TS03 FHDP

March 17, 2021

Table of Contents

Preparado por MachinaFantasma

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
from scipy import linalg

%matplotlib inline
```

'*I remember the econometrics class that Tom taught. He said, "In this course, we're going to use a lot of linear and quadratic methods. I know many people think that's not high-tech, but it put people on the moon, and that's good enough for me.*"'

Ellen McGrattan, interview with S. Bowmaker in *The Art and Practice of Economic Research*

1 LQ Life-cycle accumulation

This is an example of a life-cycle, optimal accumulation problem.

By restricting to linear constraints and quadratic objective we have a very tractable class of optimal decision rules that are linear in the state variables.

Consider a consumer who ranks sequences of consumption outcomes $\{c_t\}_{t=0}^T$ according to this

preference function:

$$\sum_{t=0}^{T} \beta^t U(C_t) + \beta^{T+1} V_{T+1}(a_{T+1}), \tag{1}$$

where the per-period utility of consumption is

$$U(C_t) = -(C_t - c^\star)^2. \tag{2}$$

The consumer seeks the maximal total payoff by varying her length-$(T+1)$ consumption sequence (plan).

Assume that $V_{T+1}(a_{T+1}) = -k(a_{T+1})^2$, where $k \geq 0$ is some arbitrarily large number.

This has an interpretation of a (quadratic) cost of bequething assets post mortem.

Let us assume that

$$y_t = \bar{y} + \kappa_1 t + \kappa_2 t^{\kappa_3}, \tag{3}$$

where the parameters (κ_1, κ_3) are all strictly positive, $\bar{y} \geq 0$ and $\kappa_2 < 0$.

We'll set $\kappa_3 = 2$.

Remark.

- Setting $k \geq 0$ is a useful device in life-cycle modelling for rationalizing positive and large amounts of assets held by retirees or people at the end of their life (see, e.g., Mariacristina de Nardi here and here.)
  - This has the interpretation of a "warm-glow" bequest motive of the consumer: She or he derives payoff from leaving assets behind (i.e., bequest to an offspring).
  - The modelling trick here has a quantitative motive: It helps fit the end point of age-asset profile in life-cycle data better.
- Note, if $k \to \infty$, then we are back to something similar to our selfish consumer who has no incentive to leave resources behind after their die.

The parameter $c^\star \in (0, \infty)$ is an ideal, reference consumption point. The consumer discounts per period payoffs by discount factor $\beta^t \in (0, 1]$. The consumer faces a sequence of budget constraints:

$$a_{t+1} + C_t = Ra_t + y_t, \qquad t = 0, 1, ..., T, \tag{4}$$

where a_t is the consumer's asset position (positive/negative denotes saving/borrowing), $R > 1$ is some given gross return on the asset, and, y_t is income-endowment.

Solution.

Combining (4) and (3) we have the sequence of budget constraints as

$$a_{t+1} = Ra_t + (\bar{y} + \kappa_1 t + \kappa_2 t^{\kappa_3}) - (C_t - c^\star) - c^\star. \tag{5}$$

Given a choice outcome $C_t = c$, (5) tells us that next period's asset outcome is conditional on state vector $x_t = (1, t, t^{\kappa_3}, a_t)$. Now when next period eventuates, the form of this problem repeats itself. Again, in next period, if the consumer takes another action $C_{t+1} = c'$, the continuation asset into

$t+2$ depends again on state vector x_{t+1}. So on and so forth. Hence, a sufficient statistic about the location or position of the decision problem at the start of each date t is the vector x_t.

If $\kappa_3 = 2$, then we have $(t+1)^2 = 1 + 2t + t^2$.

Note: You could do all this below using scalar algebra and calculus. But we'll set things up more generally as a linear-quadratic program.

Give the grey suits in the creative department of your brain a call. They'll be able to see that (5) can also be written out as a system of linear equations and hence a recursion:

$$\begin{aligned} 1 &= 1 \\ (t+1) &= 1 + (t) \\ (t+1)^2 &= 1 + 2 \times (t) + 1 \times (t)^2 \\ a_{t+1} &= (\bar{y} - c^\star) \times 1 + \kappa_1 \times (t) + \kappa_2 \times (t)^2 + R \times a_t. \end{aligned}$$

If you existed in 1999, then you'd know to take the red pill, and enter this in matrix form:

$$\begin{bmatrix} 1 \\ t+1 \\ (t+1)^2 \\ a_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 2 & 1 & 0 \\ \bar{y} - c^\star & \kappa_1 & \kappa_2 & R \end{bmatrix} \begin{bmatrix} 1 \\ t \\ t^2 \\ a_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} (C_t - c^\star). \tag{6}$$

Fast takeaway.

1. We have just shown that we can re-write the controllable, non-homogeneous scalar difference equation (5) in a state-space form:

$$x_{t+1} = Ax_t + Bu_t; \tag{7}$$

   where u_t is our control variable.

2. Observe that the objective function (1), given assumption (2), is quadratic. The constraints (6), for each date t, are linear.

1.1 From sequence to recursive problem

Claim. Bellman's principle of optimality allows us to rewrite this decision problem as:

$$W_t(x_t) = \min_{u_t} \left\{ x_t^{\mathrm{T}} Q x_t + u_t^{\mathrm{T}} R u_t + 2 u_t^{\mathrm{T}} S x_t + \beta W_{t+1}(x_{t+1}) : x_{t+1} = Ax_t + Bu_t \right\} \tag{8}$$

and for the terminal valuation, we have

$$W_{T+1}(x_{T+1}) = x_{T+1}^{\mathrm{T}} P_{T+1} x_{T+1} \equiv V_{T+1}(a_{T+1}). \tag{9}$$

Can you prove this?

Eat your greens and do your linear algebra. Then you'll see that:

$$Q = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; R = 1, \; S = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}, \; P_{T+1} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k \end{bmatrix}.$$

1.2 Backward induction

Stage $t = T$: We have $W_{T+1}(x_{T+1}) = x_{T+1}^{\mathrm{T}} P_{T+1} x_{T+1}$.

So Bellman's equation (8) at this stage is

$$W_T(x_T) = \min_{u_T} \left\{ x_T^{\mathrm{T}} Q x_T + u_T^{\mathrm{T}} R u_T + 2 u_T^{\mathrm{T}} S x_T + \beta (A x_T + B u_T)^{\mathrm{T}} P_{T+1} (A x_T + B u_T) \right\} \tag{10}$$

The minimizer is

$$u_T^* = F_T x_T, \tag{11}$$

where $F_T = -(R + \beta B^{\mathrm{T}} P_{T+1} B)^{-1} (\beta B^{\mathrm{T}} P_{T+1} A + S)$.

What is the value of this optimal response?

Plug minimizer (11) into the objective function on the RHS of (10) to get

$$W_T(x_T) = x_T^{\mathrm{T}} P_T x_T. \tag{12}$$

where

$$P_T = Q - (\beta B^{\mathrm{T}} P_{T+1} A + S)^{\mathrm{T}} (R + \beta B^{\mathrm{T}} P_{T+1} B)^{-1} (\beta B^{\mathrm{T}} P_{T+1} A + S) + \beta A^{\mathrm{T}} P_{T+1} A. \tag{13}$$

Notice how this value function is also a quadratic form?

If we can find the fixed point P_T satisfying (13), then we can compute value function W_T.

Now step back by one stage.

By induction, you can show that for each

Stage $t = T, T-1, ..., 0$: Given $W_{t+1}(x_{t+1}) = x_{t+1}^{\mathrm{T}} P_{t+1} x_{t+1}$, we have

$$W_t(x_t) = \min_{u_t} \left\{ x_t^{\mathrm{T}} Q x_t + u_t^{\mathrm{T}} R u_t + 2 u_t^{\mathrm{T}} S x_t + \beta (A x_t + B u_t)^{\mathrm{T}} P_{t+1} (A x_t + B u_t) \right\}. \tag{14}$$

The optimal regulator (i.e., the minimizer) is a time-varying, linear function of the state vector:

$$u_t^* = F_t x_t, \tag{15}$$

where $F_t = -(R + \beta B^{\mathrm{T}} P_{t+1} B)^{-1} (\beta B^{\mathrm{T}} P_{t+1} A + S)$.

What is the value of this optimal response?

Plug minimizer (11) into the objective function on the RHS of (14) to get

$$W_t(x_t) = x_t^{\mathrm{T}} P_t x_t, \tag{16}$$

where

$$P_t = Q - (\beta B^{\mathrm{T}} P_{t+1} A + S)^{\mathrm{T}} (R + \beta B^{\mathrm{T}} P_{t+1} B)^{-1} (\beta B^{\mathrm{T}} P_{t+1} A + S) + \beta A^{\mathrm{T}} P_{t+1} A. \tag{17}$$

Notice the recursivity of the form now?

If we can calculate P_t satisfying (13), then we can compute value function W_t.

1.2.1 Summary

Given terminal period payoff function indexed by P_{T+1} (which we know), we can solve by backward induction using the recursion (17) and (15).

This yields a sequence $\{W_t, h_t : t = T, T-1, ..., 0\}$ of stage- and state-dependent (quadratic) value and (linear) decision functions, where each function is

- $W : \mathbb{N} \times X \to \mathbb{R}$, where $W_t(x_t) := x_t^{\mathrm{T}} P_t x_t$;
- $h : \mathbb{N} \times X \to \mathbb{R}$, where $h_t(x_t) := F_t x_t$;

and $\mathbb{N} \ni t$ and $X \ni x_t$.

1.3 Implementation

Here's our deterministic, quadratic-trend (non-asset) income process:

[2]:
```
def y(t, ybar,  1,  2):
    """Exogenous and stylized hump-shaped income process
    as a function of time/age, t"""
    return ybar +  1*t +  2*t**2.0
```

Applying the optimal control $u_t^* = F_t x_t$ to the linear state-space model (7), we have the optimal trajectory of the state vector characterized by

$$x_{t+1} = g_t(x_t) := (A + BF_t)x_t. \tag{18}$$

We code this up as:

[3]:
```
def g(x, A, B, F):
    """Optimal transition map of x under optimal
    regulator: u(t) = F(t)x(t)"""
    x_next = (A + B@F)@x
    return x_next
```

This implements the matrix Riccatti recursion we saw in (17):

[4]:
```
def riccatti(P_next,  , A, B, Q, R, S):
    """Riccatti equation: A matrix-quadratic recursion.
    Given continuation value P_{t+1}, get P_{t} and F_{t}.
    Note: If A is a stable matrix, there is a unique long-run P!"""
    cov = R +  *(B.T)@(P_next@B)
    Z =  *(B.T)@(P_next@A) + S
    # (-ve) of policy function
    F = linalg.solve(cov, Z)     # A@x = B, x = np.linalg.inv(A)@B
    # Value function (quadratic)
    P = Q - (Z.T)@F +  *(A.T)@(P_next@A)
    return P, -F
```

An aside. For the reader familiar with the Kalman filter, you will see a similar recursion derived in the Kalman optimal projection problem. The Kalman filter is often used in forward projections

or backward smoothing in linear, Gaussian hidden Markov models of time-series statistics. This was originally stuff designed for rocket science and the research was financed by the U.S. Air Force in the 1950s.

Now, our backward-induction algorithm:

[5]:
```
def lq_backsolve(T, P_next,  , A, B, Q, R, S):
    """Solve finite-horizon, deterministic LQ program
    with horizon T, given DP model ( , A, B, Q, R, S)
    and terminal stage valuation quadratic matrix, P_next."""
    P_list = []
    F_list = []
    P_next = P_terminal
    for t in range(T, -1, -1):
        # Evaluate stage-t value and policy functions
        P, F = riccatti(P_next,  , A, B, Q, R, S)
        # Store these
        P_list.append(P) # P_t
        F_list.append(F) # F_t
        # Step backward again
        P_next = P
    # Sort in ascending t order
    P_list = P_list[::-1]
    F_list = F_list[::-1]
    return P_list, F_list
```

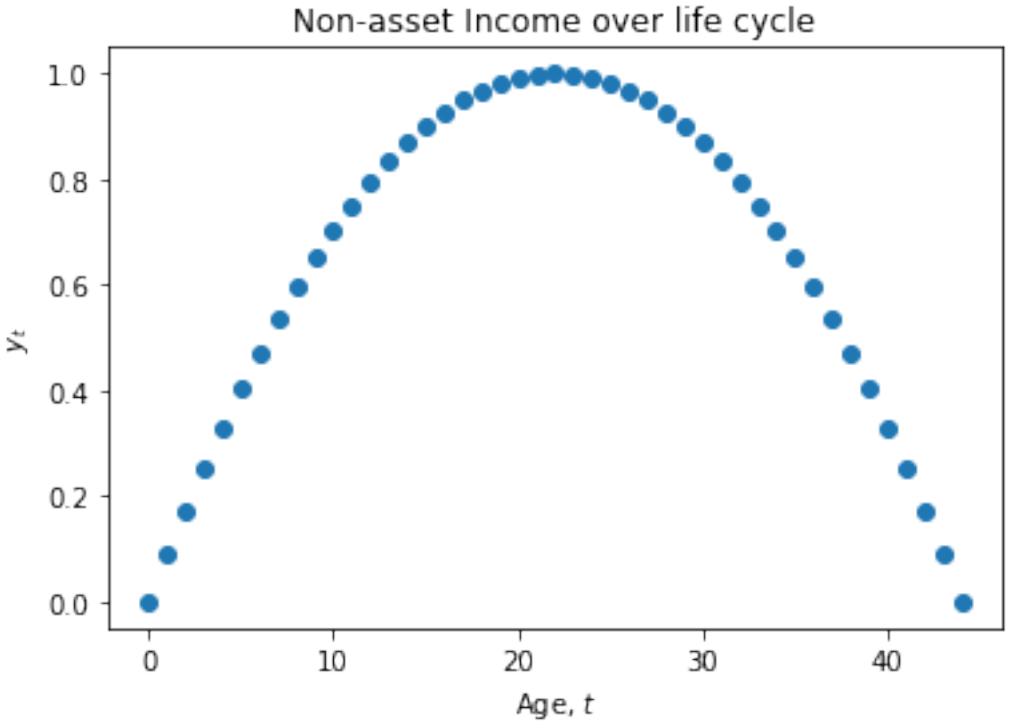
1.4 Numerical example

Let's parametrize the model:

[6]:
```
# Exogenous gross real return on asset
RR = 1.05
# Subjective discount factor
  = 1/RR
# Life-cycle horizon
T = 44
# Ideal consumption level
cstar = 2.0
# Warm-glow scale (cost)
k = 1e6
# Exogenous income process/trend
ybar = 0.0                      # intercept coeff
 1 = (T)/((T)/2.0)**2.0           # linear trend coeff
 2 = -1.0/((T)/2.0)**2.0        # quadratic trend coeff
```

Here's a peek at our deterministic (non-asset) income path of the consumer's lifecycle:

[7]:
```
plt.figure()
t_range = np.arange(T+1)
y_range = y(t_range, ybar,  1,  2)
plt.plot(t_range, y_range, 'o')
plt.xlabel('Age, $t$')
plt.ylabel('$y_{t}$')
plt.title('Non-asset Income over life cycle')
plt.show()
```



The income-age profile is an inverted-U or hump shaped path. This what is typically observed in microdata at the household level, when we sort individuals by their age groups.

Let's define the arrays making up our controllable linear state-space form (6) of the recursive budget constraint:

[8]:
```
# Linear: Constraint
A = np.array([[1.,          0.,  0., 0. ],
              [1.,          1.,  0., 0. ],
              [1.,          2.,  1., 0. ],
              [ybar-cstar,   1,   2, RR ]])

B = np.atleast_2d([0., 0., 0., -1.]).T
```

Now let's set up the matrices defining our objective function in (8):

[9]:
```
# Quadratic: Objective
Q = np.zeros((4,4))
R = 1.0
S = np.zeros((1,4))

# Terminal stage payoff function (quadratic matrix)
P_terminal = Q.copy()
P_terminal[-1,-1] = k
```

Pass these model elements to the backward-induction algorithm:

[10]:
```
PT, FT = lq_backsolve(T, P_terminal,  , A, B, Q, R, S)
```

The output consists of the sequence of matrices defining the value and optimal regulation functions $\{P_t, F_t\}_{t=0}^{T} \Rightarrow \{W_t, h_t\}_{t=0}^{T}$.

1.4.1 Analyzing and visualizing optimal plan

This code below starts off the decision problem with a consumer that has zero initial assets $a_0 = 0$.

It then loops over the agent's lifetime, with age indexed by $t = 0, 1, ..., T$.

For each age/date t, the agent uses the optimal control rule h_t to select her best consumption outcome as a function of the date and state.

This also induces a corresponding optimal transition in the asset level via mapping g_t.

We record the outcomes for each date t using Python lists:

[11]:
```
# Initial state with some positive initial asset balance
# Recall x_{t} = (1, t, t^2, a_{t})
x0 = np.atleast_2d([1., 0., 0., 0.0]).T

# Simulate path
x = x0.copy()
asset = []
asset.append(x[-1])
consumption = []
for t in range(T):
    # Evaluate optimal transition map
    x_next = g(x, A, B, FT[t])
    # Store asset outcome
    asset.append(x_next[-1])
    # Optimal control rule: consumption outcome, store
    consumption.append(cstar + FT[t]@x)
    # Update history of state outcomes
    x = x_next
# Flatten into 1D numpy arrays
asset = np.asarray(asset).ravel()
consumption = np.asarray(consumption).ravel()
```

Let's visualize the age profile of the optimal consumption plan against the lifetime non-asset income:

[12]:
```
plt.figure()
# Non-asset income profile
t_range = np.arange(T)
y_range = y(t_range, ybar, 1, 2)
plt.plot(t_range, y_range, 'o', label='$y_{t}$')
# Consumption profile
plt.plot(t_range, consumption, '--', label='$C_{t}$')
plt.xlabel('Age, $t$')
plt.title('Non-asset income and consumption over life cycle')
plt.legend()
plt.show()
```

Her optimal consumption path is *smooth*.

The fast-living years. Notice how she is consuming more than her non-asset income each period for some initial years?

That means she must be borrowing.

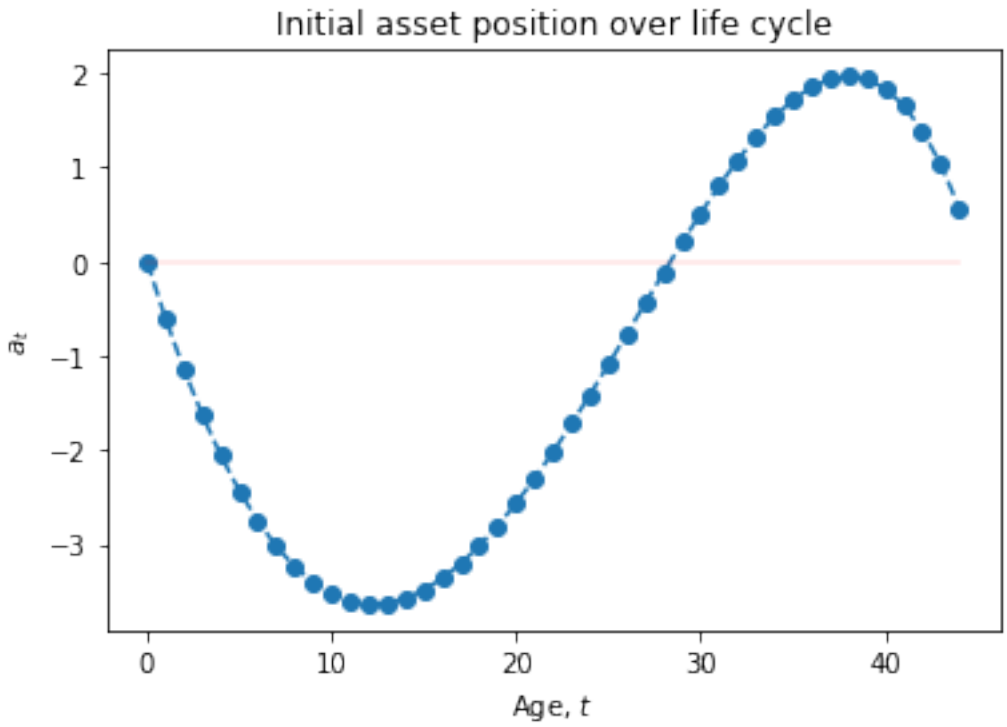
Mid-life saving. But her sequence of budget constraints must add up to imply a solvent intertemporal saving/borrowing-consumption plan. (Do the algebra and you can show this argument!) So she would have to start saving around middle age. This saving is optimally spread to cover dissaving in the initial and the latter years.

Twilight is the loneliest time of day. As she gets on in life, her income profile is falling (by assumption). We know the agent dislikes too much deviation of consumption from her ideal point $c^\star$. So she is optimally smoothing consumption. That means she has to dissave towards the end of her economic-actor life.

On the flipside of the consumer's budget constraint, we can also work out her asset position.

Here, we plot the path of date-t *initial* asset positions.

[13]:
```
plt.figure()
# Asset profile
t1_range = np.arange(asset.size)
plt.plot(t1_range, asset, '--o')
plt.plot(t1_range, np.zeros(t1_range.size), '-r', alpha=0.1)
# Finer x-tick labels
# plt.xticks(t1_range)
plt.xlabel('Age, $t$')
plt.ylabel('$a_{t}$')
plt.title('Initial asset position over life cycle')
plt.show()
```



Now we can also verify that if k is very large the terminal stage consumption is (approximately) the same as initial (non-asset plus asset) income:

[14]:
```
# Terminal consumption
C_terminal = consumption[-1]
```

[15]:
```
# Terminal income
Y_terminal = RR*asset[-1] + y(T, ybar,  1,  2)
```

So post-mortem initial asset a_{T+1} is ...

[16]:
```
Y_terminal - C_terminal
```

[16]:
```
-1.4693191954240348e-06
```

1.4.2 Exercise

1. Show that the terminal period optimal consumption in this model has this exact formula:

$$C_T = \frac{\beta k}{1+\beta k}\left(Ra_T + y_T\right).$$

2. Then show that the indirect utility at the beginning of stage $t = T$ is

$$W_T(x_T) = \frac{\beta k}{1+\beta k}\left(\bar{y} + \kappa_1 T + \kappa_2 T^2 + Ra_T\right)^2.$$

3. How does your answer in the last part relate to the matrix P_T?
4. Now can you verify that if $k \to \infty$, $a_{T+1} \to 0$?

1.5 Postscript

Additionally, if we extend this problem to include random shocks to income, for example, the solution can be viewed as a time-varying-parameter (TVP) linear, multiple-time-series stochastic process (also know as a TVP-VAR model amongst time-series econometricians).