Quantitative Macroeconomics: Project 2

Marta Oliva Riera

Problem b

1.Explain the code

0. Setting up the parameters, creating the grid points for x_t and the income shocks (ε_t).

1. Solution:

First set up an initial guess for consumption and for the value function, and then iterate. For each point in the grid, check whether the borrowing constraint is binding: if it is (when $\mu >= 0$) then set $c_t = x_t$. If the constraint is not binding, use Brent's method on the FOC to find roots and set consumption equal to the maximum between the root and a very small number.

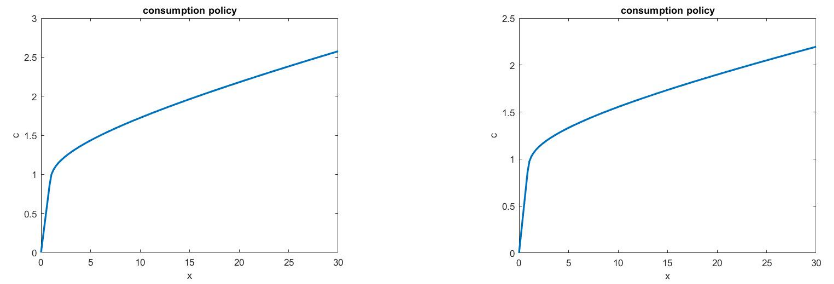
Then update the value function ($V_t = c^{-\theta}$)) using the previously obtained consumptions and finish by checking if there is convergence: if $|c_t - c_{t-1}| < E$ there is convergence and we stop, otherwise repeat until the values of consumption converge. Once there is convergence we can plot the policy function for consumption.

2. Simulation:

Run a simulation with 1100 repetitions. In each iteration, generate a value for income ($y_t = \varepsilon_t$), cash at hand ($x_t = a_t + y_t$), consumption (using function interpolation from the consumption policy function we got in the previous point) and assets next period ($a_{t+1} = (x_t - c_t)\frac{1+r}{1+g}$. If we are at an interior solution (the borrowing constraint isn't binding $c_t < x_t$ and our c_t is not the maximum of the policy function) then calculate the Euler equation error:

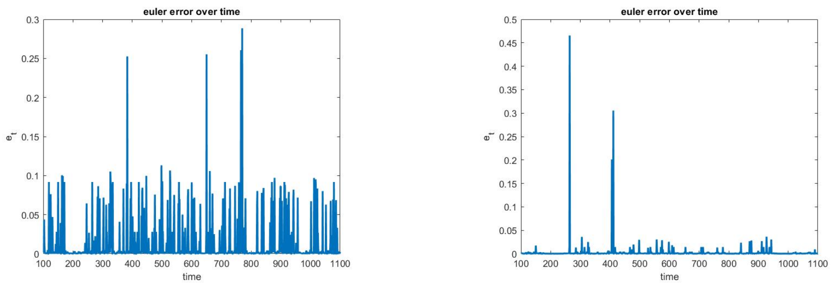
$$e_t = \frac{c_t^{-\theta} - (\beta\frac{1+r}{(1+g)^{-\theta}}V(x_t))}{c_t^{-\theta}}$$

The simulation is useful to see what values result from our policy function if we use it over a long period of time (t=1000 periods). The error evaluation inside the simulation is computing the Euler equation error, which is the percentage of mistakes we make due to our numerical approximation of the expectation - so it represents how accurate our results are.



(a) $\theta = 1$ (b) $\theta = 4$

Figure 1: Policy function with different risk aversion



(a) Variance = 0.01 (b) Variance = 0.05

Figure 2: Euler equation error with different income shock variances

2. Changing parameters

The larger the value of θ the more risk averse the individuals are, and the less steep the policy function for consumption become, as can be seen in figure 1. That means that more risk averse individuals tend to consume less, because they save more. Additionally, the iteration takes longer when we increase the relative risk aversion.

Increasing the variance of the income shocks from 0.01 to 0.05 results in changes in the Euler equation error in the simulation: the maximum error increases, but the average error for the simulation actually goes down compared to the case with lower variance, as plotted in figure 2. In the resulting graphs it is clear that the larger variance results in some much larger outliers in the error term but in a lower overall level. That means that having larger uncertainty over labour income (due to the more varied shocks) results in less errors on average in our simulation, but some which are much larger.

3. Increase the grid size

Increasing the grid size from 20 to 50 points increases the computation time, as would be expected. Also as expected, convergence of the policy function takes less iterations the more grid points we have (from 64 to 57). However, the increase in grid size actually increases the mean Euler equation error, going from 0.74 percent to 1.16 percent. That means that our numerical approximation makes more mistakes in the second case.

Problem c

Value function iteration

I used Python code to perform the value function iteration, because I modified the code I had already written for Homework 4. To do so, I added the *makegrid* function to create the grid in the same way that was done in the provided code. Then I create the grid with 50 points, set the parameters equal to those in the previous part and create the income shocks in the same manner.

As the following step I generate the returns matrix, and then I iterate over the value functions. Because we now have seven possible states of the income shock, the returns matrix becomes three dimensional (50x50x7). Having adapted the iteration process for this change, I obtain the value function below, where I have a value function for each possible future state of the labour income shock.

Convergence takes only four iterations and around 2 seconds, so it is a lot faster than the previous process.

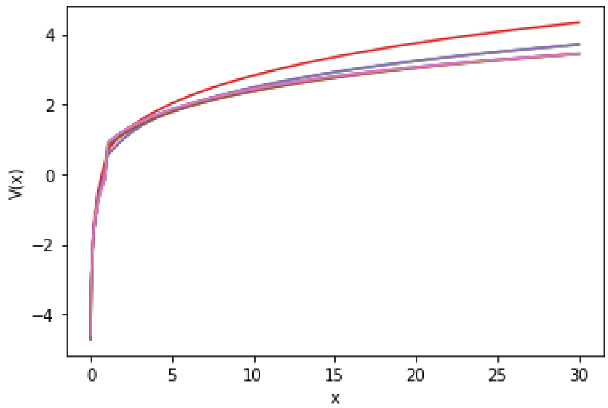


Figure 3: Value function

Howard's improvement algorithm

I was not able to do this in HW4, and I didn't have time to get it this time either.

Problem d

Here I tried to transform the code provided to solve the problem for finitely lived individuals, so $T < \infty$, but it did not work so I'll explain how I would do it in theory.

Because we have a finite horizon, we need to set a maximum age which individuals can live to, which I decided to make $T = 80$ like in the previous project. Then we go to the last period and iterate backwards, starting from the fact that we know at that at T future cash at hand will be zero, because individuals know they will die. Then:

$$(T) : x_{T+1} = 0 \Rightarrow V(x_T) = Max_{c_T} u(c_T) \to c_T = x_T$$

$$(T-1) : V(x_{T-1}) = Max_{c_{T-1}} u(c_{T-1}) + \beta E_{T-1}(1+g)^{1-\theta} V(x_T)$$

To implement this in the code we would have to set the initial condition for the last period ($x_{T+1} = 0$) and then iterate on the value functions backwards similarly to how we did in the previous project. We would also need to include something to check for the budget and borrowing constraints to be satisfied: if they are, the backwards loop can continue using the obtained c_t into the previous period, but if they are binding we should make $c_t = x_t$ and continue the loop from there.

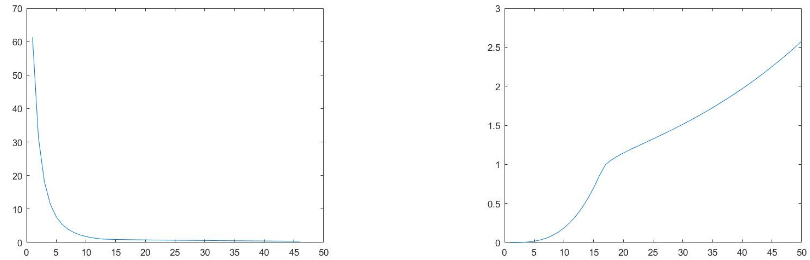
Problem e

Interpolate on Λ_x instead of V_x

Having run the provided code, figure 4 represents what the resulting Value function ($V_x = u_c = c^{-\theta}$) and $\Lambda_x = (V_x)^{\frac{-1}{\theta}} = c$ look like. In theory, by interpolating on Λ_x we would have a linear function of c instead of the more complex value function, such that interpolation should be faster - but I was not able to get this implementation to work on the code.

Exogenous grid on savings

The shadow value is evaluated up to 5000 times (100 maximum iterations on 50 grid points). The lowest value possible taken by savings is zero (if the borrowing constraint is binding $x_t = c_t$), while the largest possible is 30 (when $c_t = 0$ and x_t is at the maximum grid point).



(a) V_x (b) Λ_x

Figure 4

By working with an exogenous grid on savings instead of on cash at hand (ENDGM instead of EXOGM) we would only have consumption appear once in the FOC instead of twice, making the computation faster.

$$FOC(EXOGM) : U_c - \beta R E_{\eta'|\eta}(\frac{\delta V_{t+1}}{\delta x'}((x-c)R + y\eta')) = 0$$

$$FOC(ENDOGM) : U_c - \beta R E_{\eta'|\eta}(\frac{\delta V_{t+1}}{\delta x'}(sR + y\eta')) = 0$$

Problem f: Consumption over the life cycle - Gourinchas and Parker

This paper estimates a structural model of the life-cycle saving decisions of households which face stochastic labour income processes. It does so in two steps: first, it constructs consumption and income profiles over the working life of households in different educational and occupational groups, using data from the Consumer Expenditure Survey. Then, using those profiles and a measure of labour income uncertainty, they solve the model numerically for the optimal behaviour and aggregate to generate a simulated life-cycle consumption profile. Matching the simulated profile and the one from the data, they estimate the preference parameters for consumption by using the simulated method of moments.

In their model the optimal choice for consumption depends on lifetime resources, real interest rate, discount factor and also the expected growth rate of income. When income uncertainty and prudence are introduced, households use the one available asset to insure themselves, increasing the slope of their consumption profile. Very similar to the model that we were solving for in the problem set, they also have income as a function of a permanent and transitory component ($Y_t = P_t U_t$) which introduces uncertainty, and use the same normalization dividing by the permanent component that we used.

They have four main findings. First, the model matches the correlation between consumption and income for young ages and the concave consumption profile observed in the data. Secondly, they obtain reasonable estimates for the preference parameters. They also find evidence on the determinants of wealth accumulation: early in life households save mainly due to a precautionary savings motive arising from their larger labour income uncertainty, while households older than 40 tend to save with retirement and bequest motives in mind. The saving patterns found in the model are consistent with the forward-looking optimizing behaviour in LC models. Finally, the the estimated model finds significant age heterogeneity in consumption/savings behaviour: young consumers behave as buffer stock agents (they have a target of assets around which they fluctuate) while older ones act closer to the certainty equivalent life cycle hypothesis model.

The low holding of assets by young households is an optimal response to their expected income growth and the risk associated to labour income over the life cycle. For the older households the retirement and bequest motives have relatively more importance, consistently with CEQLCH. The two clearly distinct phases of consumer behaviour over the life cycle are at the heart of the identification procedure of the model.