sarmi.m

Ari Anisfeld

February 19, 2019

1 Define Variables and Set-up Grid (lines 1 - 470)

This code is nearly identical to armi.f90

2 Load policy functions (lines 470- 1200)

For each file year_t, rear_t
we unpack a tensors
$cons_t$ the consumption paths from armi.f90
def_t the default flags from armi.f90
$consdef_t$the alternative consumption path from rent.f90

3 Simulations

We complete 40000 simulations (indexed by j). In the following description I will refer to objects as vectors, when they are often one column of a 21 x 40000 matrix. There are several accumulation objects that are in fact 21 x 1 vectors. In the code these have a 0 at the end of their name. (Update: While the accumulation objects collect the average path, they are never used. Perhaps they were included for FRMs or diagnostic purposes).

3.1 Generate paths (lines 1200 - 1480)

Randomness is built in at this stage. The strategy is to draw a 21 x 1 vector of a random variable x drawn from a uniform distribution over [0,1].

Then compare x to some number generated in part 1.

These are used to determin indexes into the consumption / default policy functions like so

```
cons_t(1,indcash(i,j),indr(1,j),indnr(1,j),indlref(1,j),indni(1,j),indir(1,j))
```

- indcash 1:185 (points on the grid cash)
- indr 1:2 (high or low inflation)
- indnr 1:t (number of times we see low inflation)

- indlref 1:t (number of times we have a low permanent income shock)
- indni 1:t (number of times we see low house price growth ("low innovation"))
- indir 1:2 (high or low interest rates)

3.1.1 Generate path for inflation

We make 21 x 1 vectors *indr* (1 or 2), *indnr* (1 to t+1) and$simr_{:,j}$ (g_{1r} or g_{2r}).

This section of code is overly complicated. The first for-loop is not necessary, nor is the inner for-loop in the next chunk of code.

In the first case, we only do an action if i == 1, so there is no need for a loop.

In the second for-loop we want to either transition to 1 or not. So for each k, we can ask "if rn(k,1) < acum(aux,i)" and determine *indr*.

Once *indr* is established, we make $simr_{:,j}$ the path of inflation for person j by pulling the high or low state from g_r.

3.1.2 Generate real interest rate

We make 21 x 1 vector*indir* (1 or 2) depending on transition matrix (*tpi*).

3.1.3 Generate real house prices

/We make 21 x 1 vectors *indi* (1 or 2) depending on transition probs (*prh*) and *indni* (1 to t+1) counts the number of low house price growth past.

3.1.4 "Same aggregate shocks for 50 individuals"

This section of code is overly complicated. There is no need to distinguish between j <=50 and j >= 50.

For groups of 50 people assign the "aggregate shocks" generated above for the first person of that group.

This suggests that the previous code was 50 times more than necessary.

3.1.5 Generate permanent income

We make a 21 x 1 vector *indpy* (1 or 2) depending on transition prob (*pyh*) and *indlref* (1 to t + 1) counts number of low permanent incom shocks.

3.1.6 Generate labor income/house prices

We make a 21 x 1 vector *simy* (real numbers) and *simph* (real numbers).

There are two *state*s of the world with equal probability (hard coded) for *simy*.

$simy_t = labinc_t \cdot e^{eyt(indir_t, indr_t, state)} \cdot yp_2^{(indlref_t - 1)} yp_1^{(t - indlref_t)}$

$simph_t = gph_2^{(indni_t - 1)} \cdot gph_1^{(t - indni_t)}$

Note that $simph_1$ is normalized to 1. So we are taking the product of growth rates in the low state ($indni_t - 1$) and high state.

3.1.7 Generate ind forced move

We make a 21 x 1 vector *indmove* (0 or 1, with 1 indicating a move) the decision has two factors:

First, if the homeowner is in the money and the random number is below *probmove*, they move.

Second, if the random number is below $probmove \cdot ratiolock$, they move.

3.2 First period (lines 1480 - 1540)

t = 1

3.2.1 Find current cash situation.

$cash = (1 - tax) \cdot simy_t$ (Note. The first few lines of code regarding *simw* and *simpl* are redundant / over written later in the code).

Set $simcoh_t = cash$

Use *ntoi*() function to find index of cash relative to grid.

We set $simw_t = cash$

Simw appears to be hold wealth.

3.2.2 Is there default? and updates

Set $simdefc_t$ and $simcons_t$depending on default situation (indexed from policy functions).

Set $sav_t = cash - simcons_t + simdebt_t$(Note: debt = 0).

We set auxilary variables which capture the growth rate of inflation in time t and t+1

$auxprice = e^{g_{2r}^{indnr_t - 1}} \cdot e^{g_{1r}{}^{t - indnr_t}}$

$auxprice_1 = e^{g_{2r}^{indnr_{t+1} - 1}} \cdot e^{g_{1r}{}^{t - indnr_{t+1}}}$

Preliminarily set $simw_{t+1} = sav_t \cdot (1 + (e^{g_{(m)r} + g_{(k.m)i}} - 1) \cdot (1 - tax)) \cdot \frac{auxprice}{auxprice_1} - simdebt_t \cdot e^{g_{(k,m)i}}$ (Note: debt = 0).

With m, k as 1 or 2 depending on $indr_t$and$indir_t$respectively.

$simpl_t = auxprice$ (this equals 1 when t =1).

(Note in the code the following are defined redundently in the if/else statement).

$anaux = loanrepaid_t + (e^{g_{(m)r} + g_{(m,k)i}} - 1 + \theta) remdebt_t$

$mpreal_t = anaux/auxprice$

$rentpay_t = ((e^{g_{(m)r} + g_{(m,k)i}} - 1) - (e^{\mu_{ph} + \frac{\sigma_{dph}^2}{2}} \cdot e^{g_{(m)r}} - 1) + tax_p + m_p + rentalpr) \cdot simph_t \cdot house$

- In the case of no default update:

$simw_{t+1} = simw_{t+1} - \frac{anaux}{auxprice} - (tax_p + m_p) \cdot house \cdot simph_t + simy_t(1 - tax) + tax \cdot (e^{g_{(m)r} + g_{(m,k)i}} - 1 + \theta) \cdot \frac{remdebt_t}{auxprice} + tax \cdot tax_p \cdot simph_t \cdot house$

(Note: the above equation can be slightly simplified, but this how it is in the code.)

$mpnominal_t = anaux$

- In the case of default update:

Next period wealth does not have all the housing costs and benefits as above.

$simw_{t+1} = simw_{t+1} + simy_t(1 - tax) - simph_t \cdot house \cdot anaux_1$

where $anaux_1 = (e^{g_{(m)r} + g_{(m,k)i}} - 1) - (e^{\mu_{ph} + \frac{\sigma^2_{dph}}{2}} \cdot e^{g_{(m)r}} - 1) + tax_p + m_p + rentalpre$

$mpnominal_t$is not defined which seems like a potential bug (it is zero by default and will not throw an error).

3.2.3 Other causes of default or prepay

If next period wealth $simw_{t+1} < 0$ and next period house price after seller's fees is below $remdebt_{t+1}$:

$simdefc_{t+1} = simdef_{t+1} = 1$

$simw_{t+1} = 1$ (lowest acceptable cash)

If next period wealth $simw_{t+1} < 0$ and next period house price after seller's fees is above $remdebt_{t+1}$:

(Note the inline comment is misleading)

$simdefc_{t+1} = 2$

$simdef_{t+1} = 1$

$simw_{t+1} = simw_{t+1} + .94simph_{t+1} \cdot house - \frac{remdebt_{t+1}}{auxprice1}$

Finally, we ensure that the cash on hand is within our grid and set the next period def to 1 if it is currently 1.

3.3 Remaining periods

t = 2:20

3.3.1 Find current cash situation.

$simcoh_t = cash = simw_t$ with the caveat that it's on the cash grid.

3.3.2 Is there a default or prepay?

There is a lot of repeated code here, because Cocco didn't have enough RAM to run the program in one go.

If j has not defaulted previously and if $indmove_t = 1$,

We simulate a move, which is a prepay if they are over water, and a default otherwise.

In the case they prepay/sell home, we update cash to reflect the financial gains.

Set $simdefc_t$ (if j still in home) and $simcons_t$depending on default situation (indexed from policy functions).

3.3.3 Updates

Identical to above, plus an update to $simltv_t = \frac{remdebt_t}{simph_t simpl_t house}$

Finally, they fill in $simcons$ and $simdefc$ for t=21.

4 Calculate PV of profits of mortgage providers (lines 2230 - 2290)

This section is not relevant to us. It determins pvsimpi in the output file.

5 Output (lines 2290 - end)

For each simulation, a 17 by 20 "dateframe" is stacked on the output dataframe. This includes

1. id
2. age (of loan 1:20)
3. income (simy)
4. consumption (simcons)
5. savings (sav)
6. cash on hand (simcoh)
7. default / prepay flag (simdefc)
8. inflation growth (exp(gr))
9. log real rate growth (simexpgi)
10. house price (simph, normalized to 1 in t=1)
11. simpl
12. ltv (simltv)
13. remaining debt (there is no randomness here).
14. mpreal
15. rentpay
16. pvsimpi
17. simmove