About TA sections:

TAs: Ekaterina Zubova (ez268@cornell.edu), Zheyang Zhu (zz792@cornell.edu)

Section time and location: 8:40am - 9:55am Uris Hall 262 (section 201), Goldwin Smith Hall 236 (section 202)

Office hours: Tuesdays 5-7 pm in Uris Hall 451 (Ekaterina), Thursdays 5-7 pm in Uris Hall 429 (Zheyang). Other times available by appointment (just send us an email!)

Our plan for today:[1]

[1]Materials adapted from notes provided by a previous Teaching Assistant, Zhuoheng Xu.

1 Discretizing AR(1): Tauchen's Method

1.1 Idea

The essence of Tauchen's method is to discretize a continuous AR(1) process, which is typically normal, into a finite state space that captures the essence of the continuous process. By doing so, it allows for easier computation of stochastic processes in numerical models, especially in dynamic programming contexts.

Here we want to approximate the following AR(1) process with $\{z_i\}_{i=1}^n$ and $\{p_{ij}\}_{i,j=1}^{n,n}$ where p_{ij} is the transition probability from state z_i to state z_j :

$$z' = (1-\rho)\mu + \rho z + \epsilon' \quad \epsilon' \sim \text{iid} \quad N(0, \sigma_\epsilon^2)$$

1. Set n, which is the number of potential realizations of z.

2. Notice that the stationary distribution of z is $N(\mu, \sigma_z^2)$ where $\sigma_z = \frac{\sigma_\epsilon}{\sqrt{1-\rho^2}}$.

3. Set the upper bound $\bar{z}$ and the lower bound $\underline{z}$ to the support of z. Considering the symmetry of the normal distribution around μ, a natural way to set the bounds is to choose λ such that:

$$\bar{z} = \mu + \lambda\sigma_z$$

$$\underline{z} = \mu - \lambda\sigma_z$$

4. Set $\{z_i\}_{i=1}^n$ such that, $z_1 = \underline{z}$, $z_n = \bar{z}$, and all of $\{z_i\}_{i=1}^n$ are equally distanced. In other words, for $i = 1, 2, \ldots, n$:

$$z_i = \underline{z} + \frac{\bar{z} - \underline{z}}{n-1}(i-1) = \underline{z} + \frac{2\lambda\sigma_z}{n-1}(i-1)$$

5. Construct the midpoints $\{m_i\}_{i=1}^{n-1}$. m_i is constructed as follows:

$$m_i = \frac{z_{i+1} + z_i}{2}$$

6. Let's construct intervals $\{Z_i\}_{i=1}^n$ as follows:

$$Z_1 = (-\infty, m_1]$$

$$Z_i = (m_{i-1}, m_i] \quad i = 2, 3, \ldots, n-1$$

$$Z_n = (m_{n-1}, \infty)$$

7. We will approximate the transition probability p_{ij} as the probability that, conditional on z_i, $z' = (1-\rho)\mu + \rho z_i + \epsilon'$ falls into the interval j. p_{ij} can be easily computed as follows:

$$p_{ij} = \Phi\left(\frac{m_j - (1-\rho)\mu - \rho z_i}{\sigma_\epsilon}\right) - \Phi\left(\frac{m_{j-1} - (1-\rho)\mu - \rho z_i}{\sigma_\epsilon}\right) \quad j = 2, 3, \ldots, n-1$$

$$p_{i1} = \Phi\left(\frac{m_1 - (1-\rho)\mu - \rho z_i}{\sigma_\epsilon}\right)$$

$$p_{in} = 1 - \Phi\left(\frac{m_{n-1} - (1-\rho)\mu - \rho z_i}{\sigma_\epsilon}\right)$$

where $\Phi(.)$ is the CDF of $N(0, 1)$.

1.2 Intuition

The method assumes that the future value of z_{t+1} is normally distributed around the mean given by the AR(1) process, and the variance remains constant. By dividing the possible range of z_{t+1} into discrete intervals and computing transition probabilities, Tauchen's method approximates the likelihood of moving from one state to another.

If currently we are in state i such that $z_t = z_i$, then:

$$E[z_{t+1} \mid z_t = z_i] = (1-\rho)\mu + \rho z_i$$

$$Var[z_{t+1} \mid z_t = z_i] = \sigma_\epsilon^2$$

Therefore,

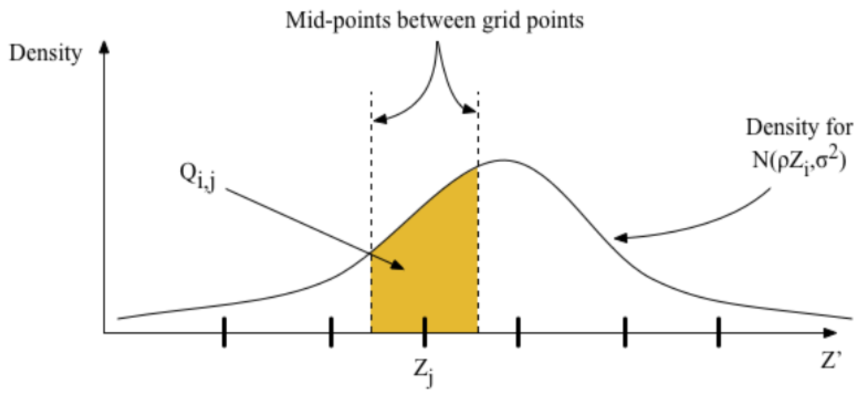
$$z_{t+1} \mid z_t \sim N((1-\rho)\mu + \rho z_i, \sigma_\epsilon^2)$$

And the transition probability $P_{i\to j}$ we want to calculate would be something like:

$$P_{i\to j} = P(z = z_j \mid z_i) \approx P(z \in \text{“}Z_j\text{”} \mid z = z_i)$$

This is basically what the algorithm is doing.

A graphical illustration of Tauchen (By Alisdair McKay):



1.3 Simulate the AR(1) as a Markov Chain

Route 1: With the transition matrix $P_{i\to j}$ in hand, you could use the MATLAB function `simulate` together with `dtmc`[2]. Please read the document yourself.

```
x = simulate(mc,numSteps,'X0',x0)
```

where `"x0"` is the initial state.

Route 2: Start at an initial state. Use a random number generator from the standard uniform distribution, and assign the draw to a destination state. (The code is provided in `section7_MATLAB.m`).

Note: There are built-in functions in MATLAB for variance, mean, and autocorrelation. These are easy to use, but you can also code up the "math formulas" yourself.

[2]https://www.mathworks.com/help/econ/dtmc.simulate.html

2 Value Function Iteration with (z, k) State Space

2.1 Remarks on how to code this

This code builds on Section 5's code, but now we have two state variables: z, k. Therefore, we need a two-dimensional grid to discretize the state space!

1. For z, we have the Tauchen grid already (say, per Pset4, we have 7 grid points).

2. For k, we can choose 100 grid points using `linspace`.

Therefore, we need to find the policy rule $k' = g(z, k)$ for each of the $7 \times 100 = 700$ grid points. And $z' = h(z, k)$ is simply the exogenous AR(1) rule.

Key point: Note that we need to compute the conditional expectation of the value function across all possible states. For example,

$$v(k, z) = \max_{k'} \{U(e^z F(k, 1) + (1-\delta)k - k') + \beta \mathbb{E}[v(k', z') \mid z]\}$$

$$\approx \max_{k'} \left\{ U(e^z F(k, 1) + (1-\delta)k - k') + \beta \sum_{z'} \pi(z' \mid z) v(k', z') \right\}$$

And in MATLAB, we can perform a dot product between two vectors:

```
value_iter(i,k,j)=log(exp(sample_space(k))*k_grid(i)^alpha-k_grid(j)+...
beta*trans(k,:)*squeeze(value(1,j,:)));
```

We here provide you with a code with a different AR(1) process and assume full depreciation. The code is also not optimized (e.g., k' is not in vector form). You need to adapt the code on your own.

```
% Loop to compute value functions iteratively, continue until sup<tol
while sup>=tol
% Rename previous iteration's value function as base value function for
% current iteration
    value(1,:,:)=value(2,:,:);
% Loop over current income shock values in sample_space
    for k=1:size_space
```

```
% Loop over current capital values in k_grid
        for i=1:grid_size
% Loop over next-period capital values in k_grid
            for j=1:grid_size
% Check if feasibility is satisfied, capital strictly
% positive (Inada conditions assumed)
                if 0<k_grid(j) && k_grid(j)<=exp(sample_space(k))*k_grid(i)^alpha
% Calculate value of objective with current-period capital
% k_grid(i), next-period capital k_grid(j), and
% current-period shock sample_space(k)
                   value_iter(i,k,j)=log(exp(sample_space(k))*k_grid(i)^alpha...
                    -k_grid(j))+beta*trans(k,:)*squeeze(value(1,j,:));
% Set value to -Inf if feasibility violated
                else
                    value_iter(i,k,j)=-Inf;
                end
            end
% Assign value(2,i,k) as maximum of value_iter(i,k,j) over j
            value(2,i,k)=max(value_iter(i,k,:));
        end
    end
% Determine sup difference between value(2,i,k) and
% value(1,i,k)
    sup=max(max(abs(value(2,:,:)-value(1,:,:))));
end
```

3 Practice Question: Multiple Endogenous States

Consider an economy with a single good, cookies, produced using capital. Cookies can be eaten as consumption, or transformed into capital goods or durable goods. The representative agent has lifetime utility given by

$$\sum_{t=0}^{\infty} \beta^t u(c_t^n, x_t)$$

where $0 < \beta < 1$. Here c_t^n is the agent's consumption of cookies, x_t is the agent's consumption of the service flow from durable goods. Here u has standard properties.

The resource constraint is

$$c_t^n + c_t^d + i_t = y_t$$

where c_t^d represents investment in durables goods (durables purchases), i_t represents investment, and y_t is the output of cookies. The capital stock and the stock of durables evolve according to

$$k_{t+1} = (1-\delta)k_t + i_t$$

$$d_{t+1} = (1-\psi)d_t + c_t^d$$

Here $0 < \delta < 1$ and $0 < \psi < 1$ are depreciation rates of capital and durables, respectively.

Output and service from the stock of durable is given by

$$y_t = f(k_t)$$

$$x_t = \phi(d_t)$$

Here f and ϕ have standard properties.

1. Write down the social planner's problem for this economy as a Bellman equation.
2. Derive the first-order conditions and the envelope conditions of the social planner.
3. Derive and interpret the Euler equation for durable goods.