Monte Carlo Simulation Using Markov Transition Probability

Dr. Guanyi Yang

St. Lawrence University

March 7, 2019

Markov Chain

Using Tauchen Method, we get discretized aggregate shock state:

$$z_t \in \{z_1, z_2, z_3, ..., z_N\}$$

with transition probability matrix:

$$\Pi = \begin{bmatrix} \pi_{11} & \pi_{12} & \cdots & \pi_{1N} \\ \pi_{21} & \pi_{22} & \cdots & \pi_{2N} \\ . & & & \\ . & & & \\ . & & & \\ \pi_{N1} & \pi_{N2} & \cdots & \pi_{NN} \end{bmatrix}$$

Using Markov Chain to simulate the economy for many periods, AKA Monte Carlo Simulation

Our goal is to simulate an economy for, say, 100 periods, and the TFP transitions from period to period according to the Markov Chain probability.

Let's arbitrarily assume a starting point. It can be any point on the *zgrid*. Let's assume that TFP in period 1 is z_3.

In period 2, the probability of landing in any z state has to follow the Markov Matrix. In our example, it has to follow the second row.

$$[\pi_{31}, \pi_{32}, ..., \pi_{3N}]$$

What this is saying is that given we are currently at state z_3, the next period probability in z_1 is π_{31}; the next period probability in z_2 is π_{32}; the probability in z_3 is π_{33};... ; in z_N is π_{3N}.

Problem: How do we let computer randomly draw a number, while following this arbitrary probability?

A numerical example

Let's explain this algorithm using a numerical example. Let *zgrid* size be 5, and the transition probability given currently :

$$[\pi_{31}, \pi_{32}, ..., \pi_{35}] = [0.0047,\ 0.1414,\ 0.5423,\ 0.2908,\ 0.0208]$$

What this is saying is that we have probability of 0.47% chance of drawing z_1, 14.14% chance of drawing z_2, and so on.

With some rounding errors, if you sum them up, you should get probability 100%. i.e. you are 100% sure that you are going to hit any one of the 5 possible next period states.

Uniform distribution

Matlab has a built-in function **rand** that randomly generates a number between $(0, 1)$, with equal probability (uniform distribution).

Uniform distribution has this quality, where if the number generated is 0.01, it also represents the cumulative probability of it. Let's say we are looking at the probability of drawing something between 0 and 0.01; uniformly distributed numbers between 0 and 1 will have 1% chance hitting this bracket.

If we want to find the probability of drawing something between 0.01 and 0.2; uniformly distributed numbers between 0 and 1 will have (0.2-0.01) $=$ 19% chance of hitting this bracket.

Random number generator following an arbitrary probability

We can exploit this feature of uniform distribution, or **rand** in Matlab in locating the random draw form Markov Chain

We can let Matlab generate random numbers between (0, 1). with any number generated below 0.0047, we consider it as drawing our z_1.

If we add $0.0047 + 0.1414 = 0.1416$. Any chance **rand** generated more than 0,0047, but less than 0.1416, we consider matlab draws state z_2.

So on and so forth.

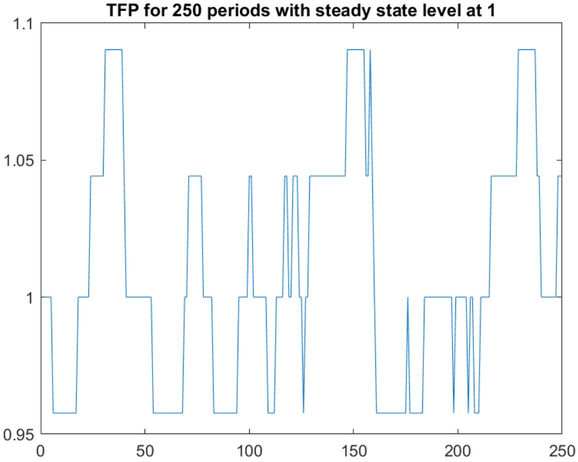
If you sum up all first four probabilities, you will get 0.9747. If **rand** draws any number between 0.9747 and 1, we consider it is drawing z_5.

Random number generator following an arbitrary probability

An example code for Matlab to manipulate the random number generator:
`r = rand` % randomly generate a number between 0 and 1, store at name r

$index = sum(r >= cumsum([0, prob_1]))$ % assume I previously give matlab a Markov Probility called $prob_1$, it returns a raw in the *zgrid* that follows the Markov Probability, and name it *index*.

Result sample



I use $\rho = 0.909; \sigma_e = 0.014; znum = 5$; and simulated 250 periods

How to simulate the economy following the TFP?

Next step, we not only want to know how the TFP fluctuates, we also want to know how the GDP and the rest of the variables in the economy fluctuate.

Remember everything in the economy is solved on the grid?

Economy with uncertainty has a two-dimension grid: $Kgrid \text{X} zgrid$.

To simulate the economy to follow our steady state decision rules, and the Monte Carlo series of TFP, we exploit this grid and direct generate results.

Therefore, first step is always to simulate the **decision rules**, i.e. **state variables**, which is **K'** (or G in your program, depending on how you name it).

Simulate the Capital decision rule

Just like how we simulate TFP, we need to assume a starting position. Remember, z_0 and K_0 are both given in the model?

Let's assume $K_0 = Kgrid(3)$ in the first period.
We know that decision rule K' or G has two dimensions *G(kgrid, zgrid)*.
Knowing first period K_0, and from previous TFP simulation, we also know the first period z_0; this directly gives us value for second period capital K_2 by letting matlab calling value that you saved from solving the steady state model

$$K(2) = G(3, 5)$$

where 3 represents first period capital grid, and 5 represents first period TFP grid.

Simulate the Capital decision rule

Now we get period 2 capital, we can use the same method, repeat to the 3rd, 4th, and all the rest of the periods, in finding our capital values each date.
One trick, however, is to also find, not only the value of K at each period, but also, the corresponding position of it on the grid.
Because we report value, but also need position for Matlab to find the future value.
In the Sakai resource tab, you will find a function written by me, help you directly find the location of the value on the grid.

$$[position] = gridposition(grid, value)$$

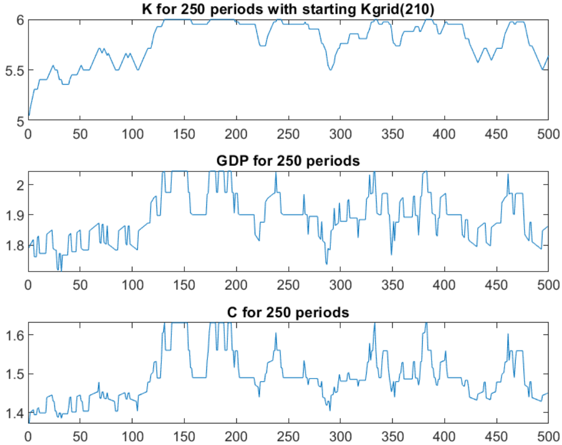
Simulate the rest of the economy

With both Capital and TFP grid values received, we need to build the rest of the economy.

- For each period, we have GDP, following directly from the production function: $Y_t = z_t K_t^{\alpha}$
- Consumption: $C_t = Y_t + (1 - d)K_t = K_{t+1}$

Notice, that you may have to adjust the dimension for C, since you are using K_{t+1} to find C_t. If you have 250 periods of K, then C can only be calculated for the first 249 periods.

Result sample



I use $\rho = 0.859; \sigma_e = 0.014; znum = 5$; and all the other parameters follow uncertainty result parameters, and simulated 500 periods.