Contents

These notes are based on the problem and code for the "Two Assets and Kinked Adjustment Costs," on Ben's Website.

1 Household's Problem

- The household solves

$$\max_{\{c_t,d_t\}_{t\geq 0}} \mathbb{E}_0 \int_0^\infty e^{-\rho t} u(c_t) dt$$

- subject to

$$\begin{aligned}\dot{b}_t &= (1-\xi) w z_t + r^b(b_t) b_t - d_t - \chi(d_t, a_t) - c_t \\ \dot{a}_t &= r^a a_t + \xi w z_t + d_t\end{aligned}$$

- with constraints

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

- where a_t, b_t denote illiquid and liquid assets, respectively, c_t is consumption, z_t is the idiosyncratic productivity which is considered to be a two-state Poisson process with intensities $\lambda(z, z')$, d_t is the depositing rate and $\chi(.,.)$ the transaction cost function. The wage is denoted by w, the return on illiquid asset is r^a and the return on liquid asset is r^b. Finally we assume that a fraction ξ of income is automatically deposited in the illiquid account (e.g. capturing automatic payroll deductions into a 401(k) account).
- Let's assume the following functional form for the adjustment cost function:

$$\chi(d, a) = \chi_0 |d| + \frac{\chi_1}{2} \left(\frac{d}{a}\right)^2 a \tag{1}$$

- The two components of the adjustment cost function have different implications for the household behaviour:
    1. the kinked cost component implies inaction,
    2. the convex component implies finite deposit rates.
- In what follows let's denote the net effect of deposit flow d on cash in hand of agent with illiquid wealth a, by the function $g(d, a)$:

$$g(d, a) = d + \chi(d, a)$$

Conditions on χ parameters:

1. First, $\chi_0, \chi_1 > 0$.
2. Then, assume $r^a < \frac{1-\chi_0}{\chi_1}$ to ensure households won't accumulate illiquid wealth to infinity.
3. Finally, assume $\chi_0 < 1$, otherwise it never makes sense to withdraw. That is, $g(d, a)$ would be non-negative on the whole domain $d \in \mathbb{R}$. (this is also implied by 2 if we want to avoid degenerate cases).

2 Recursive Formulation

The HJB equation is

$$\begin{aligned}\rho V(a,b,z) = \max_{c,d}\ & u(c) + V_b(a,b,z)((1-\xi)wz + r^b(b)b - d - \chi(d,a) - c) \\ & + V_a(a,b,z)(r^a a + \xi wz + d) \\ & + \sum_{z'} \lambda(z,z')(V(a,b,z') - V(a,b,z)) \end{aligned} \tag{2}$$

- The first order conditions are

$$\begin{aligned} u'(c) &= V_b(a,b,z) \\ V_b(a,b,z)(1+\chi_d(d,a)) &= V_a(a,b,z) \end{aligned} \tag{3}$$

- Note that $\chi_d(d,a) =$

$$\begin{cases} \chi_0 + \chi_1 d/a, & d > 0 \\ -\chi_0 + \chi_1 d/a, & d < 0 \end{cases} \tag{4}$$

- Based on the equation above, optimal deposits satisfy

$$d = \left(\frac{V_a}{V_b} - 1 + \chi_0\right)^{-} \frac{a}{\chi_1} + \left(\frac{V_a}{V_b} - 1 - \chi_0\right)^{+} \frac{a}{\chi_1} \tag{5}$$

- Or, equivalently, assuming we know the optimal consumption policy,

$$d = \left(\frac{V_a}{u'(c)} - 1 + \chi_0\right)^{-} \frac{a}{\chi_1} + \left(\frac{V_a}{u'(c)} - 1 - \chi_0\right)^{+} \frac{a}{\chi_1} \tag{6}$$

- In particular, $d = 0$ if $-\chi_0 < \frac{V_a}{V_b} - 1 < \chi_0$ (the inaction region).

3 Numerical Solution Without Drift-Splitting

Kaplan et. al. propose a scheme in which the drift for the liquid asset b is split in two parts, to upwind the finite difference in a way that is both simple (in the sense of avoiding non-linearities) and monotone. Importantly, the way the boundary conditions are handled in the accompanying code seem also to be consistent with the drift-split idea.

In this document we explain the details for an upwind implicit scheme without splitting the drift in two, and with handling of boundary conditions in the traditional sense (e.g. as done in [Achdou et al., 2022]). The aim is to try the traditional method of building monotone schemes to see whether that would be more robust, particularly at the boundaries, for other problems involving the same two-asset structure.

- Before proceeding further, let's specify three useful *special* d (and the corresponding c points when $\dot{b} = 0$:
  - $\mathbf{d_0}, \mathbf{c_0}$: correspond to $d = 0, c(d = 0, \dot{b} = 0) = (1 - \xi)wz + r^b(b)b$
  - $\underline{\mathbf{d}}, \underline{\mathbf{c}}$: correspond to $\underline{d} = \left(\frac{\chi_0 - 1}{\chi_1}\right) a$, $\underline{c} = c(d = \underline{d}, \dot{b} = 0)$. Note that by assumption 1, $\underline{d} < 0$. This is a point which maximises the cash in hand.
  - $\bar{\mathbf{d}}, \bar{\mathbf{c}}$: correspond to $\bar{d} = -(r^a a + \xi wz), \bar{c} = c\left(d = \bar{d}, \dot{b} = 0\right)$. This points give the threshold for the sign of b-drift to switch.

3.1 United We Upwind!

Here is the general algorithm to upwind without split (the traditional way); given $V^{n-1}(b_i, a_j, z_k)$ (hereafter $V^{n-1}_{i,j,k}$ for more concise notation following Kaplan et. al.)

1. Start with $c^{n,F}$ (that is, use $V_b^{n,F}$ to update the consumption policy c^n).
2. Use $c^{n,F}$ with equation (6) *subject to upwinding with respect to the drift of a* to update the deposit policy $d^{n,F}$. More specifically, if we denote by $d(c, V_a) = d(c, V_a; a_j)$ the optimal d according to equation (6),
   (a) If $d(c^{n,F}, V_a^{n,F}) > -(r^a a_j + \xi w z_k)$, then $d^{n,F} = d(c^{n,F}, V_a^{n,F})$.
   (b) If $d(c^{n,F}, V_a^{n,B}) < -(r^a a_j + \xi w z_k)$, then $d^{n,F} = d(c^{n,F}, V_a^{n,B})$.

(c) If neither of the above holds, then *stay put with respect to* a, that is, $\dot{a} = 0$ which implies $d^{n,F} = -(r^a a_j + \xi w z_k)$.

3. After solving for $d^{n,F}$ using $c^{n,F}$, now plug them back in the drift for b to check whether they're consistent. In particular, if $c^{n,F} + g(d^{n,F}, a_j) < (1-\xi)wz_k + r^b(b_i)b_i$, then we update the optimal consumption and depositing policies as: $c^n = c^{n,F}, d^n = c^{n,F}$. Proceed with updating the value function by the conventional upwinding according to drifts computed using these policies.

4. Otherwise, repeat the same procedure for $c^{n,B}$. That is, compute $d^{n,B}$ similar to above, and then if $c^{n,B} + g(d^{n,B}, a_j) > (1-\xi)wz_k + r^b(b_i)b_i$, consider $c^n = c^{n,B}, d^n = d^{n,B}$, as optimal policies, and proceed from there.

5. If none of the two options above led to consistent drift for b, then solve for the optimal policies assuming that we are put with respect to b, i.e., enforce $\dot{b} = 0$. More on that below.

3.2 Note on Implementation and Comparison with Kaplan et. al. Code

How to implement this algorithm, particularly to make it amenable to vectorisation that is important for some languages including Matlab? And how does the scheme compare with what the scheme suggested in Kaplan et. al.? For now, we ignore step 5 ($\dot{b} = 0$) and focus on steps 2-4.

- Let's rewrite our algorithm in terms of the notations used in Kaplan et. al. In particular, d^{FB} denotes the d policy resulting from applying (5) to V_b^F and V_a^B; d^{BB}, d^{BF} and d^{FF} are defined similarly, with the first letter in superscript denoting the direction of finite difference for $V_b = \frac{\partial V}{\partial b}$ and the second letter for the direction of $V_a = \frac{\partial V}{\partial a}$. Also, d^B denotes the final upwind policy for d, if the backward direction V_b^B is used for estimating the partial derivative of V with respect to b.

- Then, d^F, d^B would be given by

$$d^F = d^{FF}\mathbb{I}_{\left[d^{FF} > \bar{d}(a,z)\right]} + d^{FB}\mathbb{I}_{\left[d^{FB} < \bar{d}(a,z)\right]}$$
$$d^B = d^{BF}\mathbb{I}_{\left[d^{BF} > \bar{d}(a,z)\right]} + d^{BB}\mathbb{I}_{\left[d^{BB} < \bar{d}(a,z)\right]}$$

- Given d^F, d^B, estimation of optimal depositing policy d would be updated as

$$
\begin{aligned}
d &= d^B \mathbb{I}_{\left[c^B + g(d^B, a) > (1-\xi) wz + r^b b\right]} + d^F \mathbb{I}_{\left[c^F + g(d^F, a) < (1-\xi) wz + r^b b\right]} \\
&= d^B \mathbb{I}_{\left[\dot{b}(c^B, d^B) < 0\right]} + d^F \mathbb{I}_{\left[\dot{b}(c^F, d^F) > 0\right]}
\end{aligned}
$$

- Using the notation from Kaplan et. al., and for comparison, the last equation can be re written as

$$
d = d^B \mathbb{I}_{\left[s^{d,B} < -s^{c,B}\right]} + d^F \mathbb{I}_{\left[s^{d,F} > -s^{c,F}\right]}
$$

3.2.1 Comparison with Kaplan et. al.

- Obviously, there is no splitting. The indicators which tell us whether to use d^B or d^F are the same indicators telling us whether to use c^B or c^F. To me this sounds neater and more inline with traditional upwinding.
- The idea here is **to nest, rather than split** the two upwindings. I.e., upwinding with respect to a-drift is taken care of when forming d^B, d^F (inner level). Then at the outer level, the indicator just forms with respect to sign of $\dot{b}$.
- As apparent from above, the implementation is not more difficult, maybe even a bit shorter. The only additional element is $\bar{d}$ which can be computed outside loop. However, policy updates seem to be different. In forming indicators for d^B, d^F I take $\bar{d}(a, z)$ as the point of reference, while Kaplan et. al. take 0. Also in forming d, I take the sign of the whole s^b drift as indicator, while Kaplan et. al. take only the sign of $s^d = -g(d, a)$. So the two implementations differ in **step-wise policy update**. However, this does not necessarily imply they converge to different solutions.
- There is still one important piece left unaddressed: policies for the case of $\dot{b} = 0$, which happens at the boundaries as well as when none of the two indicators above leads to consistent b-drifts.

3.3 Updating Policies for $\dot{b} = 0$

There are cases in which we want to solve for the optimal policies, given $\dot{b} = 0$. This particularly matters for enforcing the borrowing constraint at the boundary of the grid, but also because, as discussed above, neither $V_b^{n,F}$ nor $V_b^{n,B}$ leads to consistent b-drifts.

- In this case the HJB simplifies to

$$\rho V(a,b,z) = \max_{d}\ u(c(d)) + V_a(a,b,z)(r^a a + \xi w z + d) + \sum_{z'} \lambda(z,z')(V(a,b,z') - V(a,b,z)) \tag{7}$$

- where $c(d) = (1-\xi)wz_t + r^b(b)b - g(d,a)$
- FOC for d satisfies

  $$u'(c)g'(d) = V_a(a,b,z) \implies u'\Big((1-\xi)wz + r^b(b)b - g(d,a)\Big)(1+\chi_d(d,a)) = V_a(a,b,z)$$

  with $\chi_d(d,a)$ given by (4).
- Note that looking at the interval $[\underline{d},\infty)$ there should exist a unique solution "for updating optimal d". Assuming that $V_a(a,b,z)$ is a positive number, then the LHS is zero at $\underline{d}$, and it's strictly increasing in d. So there should be a unique solution for it "unless" the solution is "lost" in the jump that happens at 0, in that case we update our approximation of the optimal d to be zero (case 2 below).
- The LHS is not only non-linear but kinked, which makes it a bit messy to solve. To solve for d non-linearly using this equation, two things should be taken care of; first whether to use V_a^F or V_a^B to properly upwind, second the jump in $(1+\chi_d)$ (artefact of the kink in adjustment comes).

Now to update the optimal policy for the case of $\dot{b} = 0$, consider these cases:

1. Either $u'(c_0)(1+\chi_0) < V_a^F$: In this case solve for optimal d on $[0,\bar{d})$.
2. Or $\frac{V_a^F}{1+\chi_0} < u'(c_0) < \frac{V_a^F}{1-\chi_0}$, in which case optimal d $= 0$.
3. Finally, if $u'(c_0)(1-\chi_0) > V_a^F$
   (a) If $\underline{d} > \bar{d}$, solve the FOC for d, again using V_a^F, on $[\underline{d}, 0]$.
   (b) Else
      i. If $u'(\bar{c})(1+\chi_d(\bar{d},a) < V_a^F$, then: solve for d on $[\underline{d},\bar{d}]$ using V_a^B.
      ii. If $u'(\bar{c})(1+\chi_d(\bar{d},a) < V_a^F$, solve for d on $[\bar{d},0]$ using V_a^F.
      iii. Otherwise, optimal $d = \bar{d}$ (that is, a-drift is zero).

Again, this can be vectorised similar to above.

3.4 Updating Policies for $\dot{a} = 0$

Which we use at a-boundaries or where neither V_a^F nor V_a^B leads to consistent policies. This bit is straightforward. Set $d = -(r^a a + \xi w z)$, then solve for c by upwinding with respect to $\dot{b}$.

Note: As long as V is concave and monotone in each of assets separately, the above scheme is unambiguous, monotone and upwind (in the conventional sense!)

References

[Achdou et al., 2022] Achdou, Y., Han, J., Lasry, J.-M., Lions, P.-L., and Moll, B. (2022). Income and wealth distribution in macroeconomics: A continuous-time approach. *The review of economic studies*, 89(1):45–86.