PhD Macroeconomics 1
11. Stochastic Dynamic Programming with Finite Horizons

Karl Whelan

School of Economics, UCD

Autumn 2023

Stochastic Dynamic Programming

- Suppose we altered our problem to be of the form:

$$\max_{u_t, x_t} E_0 \left[\sum_{t=0}^{T} \beta^t F(x_t, u_t, \epsilon_t) \right]$$

subject to

$$x_{t+1} = g(x_t, u_t, \epsilon_t)$$

where ϵ_t was a set of mean-zero random shocks from a known distribution.

- This looks like it must be a much more complicated problem but the logic of dynamic programming still holds and we have a Bellman equation of the form

$$V_t(x_t, \epsilon_t) = \max_{u_t} [F(x_t, u_t, \epsilon_t) + \beta E_t V_{t+1}(g(x_t, u_t, \epsilon_t))]$$

- The first-order conditions for this problem are obtained in a similar fashion to the deterministic case.

First-Order Conditions

- The first-order conditions take a familiar form

$$\frac{\partial F\left(x_t, u_t^*, \epsilon_t\right)}{\partial u} + \beta E_t\left[V_{t+1}'\left(g\left(x_t, u_t^*, \epsilon_t\right)\right)\frac{\partial g\left(x_t, u_t^*, \epsilon_t\right)}{\partial u}\middle| x_t\right] = 0$$

$$\begin{aligned} V_{t+1}'\left(x_{t+1}\right) \quad = \quad & \frac{\partial F\left(x_{t+1}, u_{t+1}^*, \epsilon_t\right)}{\partial x} \\ & + \beta E_{t+1}\left[V_{t+2}'\left(g\left(x_{t+1}, u_{t+1}^*, \epsilon_t\right)\right)\frac{\partial g\left(x_{t+1}, u_{t+1}^*\right)}{\partial x}\middle| x_{t+1}\right] \end{aligned}$$

Markov Uncertainty

- In our deterministic dynamic programming examples, we solved for value functions that depended on a single state variable. We will now look at examples in which the value function also depends on a second state variable, one that evolves exogenously and stochastically over time according to a Markov process.
- Specifically, we formulate our problem to include a stochastic state variable s_t so that our value function is

$$V_t(x_t, s_t) = \max_{u_t} \left[F(x_t, u_t, s_t) + \beta E_t \left[V_{t+1}(g(x_t, u_t, s_t), s_{t+1}) \mid s_t \right] \right.$$

- Because of the assumption that s_t is a Markov process, we don't need to condition on anything other than s_t when formulating probabilities of different possible values of s_{t+1}.
- Because of uncertainty, we cannot any longer be sure what the value function for next period will look like once we have chosen a value for our control variable, u_t. We need to average over all the possible values of s_{t-1} given the current value of s_t How would we calculate $E_t \left[V_{t+1}(g(x_t, u_t), s_{t+1}) \middle| s_t \right]$? This looks complicated but we can explain how to calculate it using a simplified example.

Two-State Markov Chain Uncertainty

- As we have discussed, in numerical implementations, it is not possible to calculate the value function at every possible value of the state variables.
- So let us decide to only (at first) calculate the value function for a limited number of values of our endogenous state variable x_t (assets in the life-cycle model). Call these values $(\gamma_1, \gamma_2, ..., \gamma_K)$.
- Let's also assume that s_t can take values of either $s_t = \mu_1$ or $s_t = \mu_2$ and its changes over time are determined by a 2-state Markov chain with transition matrix Π, with entries π_{ij} (the i'th row and j'th column) describing the probability of moving from $s_t = \mu_i$ to $s_{t+1} = \mu_j$.

$$\Pi = \begin{pmatrix} \pi_{11} & \pi_{12} \\ \pi_{21} & \pi_{22} \end{pmatrix}$$

where

$$\pi_{11} + \pi_{12} = \pi_{21} + \pi_{22} = 1$$

- Let's figure out how to calculate $E_t \left[V_{t+1} \left(\gamma_m, s_{t+1} \right) \middle| s_t \right]$ for all the values of γ_m on our grid.

Using Matrices to Store Value Functions

- Let

$$V_{mj}^{t+1} = V_{t+1}\left(x_{t+1} = \gamma_m, \mid s_{t+1} = \mu_j\right)$$

- We can summarise the value function at time $t+1$ with the following $K \times 2$ matrix.

$$V^{t+1} = \begin{pmatrix} V_{11}^{t+1} & V_{12}^{t+1} \\ V_{21}^{t+1} & V_{22}^{t+1} \\ . & . \\ . & . \\ . & . \\ V_{K1}^{t+1} & V_{K2}^{t+1} \end{pmatrix}$$

- We can use the elements of this matrix to calculate all the expected values $E_t \left[V_{t+1}\left(\gamma_m, s_{t+1}\right) \mid s_t \right]$ which we require.
- If $s_t = \mu_1$, then the expected value once $x_{t+1} = \gamma_m$ is

$$E_t V_{t+1}\left[\left(x_{t+1} = \gamma_m, \mid s_t = \mu_1\right)\right] = \pi_{11} V_{m1}^{t+1} + \pi_{12} V_{m2}^{t+1}$$

- If $s_t = \mu_2$, then this expected value is

$$E_t V_{t+1}\left[\left(x_{t+1} = \gamma_m, \mid s_t = \mu_2\right)\right] = \pi_{21} V_{m1}^{t+1} + \pi_{22} V_{m2}^{t+1}$$

Calculating Expected Values for States

- This means we can summarise all of the values for $E_t\left[V_{t+1}\left(\gamma_m, s_{t+1}\right)\middle|\, s_t\right]$ in a $K \times 2$ matrix, with the first column showing the expected values when $s_t = \mu_1$ and the second column showing the expected values when $s_t = \mu_2$.

$$\begin{pmatrix} \pi_{11} V_{11}^{t+1} + \pi_{12} V_{12}^{t+1} & \pi_{21} V_{11}^{t+1} + \pi_{22} V_{12}^{t+1} \\ \pi_{11} V_{21}^{t+1} + \pi_{12} V_{12}^{t+1} & \pi_{21} V_{21}^{t+1} + \pi_{22} V_{22}^{t+1} \\ . & . \\ . & . \\ . & . \\ \pi_{11} V_{K1}^{t+1} + \pi_{12} V_{K2}^{t+1} & \pi_{21} V_{K1}^{t+1} + \pi_{22} V_{K2}^{t+1} \end{pmatrix}$$

- This matrix can be simplified to be

$$\begin{pmatrix} V_{11}^{t+1} & V_{12}^{t+1} \\ V_{21}^{t+1} & V_{22}^{t+1} \\ . & . \\ . & . \\ . & . \\ V_{K1}^{t+1} & V_{K2}^{t+1} \end{pmatrix} \begin{pmatrix} \pi_{11} & \pi_{21} \\ \pi_{12} & \pi_{22} \end{pmatrix} = V^{t+1}\Pi'$$

Numerical Calculation for the Expected Value Function

- This gives a method for calculating the expected future values $E_t \left[V_{t+1} \left(\gamma_m, s_{t+1} \right) \middle| \, s_t \right]$ at each of the points on our x_{t+1} grid.
  - Stack all the possible value function outcomes at time $t+1$ into a matrix with rows for each value of x_{t+1} on our grid and columns for each value of s_{t+1}.
  - Then multiply this value function matrix by the transpose of the Markox transition matrix for s_t.
  - This fully generalises to the case of an N-state Markov chain.
- This makes the code for the stochastic version of our programme relatively simple. A single matrix multiplication is doing all the extra calculations associated with the problem being stochastic rather than deterministic.
- With these expected future values calculated, the optimisation problem for choosing u_t is now just as it was before: Optimal decision making trades off the impact of a higher u_t on today's payoff $F\left(x_t, u_t\right)$ against its impact on tomorrow's expected value function $E_t \left[V_{t+1} \left(g\left(x_t, u_t\right), s_{t+1} \right) \middle| \, s_t \right]$.

An Example: Consumption and Savings with Stochastic Income

- We will implement these methods to solve a model in which a consumer lives for T periods.
- Each period, the consumer's income is either one or zero with transitions between states determined by a Markov chain.
- We will start with the case in which there is an equal chance each period of income being one or zero. We implement this via a two-state Markov chain with transition probabilites both equal to 0.5.
- There is no retirement so this is not a motivation for building up assets in this model.
- Assets are built up solely due to precautionary savings and these get run down at the end of life.
- Let's look at the programme we use to solve this model.

Program Set Up: Specify Parameters Including the Markov Chain.

```
%%
% 1. Model Parameters
T     = 60 ;                    % Finite Horizon
gamma = 1.5;                    % utility parameter
r     = 1/0.94 -1;              % Interest rate. Set for beta =0.94 to be the "random walk" benchmark.
%%
% 2. Building the Asset and Income Grids
StartingA      = 0;
nagrid1        = 300;
nagrid2        = 10000;
amax           = 30;  % Set this so as not to constrain anyone in the simulation
amin           = 0;
agrid          = linspace(amin,amax,nagrid1)';
agrid2         = linspace(amin,amax,nagrid2)';

nygrid = 2;
Income = [0 1] ;
p = 0.5;    % Probability of remaining unemployed
q = 0.5;    % Probability of remaining employed
% Probability of remaining in the same state
% This is equivalent to an iid process.
Transition = [p 1-q; 1-p q] ;
```

Policy Functions for Stochastic Dynamic Programming

- In our previous programme, we calculated "policy rules" for consumption i.e. for each age, we calculated the optimal level of consumption given the starting amount of assets they had. This meant the central part of the programme had a double loop: Looping over time and then for each period looping over starting assets.
- This programme uses a triple loop: Looping over time, over start-of-period assets and over the one-zero income variable.
- This produces a set of policy rules for optimal consumption at each point in time given start-of-period assets and the realisation of income.
- Note the use again of a double grid. We initially calculate the value functions over a grid of size 300 but then interpolate that over a grid of size 10,000. The grid of size 10,000 is then used to calculate optimal consumption rules.
- These rules are then combined with Markov chain simulations of s_t to simulate consumption and assets over a 60 period lifetime for 100,000 different simulated households for the final value of β examined. We then calculate the average path for consumption and assets across all of these simulations.

Program Set Up: Policy Rules For Each Point on the Asset Grids and Each Value of Income and Each Point in Time

```
% 3. Initialising the matrices
% Some matrices use the smaller grid
vT          = zeros(nagrid1,nygrid);
tv          = zeros(nagrid1,nygrid);
optimalj    = zeros(nagrid1,nygrid);
c_optimal   = zeros(nagrid1,nygrid);
a_optimal   = zeros(nagrid1,nygrid);

% And some use the larger grid
c           = zeros(nagrid2,nygrid);
valuefunc   = zeros(nagrid2,nygrid);
v           = zeros(nagrid2,nygrid);

numbeta  = 5;
Apathmean = NaN(T+1,numbeta);
Cpathmean = NaN(T+1,numbeta);
Ypathmean = NaN(T+1,numbeta);

for g=1:numbeta
beta = 0.9 + 0.02*(g-1);

% Making value function, optimal consumption, optimal asset matrices for
% each period
for k = 1:T
v(:,:,k)          = zeros(nagrid2,nygrid);
tv(:,:,k)         = zeros(nagrid1,nygrid);
c_optimal(:,:,k)  = zeros(nagrid1,nygrid);
a_optimal(:,:,k)  = zeros(nagrid1,nygrid);
end
```

Main Programme Code is Very Similar

```
% 4. Final Period
% Set the terminal value function to consuming all assets and income
for n=1:nygrid
for i=1:nagrid1
c_optimal(i,n,T)  =  agrid(i)+Income(n);
vT(i,n)          =  ((c_optimal(i,n,T))^(1-gamma)  )/(1-gamma);
end

% Interpolating the final-period value function over a finer grid
v(:,n,T) = interp1(agrid,vT(:,n),agrid2,'linear','extrap');
end % n loop

%%
% 5. Working backwards in time to calculate optimal policy functions
for t=T-1:-1:1
for n=1:nygrid
for i=1:nagrid1
% Making a matrix of possible consumption levels given initial capital of
% agrid(i) and each level of assets on the finer grid
c                           = agrid(i) + Income(n) - agrid2 / (1+r);
c(c<0)                      = NaN;  % Not allowing negative values of consumption
util                        = (c.^(1-gamma)-1)/(1-gamma);
valuefunc                   = util + beta*v(:,:,t+1)*Transition;

% Choosing the optimal level of capital (and thus consumption) from agrid2 given initial capital of
% agrid(i) for each level of Income
[maxvalue , index]          = max(valuefunc);
tv(i,n,t)                   = maxvalue(1,n);
optimala_ind(i,n)           = index(1,n);
c_optimal(i,n,t)            = c(optimala_ind(i,n));
a_optimal(i,n,t)            = agrid2(optimala_ind(i,n));
end % i loop

% Again interpolating the new value function over a finer grid
v(:,n,t) = interp1(agrid,tv(:,n,t),agrid2,'linear','extrap');
end % n loop

fprintf('Solved period %d of %d.\n',t, T)
end % time period loop
```

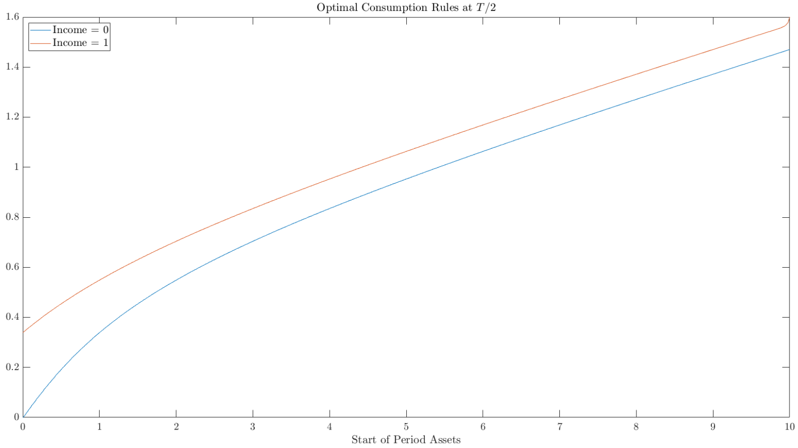
Simulating the Solved Model 100,000 Times For a Fixed β

```
% 6. Stochastic Simulation With Various Values of beta
mc = dtmc(Transition');
time = linspace(1,T+1,T+1);
NH = 100000;

Cpath = NaN(T+1,NH) ;
Apath = NaN(T+1,NH) ;
Ypath = NaN(T+1,NH) ;

Apath(1,:) = 0;

for h = 1:NH
State = simulate(mc,T+500);
for t=1:T-1
    PC           = c_optimal(:,State(t+500),t) ;
    Cpath(t,h)   = max(0, interp1(agrid, PC, Apath(t,h),'linear','extrap') );
    Ypath(t,h)   = Income(State(t+500));
    Apath(t+1,h) = (1+r)*(Apath(t,h) - Cpath(t,h) + Ypath(t,h) );
end

Cpath(T,h) = Apath(T,h) + Ypath(T,h);
Apath(T+1,h) = 0;
end % h loop

for t=1:T+1
Apathmean(t,g) = mean(Apath(t,:));
Cpathmean(t,g) = mean(Cpath(t,:));
Ypathmean(t,g) = mean(Ypath(t,:));
end
```

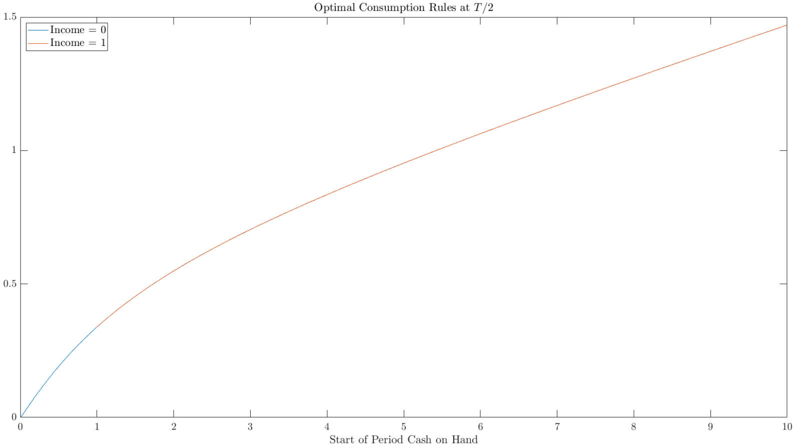
Consumption and Cash-on-Hand

- Our programme calculates policy rules for optimal consumption at each point in time given start-of-period assets and the realisation of income.
- We have set $T = 60$. Let's take a look at the policy rules at mid-life, $t = 30$.
- The graph on the next page shows the consumption spending rules at $t = 30$ for the high income state ($Y_t = 1$) and for the low income state ($Y_t = 0$).
- Not surprisingly, people consume more when the get the good draw for income than when they get the bad one. They also consume more if they have more assets at the start of the period.
- This might look like there is quite different optimal policy rules depending on the outcome for income but underlying behaviour can actually be modelled as a function of cash-on-hand, i.e. starting period assets plus income.
- The chart on the following page shows that once we plot the consumption rules against cash on hand, the behaviour is really the same in both states. This because the households are equally likely to be in each state next period.
- Note that the MPC from cash-on-hand is quite high at low levels of cash-on-hand but tails off as people build up a larger buffer.

Consumption Decision Rules at Mid-Life For Both Income States $r = 1/0.94 - 1$, $\beta = 0.90$, $\gamma = 1.5$.



Consumption Decision Rules at Mid-Life For Both Income States $r = 1/0.94 - 1$, $\beta = 0.90$, $\gamma = 1.5$.



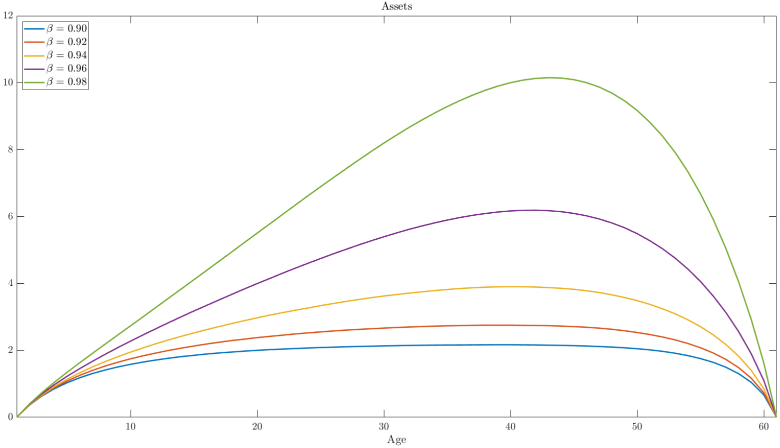
Average Paths for Consumption and Assets

- Previously, we discussed how uncertainty about the future could lead to consumption having an upward tilt. The optimal behaviour involved protecting against this uncertainty by building up a stock of "precautionary savings".
- The graphs a few pages down show the average paths of assets and consumption across 100,000 simulated households for five different discount rates with our iid income process.
- We set $r = 1/0.94 - 1$, so without uncertainty and with no borrowing constraints, we would see the following
  - Those with $\beta > 0.94$ would have growing consumption, financed by building up assets before running them down to zero.
  - Those with $\beta = 0.94$ would have perfectly smooth consumption with assets averaging zero at all times.
  - Those with $\beta < 0.94$ would have falling consumption, financed by borrowing before paying this off later at the expense of lower consumption.
- Let's see what happens with our uncertainty model (which also incorporates borrowing constraints).

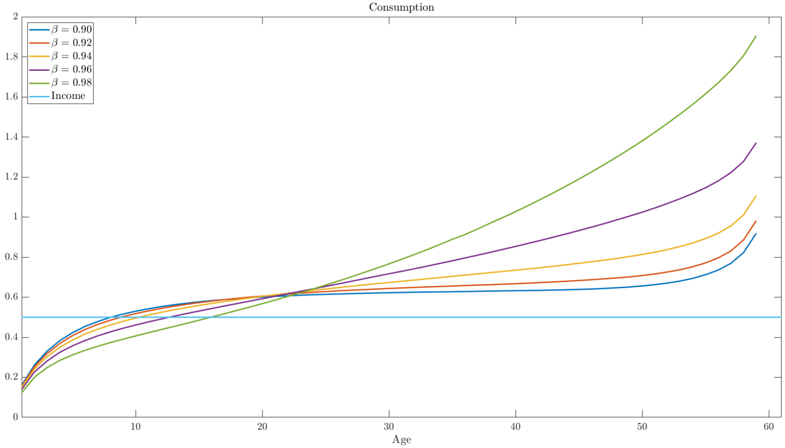
Precautionary Savings Under Uncertainty

- In our simulation, with uncertain income and negative assets ruled out, we see the following:
  - For those with $\beta > 0.94$, things are not so different from a model with certainty about income. They build up large stocks of savings and run them down. There is a burst of particularly high average consumption near the end of life as precautionary savings are not needed as much anymore.
  - For those with $\beta = 0.94$, consumption grows as they age. This is particularly true when young (because they are building up precautionary savings stocks) and when old (when they decide they no longer need those stocks).
  - For those with $\beta < 0.94$, who would have preferred a downward-sloping consumption profile, there is an upward-sloping consumption early in life as they build up precautionary savings buffers, then average consumption and assets flatten out before a late consumption burst running down their precautionary savings.
- These charts show that precautionary savings due to uncertainty can generate very different outcomes from models based on perfect certainty.

Average Time Path for Assets With IID Income Draws Averaging 0.5. $r = 1/0.94 - 1$, $\gamma = 1.5$.



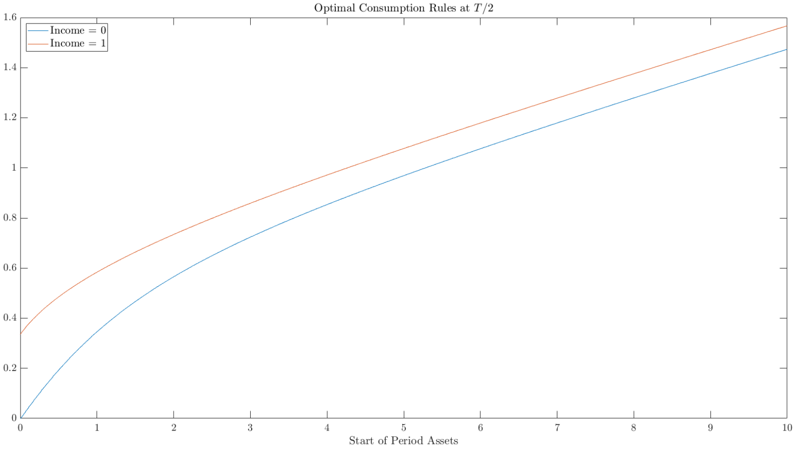
Average Time Path for Consumption and Income With IID Income Draws Averaging 0.5. $r = 1/0.94 - 1$, $\gamma = 1.5$.



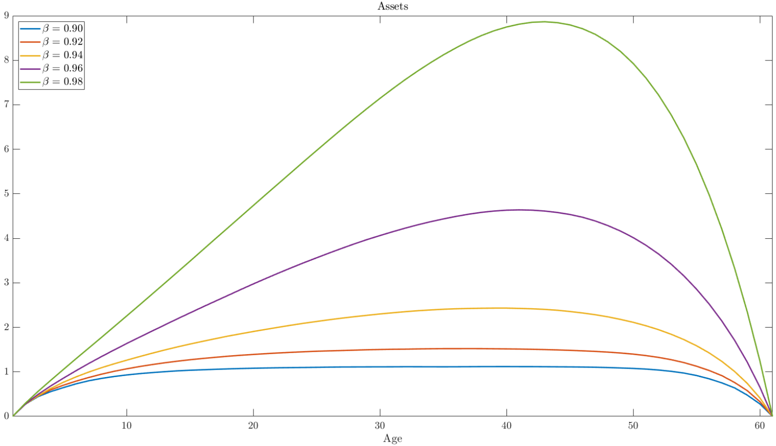
A More Realistic Model of Income and Unemployment

- The income process we have just used is not very realistic. Whether people receive income this period has no influence on whether they will receive income next period.
- In reality, once people are in employment and receiving income, they are likely to continue being in employment next period.
- I have replaced the iid income model with a model in which the probability of remaining employed next period once in employment in 90 percent and the probability of remaining unemployed once out of a job (we still assume zero income in this state) is 50 percent. I have calibrated the income people earn while employed so that average income is still equal to 0.5, matching the previous model.
- The decision rules at $t = 30$ generated by this process is on the next page. Employed people have more security so they consume more from cash-on-hand than unemployed people.
- The following two pages show the average time paths for assets. While qualitatively similar to the previous ones, the level of precautionary savings is lower because the income process generates less uncertainty.

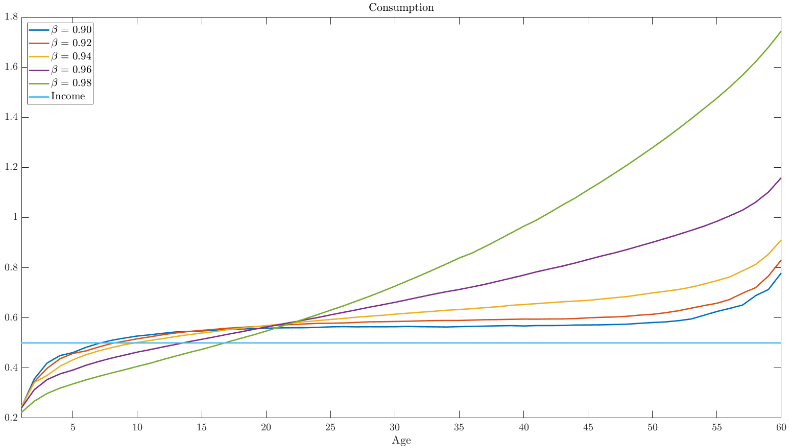
Consumption Decision Rules at Mid-Life With a Two-State "Unemployment" Process $r = 1/0.94 - 1$, $\beta = 0.90$, $\gamma = 1.5$.



Average Time Path for Assets With "Unemployment" Model $r = 1/0.94 - 1$, $\gamma = 1.5$.



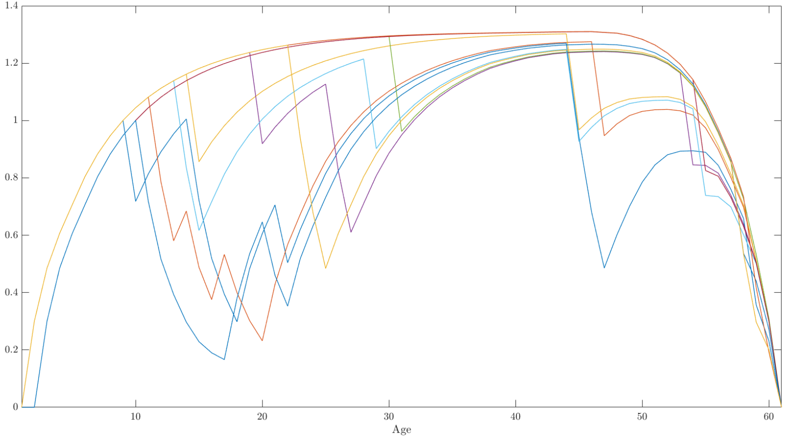
Average Time Path for Consumption With “Unemployment” Model $r = 1/0.94 - 1$, $\gamma = 1.5$



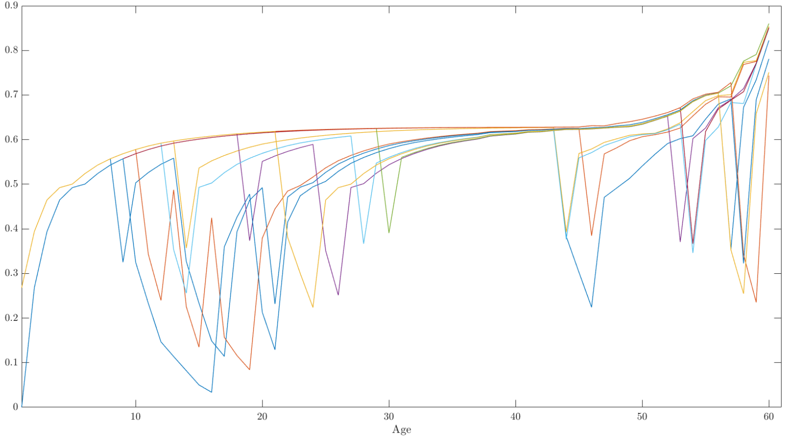
Buffer Stock Consumption Behaviour

- It isn't surprising that the "patient" households (those with $\beta > 0.94$ in this case) build up asset stocks: They also did so in the model without uncertainty.
- But impatient households ($\beta < 0.94$ in this case) are behaving differently. On average, we see them building up a stock of assets that is only run down late in life.
- This phenomenon was labelled "buffer stock saving" in a famous 1992 paper by Chris Carroll.
- To give a sense of how buffer-stock households behave, the next few charts show 10 sample lifetime paths for assets and consumptions from a simulation of this model with $\beta = 0.9$.
- We see those who get lucky and largely avoid unemployment quickly build up a buffer stock of savings and maintain constant consumption and savings. Those who get hit with spells of unemployment reduce consumption while unemployed, using their buffer to finance spending, and then look to build their buffers back up.
- Averaging over all of this behaviour gives the smooth profiles for assets and consumption for $\beta = 0.9$ that we saw in the previous charts.

Sample Time Paths for Assets $r = 1/0.94 - 1$, $\beta = 0.90$, $\gamma = 1.5$



Sample Time Paths for Consumption $r = 1/0.94 - 1$, $\beta = 0.90$, $\gamma = 1.5$.



Analytics of the Buffer Stock Theory

- Where does this buffer stock behaviour emerge from?
- The analytics of the buffer stock behaviour can be explained using the consumption Euler equation. In this model, the Euler equation is

$$U'(C_t) = \beta(1+r)E_t(U'(C_{t+1}))$$

- We discussed before how, for utility functions displaying "prudence" (positive third derivatives) like the one we are using, we have

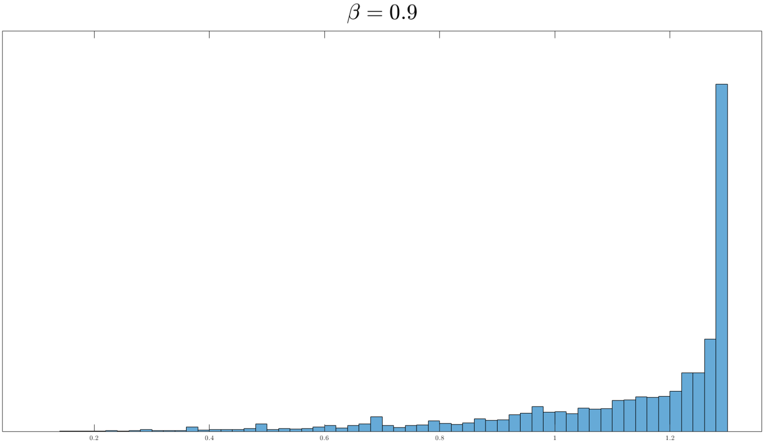
$$E_t[U'(C_{t+1})] > U'(E_t C_{t+1})$$

  and the difference between the two sides increases as the uncertainty increases.
- Note that the uncertainty that matters is driven by uncertainty about next period's level of consumption. Households that have built up their savings buffer stock can smooth consumption, so they have less uncertainty at time t about C_{t+1} and they can smooth consumption.
- But households with insufficient buffers know that another negative shock will reduce their consumption while a positive shock will increase it. This greater uncertainty induces a larger upward tilt in consumption, which means lower consumption today, thus building their buffers back up.

Wealth Inequality

- There has been an increased focus on inequality in recent years. Macroeconomic models like this can be used to explore the forces driving inequality.
- We will not have time to explore the large literature on macroeconomic models with heterogenous agent but the building blocks of these models have similarities to this model.
- When we run repeated simulations of the model, we can get a sense of how *ex ante* identical households can end up having higher or lower wealth due to good or bad luck.
- For example, the graph on the next page shows the cross-sectional distribution for our 100,000 simulations of assets at age 30. We can see a big right tail of people who have been lucky and avoided unemployment.
- We could generalise this analysis by including variations in inherited wealth and explore the extent to which models like this can replicate empirical wealth distributions.

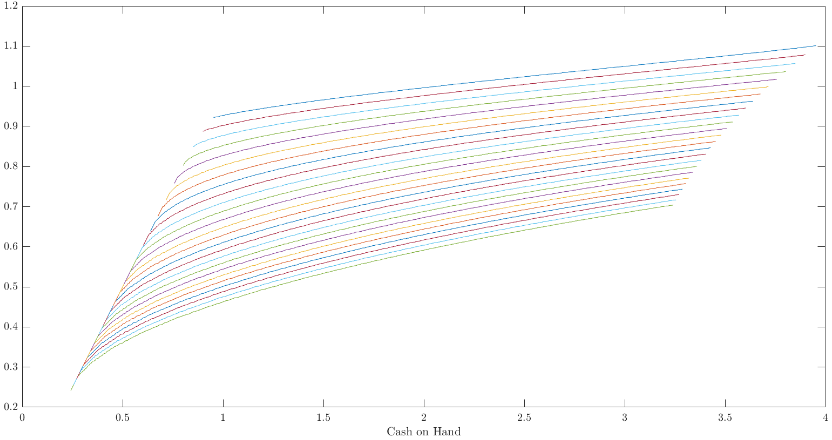
Cross-Sectional Distribution of Assets at Age 30 Across 100,000 Time Paths, $\beta = 0.9$



Approximating AR(1) Income Processes with Many State Markov Chains

- We can also simulate models like this using AR(1) style income processes.
- Earlier we showed how the Rowenhurst method can be used to allow a Markov chain to approximate an AR(1) process.
- The graph on the next page shows the consumption policy rules that emerge from running our model with a 25-state Markov chain approximating the log of income as an AR(1) process with $\rho = 0.9$. In this case, we set $\beta = \frac{1}{1+r}$.
- The higher the level of income, the higher the propensity to consume from cash on hand is.
- We also see that MPCs from cash on hand are very high at low levels of cash on hand and then flatten out fairly dramatically as cash on hand levels rise.
- The final graph provides sample paths from a model with 50 income states and impatient households. Lots of interesting dynamics are evident. Consumption is smoother than income but there are various occasions when households have fallen below their buffer stock levels of savings and consumption moves closely in line with income.

Mid-Life Consumption Rules for Various Incomes with a Many-State Markov Process ($\beta = \frac{1}{1+r}$)



Sample Time Paths for Consumption and Income with a 50-State Markov Process ($\beta = 0.9$, $r = 1/0.94 - 1$)