“Fundamentals of Heterogeneous Agent Models” Examples from Household Life-Cycle Models

James Graham (University of Sydney)

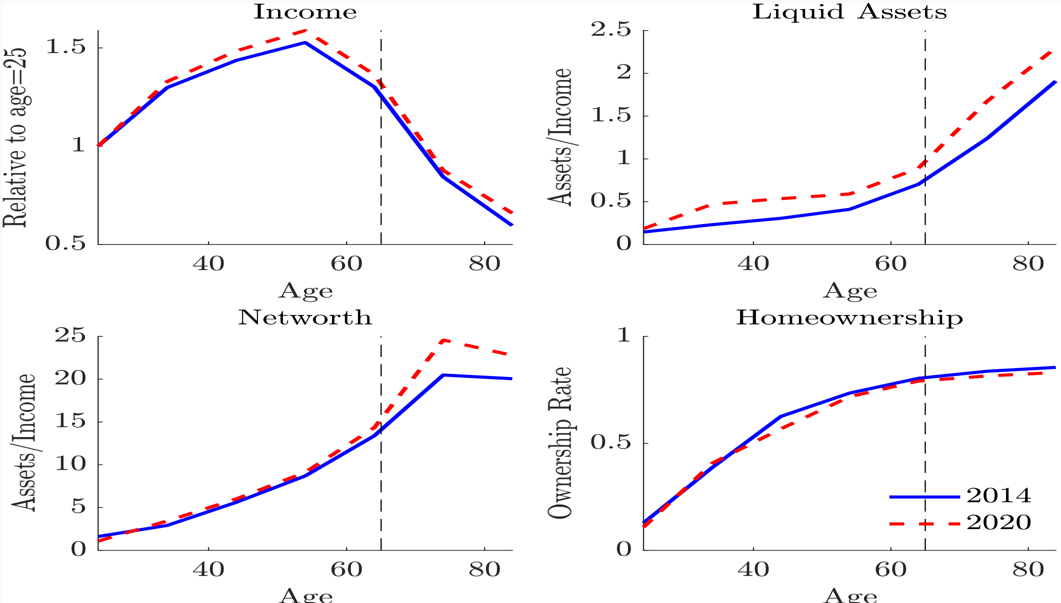
Continuing Education in Macroeconometrics Workshop
8 November, 2023

Introduction

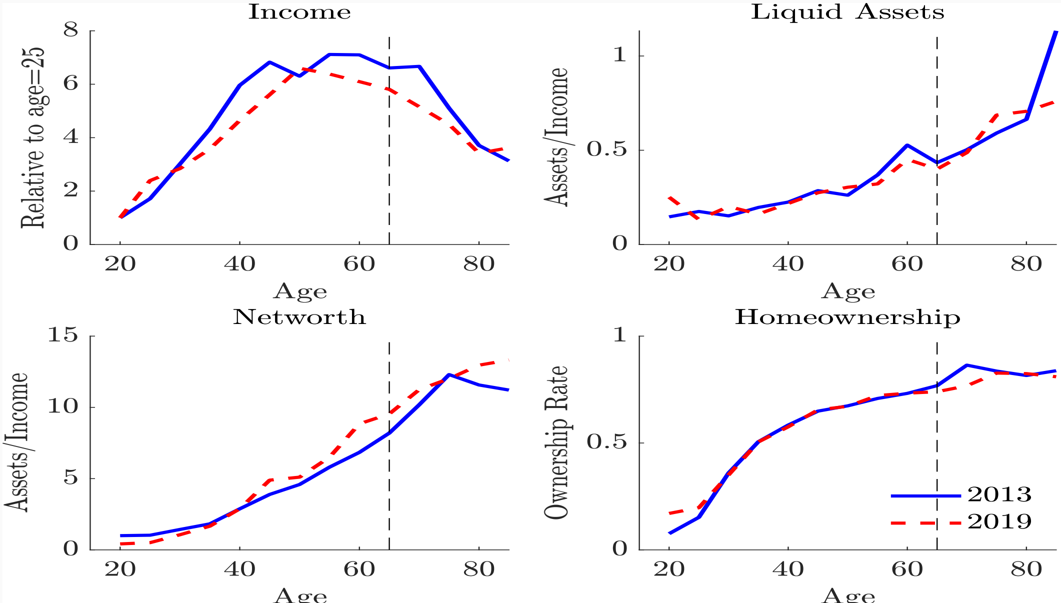
Motivation

- Many research questions ask: "how does **X** vary over the life-cycle?"
- Typically want to understand some combination of income and wealth dynamics
- Many varied examples from recent research in Australia:
  - Age, industry, and unemployment risk during lock-downs (Graham and Ozbilgin, 2021)
  - Housing, landlords, and negative gearing (Cho, Li, Uren, 2023)
  - Income contingent student loans and life-cycle income (Hua and Kudrna, 2023)
  - Life-cycle earnings risk and insurance (Tin and Tran, 2023)
  - Means-testing pension income (Kudrna, Tran, and Woodland, 2023)
- Where to start? Publicly available cross-section or panel data:
  - Survey of Income and Housing (cross-section; Australia)
  - HILDA (panel; Australia)
  - Survey of Consumer Finances (cross-section; USA)
  - PSID (panel; USA)

Survey of Income and Housing (Australia)



Survey of Consumer Finances (USA)



Codes and Toolboxes for Solving Heterogeneous Agent Models

- In this lecture, build up the basics of life-cycle modeling techniques
  - All codes and lecture slides available on my webpage
- But many other useful resources out there on the web:
  - Robert Kirkby's VFI Toolkit: Solve everything on a grid!
  - Chris Carroll's Econ-ARK: library of heterogeneous agent models
  - Benjamin Moll's various codes: heterogeneous agent models in continuous time

A Simple Life-Cycle Model Solved By Hand

A Simple Life-Cycle Model Solved By Hand

- Let's start from a simple model that we can solve by hand

- Consider an individual household that maximizes life-time utility
- Chooses consumption and savings subject to life-time income

$$\begin{aligned}
\max_{c_1,c_2,c_3,a_2,a_3} \quad & \log c_1 + \beta \log c_2 + \beta^2 \log c_3 \\
\text{s.t.} \quad & c_1 + a_2 = y_1 + a_1 \\
& c_2 + a_3 = y_2 + a_2(1+r) \\
& c_3 = y_3 + a_3(1+r)
\end{aligned}$$

- Life-cycle outcomes depend on income $\{y_1, y_2, y_3\}$, parameters $\{\beta\}$, and rates $\{r\}$

A Simple Life-Cycle Model Solved By Hand

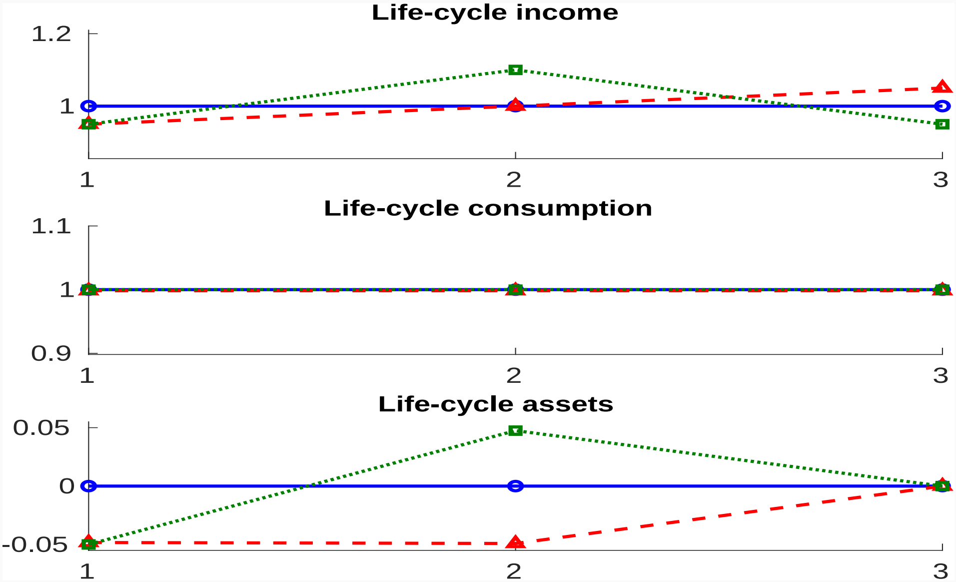
- Taking FOCs yields following consumption functions:

$$c_1 = \frac{1}{1+\beta+\beta^2}\left(a_1 + y_1 + \frac{y_2}{1+r} + \frac{y_3}{(1+r)^2}\right)$$
$$c_2 = \frac{\beta(1+r)}{1+\beta+\beta^2}\left(a_1 + y_1 + \frac{y_2}{1+r} + \frac{y_3}{(1+r)^2}\right)$$
$$c_3 = \frac{\beta^2(1+r)^2}{1+\beta+\beta^2}\left(a_1 + y_1 + \frac{y_2}{1+r} + \frac{y_3}{(1+r)^2}\right)$$

- And the following savings functions:

$$a_2 = y_1 - c_1 = \frac{\beta+\beta^2}{1+\beta+\beta^2}(a_1+y_1) - \frac{1}{1+\beta+\beta^2}\left(\frac{y_2}{(1+r)} + \frac{y_3}{(1+r)^2}\right)$$
$$a_3 = y_2 + a_2(1+r) - c_2 = \frac{\beta^2(1+r)}{1+\beta+\beta^2}\left(a_1 + y_1 + \frac{y_2}{(1+r)}\right) - \frac{1+\beta}{1+\beta+\beta^2}\frac{y_3}{(1+r)}$$

Model Life-Cycle Profiles [Code: mod0_analytical.m]



Simple Numerical Solutions

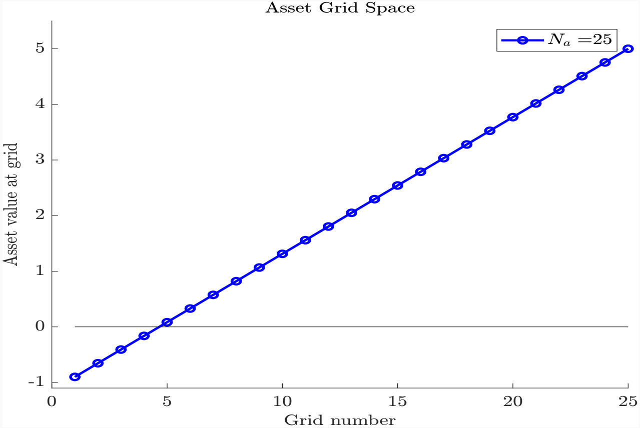
Simple Numerical Solutions of the Benchmark Model

- Need to prepare ourselves to solve more complex models
- Prepare our model for numerical solution methods
- Re-cast our simple model as a **Value Function Problem**

$$
\begin{aligned}
V_j(a) = \max_{c_j, a_{j+1}} \; & \log c_j + \beta V_{j+1}(a_{j+1}) \\
\text{s.t.} \quad & c_j + a_{j+1} = y_j + a(1+r)
\end{aligned}
$$

- Important objects:
  - $V_j(\cdot)$: is the **value function** at age j
  - $V_{j+1}(\cdot)$: is the **next-period value function**
  - a: assets are the **current state variable** for the problem
  - a_{j+1}: assets chosen as the **next-period state variable**

- Note that we are now solving for sets of **functions**: $\{V_j(a)\}_{j=1}^{3}$, $\{a_{j+1}(a)\}_{j=1}^{2}$, $\{c_j(a)\}_{j=1}^{3}$
- Our functions defined on the asset space $a \in \mathcal{R}$
- But to solve on a computer/numerically, we **discretize** this space
- For example, set upper and lower bounds $\{\underline{a}, \cdots, \overline{a}\}$, and a number of grid points N_a



Numerical Solution and "Life on the Grid"

- So now let's force the whole problem onto our grid:

$$V_j\left(\begin{bmatrix}\underline{a}\\ \vdots \\ \overline{a}\end{bmatrix}\right) = \max_{a_{j+1}} \log c_j + \beta V_{j+1}(a_{j+1})$$

$$\text{s.t.} \quad c_j = y_j + \begin{bmatrix}\underline{a}\\ \vdots \\ \overline{a}\end{bmatrix}(1+r) - a_{j+1}$$

$$a_{j+1} \in \{\underline{a}, \cdots, \overline{a}\}$$

- Notice that value functions $V_j(a)$, consumption functions $c_j(a)$, and asset choices $a_{j+1}(a)$ all live on the asset grid $\{\underline{a}, \cdots, \overline{a}\}$
- Since value function lives on the grid, V_{j+1} is known for all choices $a_{j+1} \in \{\underline{a}, \cdots, \overline{a}\}$

- Start from the **terminal value function**:

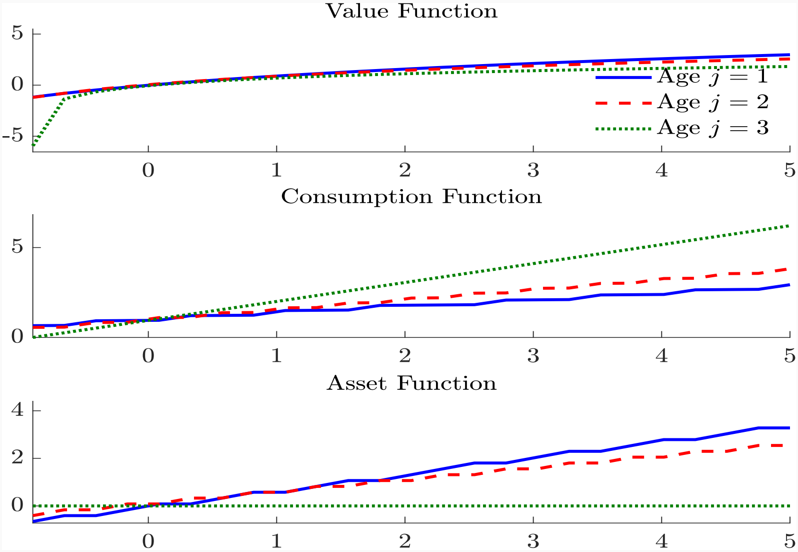
$$V_3(a) = \log c_3$$
$$\text{s.t.} \quad c_3 = y_3 + a(1+r)$$

```
%----------------------------
%% 1. Solve final period of life first: j = J
%----------------------------

jj = J;  % Set final age

aprime_j(:,jj) = zeros(Na, 1);          % No saving in last period of life
cons_j(:,jj)   = y_j(jj) + agrid.*(1+r);    % Consume everything in last period of life
V_j(:,jj)      = log( cons_j(:,jj) );       % Utility over consumption

% Often want to ensure that V_j is never -inf or nan.
scale      = -1e+10;     % Very large negative number
V_j(:,jj) = (cons_j(:,jj)>0).*V_j(:,jj) + (cons_j(:,jj)<=0).*scale;
```

- Now solve model via **backward iteration** on the value function

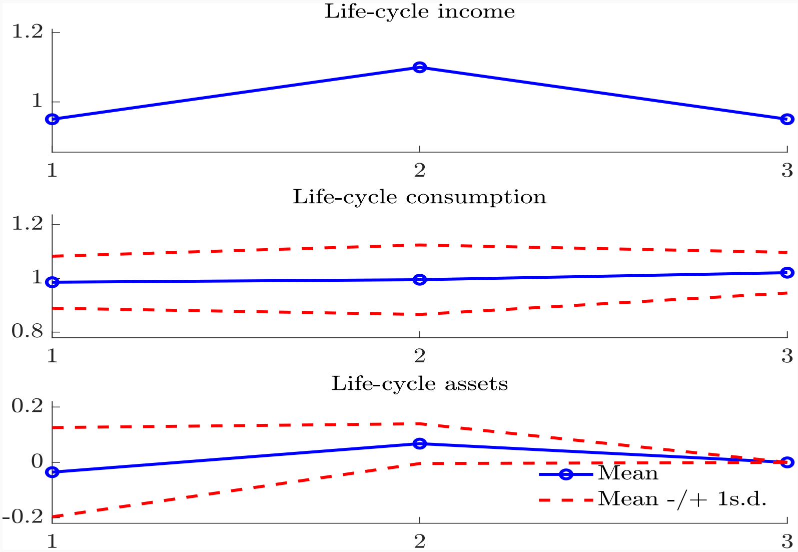
```
% Auxilliary functions
util_func = @(cons) log( cons );

% Loop *backwards* through ages
for jj = J-1:-1:1
    % Loop across each asset grid
    for aa = 1:Na
        aprime      = agrid;                                % Suppose savings choice comes from the asset grid
        cons        = y_j(jj) + agrid(aa).*(1+r) - aprime;  % Get consumption from the budget constraint
        cons        = (cons>0).*cons + (cons<=0).*1e-10;    % Trick to make sure consumption always positive

        V_jplus1   = V_j( :, jj+1); % Get tomorrow's value function where aprime=current asset grid
        V_on_agrid = util_func(cons) + beta.*V_jplus1;    % Compute today's value function

        % Find index on asset grid corresponding to maximum value function value
        [~, ix_max] = max(V_on_agrid);

        % Get optimal functions using position on asset grid with maximum value
        aprime_j(aa,jj) = aprime(ix_max);
        cons_j(aa,jj)   = cons(ix_max);
        V_j(aa,jj)      = V_on_agrid(ix_max);
    end
end
```

Policy Functions Across the Asset Grid [Code: `mod1_assets.m`]

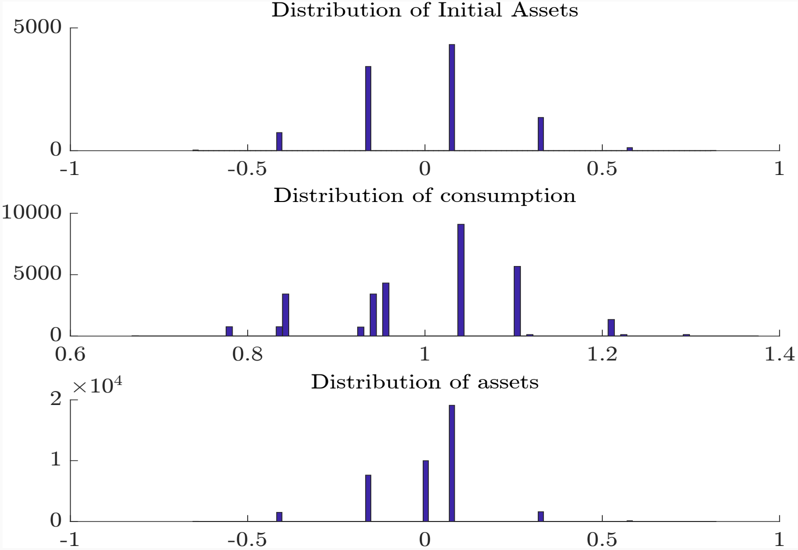


- Let's simulate a panel of households over the life-cycle

```
% Initialize sim vectors
sim_assets = zeros(Npop, J);
sim_cons   = zeros(Npop, J);

% Draw sample of initial assets
mu_a  = 0;     % Mean of asset dist in period 1
sig_a = 0.2;     % std dev of asset dist in period 1
sim_assets(:,1) = mu_a + sig_a.*randn(Npop,1);

% Force assets onto grid
[~,ix_max] = min(abs(agrid-sim_assets(:,1)'));
sim_assets(:,1) = agrid(ix_max);

% Simulate panel one age group at a time
for jj = 1:J
    % Loop over each household
    for nn=1:Npop
        % Get position in asset grid
        ix_max = (sim_assets(nn,jj)==agrid);

        % Find choice of assets/consumption for each household at each grid point
        sim_assets(nn,jj+1) = aprime_j(ix_max,jj);
        sim_cons(nn,jj)     = cons_j(ix_max,jj);
    end
end
```


Life-cycle income
1.2
1
1
2
3
Life-cycle consumption
1.2
1
0.8
1
2
3
Life-cycle assets
0.2
0
-0.2
1
2
3
Mean
Mean -/+ 1s.d.

Simulated Distributions [Code: `mod1_assets.m`]



Useful Computational Methods

Useful Computational Methods: Function interpolation

- Rather than force every function onto the asset grid, want to allow evaluation of functions off-grid i.e. between grid points in the state space
- Consider a value function on the grid:

$$V_j\left(\begin{bmatrix}\underline{a}\\ \vdots \\ \overline{a}\end{bmatrix}\right)$$

- Want to evaluate function at $\tilde{a}$ that lies between two asset grid points $[a_{[i]}, a_{[i+1]}]$
- Linear interpolation (i.e. weighted average of values at bounding grid points):

$$V_j(\tilde{a}) = V_j(a_{[i]}) \times \left(\frac{a_{[i+1]} - \tilde{a}}{a_{[i+1]} - a_{[i]}}\right) + V_j(a_{[i+1]}) \times \left(\frac{\tilde{a} - a_{[i]}}{a_{[i+1]} - a_{[i]}}\right)$$

Useful Computational Methods: Fast Asset Choices

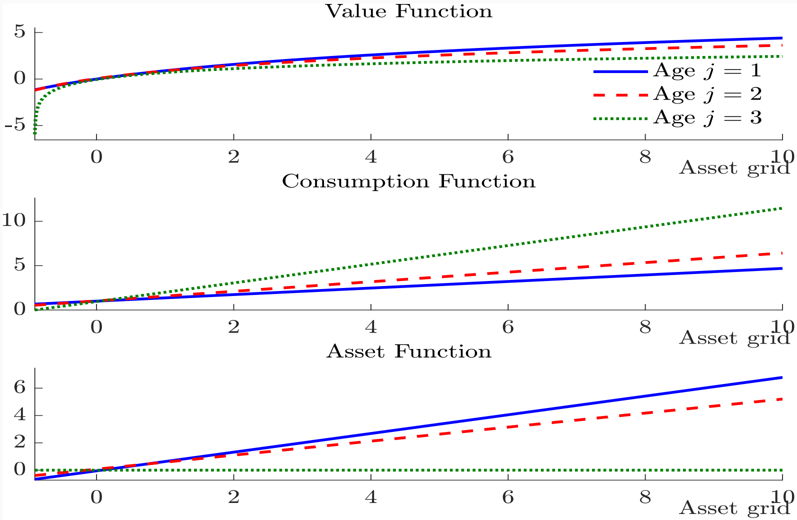
- Asset choices on the grid too slow:
    - At every grid point in the state vector (a) (=N_a grids)
    - Need to test value function at every grid point: $a_{j+1} \in \{a_{[1]}, \cdots, a_{[N_a]}\}$ (=N_a grids)
    - Find maximum across the values at each grid point (=$N_a \times N_a$ test points)
- Fast and robust method: Golden Section Search
    - Essentially, aim to bracket the maximum value between test points
    - At every grid point in the state vector (a, y) (=$N_a N_y$ grids)
    - Need only evaluate value function at small number of test points
    - Household can now choose assets freely: $a_{[1]} \leq a_{j+1} \leq a_{[N_a]}$
- Use Matlab function: `goldenx.m`

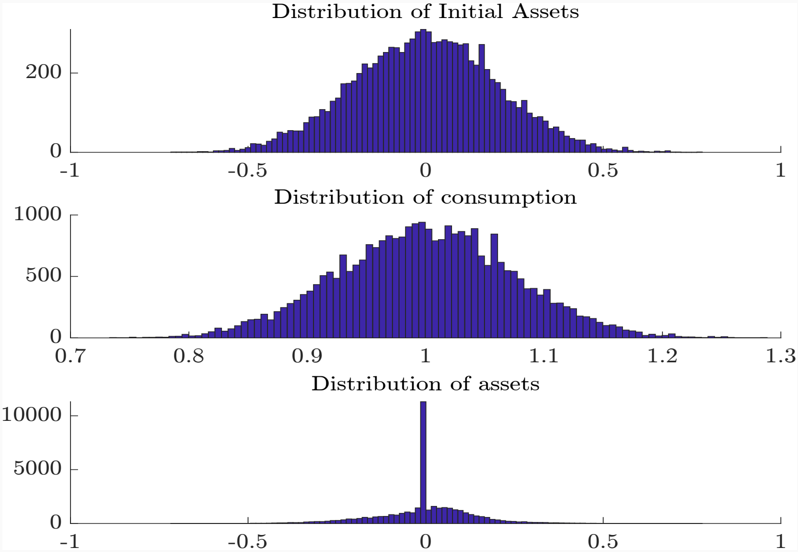
- Consider solving same model but with different numbers of grid points
- Compare speed: (1) model on grid, (2) interpolation+Golden Section Search

	Number of grid points, N_a			
	100	1,000	10,000	50,000
Model on grid	0.005s	0.05s	1.5s	26.7s
Interpolation+GSS	0.013s	0.02s	0.05s	0.17s

Policy Functions Across the Asset Grid [Code: mod2_interpolation.m]

- With more grid points ($N_a = 1000$), solved policy functions are much smoother




Distribution of Initial Assets
200
0
-1
-0.5
0
0.5
1
Distribution of consumption
1000
500
0
0.7
0.8
0.9
1
1.1
1.2
1.3
Distribution of assets
10000
5000
0
-1
-0.5
0
0.5
1

Extending Numerical Solutions to Multi-Dimensional Grids

Numerical Solution and "Life on Grid^2"

- We can easily generalize the model to think about multi-dimensional grids
- For example, suppose we want to consider the possibility of different assets and productivities (a, z)
- Consider second **discretized space** for productivity: $\{\underline{z}, \cdots, \bar{z}\}$ with N_z grid points
- Now need to create **all possible combinations** of (a, z)
  - Asset grid points $a_{[i]}$ for $i \in 1, \cdots, N_a$
  - Income grid points $z_{[k]}$ for $k \in 1, \cdots, N_z$

Numerical Solution and "Life on Grid²"

- Picture entire (a, z)-grid as combinations
- Note: *"the asset grid moves faster than productivity grid"*

$$\begin{bmatrix} a_{[1]}, & z_{[1]} \\ a_{[2]}, & z_{[1]} \\ \vdots & \vdots \\ a_{[N_a]}, & z_{[1]} \\ a_{[1]}, & z_{[2]} \\ a_{[2]}, & z_{[2]} \\ \vdots & \vdots \\ a_{[N_a]}, & z_{[2]} \\ \vdots & \vdots \\ a_{[1]}, & z_{[N_z]} \\ a_{[2]}, & z_{[N_z]} \\ \vdots & \vdots \\ a_{[N_a]}, & z_{[N_z]} \end{bmatrix}$$

Numerical Solution and "Life on the Grid[2]"

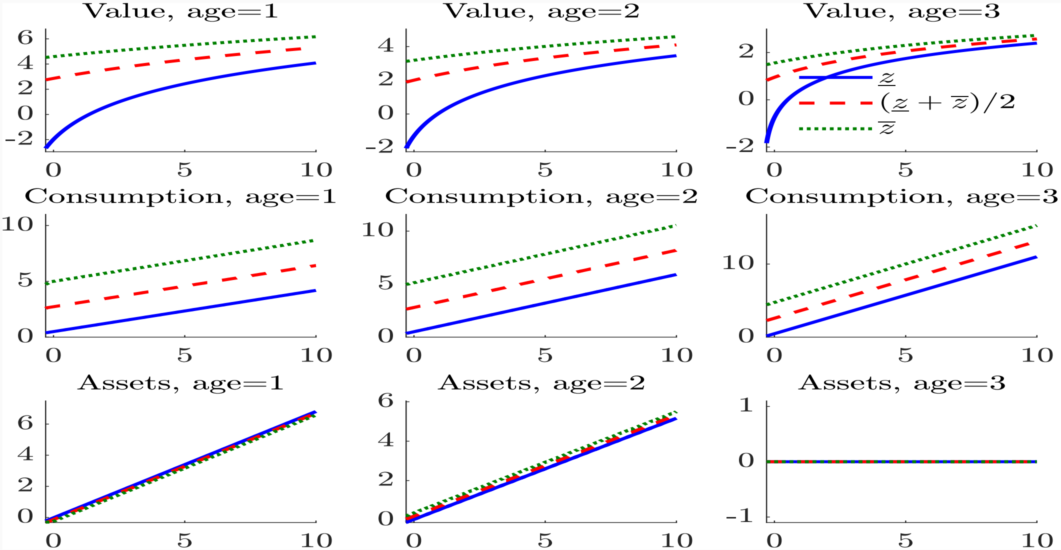
- So the new problem on the grid would be:

$$V_j\left(\begin{bmatrix} a_{[1]}, & z_{[1]} \\ a_{[2]}, & z_{[1]} \\ \vdots & \vdots \\ a_{[N_a]}, & z_{[N_y]} \end{bmatrix}\right) = \max_{a_{j+1}} \log c_j + \beta V_{j+1}(a_{j+1}, z_{j+1})$$

$$\text{s.t.} \quad c_j = y_j \times \begin{bmatrix} z_{[1]} \\ z_{[1]} \\ \vdots \\ z_{[N_y]} \end{bmatrix} + \begin{bmatrix} a_{[1]} \\ a_{[2]} \\ \vdots \\ a_{[N_a]} \end{bmatrix} (1+r) - a_{j+1}$$

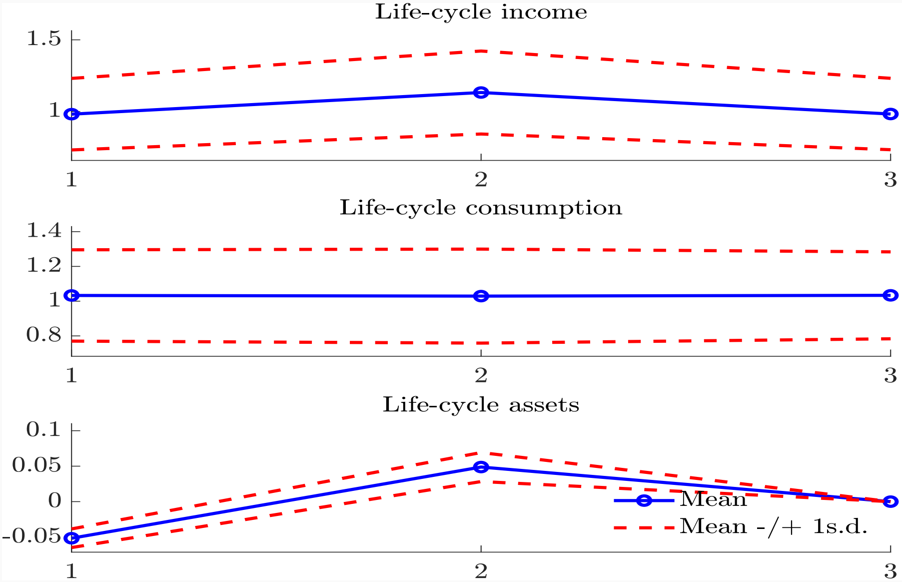
- Require some kind of rule for determining next period productivity z_{j+1}
- For now, suppose household believes constant income through time: $z_{j+1} = z$

Policy Functions Across the Gridspace [Code: mod3_income.m]



Simulated Life-Cycle

- Again let's simulate a panel of households
- Let's hold initial assets constant at $a_0 = 0$
- But simulate productivity z_j each period from a log-normal distribution
- Interpolate functions over assets and productivity
- Simulate 1000 households, compute statistics over the life-cycle


Life-cycle income
1.5
1
1
2
3
Life-cycle consumption
1.4
1.2
1
0.8
1
2
3
Life-cycle assets
0.1
0.05
0
-0.05
1
2
3
Mean
Mean -/+ 1s.d.

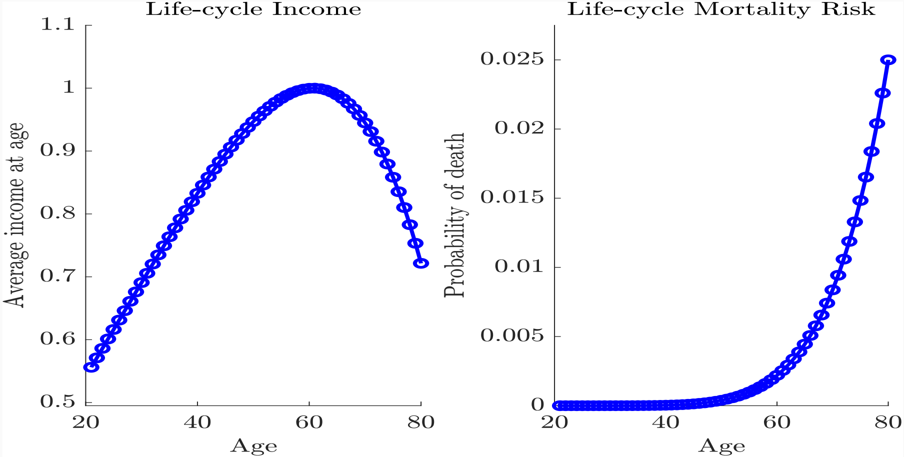
Adding Model Ingredients: Expanding our Repertoire

Common Model Ingredients

- Now consider mixing in more model ingredients
  - Longer life + mortality risk
  - Bequests
  - Income uncertainty + Borrowing constraints
  - Retirement
- Other extensions to consider...
  - Education choices
  - Occupation choices
  - Health
  - Investment activity
  - Housing choices
  - Migration
  - Household formation
  - Inter-generational interactions

Longer life and mortality risk

- Assume that households now live for J periods (e.g. $J = 60$)
- Assume “hump-shaped” profile of income y_j
- Assume households die with π_j, with increasing mortality risk with age

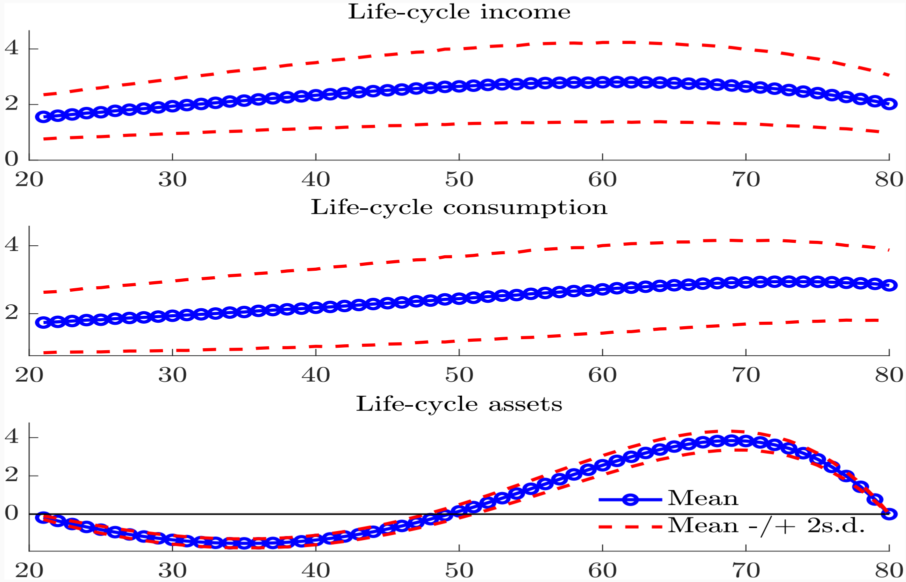


- Modify our model:

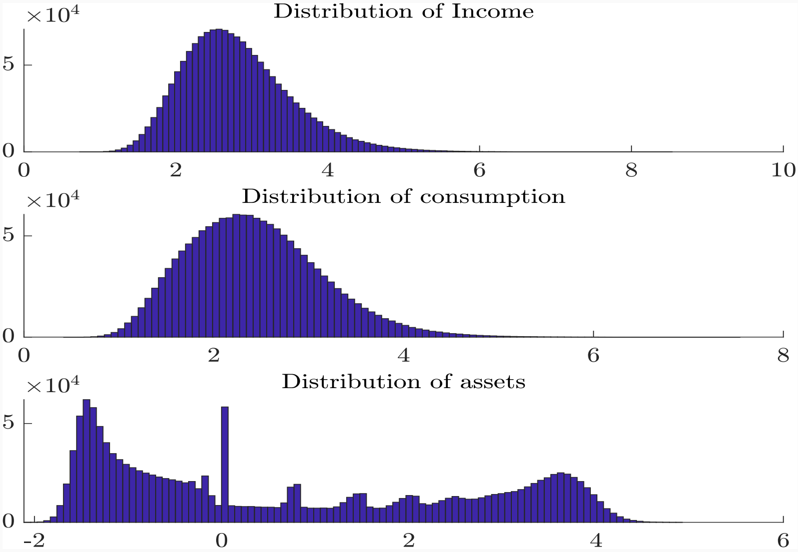
$$V_j(a,z) = \max_{c_j, a_{j+1}} \log c_j + \underbrace{\beta(1-\pi_j)V_{j+1}(a_{j+1}, z_{j+1})}_{\text{Future value given survival}} + \underbrace{\beta\pi_j \times 0}_{\text{Future value following death}}$$

$$\text{s.t.} \quad c_j + a_{j+1} = y_j \times z + a(1+r)$$

- And where we now solve for $j = [1, \cdots, J]$


Life-cycle income
4
2
0
20
30
40
50
60
70
80
Life-cycle consumption
4
2
20
30
40
50
60
70
80
Life-cycle assets
4
2
0
Mean
Mean -/+ 2s.d.
20
30
40
50
60
70
80

Simulated Distributions [Code: mod4_longerlife.m]



Bequests

Bequests

- Households also motivated by desire to provide for their children
- Several ways to model transfers of resources to children:
  - Inter-vivos transfers: active choice of bequests in each period of life
  - Bequests: passive transfer of remaining assets at death
- Bequests may be modelled to capture real world transfers at the end of life
- But more common to use bequests as mechanism to increase saving in old age

Bequests

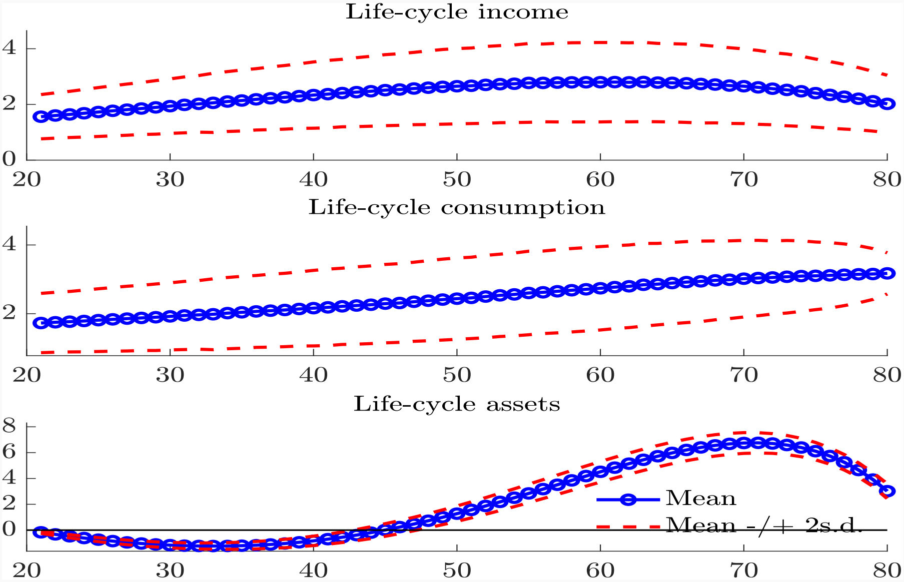
- Modify our model:

$$V_j(a,z) = \max_{c_j, a_{j+1}} \log c_j + \underbrace{\beta(1-\pi_j)V_{j+1}(a_{j+1}, z_{j+1})}_{\text{Future value given survival}} + \underbrace{\beta\pi_j W(a_{j+1})}_{\text{Future value following death}}$$

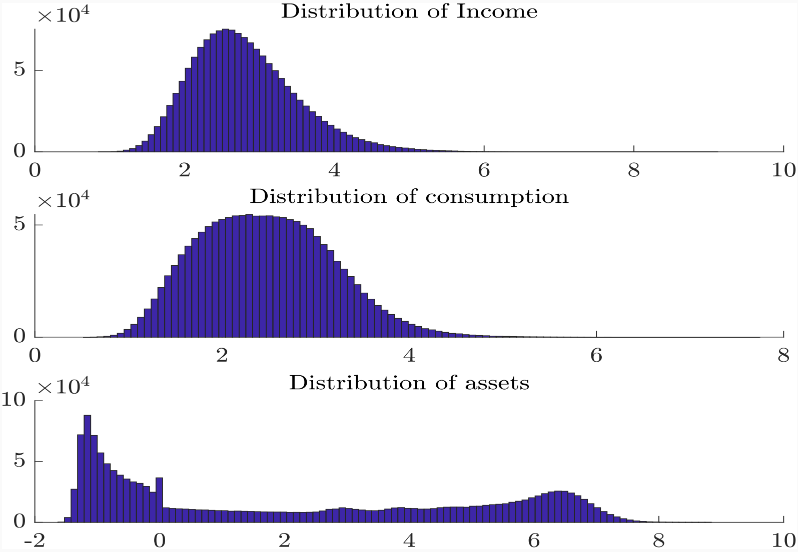
$$\text{s.t.} \quad c_j + a_{j+1} = y_j \times z + a(1+r)$$

$$W(a_{j+1}) = \psi \log a_{j+1}$$

- Bequests function $W(a_{j+1})$ captures utility of leaving wealth behind at death
- ψ captures strength of bequest motive


Life-cycle income
4
2
0
20 30 40 50 60 70 80
Life-cycle consumption
4
2
20 30 40 50 60 70 80
Life-cycle assets
8
6
4
2
0
20 30 40 50 60 70 80
Mean
Mean -/+ 2s.d.

Simulated Distributions [Code: `mod5_bequests.m`]



Income Uncertainty+Borrowing Constraints

Income Uncertainty+Borrowing Constraints

- So far, households oblivious to uncertainty in their income
- Now, suppose households ***aware*** that productivity follows a stochastic process
- Most common to assume a Markov chain. Can be generated in many ways e.g.:
    - Transitions between employment and unemployment
    - Discretized AR(1) process (see Tauchen, 1986; Rouwenhorst, 1995)
- Markov chain details:
    - Productivity can from a given set of values:

$$z \in \{z_1, z_2, \cdots, z_{N_z}\}$$

    - Probability transition matrix from $z \to z_{j+1}$

$$\Gamma_{z,z'} = \begin{bmatrix} \gamma_{1,1} & \gamma_{1,2} & \cdots & \gamma_{1,N_z} \\ \gamma_{2,1} & \gamma_{2,2} & \cdots & \gamma_{2,N_z} \\ \vdots & & & \vdots \\ \gamma_{N_z,1} & \gamma_{N_z,2} & \cdots & \gamma_{N_z,N_z} \end{bmatrix}$$

- Drop mortality risk and bequests for ease of notation
- Update our problem to account for uncertainty
- Also incorporate an explicit borrowing constraint: $a_{j+1} >= 0$

$$V_j(a, z) = \max_{c_j, a_{j+1}} \log c_j + \beta \underbrace{\mathbb{E}\left[V_{j+1}(a_{j+1}, z_{j+1})|z\right]}_{\text{Conditional expected value}}$$

$$\begin{aligned} \text{s.t.} \quad & c_j + a_{j+1} = y_j \times z + a(1+r) \\ & a_{j+1} \geq 0 \\ & z_{j+1} \sim \Gamma_{z,z'} \end{aligned}$$

- $\mathbb{E}$ characterizes expectations over evolution of z_{j+1} given the Markov chain

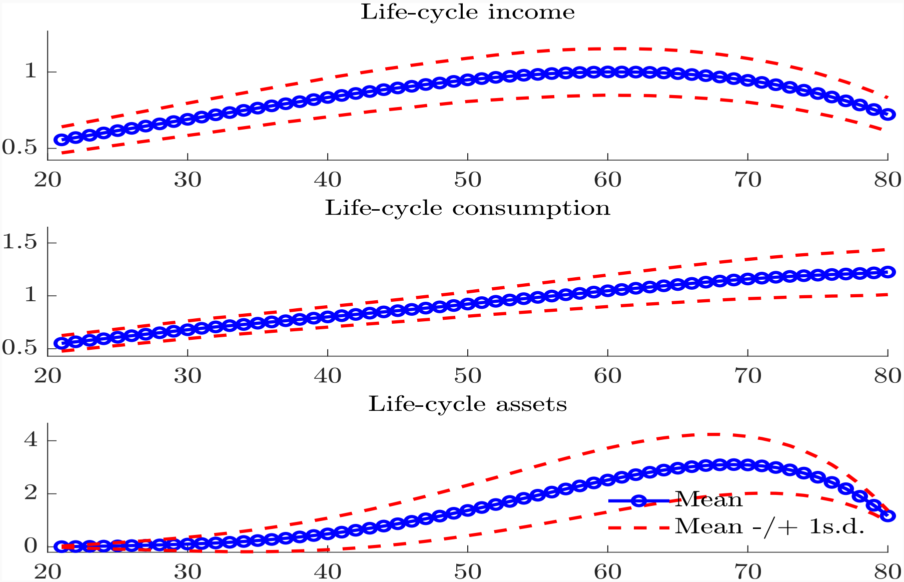
- Using our state-space notation:

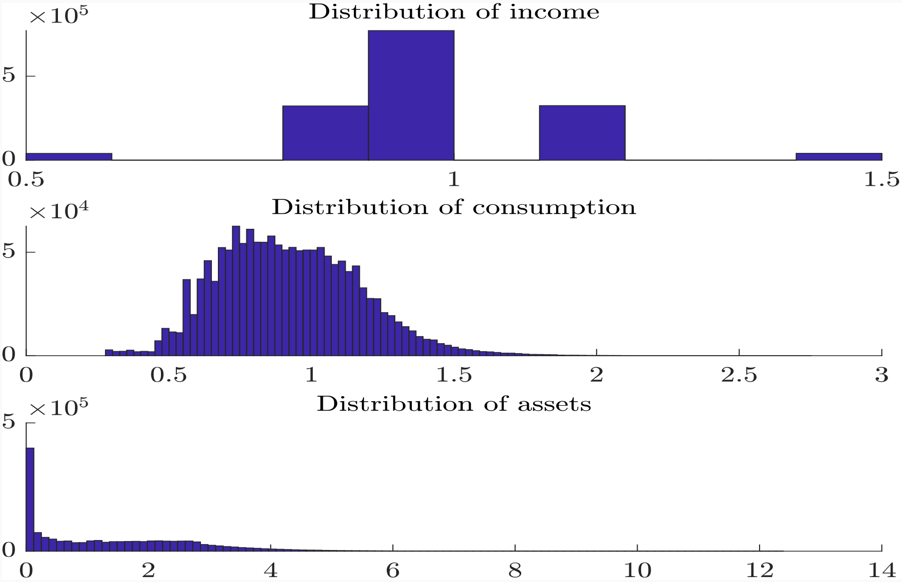
$$V_j\left(\begin{bmatrix} a_{[1]}, & z_{[1]} \\ a_{[2]}, & z_{[1]} \\ \vdots & \vdots \\ a_{[N_a]}, & z_{[N_y]} \end{bmatrix}\right) = \max_{c_j, a_{j+1}} \log c_j + \beta \times \underbrace{\overbrace{Q}^{\text{trans. prob. matrix}} \times V_{j+1}\left(\begin{bmatrix} a_{j+1}(a_{[1]}, z_{[1]}), & z_{[1]} \\ a_{j+1}(a_{[2]}, z_{[1]}), & z_{[1]} \\ \vdots & \vdots \\ a_{j+1}(a_{[N_a]}, z_{[N_y]}), & z_{[N_y]} \end{bmatrix}\right)}_{=\mathbb{E}\left[V_{j+1}(a_{j+1}, z_{j+1})|z\right]}$$

- The transition probability matrix Q is given by:

$$Q = \Gamma_{z,z'} \otimes \mathbb{I}_{N_a} = \begin{bmatrix} \gamma_{1,1} \times \mathbb{I}_{N_a} & \gamma_{1,2} \times \mathbb{I}_{N_a} & \cdots & \gamma_{1,N_z} \times \mathbb{I}_{N_a} \\ \gamma_{2,1} \times \mathbb{I}_{N_a} & \gamma_{2,2} \times \mathbb{I}_{N_a} & \cdots & \gamma_{2,N_z} \times \mathbb{I}_{N_a} \\ \vdots & & & \vdots \\ \gamma_{N_z,1} \times \mathbb{I}_{N_a} & \gamma_{N_z,2} \times \mathbb{I}_{N_a} & \cdots & \gamma_{N_z,N_z} \times \mathbb{I}_{N_a} \end{bmatrix}$$

- Tips and tricks:
    - Note that Q is a very **sparse matrix**
    - Use sparse matrix methods in computation: `Q = kron(Gamma_z, speye(Na))`
    - Dramatically cut memory use and speed up computation!
    - Pre-compute Q matrix just once, before iterating over the value function
    - Saves costly computation steps!
    - Interpolate over the **expected future value function**, $\mathbb{E}[V_{j+1}]$
    - Expected value function is a smooth object, easier to approximate (i.e. interpolate over) than the underlying value function itself
    - After solving household problem, need to draw from the Markov chain to simulate household productivity


Life-cycle income
1
0.5
20
30
40
50
60
70
80
Life-cycle consumption
1.5
1
0.5
20
30
40
50
60
70
80
Life-cycle assets
4
2
0
20
30
40
50
60
70
80
Mean
Mean -/+ 1s.d.


Distribution of income
$\times 10^5$
5
0
0.5
1
1.5
Distribution of consumption
$\times 10^4$
5
0
0
0.5
1
1.5
2
2.5
3
Distribution of assets
$\times 10^5$
5
0
0
2
4
6
8
10
12
14

Retirement

Retirement

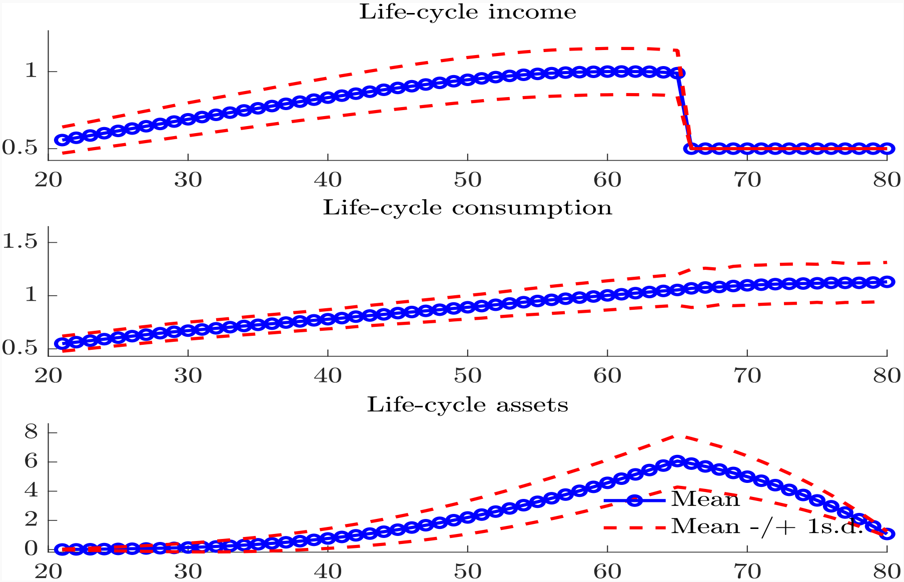
- So far, households earn deterministic life-cycle income + stochastic labour market income
- Now we want to account for retirement
- Many ways to do this, some more realistic than others
    - Exogenous retirement date + simple retirement income
    - Exogenous retirement date + accumulated retirement savings (e.g. superannuation)
    - Endogenous choice of retirement date

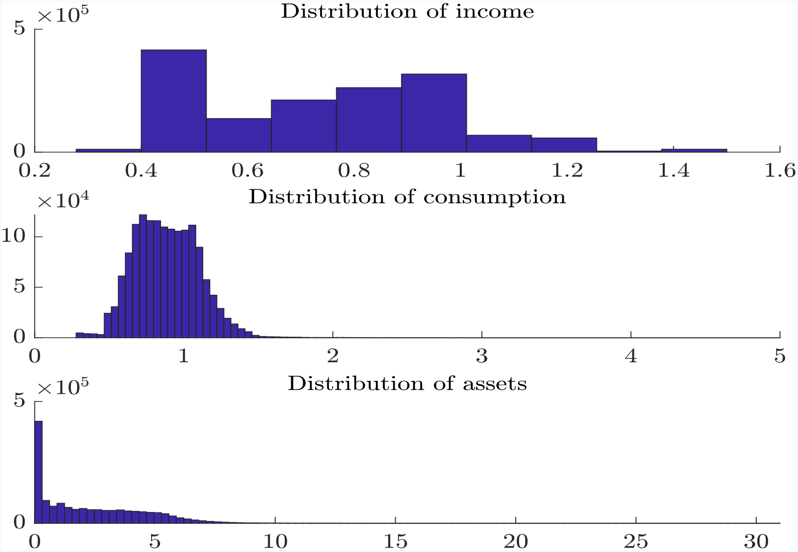
Retirement 1:
Exogenous retirement

- Suppose all households receive a constant old-age pension, ω_{ret}

$$
\begin{aligned}
V_j(a,z) = \max_{c_j, a_{j+1}} \; & \log c_j + \beta \mathbb{E}_j \left[V_{j+1}(a_{j+1}, z_{j+1}) | z \right] \\
\text{s.t.} \quad & c_j + a_{j+1} = m_j(z) + a(1+r) \\
& a_{j+1} \geq 0 \\
& z_{j+1} \sim \Gamma_{z,z'} \\
& m_j(z) = \begin{cases} y_j \times z & \text{if } j \leq J_{ret} \\ \omega_{ret} & \text{if } j > J_{ret} \end{cases}
\end{aligned}
$$

- Note: model code has age-dependent expectations, since productivity stops evolving during retirement


Life-cycle income
1
0.5
20
30
40
50
60
70
80
Life-cycle consumption
1.5
1
0.5
20
30
40
50
60
70
80
Life-cycle assets
8
6
4
2
0
20
30
40
50
60
70
80
Mean
Mean -/+ 1s.d.


Distribution of income
×10^5
5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
Distribution of consumption
×10^4
10
5
0
0
1
2
3
4
5
Distribution of assets
×10^5
5
0
0
5
10
15
20
25
30

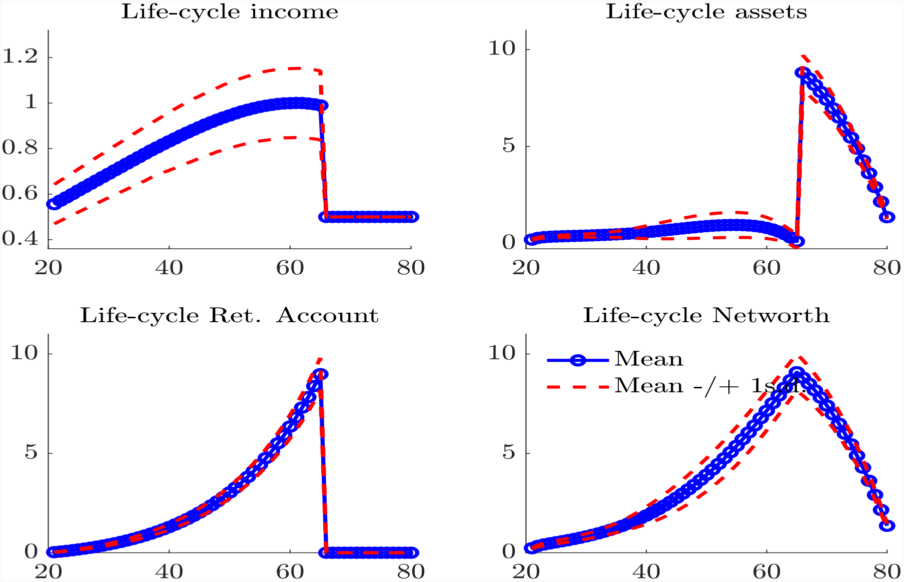
Retirement 2: Exogenous Retirement + Superannuation Accounts

Exogenous retirement + retirement account

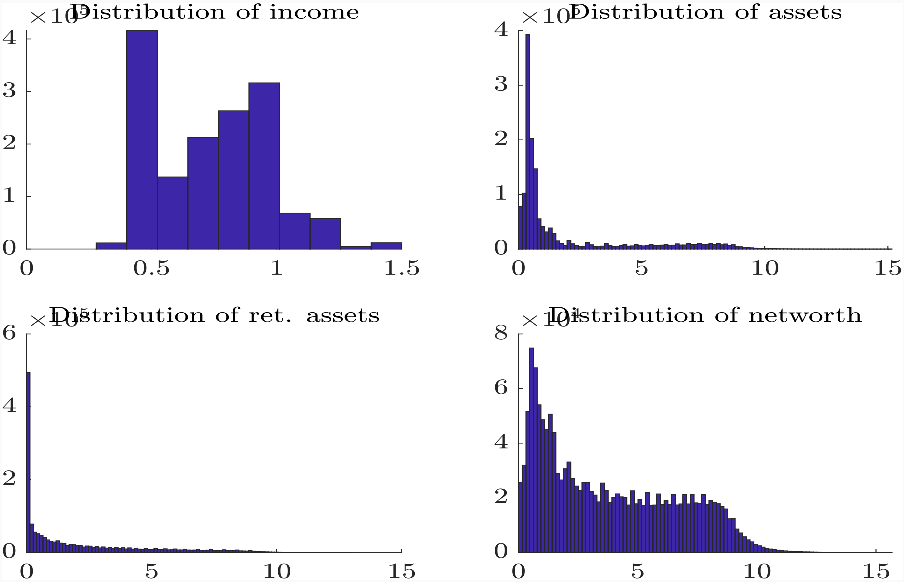
- Add new state variable k to track retirement account
- Fixed contribution rate τ out of working life income
- Fixed draw-down rate κ during retirement

$$
\begin{aligned}
V_j(a,k,z) = \max_{c_j, a_{j+1}} \ & \log c_j + \beta \mathbb{E}_j \left[V_{j+1}(a_{j+1}, k_{j+1}, z_{j+1}) | z \right] \\
\text{s.t.} \quad & c_j + a_{j+1} = m_j(k,z) + a(1+r) \\
& a_{j+1} \geq 0 \\
& z_{j+1} \sim \Gamma_{z,z'} \\
& m_j(k,z) = \begin{cases} (1-\tau) \times y_j \times z & \text{if } j \leq J_{ret} \\ \omega_{ret} + \kappa(1+r)k & \text{if } j > J_{ret} \end{cases} \\
& k_{j+1} = \begin{cases} k(1+r) + \tau \times y_j \times z & \text{if } j \leq J_{ret} \\ (1-\kappa)(1+r)k & \text{if } j > J_{ret} \end{cases}
\end{aligned}
$$

Simulated Life-Cycle [Code: mod8_retirementassets.m]



Simulated Distributions [Code: mod8_retirementassets.m]



Retirement 3: Endogenous Retirement

Endogenous retirement (via Discrete Choice Problem)

- Household problem when retired is:

$$V_j^R(a,z) = \max_{c_j, a_{j+1} \geq 0} \log c_j + \beta V_{j+1}^R(a_{j+1}, z)$$

$$\text{s.t.} \quad c_j + a_{j+1} = \omega + a(1+r)$$

- Household problem when working is:

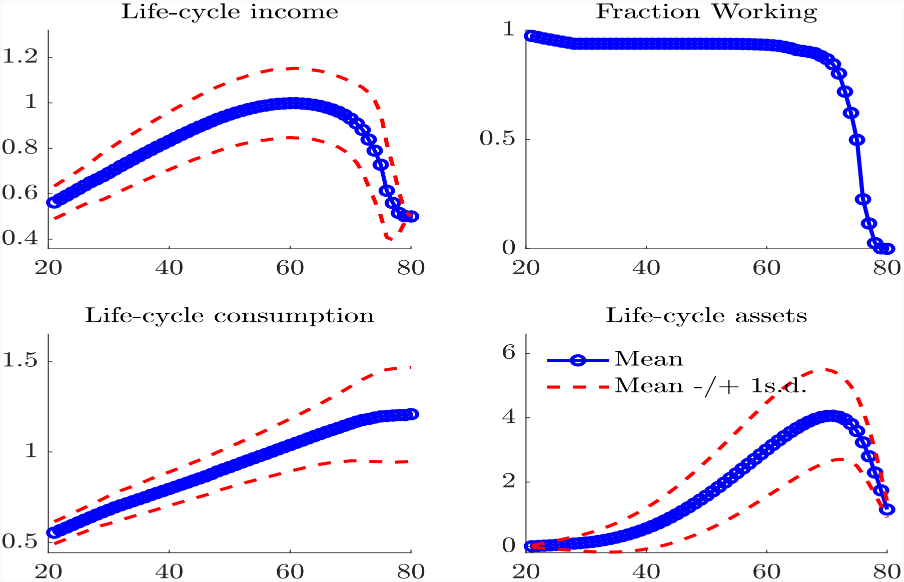
$$V_j^W(a,z) = \max_{c_j, a_{j+1} \geq 0} \log c_j \overbrace{-\chi}^{\text{work disutility}} + \beta \overbrace{\mathbb{E}\left[\tilde{V}_{j+1}(a_{j+1}, z_{j+1})|z\right]}^{\text{future choice of work vs. retire}}$$

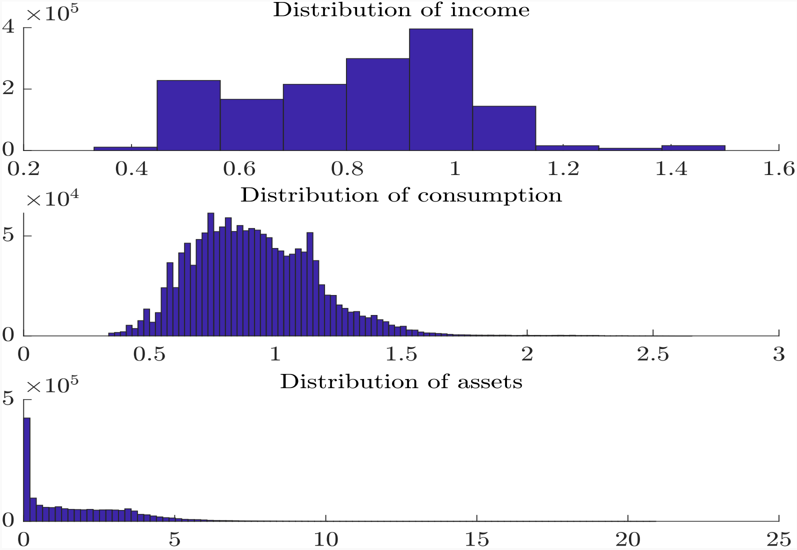
$$\text{s.t.} \quad c_j + a_{j+1} = y_j \times z + a(1+r)$$

$$z_{j+1} \sim \Gamma_{z,z'}$$

- While working, household makes **discrete choice** over working and retirement

$$\tilde{V}_j(a,z) = \max\left\{V_j^W(a,z), V_j^R(a,z)\right\}$$


Life-cycle income
1.2
1
0.8
0.6
0.4
20
40
60
80
Fraction Working
1
0.5
0
20
40
60
80
Life-cycle consumption
1.5
1
0.5
20
40
60
80
Life-cycle assets
Mean
Mean -/+ 1s.d.
6
4
2
0
20
40
60
80


Distribution of income
×10^5
4
2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
Distribution of consumption
×10^4
5
0
0
0.5
1
1.5
2
2.5
3
Distribution of assets
×10^5
5
0
0
5
10
15
20
25

Conclusion

Conclusion

- Heterogeneous agent life-cycle models an incredibly useful tool for macro research
- Useful for any and all questions addressing life-cycle economics
- The models are easily extended
- Models are typically very stable - useful for code development, model calibration, and research experiments
- Plenty of examples and code out there ready to try!