more general setups

- non-convexities
- stopping time problems (no need for threshold rules)
- multiple assets
- aggregate shocks

What you'll be able to do at end of this lecture

- Joint distribution of income and wealth in Aiyagari model



What you'll be able to do at end of this lecture

- Experiment: effect of one-time redistribution of wealth

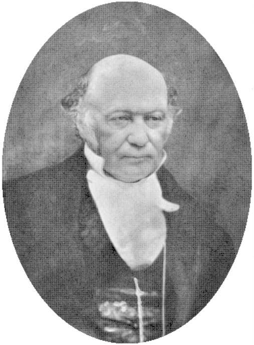
What you'll be able to do at end of this lecture

Video of convergence back to steady state

`https://www.dropbox.com/s/op5u2nlifmmer2o/distribution_tax.mp4?dl=0`

Review: HJB Equations

Hamilton-Jacobi-Bellman Equation: Some "History"



(a) William Hamilton

(b) Carl Jacobi

(c) Richard Bellman

- Aside: why called "dynamic programming"?
- Bellman: *"Try thinking of some combination that will possibly give it a pejorative meaning. It's impossible. Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to. So I used it as an umbrella for my activities."* `http://en.wikipedia.org/wiki/Dynamic_programming#History`

Hamilton-Jacobi-Bellman Equations

- Pretty much all deterministic optimal control problems in continuous time can be written as

$$v(x_0) = \max_{\{\alpha(t)\}_{t\geq 0}} \int_0^\infty e^{-\rho t} r(x(t), \alpha(t))\, dt$$

subject to the law of motion for the state

$$\dot{x}(t) = f(x(t), \alpha(t)) \quad \text{and} \quad \alpha(t) \in A$$

for $t \geq 0$, $x(0) = x_0$ given.

- $\rho \geq 0$: discount rate
- $x \in X \subseteq \mathbb{R}^m$: state vector
- $\alpha \in A \subseteq \mathbb{R}^n$: control vector
- $r : X \times A \to \mathbb{R}$: instantaneous return function

Example: Neoclassical Growth Model

$$v(k_0) = \max_{\{c(t)\}_{t\geq 0}} \int_0^\infty e^{-\rho t} u(c(t))dt$$

subject to

$$\dot{k}(t) = F(k(t)) - \delta k(t) - c(t)$$

for $t \geq 0$, $k(0) = k_0$ given.

- Here the state is $x = k$ and the control $\alpha = c$
- $r(x, \alpha) = u(\alpha)$
- $f(x, \alpha) = F(x) - \delta x - \alpha$

Generic HJB Equation

- How to analyze these optimal control problems? Here: "cookbook approach"

- Result: the value function of the generic optimal control problem satisfies the Hamilton-Jacobi-Bellman equation

$$\rho v(x) = \max_{\alpha \in A} \; r(x, \alpha) + v'(x) \cdot f(x, \alpha)$$

- In the case with more than one state variable $m > 1$, $v'(x) \in \mathbb{R}^m$ is the gradient vector of the value function.

Example: Neoclassical Growth Model

- "cookbook" implies:

$$\rho v(k) = \max_{c} \; u(c) + v'(k)(F(k) - \delta k - c)$$

- Proceed by taking first-order conditions etc

$$u'(c) = v'(k)$$

- ▸ Derivation from discrete time Bellman equation

Poisson Uncertainty

- Easy to extend this to stochastic case. Simplest case: two-state Poisson process
- Example: RBC Model. Production is $Z_t F(k_t)$ where $Z_t \in \{Z_1, Z_2\}$ Poisson with intensities λ_1, λ_2
- Result: HJB equation is

$$\rho v_i(k) = \max_c \; u(c) + v_i'(k)[Z_i F(k) - \delta k - c] + \lambda_i [v_j(k) - v_i(k)]$$

for $i = 1, 2, j \neq i$.

- Derivation similar as before

Some general, somewhat philosophical thoughts

- MAT 101 way ("first-order ODE needs one boundary condition") is not the right way to think about HJB equations
- these equations have very special structure which you should exploit when analyzing and solving them
- Particularly true for computations
- Important: all results/algorithms apply to problems with more than one state variable, i.e. it doesn't matter whether you solve ODEs or PDEs

A Textbook Heterogeneous Agent Model

Households

are heterogeneous in their wealth a and income y, solve

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

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

$$y_t \in \{y_1, y_2\} \text{ Poisson with intensities } \lambda_1, \lambda_2$$

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

- 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

later: carries over to y_t = general diffusion process.

Equations for Stationary Equilibrium

$$\rho v_j(a) = \max_c \; u(c) + v_j'(a)(y_j + ra - c) + \lambda_j(v_{-j}(a) - v_j(a)) \qquad \text{(HJB)}$$

$$0 = -\frac{d}{da}[s_j(a)g_j(a)] - \lambda_j g_j(a) + \lambda_{-j} g_{-j}(a), \qquad \text{(KF)}$$

$$s_j(a) = y_j + ra - c_j(a) = \text{saving policy function from (HJB)},$$

$$\int_{\underline{a}}^{\infty} (g_1(a) + g_2(a))da = 1, \quad g_1, g_2 \geq 0$$

$$S(r) := \int_{\underline{a}}^{\infty} ag_1(a)da + \int_{\underline{a}}^{\infty} ag_2(a)da = B, \qquad B \geq 0 \qquad \text{(EQ)}$$

- The two PDEs (HJB) and (KF) together with (EQ) fully characterize stationary equilibrium Derivation of (HJB) (KF)

Transition Dynamics

- Needed whenever initial condition $\neq$ stationary distribution
- Equilibrium still coupled systems of HJB and KF equations...
- ... but now time-dependent: $v_j(a, t)$ and $g_j(a, t)$
- See paper for equations
- Difficulty: the two PDEs run in opposite directions in time
  - HJB looks forward, runs backwards from terminal condition
  - KF looks backward, runs forward from initial condition

Numerical Solution of HJB Equations

Finite Difference Methods

- See `http://www.princeton.edu/~moll/HACTproject.htm`
- Explain using neoclassical growth model, easily generalized to heterogeneous agent models

$$\rho v(k) = \max_c \; u(c) + v'(k)(F(k) - \delta k - c)$$

- Functional forms

$$u(c) = \frac{c^{1-\sigma}}{1-\sigma}, \quad F(k) = k^\alpha$$

- Use finite difference method
  - Two MATLAB codes

    `http://www.princeton.edu/~moll/HACTproject/HJB_NGM.m`

    `http://www.princeton.edu/~moll/HACTproject/HJB_NGM_implicit.m`

Barles-Souganidis

- There is a well-developed theory for numerical solution of HJB equation using finite difference methods
- Key paper: Barles and Souganidis (1991), "Convergence of approximation schemes for fully nonlinear second order equations

  `https://www.dropbox.com/s/vhw5qqrczw3dvw3/barles-souganidis.pdf?dl=0`
- Result: finite difference scheme "converges" to unique viscosity solution under three conditions
  1. monotonicity
  2. consistency
  3. stability
- Good reference: Tourin (2013), "An Introduction to Finite Difference Methods for PDEs in Finance"
- Background on viscosity soln's: "Viscosity Solutions for Dummies"

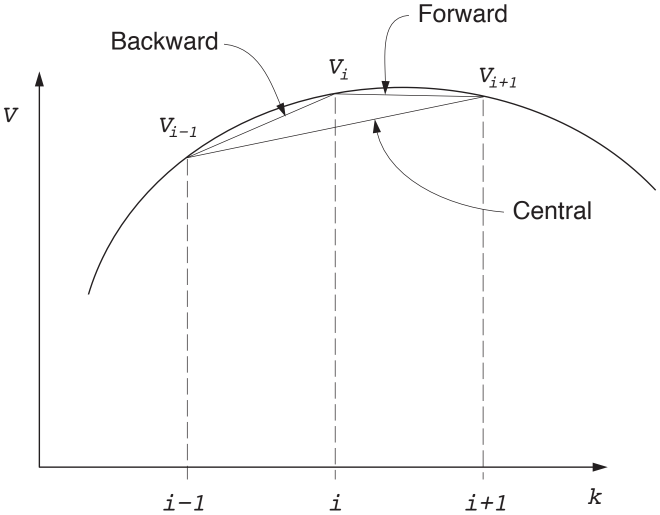
  `http://www.princeton.edu/~moll/viscosity_slides.pdf`

Finite Difference Approximations to $v'(k_i)$

- Approximate $v(k)$ at I discrete points in the state space, k_i, $i = 1, ..., I$. Denote distance between grid points by Δk.
- Shorthand notation
$$v_i = v(k_i)$$
- Need to approximate $v'(k_i)$.
- Three different possibilities:

$$v'(k_i) \approx \frac{v_i - v_{i-1}}{\Delta k} = v'_{i,B} \qquad \text{backward difference}$$
$$v'(k_i) \approx \frac{v_{i+1} - v_i}{\Delta k} = v'_{i,F} \qquad \text{forward difference}$$
$$v'(k_i) \approx \frac{v_{i+1} - v_{i-1}}{2\Delta k} = v'_{i,C} \qquad \text{central difference}$$

Finite Difference Approximations to $v'(k_i)$



Finite Difference Approximation

FD approximation to HJB is

$$\rho v_i = u(c_i) + v_i'[F(k_i) - \delta k_i - c_i] \qquad (*)$$

where $c_i = (u')^{-1}(v_i')$, and v_i' is one of backward, forward, central FD approximations.

Two complications:

1. which FD approximation to use? "Upwind scheme"
2. $(*)$ is extremely non-linear, need to solve iteratively: "explicit" vs. "implicit method"

My strategy for next few slides:

- what works
- slides on my website: why it works (Barles-Souganidis)

Which FD Approximation?

- Which of these you use is extremely important
- Best solution: use so-called "upwind scheme." Rough idea:
  - forward difference whenever drift of state variable positive
  - backward difference whenever drift of state variable negative
- In our example: define

$$s_{i,F} = F(k_i) - \delta k_i - (u')^{-1}(v'_{i,F}), \quad s_{i,B} = F(k_i) - \delta k_i - (u')^{-1}(v'_{i,B})$$

- Approximate derivative as follows

$$v'_i = v'_{i,F}\mathbf{1}_{\{s_{i,F}>0\}} + v'_{i,B}\mathbf{1}_{\{s_{i,B}<0\}} + \bar{v}'_i\mathbf{1}_{\{s_{i,F}<0<s_{i,B}\}}$$

where $\mathbf{1}_{\{\cdot\}}$ is indicator function, and $\bar{v}'_i = u'(F(k_i) - \delta k_i)$.

- Where does $\bar{v}'_i$ term come from? Answer:
  - since v is concave, $v'_{i,F} < v'_{i,B}$ (see figure) $\Rightarrow s_{i,F} < s_{i,B}$
  - if $s'_{i,F} < 0 < s'_{i,B}$, set $s_i = 0 \Rightarrow v'(k_i) = u'(F(k_i) - \delta k_i)$, i.e. we're at a steady state.

Sparsity

- Recall discretized HJB equation

$$\rho v_i = u(c_i) + v_i' \times (F(k_i) - \delta k_i - c_i), \quad i = 1, ..., I$$

- This can be written as

$$\rho v_i = u(c_i) + \frac{v_{i+1} - v_i}{\Delta k} s_{i,F}^+ + \frac{v_i - v_{i-1}}{\Delta k} s_{i,B}^-, \quad i = 1, ..., I$$

Notation: for any x, $x^+ = \max\{x, 0\}$ and $x^- = \min\{x, 0\}$

- Can write this in matrix notation

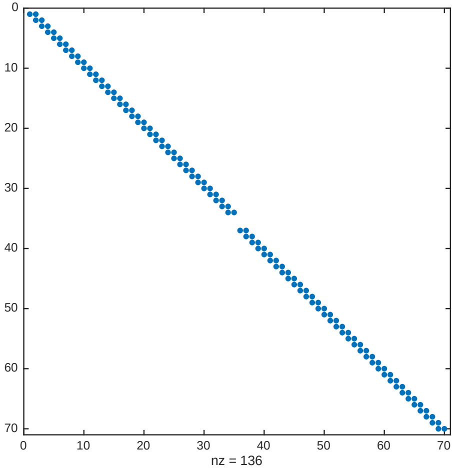
$$\rho v_i = u(c_i) + \begin{bmatrix} -\frac{s_{i,B}^-}{\Delta k} & \frac{s_{i,B}^-}{\Delta k} - \frac{s_{i,F}^+}{\Delta k} & \frac{s_{i,F}^+}{\Delta k} \end{bmatrix} \begin{bmatrix} v_{i-1} \\ v_i \\ v_{i+1} \end{bmatrix}$$

and hence

$$\rho \mathbf{v} = \mathbf{u} + \mathbf{A}\mathbf{v}$$

where **A** is $I \times I$ (I= no of grid points) and looks like...

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



The matrix **A**

- FD method approximates process for k with discrete Poisson process, **A** summarizes Poisson intensities
  - entries in row i:

$$\begin{bmatrix} \underbrace{-\frac{s_{i,B}^{-}}{\Delta k}}_{\text{inflow}_{i-1}\geq 0} & \underbrace{\frac{s_{i,B}^{-}}{\Delta k} - \frac{s_{i,F}^{+}}{\Delta k}}_{\text{outflow}_{i}\leq 0} & \underbrace{\frac{s_{i,F}^{+}}{\Delta k}}_{\text{inflow}_{i+1}\geq 0} \end{bmatrix} \begin{bmatrix} v_{i-1} \\ v_i \\ v_{i+1} \end{bmatrix}$$

  - negative diagonals, positive off-diagonals, rows sum to zero:
  - tridiagonal matrix, very sparse

- **A** (and **u**) depend on **v** (nonlinear problem)

$$\rho \mathbf{v} = \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v})\mathbf{v}$$

- Next: iterative method...

Iterative Method

- Idea: Solve FOC for given v^n, update v^{n+1} according to

$$\frac{v_i^{n+1} - v_i^n}{\Delta} + \rho v_i^n = u(c_i^n) + (v^n)'(k_i)(F(k_i) - \delta k_i - c_i^n) \qquad (*)$$

- Algorithm: Guess v_i^0, $i = 1, \ldots, I$ and for $n = 0, 1, 2, \ldots$ follow
  1. Compute $(v^n)'(k_i)$ using FD approx. on previous slide.
  2. Compute c^n from $c_i^n = (u')^{-1}[(v^n)'(k_i)]$
  3. Find v^{n+1} from $(*)$.
  4. If v^{n+1} is close enough to v^n: stop. Otherwise, go to step 1.
- See `http://www.princeton.edu/~moll/HACTproject/HJB_NGM.m`
- Important parameter: Δ = step size, cannot be too large ("CFL condition").
- Pretty inefficient: I need 5,990 iterations (though quite fast)

Efficiency: Implicit Method

- Efficiency can be improved by using an "implicit method"

$$\frac{v_i^{n+1} - v_i^n}{\Delta} + \rho v_i^{n+1} = u(c_i^n) + (v_i^{n+1})'(k_i)[F(k_i) - \delta k_i - c_i^n]$$

- Each step n involves solving a linear system of the form

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

$$\left((\rho + \tfrac{1}{\Delta})\mathbf{I} - \mathbf{A}(\mathbf{v}^{\mathrm{n}})\right)\mathbf{v}^{n+1} = \mathbf{u}(\mathbf{v}^{\mathrm{n}}) + \tfrac{1}{\Delta}\mathbf{v}^n$$

- but $\mathbf{A}(\mathbf{v}^{\mathrm{n}})$ is super sparse $\Rightarrow$ super fast
- See `http://www.princeton.edu/~moll/HACTproject/HJB_NGM_implicit.m`
- In general: implicit method preferable over explicit method
    1. stable regardless of step size Δ
    2. need much fewer iterations
    3. can handle many more grid points

Implicit Method: Practical Consideration

- In Matlab, need to explicitly construct **A** as sparse to take advantage of speed gains
- Code has part that looks as follows

```
X = -min(mub,0)/dk;
Y = -max(muf,0)/dk + min(mub,0)/dk;
Z = max(muf,0)/dk;
```

- Constructing full matrix – slow

```
for i=2:I-1
    A(i,i-1) = X(i);
    A(i,i) = Y(i);
    A(i,i+1) = Z(i);
end
A(1,1)=Y(1); A(1,2) = Z(1);
A(I,I)=Y(I); A(I,I-1) = X(I);
```

- Constructing sparse matrix – fast

```
A =spdiags(Y,0,I,I)+spdiags(X(2:I),-1,I,I)+spdiags([0;Z(1:I-1)],1,I,I);
```

Relation to Kushner-Dupuis "Markov-Chain Approx"

- There's another common method for solving HJB equation: "Markov Chain Approximation Method"
  - Kushner and Dupuis (2001) "Numerical Methods for Stochastic Control Problems in Continuous Time"
  - effectively: convert to discrete time, use value fn iteration
- FD method not so different: also converts things to "Markov Chain"

$$\rho v = u + \mathbf{A}v$$

- Connection between FD and MCAM
  - see Bonnans and Zidani (2003), "Consistency of Generalized Finite Difference Schemes for the Stochastic HJB Equation"
  - also shows how to exploit insights from MCAM to find FD scheme satisfying Barles-Souganidis conditions
- Another source of useful notes/codes: Frédéric Bonnans' website

`http://www.cmap.polytechnique.fr/~bonnans/notes/edpfin/edpfin.html`

Non-Convexities

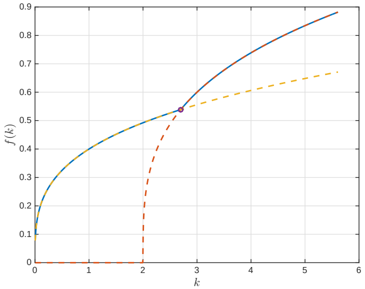
Non-Convexities

- Consider growth model

$$\rho v(k) = \max_c \; u(c) + v'(k)(F(k) - \delta k - c).$$

- But drop assumption that F is strictly concave. Instead: "butterfly"

$$\begin{aligned} F(k) &= \max\{F_L(k), F_H(k)\}, \\ F_L(k) &= A_L k^\alpha, \\ F_H(k) &= A_H((k-\kappa)^+)^\alpha, \quad \kappa > 0,\ A_H > A_L \end{aligned}$$

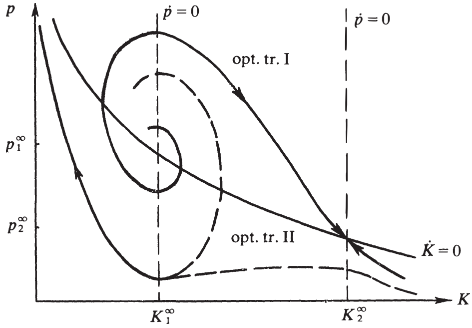


Standard Methods

- Discrete time: first-order conditions

$$u'(F(k) - \delta k - k') = \beta v'(k')$$

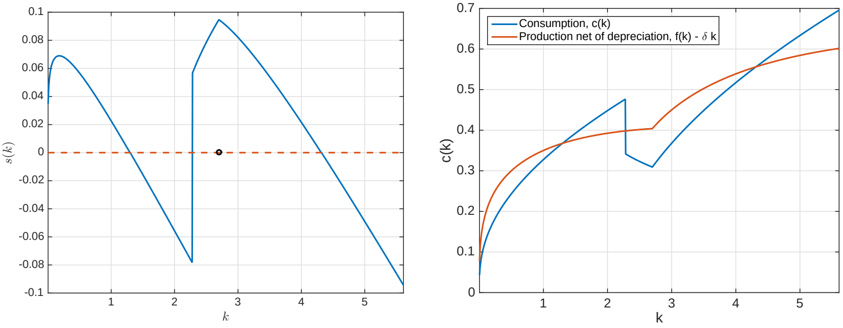
  no longer sufficient, typically multiple solutions
  - some applications: sidestep with lotteries (Prescott-Townsend)
- Continuous time: Skiba (1978)



Instead: Using Finite-Difference Scheme

Nothing changes, use same exact algorithm as for growth model with concave production function

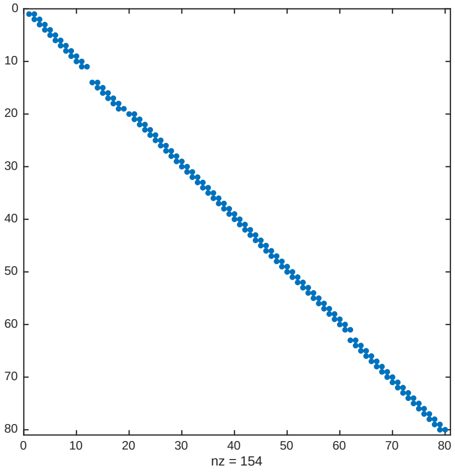
`http://www.princeton.edu/~moll/HACTproject/HJB_NGM_skiba.m`



(a) Saving Policy Function

(b) Consumption Policy Function

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



Appendix

Derivation from Discrete-time Bellman ▸ Back

- Time periods of length Δ
- discount factor
$$\beta(\Delta) = e^{-\rho\Delta}$$
- Note that $\lim_{\Delta \to 0} \beta(\Delta) = 1$ and $\lim_{\Delta \to \infty} \beta(\Delta) = 0$.
- Discrete-time Bellman equation:

$$v(k_t) = \max_{c_t} \; \Delta u(c_t) + e^{-\rho\Delta} v(k_{t+\Delta}) \quad \text{s.t.}$$

$$k_{t+\Delta} = \Delta[F(k_t) - \delta k_t - c_t] + k_t$$

Derivation from Discrete-time Bellman

- For small Δ (will take $\Delta \to 0$), $e^{-\rho\Delta} \approx 1 - \rho\Delta$

$$v(k_t) = \max_{c_t} \ \Delta u(c_t) + (1 - \rho\Delta) v(k_{t+\Delta})$$

- Subtract $(1 - \rho\Delta) v(k_t)$ from both sides

$$\rho\Delta v(k_t) = \max_{c_t} \ \Delta u(c_t) + (1 - \Delta\rho)[v(k_{t+\Delta}) - v(k_t)]$$

- Divide by Δ and manipulate last term

$$\rho v(k_t) = \max_{c_t} \ u(c_t) + (1 - \Delta\rho) \frac{v(k_{t+\Delta}) - v(k_t)}{k_{t+\Delta} - k_t} \frac{k_{t+\Delta} - k_t}{\Delta}$$

Take $\Delta \to 0$

$$\rho v(k_t) = \max_{c_t} \ u(c_t) + v'(k_t)\dot{k}_t$$

Derivation of Poisson KF Equation ▸ Back

- Work with CDF (in wealth dimension)

$$G_j(a,t) := \Pr(\tilde{a}_t \le a, \tilde{y}_t = y_j)$$

- Income switches from y_j to y_{-j} with probability $\Delta\lambda_j$
- Over period of length Δ, wealth evolves as $\tilde{a}_{t+\Delta} = \tilde{a}_t + \Delta s_j(\tilde{a}_t)$
- Similarly, answer to question "where did $\tilde{a}_{t+\Delta}$ come from?" is

$$\tilde{a}_t = \tilde{a}_{t+\Delta} - \Delta s_j(\tilde{a}_{t+\Delta})$$

- Momentarily ignoring income switches and assuming $s_j(a) < 0$

$$\Pr(\tilde{a}_{t+\Delta} \le a) = \underbrace{\Pr(\tilde{a}_t \le a)}_{\text{already below } a} + \underbrace{\Pr(a \le \tilde{a}_t \le a - \Delta s_j(a))}_{\text{cross threshold } a} = \Pr(\tilde{a}_t \le a - \Delta s_j(a))$$

- Fraction of people with wealth below a evolves as

$$\begin{aligned}\Pr(\tilde{a}_{t+\Delta} \le a, \tilde{y}_{t+\Delta} = y_j) = (1 - \Delta\lambda_j) \Pr(\tilde{a}_t \le a - \Delta s_j(a), \tilde{y}_t = y_j) \\ + \Delta\lambda_{-j} \Pr(\tilde{a}_t \le a - \Delta s_{-j}(a), \tilde{y}_t = y_{-j})\end{aligned}$$

- Intuition: if have wealth $< a - \Delta s_j(a)$ at t, have wealth $< a$ at $t + \Delta$

Derivation of Poisson KF Equation

- Subtracting $G_j(a,t)$ from both sides and dividing by Δ

$$\frac{G_j(a,t+\Delta)-G_j(a,t)}{\Delta}=\frac{G_j(a-\Delta s_j(a),t)-G_j(a,t)}{\Delta}$$
$$-\lambda_j G_j(a-\Delta s_j(a),t)+\lambda_{-j}G_{-j}(a-\Delta s_{-j}(a),t$$

- Taking the limit as $\Delta \to 0$

$$\partial_t G_j(a,t)=-s_j(a)\partial_a G_j(a,t)-\lambda_j G_j(a,t)+\lambda_{-j}G_{-j}(a,t)$$

where we have used that

$$\lim_{\Delta\to 0}\frac{G_j(a-\Delta s_j(a),t)-G_j(a,t)}{\Delta}=\lim_{x\to 0}\frac{G_j(a-x,t)-G_j(a,t)}{x}s_j(a)$$
$$=-s_j(a)\partial_a G_j(a,t)$$

- Intuition: if $s_j(a)<0$, $\Pr(\tilde{a}_t\le a,\tilde{y}_t=y_j)$ increases at rate $g_j(a,t)$
- Differentiate w.r.t. a and use $g_j(a,t)=\partial_a G_j(a,t)\Rightarrow$

$$\partial_t g_j(a,t)=-\partial_a[s_j(a,t)g_j(a,t)]-\lambda_j g_j(a,t)+\lambda_{-j}g_{-j}(a,t)$$

Heterogeneous Agent Models in Continuous Time
Part II

Benjamin Moll
Princeton

Rochester, 2 March 2017

Outline

Lecture 1

1. Refresher: HJB equations
2. Textbook heterogeneous agent model
3. Numerical solution of HJB equations
4. Models with non-convexities (Skiba)

Lecture 2

1. Analysis and numerical solution of heterogeneous agent model
2. Transition dynamics/MIT shocks
3. Stopping time problems
4. Models with multiple assets (HANK)

"When Inequality Matters for Macro and Macro Matters for Inequality"

1. Aggregate shocks via perturbation (Reiter)
2. Application to consumption dynamics

Analysis and Numerical Solution of Heterogeneous Agent Model

Recall Textbook Heterogeneous Agent Model

$$\rho v_j(a) = \max_c \; u(c) + v_j'(a)(y_j + ra - c) + \lambda_j(v_{-j}(a) - v_j(a)) \qquad \text{(HJB)}$$

$$0 = -\frac{d}{da}[s_j(a)g_j(a)] - \lambda_j g_j(a) + \lambda_{-j}g_{-j}(a), \qquad \text{(KF)}$$

$$s_j(a) = y_j + ra - c_j(a) = \text{saving policy function from (HJB)},$$

$$\int_{\underline{a}}^{\infty}(g_1(a) + g_2(a))da = 1, \quad g_1, g_2 \geq 0$$

$$S(r) := \int_{\underline{a}}^{\infty} a g_1(a)da + \int_{\underline{a}}^{\infty} a g_2(a)da = B, \qquad B \geq 0 \qquad \text{(EQ)}$$

- The two PDEs (HJB) and (KF) together with (EQ) fully characterize stationary equilibrium Derivation of (HJB) (KF)

Borrowing Constraints?

- Q: where is borrowing constraint $a \geq \underline{a}$ in (HJB)?
- A: "in" boundary condition
- Result: v_j must satisfy

$$v_j'(\underline{a}) \geq u'(y_j + r\underline{a}), \quad j = 1, 2 \tag{BC}$$

- Derivation:
  - the FOC still holds at the borrowing constraint

$$u'(c_j(\underline{a})) = v_j'(\underline{a}) \tag{FOC}$$

  - for borrowing constraint not to be violated, need

$$s_j(\underline{a}) = y_j + r\underline{a} - c_j(\underline{a}) \geq 0 \tag{$*$}$$

  - (FOC) and ($*$) $\Rightarrow$ (BC).

- See slides on viscosity solutions for more rigorous discussion
`http://www.princeton.edu/~moll/viscosity_slides.pdf`

Plan

- New theoretical results:

    1. analytics: consumption, saving, MPCs of the poor
    2. closed-form for wealth distribution with 2 income types
    3. unique stationary equilibrium if IES ≥ 1 (sufficient condition)

  Note: for 1. and 2. analyze partial equilibrium with $r < \rho$

- Computational algorithm:
    - problems with non-convexities
    - transition dynamics

Result 1: Consumption, Saving Behavior of the Poor

Behavior near borrowing constraint depends on two factors

1. tightness of constraint
2. properties of u as $c \to 0$

Assumption 1:
As $a \to \underline{a}$, coefficient of absolute risk aversion $R(c) = -u''(c)/u'(c)$ remains finite

$$\underline{R} := -\lim_{a \to \underline{a}} \frac{u''(y_1 + ra)}{u'(y_1 + ra)} < \infty$$

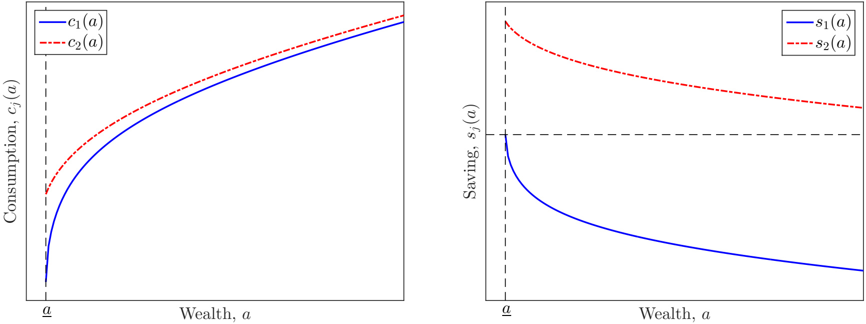
- sufficient condition for A1: borrowing constraint is tighter than "natural borrowing constraint" $\underline{a} > -y_1/r$
- e.g. with CRRA utility

$$u(c) = \frac{c^{1-\gamma}}{1-\gamma} \quad \Rightarrow \quad \underline{R} = \frac{\gamma}{y_1 + r\underline{a}}$$

- but weaker: e.g. A1 satisfied with $\underline{a} = -y_1/r$ and $u(c) = -e^{-\theta c}/\theta$

Result 1: Consumption, Saving Behavior of the Poor

Rough version of Proposition: under A1 policy functions look like this



Result 1: Consumption, Saving Behavior of the Poor

Proposition: Assume $r < \rho$, $y_1 < y_2$ and that A1 holds. The solution to (HJB) has following properties:

1. $s_1(\underline{a}) = 0$ but $s_1(a) < 0$ all $a > \underline{a}$: only households exactly at the borrowing constraint are constrained
2. Saving and consumption policy functions close to $a = \underline{a}$ satisfy

$$s_1(a) \sim -\sqrt{2\nu_1}\sqrt{a - \underline{a}}$$

$$c_1(a) \sim y_1 + ra + \sqrt{2\nu_1}\sqrt{a - \underline{a}}$$

$$c_1'(a) \sim r + \frac{1}{2}\sqrt{\frac{\nu_1}{2(a - \underline{a})}}$$

$$\nu_1 = \frac{(\rho - r)u'(\underline{c}_1) + \lambda_1(u'(\underline{c}_1) - u'(\underline{c}_2))}{-u''(\underline{c}_1)}$$

Note: "$f(a) \sim g(a)$" means $\lim_{a \to \underline{a}} f(a)/g(a) = 1$, "$f$ behaves like g close to $\underline{a}$"

Result 1: Consumption, Saving Behavior of the Poor

Corollary: The wealth of worker who keeps y_1 converges to borrowing constraint in finite time at speed governed by ν_1:

$$a(t) - \underline{a} \sim \frac{\nu_1}{2}(T-t)^2, \quad 0 \leq t \leq T, \quad \text{where}$$

$$T := \sqrt{\frac{2(a_0 - \underline{a})}{\nu_1}} = \text{“hitting time”}$$

Proof: integrate $\dot{a}(t) = -\sqrt{2\nu_1}\sqrt{a(t) - \underline{a}}$

And have analytic solution for speed

$$\nu_1 = \frac{(\rho - r)u'(\underline{c}_1) + \lambda_1(u'(\underline{c}_1) - u'(\underline{c}_2))}{-u''(\underline{c}_1)}$$

$$\approx (\rho - r)\text{IES}(\underline{c}_1)\underline{c}_1 + \lambda_1(\underline{c}_2 - \underline{c}_1)$$

Result 2: Stationary Wealth Distribution

- Recall equation for stationary distribution

$$0 = -\frac{d}{da}[s_j(a)g_j(a)] - \lambda_j g_j(a) + \lambda_{-j}g_{-j}(a) \qquad \text{(KF)}$$

- **Lemma:** the solution to (KF) is

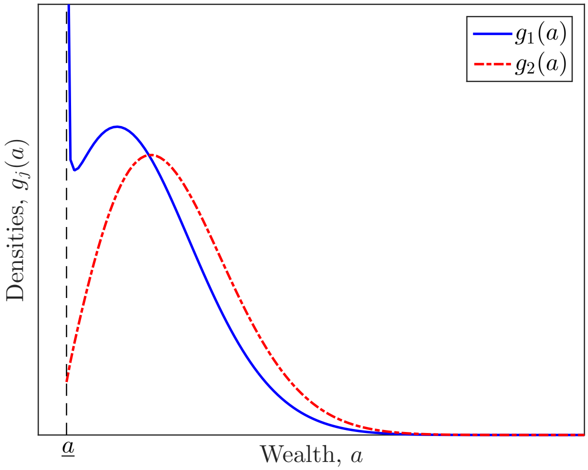
$$g_i(a) = \frac{\kappa_j}{s_j(a)} \exp\left(-\int_{\underline{a}}^{a}\left(\frac{\lambda_1}{s_1(x)} + \frac{\lambda_2}{s_2(x)}dx\right)\right)$$

  with κ_1, κ_2 pinned down by g_j's integrating to one

- Features of wealth distribution:
  - Dirac point mass of type y_1 individuals at constraint $G_1(\underline{a}) > 0$
  - thin right tail: $g(a) \sim \xi(a_{\max} - a)^{\lambda_2/\zeta_2 - 1}$, i.e. not Pareto
  - see paper for more

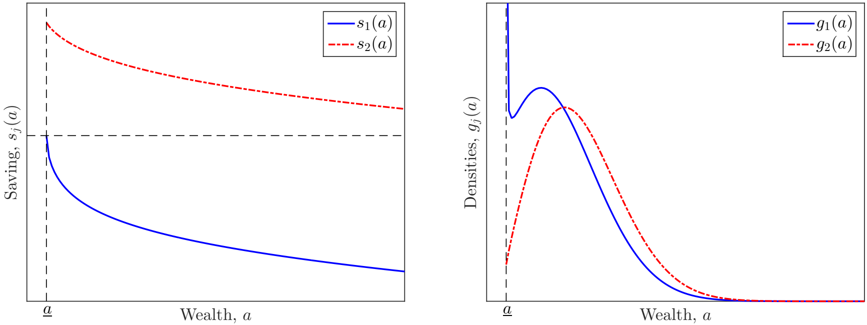
- Later in paper: extension with Pareto tail (Benhabib-Bisin-Zhu)

Result 2: Stationary Wealth Distribution

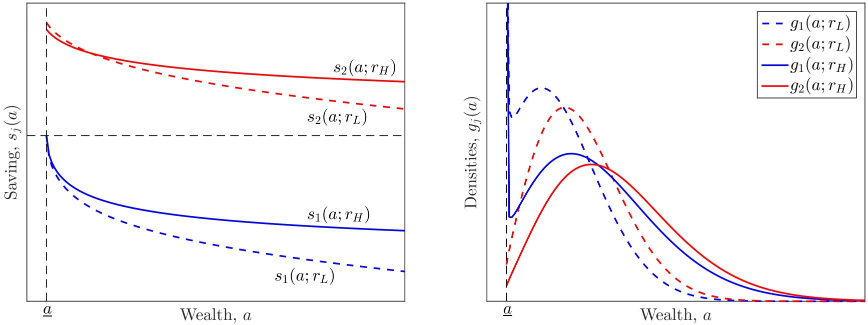


Note: in numerical solution, Dirac mass = finite spike in density

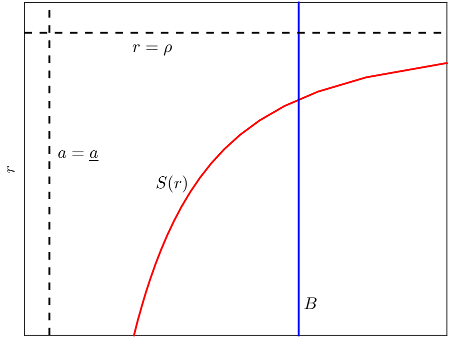
General Equilibrium: Existence and Uniqueness



Increase in r from r_L to $r_H > r_L$



Stationary Equilibrium



Asset Supply $\quad S(r) = \int_{\underline{a}}^{\infty} a g_1(a; r) da + \int_{\underline{a}}^{\infty} a g_2(a; r) da$

- **Proposition:** a stationary equilibrium exists
- **Proposition:** if $\text{IES}(c) \geq 1$ for all c and no borrowing $a \geq 0$, stationary equilibrium is unique

Computations for Heterogeneous Agent Model

Computations for Heterogeneous Agent Model

- Hard part: HJB equation. But already know how to do that.
- Easy part: KF equation. Once you solved HJB equation, get KF equation "for free"
- System to be solved

$$\rho v_1(a) = \max_c \; u(c) + v_1'(a)(y_1 + ra - c) + \lambda_1(v_2(a) - v_1(a))$$

$$\rho v_2(a) = \max_c \; u(c) + v_2'(a)(y_2 + ra - c) + \lambda_2(v_1(a) - v_2(a))$$

$$0 = -\frac{d}{da}[s_1(a)g_1(a)] - \lambda_1 g_1(a) + \lambda_2 g_2(a)$$

$$0 = -\frac{d}{da}[s_2(a)g_2(a)] - \lambda_2 g_2(a) + \lambda_1 g_1(a)$$

$$1 = \int_{\underline{a}}^{\infty} g_1(a)da + \int_{\underline{a}}^{\infty} g_2(a)da$$

$$0 = \int_{\underline{a}}^{\infty} ag_1(a)da + \int_{\underline{a}}^{\infty} ag_2(a)da \equiv S(r)$$

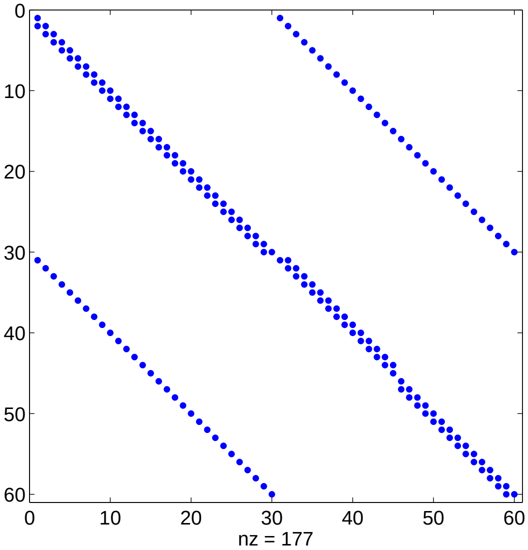
Computations for Heterogeneous Agent Model

- As before, discretized HJB equation is

$$\rho \mathbf{v} = \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v})\mathbf{v} \qquad \text{(HJBd)}$$

- $\mathbf{A}$ is $N \times N$ transition matrix
  - here $N = 2 \times I$, I=number of wealth grid points
  - $\mathbf{A}$ depends on $\mathbf{v}$ (nonlinear problem)
  - solve using implicit scheme

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



Computing the FK Equation

- Equations to be solved

$$0 = -\frac{d}{da}[s_1(a)g_1(a)] - \lambda_1 g_1(a) + \lambda_2 g_2(a)$$
$$0 = -\frac{d}{da}[s_2(a)g_2(a)] - \lambda_2 g_2(a) + \lambda_1 g_1(a)$$

  with $1 = \int_{\underline{a}}^{\infty} g_1(a)da + \int_{\underline{a}}^{\infty} g_2(a)da$
- Actually, super easy: discretized version is simply

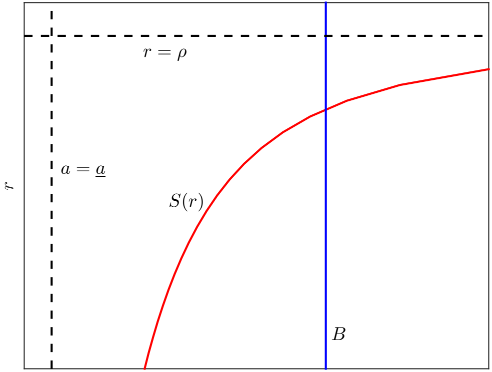
$$0 = \mathbf{A}(\mathbf{v})^{\mathsf{T}}\mathbf{g} \quad \text{(KFd)}$$

  - eigenvalue problem
  - get KF for free, one more reason for using implicit scheme
- Why transpose?
  - operator in (HJB) is “adjoint” of operator in (KF)
  - “adjoint” = infinite-dimensional analogue of matrix transpose
- In principle, can use similar strategy in discrete time

Finding the Equilibrium Interest Rate

Use bisection method

- increase r whenever $S(r) < B$
- decrease r whenever $S(r) > B$



A Model with a Continuum of Income Types

- Assume idiosyncratic income follows diffusion process

$$dy_t = \mu(y_t)dt + \sigma(y_t)dW_t$$

- Reflecting barriers at $\underline{y}$ and $\bar{y}$

$$\rho v(a,y) = \max_c \; u(c) + \partial_a v(a,y)(y + ra - c) + \mu(y)\partial_y v(a,y) + \frac{\sigma^2(y)}{2}\partial_{yy} v(a,y)$$

$$0 = -\partial_a[s(a,y)g(a,y)] - \partial_y[\mu(y)g(a,y)] + \frac{1}{2}\partial_{yy}[\sigma^2(y)g(a,y)]$$

$$1 = \int_0^\infty \int_{\underline{a}}^\infty g(a,y)dady$$

$$0 = \int_0^\infty \int_{\underline{a}}^\infty ag(a,y)dady =: S(r)$$

- Borrowing constraint: $\partial_a v(\underline{a}, y) \geq u'(y + r\underline{a})$, all y
- reflecting barriers (see e.g. Dixit “Art of Smooth Pasting”)

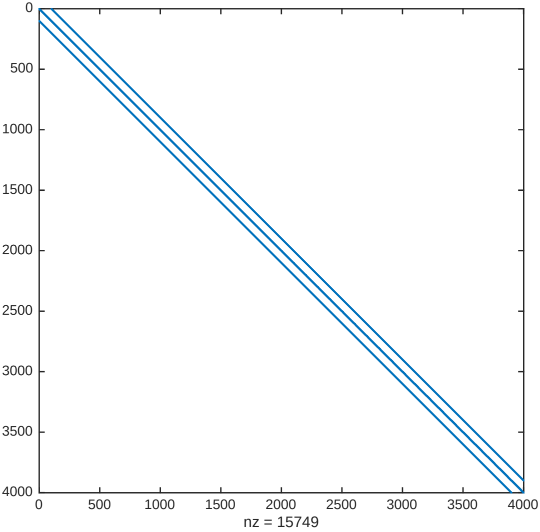
$$0 = \partial_y v(a, \underline{y}) = \partial_y v(a, \bar{y})$$

It doesn't matter whether you solve ODEs or PDEs

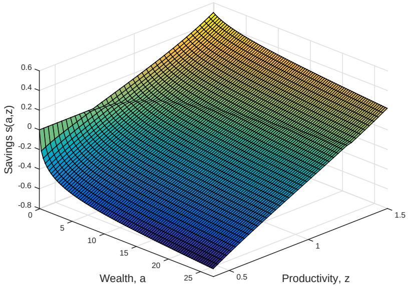
$\Rightarrow$ everything generalizes

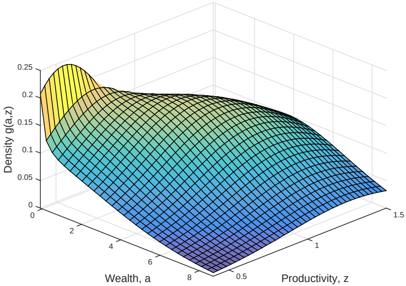
`http://www.princeton.edu/~moll/HACTproject/huggett_diffusion_partialeq.m`

Visualization of $\mathbf{A}$ (output of `spy(A)` in Matlab)



Saving Policy Function and Stationary Distribution





Summary: Stationary Equilibrium

- Can always write as

$$
\begin{aligned}
\rho \mathbf{v} &= \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v}, \mathbf{p})\mathbf{v} \\
0 &= \mathbf{A}(\mathbf{v}, \mathbf{p})^{\top} \mathbf{g} \\
0 &= \mathbf{F}(\mathbf{p}, \mathbf{g})
\end{aligned}
$$

  where $\mathbf{p}$ is a vector of prices.

Accuracy of Finite Difference Method

Accuracy of Finite Difference Method?

Two experiments:

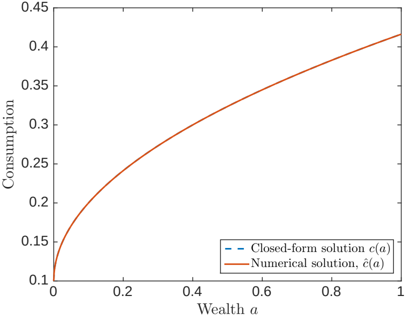
1. special case: comparison with closed-form solution
2. general case: comparison with numerical solution computed using very fine grid

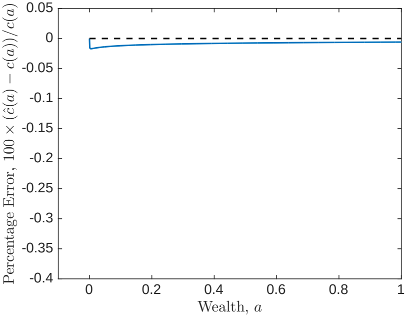
Accuracy of Finite Difference Method, Experiment 1

- see `http://www.princeton.edu/~moll/HACTproject/HJB_accuracy1.m`
- Achdou et al. (2017) get closed-form solution if
  - exponential utility $u'(c) = c^{-\theta c}$
  - no income risk and $r = 0$ so that $\dot{a} = y - c$ (and $a \geq 0$)

$$\Rightarrow \qquad s(a) = -\sqrt{2\nu a}, \qquad c(a) = y + \sqrt{2\nu a}, \qquad \nu := \frac{\rho}{\theta}$$

- Accuracy with $I = 1000$ grid points ($\widehat{c}(a)$ = numerical solution)



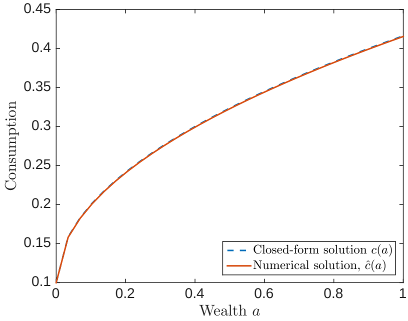


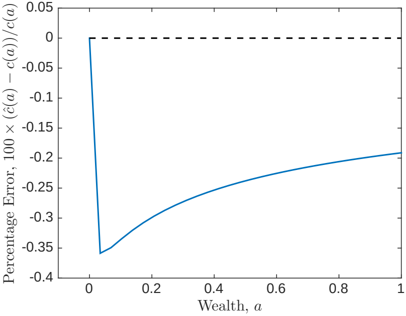
Accuracy of Finite Difference Method, Experiment 1

- see `http://www.princeton.edu/~moll/HACTproject/HJB_accuracy1.m`
- Achdou et al. (2017) get closed-form solution if
  - exponential utility $u'(c) = c^{-\theta c}$
  - no income risk and $r = 0$ so that $\dot{a} = y - c$ (and $a \geq 0$)

$$\Rightarrow \qquad s(a) = -\sqrt{2\nu a}, \qquad c(a) = y + \sqrt{2\nu a}, \qquad \nu := \frac{\rho}{\theta}$$

- Accuracy with $I = 30$ grid points ($\widehat{c}(a)$ = numerical solution)





Accuracy of Finite Difference Method, Experiment 2

- see `http://www.princeton.edu/~moll/HACTproject/HJB_accuracy2.m`
- Consider HJB equation with continuum of income types

$$\rho v(a,y) = \max_c \; u(c) + \partial_a v(a,y)(y + ra - c) + \mu(y)\partial_y v(a,y) + \frac{\sigma^2(y)}{2}\partial_{yy} v(a,y)$$

- Compute twice:
  1. with very fine grid: $I = 3000$ wealth grid points
  2. with coarse grid: $I = 300$ wealth grid points

  then examine speed-accuracy tradeoff (accuracy = error in agg C)

	Speed (in secs)	Aggregate C
$I = 3000$	0.916	1.1541
$I = 300$	0.076	1.1606
row 2/row 1	0.0876	1.005629

- i.e. going from $I = 3000$ to $I = 300$ yields $> 10\times$ speed gain and 0.5% reduction in accuracy (but note: even $I = 3000$ very fast)
- Other comparisons? Feel free to play around with `HJB_accuracy2.m`

Transition Dynamics/MIT Shocks

Transition Dynamics

Do Aiyagari version of the model

$$r(t) = F_K(K(t), 1) - \delta, \qquad w(t) = F_L(K(t), 1) \tag{P}$$

$$K(t) = \int a g_1(a,t)da + \int a g_2(a,t)da \tag{K}$$

$$\begin{aligned} \rho v_j(a,t) = &\max_c \; u(c) + \partial_a v_j(a,t)(w(t)z_j + r(t)a - c) \\ &+ \lambda_j(v_{-j}(a,t) - v_j(a,t)) + \partial_t v_j(a,t), \end{aligned} \tag{HJB}$$

$$\partial_t g_j(a,t) = -\partial_a[s_j(a,t)g_j(a,t)] - \lambda_j g_j(a,t) + \lambda_{-j} g_{-j}(a,t), \tag{KF}$$

$$s_j(a,t) = w(t)z_j + r(t)a - c_j(a,t), \quad c_j(a,t) = (u')^{-1}(\partial_a v_j(a,t))$$

- Given initial condition $g_{j,0}(a)$, the two PDEs (HJB) and (KF) together with (P) and (K) fully characterize equilibrium.

Transition Dynamics

- Recall discretized equations for stationary equilibrium

$$\rho \mathbf{v} = \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v})\mathbf{v}$$
$$0 = \mathbf{A}(\mathbf{v})^{\top}\mathbf{g}$$

- Transition dynamics
    - denote $v_{i,j}^n = v_j(a_i, t^n)$ and stack into $\mathbf{v}^n$
    - denote $g_{i,j}^n = g_j(a_i, t^n)$ and stack into $\mathbf{g}^n$

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

- Terminal condition for $\mathbf{v}$: $\mathbf{v}^N = \mathbf{v}_\infty$ (steady state)
- Initial condition for $\mathbf{g}$: $\mathbf{g}^1 = \mathbf{g}_0$.

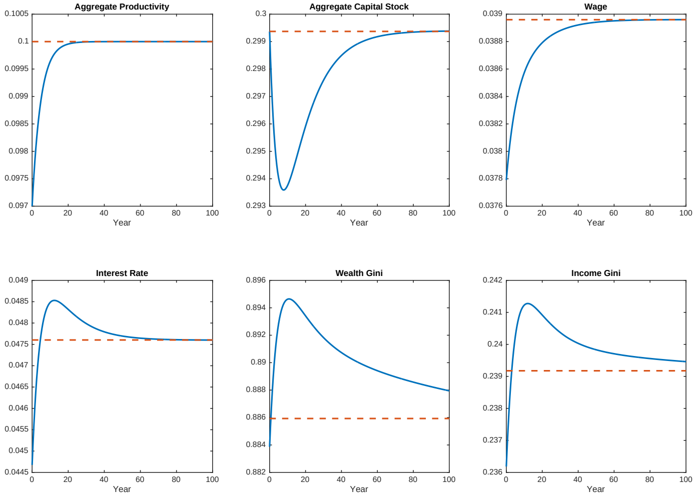
Transition Dynamics

- (HJB) looks forward, runs backwards in time
- (KF) looks backward, runs forward in time
- Algorithm: Guess $K^0(t)$ and then for $\ell = 0, 1, 2, ...$
  1. find prices $r^\ell(t)$ and $w^\ell(t)$
  2. solve (HJB) backwards in time given terminal cond'n $v_{j,\infty}(a)$
  3. solve (KF) forward in time given given initial condition $g_{j,0}(a)$
  4. Compute $S^\ell(t) = \int a g_1^\ell(a,t)da + \int a g_2^\ell(a,t)da$
  5. Update $K^{\ell+1}(t) = (1-\xi)K^\ell(t) + \xi S^\ell(t)$ where $\xi \in (0, 1]$ is a relaxation parameter

An MIT Shock

- Modification: $Y_t = F_t(K, L) = A_t K^\alpha L^{1-\alpha}$, $dA_t = \nu(\bar{A} - A_t)dt$

`http://www.princeton.edu/~moll/HACTproject/aiyagari_poisson_MITshock.m`



Stopping Time Problems

Stopping Time Problems

- In lots of problems in economics, agents have to choose an optimal stopping time
- Quite often these problems entail some form of non-convexity
- Examples:
  - how long should a low productivity firm wait before it exits an industry?
  - how long should a firm wait before it resets its prices?
  - when should you exercise an option?
  - etc... Stokey's book is all about these kind of problems
- These problems are very awkward in discrete time because you run into integer problems
- Big payoff from working in continuous time
- Next: flexible algorithm for solving such problems, also works if don't have simple threshold rules and with states > 1

Exercising an Option (Stokey, Ch. 6)

- Plant has profits
$$\pi(x_t)$$
- x_t: state variable = stand in for demand, plant capacity etc
$$dx_t = \mu(x_t)dt + \sigma(x_t)dW_t$$
where $dW_t := \lim_{\Delta t \to 0} \varepsilon\sqrt{\Delta t}, \varepsilon \sim \mathcal{N}(0,1)$
- Can shut down plant at any time, get scrap value $S(x_t)$, but cannot reopen
- Problem: choose stopping time τ to solve
$$v(x_0) = \max_{\tau \geq 0} \left\{ \mathbb{E}_0 \int_0^\tau e^{-\rho t}\pi(x_t)dt + e^{-\rho\tau}S(x_\tau) \right\}$$
- Assumptions to make sure $\tau^* < \infty$:
$$\pi'(x) > 0,\ \mu(x) < 0,\ \lim_{x \downarrow -\infty} \left(\frac{\pi(x)}{\rho} - S(x) \right) < 0,\ \lim_{x \uparrow +\infty} \left(\frac{\pi(x)}{\rho} - S(x) \right) >$$
- Analytic solution if $\mu(x) = \bar{\mu}, \sigma(x) = \bar{\sigma}, S(x) = \bar{S}$, but not in general

Exercising an Option: Standard Approach

- Assume scrap value is independent of x: $S(x) = \bar{S}$
- Optimal policy = threshold rule: exit if x_t falls below $\underline{x}$
- Standard approach (see e.g. Stokey, Ch.6):

$$\rho v(x) = \pi(x) + \mu(x)v'(x) + \frac{\sigma^2(x)}{2}v''(x), \qquad x > \underline{x}$$

with "value matching" and "smooth pasting" at $\underline{x}$:

$$v(\underline{x}) = \bar{S}, \qquad v'(\underline{x}) = 0$$

- but things more complicated if S depends on x or if dimension > 1
- $\Rightarrow$ can't use threshold property
- want algorithm that works also in those cases

Exercising an Option: HJBVI Approach

- Denote X = set of x such that don't exit:

$$x \in X: \quad v(x) \geq S(x), \quad \rho v(x) = \pi(x) + \mu(x)v'(x) + \frac{\sigma^2(x)}{2}v''(x)$$

$$x \notin X: \quad v(x) = S(x), \quad \rho v(x) \geq \pi(x) + \mu(x)v'(x) + \frac{\sigma^2(x)}{2}v''(x)$$

- Can write compactly as:

$$\min\left\{\rho v(x) - \pi(x) - \mu(x)v'(x) - \frac{\sigma^2(x)}{2}v''(x), v(x) - S(x)\right\} = 0 \quad (*)$$

- Note: have used that following two statements are equivalent
  1. for all x, either $f(x) \geq 0, g(x) = 0$ or $f(x) = 0, g(x) \geq 0$
  2. $\min\{f(x), g(x)\} = 0$ for all x
- $(*)$ is called "HJB variational inequality" (HJBVI)
- Important: did not impose smooth pasting
  - instead, it's a result: if $\bar{S}$, can prove that $(*)$ implies $v'(\underline{x}) = 0$
  - see e.g. Oksendal `http://th.if.uj.edu.pl/~gudowska/dydaktyka/Oksendal.pdf` (who calls "smooth pasting" "high contact (or smooth fit) principle")

Finite Difference Scheme for solving HJBVI

- Codes

  http://www.princeton.edu/~moll/HACTproject/option_simple_LCP.m,

  http://www.mathworks.com/matlabcentral/fileexchange/20952

- Main insight: discretized HJBVI = Linear Complementarity Problem (LCP) https://en.wikipedia.org/wiki/Linear_complementarity_problem

- Prototypical LCP: given matrix $\mathbf{B}$ and vector q, find z such that

$$
\begin{aligned}
z'(\mathbf{B}z + q) &= 0 \\
z &\geq 0 \\
\mathbf{B}z + q &\geq 0
\end{aligned}
$$

- There are many good LCP solvers in Matlab and other languages

- Best one I've found if $\mathbf{B}$ large but sparse (Newton-based):

  http://www.mathworks.com/matlabcentral/fileexchange/20952

Finite Difference Scheme for solving HJBVI

- Recall HJBVI

$$\min\left\{\rho v(x) - \pi(x) - \mu(x)v'(x) - \frac{\sigma^2(x)}{2}v''(x), v(x) - S(x)\right\} = 0$$

- Without exit, discretize as

$$\rho v_i = \pi_i + \mu_i (v_i)' + \frac{\sigma_i^2}{2}(v_i)'' \qquad \Leftrightarrow \qquad \rho v = \pi + \mathbf{A}v$$

- With exit:

$$\min\{\rho v - \pi - \mathbf{A}v, v - S\} = 0$$

- Equivalently:

$$\begin{aligned}(v - S)'(\rho v - \pi - \mathbf{A}v) &= 0\\ v &\geq S\\ \rho v - \pi - \mathbf{A}v &\geq 0\end{aligned}$$

- But this is just an LCP with $z = v - S$, $\mathbf{B} = \rho\mathbf{I} - \mathbf{A}$, $q = -\pi + \mathbf{B}$!!

Generalization: Menu Cost Model

- Work in progress: menu cost model (Golosov-Lucas) via HJBVI
  - HANK + menu cost model + aggregate shocks

Multiple Assets

Solution Method in Deterministic Version

$$\max_{\{c_t, d_t\}_{t \geq 0}} \int_0^\infty e^{-\rho t} u(c_t) dt \quad \text{s.t.}$$

$$\dot{b}_t = y + r^b b_t - d_t - \chi(d_t, a_t) - c_t$$

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

$$a_t \geq \underline{a}, \quad b_t \geq \underline{b}$$

- a_t: illiquid assets
- b_t: liquid assets
- c_t: consumption
- y: individual income
- d_t: deposits into illiquid account
- χ: transaction cost function $\chi(d, a) = \chi_0 |d| + \frac{\chi_1}{2} \left(\frac{d}{a}\right)^2 a$

No uncertainty, but easily extended to y=Markov process

How to “upwind” with two endogenous states

- HJB equation

$$\rho v(a,b) = \max_c \; u(c) + \partial_b v(a,b)(y + r^b b - d - \chi(d,a) - c) \\ + \partial_a v(a,b)(d + r^a a)$$

- FOC for d: $(1+\chi_d(d,a))\partial_b v = \partial_a v$

$$\Rightarrow \quad d = \left(\frac{\partial_a v}{\partial_b v} - 1 + \chi_0\right)^{-} \frac{a}{\chi_1} + \left(\frac{\partial_a v}{\partial_b v} - 1 - \chi_0\right)^{+} \frac{a}{\chi_1}$$

- Applying standard upwind scheme

$$\rho v_{i,j} = u(c_i) + \frac{v_{i+1,j} - v_{i,j}}{\Delta b}(s^b_{i,j})^+ + \frac{v_{i,j} - v_{i-1,j}}{\Delta b}(s^b_{i,j})^+ \\ + \frac{v_{i,j+1} - v_{i,j}}{\Delta a}(s^a_{i,j})^+ + \frac{v_{i,j} - v_{i,j-1}}{\Delta a}(s^a_{i,j})^-$$

where e.g. $s^b_{i,j} = y + r^b b_i - d_{i,j} - \chi(d_{i,j}, a_j) - c_{i,j}$

- Hard: $d_{i,j}$ depends on forward/backward choice for $\partial_b v_{i,j}, \partial_a v_{i,j}$

How to "upwind" with two endogenous states

- Convenient trick: "splitting the drift"

$$\begin{aligned}\rho v(a,b) = \max_c \; u(c) &+ \partial_b v(a,b)(y + r^b b - c) \\ &+ \partial_b v(a,b)(-d - \chi(d,a)) \\ &+ \partial_a v(a,b) d \\ &+ \partial_a v(a,b) r^a a\end{aligned}$$

  and upwind each term separately
- Can check this satisfies Barles-Souganidis monotonicity condition
- For an application, see

  `http://www.princeton.edu/~moll/HACTproject/two_asset_kinked.pdf`

  `http://www.princeton.edu/~moll/HACTproject/two_asset_kinked.m`

  Subroutines

  `http://www.princeton.edu/~moll/HACTproject/two_asset_kinked_cost.m`

  `http://www.princeton.edu/~moll/HACTproject/two_asset_kinked_FOC.m`