Computational Appendix

1 Introduction

This article briefly explains the usage of the program codes that compute our calibrated economy and quantitative exercises. The outline of this article is as follows. The second section describes our calibration strategy of aggregate and idiosyncratic shocks. In section 3, we provide the explanation of what each program code does. Section 4 gives the instruction of how to replicate all of the tables listed in our paper.

2 Calibration Programs

Our calibration strategy is based on Alvarez and Jermann (2001). The first 4 moments pin down the aggregate shocks and the aggregate transition matrix. The final 2 moments identify the idiosyncratic shocks and the idiosyncratic transition matrix. The calibration program codes are in "CalibrationProgramCode" folder.

2.1 Aggregate Shocks

Consider a two state first order Markov Chian for aggregate shocks. The first aggregate state is a recession and the second aggregate state is an expansion. We calibrate this aggregate process by 4 moments, M1 through M4, directly from Alvarez and Jermann (2001) and Mehra and Prescott (1985), expect there is no predictability of aggregate shock. These 4 moments include (1) the first order auto-correlation of aggregate consumption growth (ii) the relative frequency between expansion and recession (iii) average consumption growth rate (iv) the standard deviation for consumption growth rate.

The program code, "calibration_agg.m", give the following results. The transition probability

$$\pi(z'|z) = \begin{bmatrix} 0.2740 & 0.7260 \\ 0.2740 & 0.7260 \end{bmatrix}$$

and the two aggregate consumption growth shocks

$$\begin{aligned} z_R &= 0.9602 \\ z_H &= 1.0402 \end{aligned}$$

2.2 Idiosyncratic shocks

We also consider a two state first order Markov Chian for idiosyncratic shocks. The first state is low and the second state is high. We calibrate this aggregate process by 2 moments, M5 and M6, which are the standard deviation of idiosyncratic shock and the first order auto-correlation of the shocks respectively.

The program code, "calibration_idio.m', gives the following results. The transition probability

$$\pi(z'|z) = \begin{bmatrix} 0.9450 & 0.0550 \\ 0.0550 & 0.9450 \end{bmatrix}$$

and the two state of idiosyncratic shocks (normalized the mean to 1)

$$\begin{aligned} \eta_L &= 0.3894 \\ \eta_H &= 1.6106 \end{aligned}$$

3 Main Programs

The main programs consist of three parts. The first part of codes setups the computational environment, the second part solves the equilibrium prices and the final part delivers all results reported in our paper. All program codes in this section are in the folder named "MainProgramCode".

3.1 Part I: Environment Program

There is only one program file in the first part. The program is called "environment.m", which does the followings.

1. Define parameter values, aggregate shock process, idiosyncratic shock process and history of aggregate shocks.
2. Compute the implied stationary discount factor, the twisted probability of all aggregate states and idiosyncratic states in stationary economy.
3. Create transition matrices describing law of motion in aggregate truncated history states as well as idiosyncratic states.
4. Make an initial guess of the growth rate of aggregate multiplier.
5. Make a linear grid points on the domain of policy functions and saving functions. We use consumption share the state variable in addition to the truncated aggregate history state and current idiosyncratic state.
6. Create a simulation panel of aggregate and idiosyncratic shocks.

3.2 Part II: H-update Program

The major program code of part II solves for the stochastic discount factor defined on the equation (13) in the Web Appendix available online. The computational algorithm is also described in Web Appendix . The file name of this program is "H_update.m". By equation (13), the stochastic discount factor is known once the growth rate of aggregate multiplier, $(h_{t+1}(z^{t+1})/h_t(z^t))$, is found. Some details of this program code are described as follows.

1. Define the fraction of traders, leverage ratio, portfolio target share as well as the length of non-rebalancing periods.

2. Given the guess of the growth rate of aggregate multiplier, "H", compute the implied state contingent prices, risk free rate, leveraged equity return, price dividend ratio, portfolio return for crb traders. We will verify the guess of "H" by checking the excess supply of goods after the simulation part in the end of the program. Update the "H" until the excess supply is sufficiently close to zero.

3. Make an initial guess for saving functions for each type of traders. The saving function depends on aggregate history states, idiosyncratic states and current consumption share of the agent.

4. Solve for the optimal saving function and policy function for each type of traders. To solve these functions, the measurability constraints and solvency constraints are imposed directly on the saving function and the optimal law of motion in consumption share (from first order conditions) is employed to the intertemporal budget constraints. For the details, please refer to the appendix of Chien, Cole and Lustig (2011).

5. Simulate the economy for T periods and $N-1$ agents for each type of traders. Compute the implied "H" matrix and update the new guess of "H" until the new "H" is sufficiently close to the guess.

6. Save all results including optimal saving functions, consumption policy functions and equilibrium prices.

3.3 Part III: Programs for Reported Results

This set of program codes computes the results of asset pricing, consumption statistics and wealth statistics. These program codes include asset pricing program, consumption program and wealth program.

3.3.1 Asset Pricing Program

The program code named "Ap_function.m" compute the asset pricing results. It delivers the followings.

1. Compute the pricing kernel, the upper bound of Sharpe ratio, standard deviation of market price of risk, risk free rate and standard deviation of risk free rate.

2. Compute the pd ratio of aggregate consumption claim.

3. From section 4.2, the return of leverage equity claim is given by the following formula

$$\begin{aligned} R^D_{t+1,t}[\{D\}] &= \frac{V_{t+1}[\{D\}] + D_{t+1}}{V_t[\{D\}]} \\ &= \frac{(1-\gamma)V_{t+1}[\{Y\}] + (1-\gamma)Y_{t+1} - R_{t+1,t}[1]B^s_t}{V_t[\{D\}]} \\ &= \frac{(1-\gamma)V_{t+1}[\{Y\}] + (1-\gamma)Y_{t+1}}{(1-L)(1-\gamma)V_t[\{Y\}]} - \frac{R_{t+1,t}[1]B^s_t}{V_t[\{D\}]} \\ &= \frac{1}{(1-L)}\left(\frac{V_{t+1}[\{Y\}] + Y_{t+1}}{V_t[\{Y\}]}\right) - \psi R_{t+1,t}[1] \end{aligned}$$

   where $L = \frac{\psi}{1+\psi}$

4. Simulate the moments of asset pricing and produce results.

3.4 Consumption Program

The household consumption and portfolio return statistics are computed by a program code named "H_update_consumption.m". It does the followings.

1. Given equilibrium prices, simulate the consumption, saving and portfolio data of different types of traders.

2. Compute consumption statistics and portfolio investment statistics to each type of traders. These statistics include: standard deviation of individual consumption growth rate, standard deviation of group consumption growth rate, average excess return of individual wealth, volatility of excess return of individual wealth, Sharpe ratio of individual portfolio return.

3. Report the statistics results of household consumption and portfolio return.

3.5 Wealth Program

The wealth and portfolio statistics are computed by a program code named "H_update_wealth_return.m". It does the followings.

1. Given equilibrium prices, simulate consumption, saving and portfolio data of different types of traders.

2. Compute wealth statistics for each type of traders. These statistics include: average wealth, standard deviation of average wealth, average wealth ratio, standard deviation of wealth ratio, average equity share, standard deviation of equity share, auto-correlation of equity share.

3. Report the statistics results of household wealth.

4 Program Codes for Tables

For replication purpose of our tables, we provide all necessary program codes for producing each table listed in our paper. For each table, there is a corresponding folder, which contains some of the program codes listed and explained by section 3.

1. **Table 1**: Please refer to the program code in the folder named "Table I". To produce this table, simply run the asset pricing program code, AP_function.m.

2. **Table 2**: The results of table II can be obtained by running the asset pricing program code, AP_function.m, which is in the folder named "Table II".

3. **Table 3**: The asset pricing program code, AP_function_table3.m, reports the results of table III. The related files can be found in the folder named "Table III".

4. **Table 4**: The program codes produce Table IV are in the folder called "Table IV'. The results of first and second panels are done by the program, H_update_Consumption.m, and the third panel is by H_update_wealth_return.m.

5. **Table 5**: Please refer to "Table V" folder. There are three panels in this table. The results of each panel can be obtained by running the asset pricing program code, AP_function.m, under each separate sub-folder.

6. **Table 6**: "Table VI" folder has the program code of producing statistics in table VI. Simply run the asset pricing program code, AP_function.m, to get the results.