1 Solving the savings problem for an individual consumer

The problem is to maximize

$$
E \left[ \sum_{t=0}^{\infty} \beta^t u(c_t) \right]
$$

subject to

$$
a_{t+1} + c_t \leq e_t + ra_t
$$

where \( \{e_t\} \) is a stochastic process and \( \{c_t\} \) is constrained to be adapted to the filtration generated by this process.

Of course, \( a_0 \geq 0 \) is given. We will also impose, as a constraint, that \( a_t \geq g \) for all \( t \). We call this a “borrowing limit”, a “credit constraint” or a “liquidity constraint”.

Our stochastic Euler equation becomes

$$
u'(c_t) = \beta r E_t [u'(c_{t+1})].$$
Now suppose $e_t$ is a time-homogeneous Markov process. Then there is an optimal savings function

$$a_{t+1} = h(a_t, e_t).$$

Associated with this savings function, there is a consumption function

$$c(a, e) = e + ra - h(a, e).$$

We can write down a functional equation for $c$, assuming CRRA period utility.

$$[c(a, e)]^{-\sigma} = \beta r \mathbb{E} \left[ [c(ra + e - c(a, e), e')]^{-\sigma} \right].$$

But, as it turns out, it will be more convenient to solve for the savings function rather than the consumption function.

I will present two main approaches (with subvariants) here to solving this problem numerically. Both rely on approximating the $e$ process by a finite-state Markov chain. A good way to do that is described in Rouwenhorst (1995) and also in Kopecky and Suen (2010). To fix ideas, let’s be very specific and assume that \{e_t\} is a two-state Markov chain so that $e_t \in \{e^\ell, e^h\}$ and that the transition probabilities are given by

$$\Gamma = \begin{bmatrix} \gamma & 1 - \gamma \\ 1 - \gamma & \gamma \end{bmatrix}$$

where $2\gamma - 1$ is the autocorrelation.

We have two functional “Euler” equations as follows.

1. In the low earnings state:

$$[e^\ell + ra - h(a, e^\ell)]^{-\sigma} = \beta r \left\{ \gamma \left[ e^\ell + rh(a, e^\ell) - h(h(a, e^\ell), e^\ell) \right]^{-\sigma} + \\
+ (1 - \gamma) \left[ e^h + rh(a, e^h) - h(h(a, e^h), e^h) \right]^{-\sigma} \right\}$$

2. In the high earnings state:

$$[e^h + ra - h(a, e^h)]^{-\sigma} = \beta r \left\{ \gamma \left[ e^h + rh(a, e^h) - h(h(a, e^h), e^h) \right]^{-\sigma} + \\
+ (1 - \gamma) \left[ e^\ell + rh(a, e^\ell) - h(h(a, e^\ell), e^\ell) \right]^{-\sigma} \right\}$$

These two equations should ideally hold everywhere in the state space except (possibly) where the borrowing constraint binds.
1.1 Exogenous grid

The first approach to finding approximate savings functions begins by defining a grid over current assets $a$:

$$a \in \{a^1, a^2, \ldots, a^N\}$$

The idea now is to find (interpolating) functions $\hat{h}(a; \theta^\ell)$ and $\hat{h}(a; \theta^h)$ (where $\theta^\ell, \theta^h \in \mathbb{R}^N$) and insist (if possible) that these functions satisfy the respective Euler equations at the grid points $a^1, a^2, \ldots, a^N$ (but remember the borrowing constraint!). Without loss of much generality, suppose $\hat{h}(a^k; \theta) = \theta_k$ for all $k$.

(To be painfully explicit, $\hat{h}(a; \theta^\ell)$ is supposed to be an approximation of $h(a, e^\ell)$ and $\hat{h}(a; \theta^h)$ is supposed to be an approximation of $h(a, e^h)$.)

A surprisingly popular, but very inefficient, way of finding the vectors $\theta^\ell$ and $\theta^h$ is the following. Denote the most recent guesses by $\theta^\ell_0$ and $\theta^h_0$ and define $\theta^0 = (\theta^\ell_0, \theta^h_0)$. To update $\theta^\ell_k$ (for each $k = 1, 2, \ldots, N$) solve, for $y$, the equation

$$[e^\ell + ra_k - y]^{-\sigma} = \beta r \left\{ \gamma \left[ e^\ell + ry - \hat{h}(y; \theta^\ell_0) \right]^{-\sigma} + (1 - \gamma) \left[ e^h + ry - \hat{h}(y; \theta^h_0) \right]^{-\sigma} \right\}$$

and similarly for high state. To solve this equation, if indeed you want to pursue this approach, I would suggest something like the following. Evaluate the foc at the gridpoints $y = a^1, a^2, \ldots$ until it changes sign. Suppose this happens between $a^m$ and $a^{m+1}$. Then you know that the agent’s preferred solution lies between $a^m$ and $a^{m+1}$. Now if the foc never changes sign, you are finished. The agent would like to borrow more, but that is not allowed so you set the updated value of the savings function equal to $a$ at this point. Suppose without loss of generality that the foc is increasing. Then if it is positive at $a^1$ then you are done. Otherwise (if $y > a$), use bisection with lower bound $a^m$ and upper bound $a^{m+1}$ to solve for that $y$ which sets the foc to zero. This approach really does require interpolation.

Alternatively, once you have found $m$ as defined above, you can use a secant approximation of the foc, setting

$$y = \frac{a^m f^\ell_k(a^{m+1}; \theta^0) - a^{m+1} f^\ell_k(a^m; \theta^0)}{f^h_k(a^{m+1}; \theta^0) - f^h_k(a^m; \theta^0)}$$

where $f^\ell_k(y; \theta^0)$ is the foc function at $a = a^k$ and $e = e^\ell$. (Similarly of course for $e = e^h$.) Notice that this approach does not require any interpolation in calculating tomorrow’s savings!
To be painfully explicit:

\[ f^\ell_k(y; \theta^0) = [e^\ell + ra - y]^{-\sigma} = \beta r \left\{ \gamma \left[ e^\ell + ry - \hat{h}(y; \theta^\ell, 0) \right]^{-\sigma} + \right. \]

\[ + (1 - \gamma) \left[ e^h + ry - \hat{h}(y; \theta^h, 0) \right]^{-\sigma} \right\} \]

An alternative to updating each element \( \theta^\ell_k \) and \( \theta^h_j \) separately is to try to update them all simultaneously. One can think of Equations (2) and (2), evaluated using \( \hat{h} \) rather than the “true” savings function \( h \) as a 2N-dimensional system of equations in the unknown vectors \( \theta^\ell \) and \( \theta^h \). To be explicit, let’s write down that system of equations. In the low earnings state:

\[ [e^\ell + ra - \hat{h}(a^\ell; \theta^\ell)]^{-\sigma} = \beta r \left\{ \gamma \left[ e^\ell + r\hat{h}(a^\ell; e^\ell) - \hat{h}(\hat{h}(a^\ell; \theta^\ell); \theta^\ell) \right]^{-\sigma} + \right. \]

\[ + (1 - \gamma) \left[ e^h + r\hat{h}(a^h; \theta^\ell) - \hat{h}(\hat{h}(a^h; \theta^\ell); \theta^\ell) \right]^{-\sigma} \right\} \]

In the high earnings state:

\[ [e^h + ra - \hat{h}(a^h; \theta^h)]^{-\sigma} = \beta r \left\{ \gamma \left[ e^h + r\hat{h}(a^h; \theta^h) - \hat{h}(\hat{h}(a^h; \theta^h); \theta^h) \right]^{-\sigma} + \right. \]

\[ + (1 - \gamma) \left[ e^\ell + r\hat{h}(a^\ell; \theta^h) - \hat{h}(\hat{h}(a^\ell; \theta^h); \theta^h) \right]^{-\sigma} \right\} \]

for \( k = 1, 2, \ldots, N \). (Notice that this is 2N equations in 2N unknowns.) Of course, these equations won’t hold at all gridpoints. In particular, if \( e = e^\ell \) and \( a \) is sufficiently close to \( a \) then the foc has no feasible solution. So for those \( k \) for which that’s true you have to drop the associated equations and replace them by \( \theta^\ell_k = a \) for those \( k \).

But how should the gridpoints be selected? Where the nonlinearities are! You may want to update the gridpoints as you learn more about the savings function. But it is useful to know a few qualitative properties of the solution. They will be based on the supposition that \( \beta r < 1 \) but not by much.

1. When \( a \) is big but not too big, the borrowing constraint becomes less relevant and \( a' = a \) is approximately optimal since \( \beta r \approx 1 \).

2. As \( a \) grows, even when \( e = e^h \), optimal \( a' \) eventually goes below the 45 degree line so that \( a' < a \). This is because \( \beta r < 1 \) and it implies that assets have a natural upper bound.
3. When $a$ is very small and $e = e^{\ell}$, then the agent knows that times will eventually get better so she wants to borrow, i.e.
reduce asset holdings so that $a' < a$ unless $a = a$ in which case $a' = a$.

4. When $a$ is not too big and $e = e^h$ the agent knows that things will eventually get worse and would therefore like to save, i.e. $a' > a$.

The hardest job is to locate the kink where $h(a, e^\ell) > 0$ for the first time. It is typically close to $a$, but it is crucial for the proof of convergence that it is strictly greater than $a$, because this means that you reach the state $(a, e^\ell)$ in finitely many steps with a probability bounded away from zero.

### 1.2 The method of endogenous grid

For a savings problem where the interest rate is given, the method of endogenous grid\(^1\) is the fastest and cleverest that I am aware of. Alas, it forces one to update point by point, but this updating is lightning fast thanks to the fact that it is done analytically. (We saw this in Lecture Notes 2, here it is again, this time with a little bit more detail.)

Consider the first order condition

$$
[e^{\ell} + ra - y]^{-\sigma} = \beta r \left\{ \gamma \left[ e^{\ell} + ry - \hat{h}(y; \theta^{\ell,0}) \right]^{-\sigma} +
+(1 - \gamma) \left[ e^{h} + ry - \hat{h}(y; \theta^{h,0}) \right]^{-\sigma} \right\} = 0
$$

This is a messy function of $y$. Certainly no analytical solution is available. But suppose $y$ is given. Then solving for $a$ is a cinch! Using brutal but useful short hand notation, we can write the above foc as

$$
[e^{\ell} + ra - y]^{-\sigma} = A^{\ell}(y; \theta^0)
$$

Evidently

$$
a = \frac{[A^{\ell}(y; \theta^0)]^{-1/\sigma} - e^\ell + y}{r}.
$$

On this idea we can base an algorithm. We begin by defining, once and for all, a grid not for assets but for savings. Denote this savings grid by $y = \{y^1, y^2, \ldots, y^n\}$ where it is of course a

---

\(^1\) See Carroll (2005).
good idea to set \( y^1 = \overline{a} \) and \( y^n = \overline{a} \). The corresponding grid for assets \( a = \{a^1, a^2, \ldots, a^n\} \) is, as the name of the method suggests, computed. Indeed there will be one for the low earnings state, \( a^\ell \) and one for the high earnings state, \( a^h \).

We are now in a position to give a formal description of the endogenous grid algorithm. We begin by defining

\[
A^\ell(y; \theta^0) = \beta r \left\{ \gamma \left[ e^\ell + ry - \hat{h}(y; \theta^\ell,0) \right]^{-\sigma} + \\
+ (1 - \gamma) \left[ e^h + ry - \hat{h}(y; \theta^h,0) \right]^{-\sigma} \right\} = 0
\]

and similarly for \( A^h(y; \theta^0) \).

We then not that, so far, we have written our interpolating functions as \( f(x; \theta) \). In this context, however, it is more useful to write \( \hat{f}(x; x, y) \) where \( x \) is a grid on the domain and \( y \) is the corresponding vector grid on the range of the approximated function \( f \). In our case, we write, instead of \( \hat{h}(a; \theta^\ell), \hat{h}(a; a^\ell, a^\ell') \) and similarly for the high earnings state.

The algorithm then proceeds as follows.

1. Fix, once and for all, a grid \( a' \).
2. Initialize \( a^\ell \) and \( a^h \).
3. Update according to

\[
a^\ell_{1,k} = \frac{[A^\ell(a^\ell_k; a^\ell,0, a^h,0, a')]^{-1/\sigma} - e^\ell}{r} + a^\ell_k,
\]

and

\[
a^h_{1,k} = \frac{[A^h(a^\ell_k; a^\ell,0, a^h,0, a')]^{-1/\sigma} - e^h}{r} + a^\ell_k.
\]

What about the borrowing constraint? That pretty much takes care of itself! You will find that, endogenously, \( a^1 > \overline{a} \). To evaluate \( \hat{h}(a, e^\ell) \) below that, just add another gridpoint \( a^\ell_0 = \overline{a} \) with associated value \( a^\ell_0 = \overline{a} \).
Computing the invariant distribution

Huggett (1993) is, alas, not helpful on this point. Here are two alternative, superior approaches.\(^2\)

Denote the endogenous (continuous) state variable by \(a\) and the exogenous (discrete) state variable by \(\lambda\). We assume that \(\lambda\) lives on a finite set \(\{\lambda^1, \lambda^2, \ldots, \lambda^m\}\) and that we have defined a grid for \(a\) via \(\{a^1, a^2, \ldots, a^n\}\). (Occasionally we will use the notation \(a^1 = a\) and \(a^n = \bar{a}\).) Note that this grid should be much finer than the one used to approximate the optimal decision rule \(h(a, \lambda)\).

2.1 Interpolating the distribution function

The idea here is to find a piecewise linear approximation of the distribution function by iterating on

\[
F(a', \lambda') = \sum_{k=1}^{m} \pi(\lambda' | \lambda) F(h^{-1}(a', \lambda_k), \lambda_k)
\]

where we define

\[
h^{-1}(a', \lambda) = \sup\{a \in [\underline{a}, \bar{a}] : h(a, \lambda) = a'\}
\]

and we adopt the convention \(\sup \emptyset = -\infty\).

First we need an initial distribution \(F_{i,j}^0\). One choice is to make \(F^0\) uniform over the endogenous dimension of the state space and set

\[
F_{i,j}^0 = \frac{a^i - a^1}{a^n - a^1} \pi^*_j
\]

where \(\pi^*_j\) is the \(j\)th element of the stationary distribution for \(\lambda\) where the meaning of the notation \(F_{i,j}\) is made explicit by the following piecewise interpolation rule.

\[
\hat{F}(a, \lambda_j) = F_{i,j} + \frac{F_{i+1,j} - F_{i,j}}{a^{i+1} - a^i} (a - a^i) \quad \text{for } a^i \leq a \leq a^{i+1}.
\]

Notice that nothing in the method described here forces us to use piecewise linear interpolation; this particular interpolation rule is just there to fix ideas. You do need some interpolation rule, however, and we write \(\hat{F}(a, \lambda)\) to denote the interpolating function. Notice that we have

\(^2\) This section is plagiarized from [http://www.karenkopecky.net/Teaching/eco613614/](http://www.karenkopecky.net/Teaching/eco613614/).
suppressed the obvious dependence on the parameters $F_{i,j}$. The updating rule is

$$F_{i,j}^1 = \sum_{k=1}^{m} \pi(\lambda_k|\lambda_j) \tilde{F}^0(h^{-1}(a^i, \lambda^k), \lambda_k).$$

In each iteration, two conditions should be imposed. First, if $h^{-1}(a^i, \lambda^k) < a$, then $F_{i,k}^1 = 0$. Second, if $h^{-1}(a^i, \lambda^k) \geq a$, then $F_{i,k}^1 = \pi_k$.

### 2.2 Discretizing the density function

Initialize $f_{i,j}^0 = 0$ for all $i$ and $j$ but set $f_{1,1}^0 = 1$, i.e. concentrate all the mass at a single point. Or make it uniform.

The updating rule is:

1. Set $f_{i,j}^1 = 0$ for all $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

2. For each $k = 1, 2, \ldots, m$, each $l = 1, 2, \ldots, n$ and each $j = 1, 2, \ldots, m$, set

$$f_{i,j}^1 = f_{i,j}^1 + \pi(\lambda^k|\lambda^j) \frac{a_{i+1} - h(a^i, \lambda^k)}{a_{i+1} - a^i} f_{l,k}^0$$

$$f_{i+1,j}^1 = f_{i+1,j}^1 + \pi(\lambda^k|\lambda^j) \frac{h(a^i, \lambda^k) - a^i}{a_{i+1} - a^i} f_{i,k}^0$$

where $i$ is such that $a^i \leq h(a^i, \lambda^k) \leq a^{i+1}$.

At each iteration, you may want to make sure that

$$\sum_{i=1}^{n} \sum_{j=1}^{m} f_{i,j}^1 = 1.$$

### 3 Finding the equilibrium interest rate

To compute total assets for a given distribution function, you may want to make use of the following result. If a non-negative random variable $X$ is distributed according to $F$, then its expected value equals

$$\mathbb{E}[X] = \int_{0}^{\infty} x dF(x) = \int_{0}^{\infty} [1 - F(x)] dx.$$
(You can prove this using integration by parts on the real line or by integrating over the sample space and using Fubini’s theorem.) If $X$ has a lower bound $a$ (perhaps a negative one), the formula becomes

$$E[X] = a + \int_a^{\infty} [1 - F(x)] dx$$

and if it also has an upper bound $b$, then this reduces to

$$E[X] = b - \int_a^b F(x) dx.$$

If you have the discretized density function instead, your life is much easier. Then aggregate/average assets are

$$\sum_{i=1}^n \sum_{j=1}^m f_{i,j} a_i.$$ 

In a competitive equilibrium, $r$ should be such that aggregate assets are zero.

References


