LNU
Spring 2017

Introduction to Dynamic Economic Models
Instructor: Mykola Babiak
Date: April 28, 2017

Exercise Session 5

1. Consider the stochastic growth model where the social planner chooses sequences of consumption and next-period capital stock, $\{c_t, k_{t+1}\}_{t=0}^{\infty}$, to maximize an objective function

$$V(k_0) = \mathbb{E}_0 \left[\sum_{t=0}^{\infty} \beta^t \ln c_t \right]$$

subject to the constraints:

$$c_t + k_{t+1} \leq (1 + z_t) f(k_t) + (1 - \delta) k_t,$$
$$c_t > 0, \; k_t > 0, \; k_0 \quad \text{is given,}$$

where z_t is a Markov chain which takes only two values $z_1 = z_L$ and $z_2 = z_H > z_L > 0$ and the transition matrixis P with $P_{11} = q; P_{22} = p$ and $p = q$. Further, $k_0 > 0, f(k_t) = Ak_t^{\alpha}$ and $\beta \in (0,1), \alpha \in (0,1), \delta \in (0,1), A > 0$.

   (a) Formulate the problem recursively in such a way that next-period capital k_{t+1} is being chosen. What are the control and state variables in this problem?

   (b) Derive the FOC and the ET condition. Substitute the ET condition into the FOC in order to get rid of the derivative of the value function, the obtained equation is called the Euler equation.

   (c) Using the Euler equation prove that the deterministic steady state value of capital is given by the following formula

$$k_{ss} = \left[\frac{\alpha\beta(1 + z_{ss})A}{1 - \beta + \beta\delta} \right]^{\frac{1}{1-\alpha}}$$

   where $1 + z_{ss}$ is a steady state value of technology.

2. Use the algorithm for the deterministic case in Appendix and modify it to produce the code of the program1 in MATLAB for solving the stochastic growth model above by the use of the value function iteration method.

Assume the following:

- 2 states $[z_1 = z_L, z_2 = z_H]$ for productivity level
- 2 $(n \times 1)$ vectors v_j, $j = 1, 2$

$$v_j(i) = v(a_i, z_j) \qquad i = 1, ..., n$$

- 2 $(n \times n)$ matrices R_j, $j = 1, 2$

$$R_j(i,h) = u\left[(1+z_j) A k_i^{\alpha} + (1-\delta) k_i - k_h^{+}\right] \quad i = 1, ..., n; \; h = 1, ..., n$$

- operator $T : [v_1, v_2] \rightarrow [Tv_1, Tv_2]$

$$\begin{aligned} Tv_1 &= \max\{R_1 + \beta P_{11} 1_n v_1' + \beta P_{12} 1_n v_2'\} \\ Tv_2 &= \max\{R_2 + \beta P_{21} 1_n v_1' + \beta P_{22} 1_n v_2'\} \end{aligned}$$

or

$$\begin{bmatrix} Tv_1 \\ Tv_2 \end{bmatrix} = \max\left\{\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} + \beta (P \otimes 1_n) \begin{bmatrix} v_1' \\ v_2' \end{bmatrix}\right\}$$

3. Assign the values of model parameters: $A = 1$, $\alpha = 0.35$, $\beta = 0.55$, $\delta = 1$, and $\sigma = 2$, choose a relative error tolerance level, $\varepsilon = 10^{-6}$, and the maximum number of iterations (maxiter) to 300 - for the case when the method does not converge.

4. In order to approximate the $AR(1)$ process $z_{t+1} = \rho z_t + \sigma_\varepsilon \varepsilon_{t+1}$ with $\varepsilon_{t+1} \sim N(0,1)$ with $\rho = 0.95$ and $\sigma_\varepsilon = 0.007$, use the following formulas

$$\begin{aligned} \frac{1}{2}(z_H - z_L) &= 0, \\ \sigma_\varepsilon^2 &= p(1-p)(z_H - z_L)^2, \\ p &= \frac{1+\rho}{2}. \end{aligned}$$

5. (Bonus Question) Proof that the Markov chain given by the parameters (z_L, z_H, p) approximates the AR(1) with the autocorrelation parameter ρ and the shock innovation variance σ_ε^2.

1 Appendix

Value function iteration algorithm

- Step 0. Clear the internal memory in MATLAB and start the stopwatch time

```
clear all
tic
```

- Step 1. Assign the values of model parameters: $A = 1$, $\alpha = 0.35$, $\beta = 0.55$, $\delta = 1$, and $\sigma = 2$, choose a relative error tolerance level, $\varepsilon = 10^{-6}$, and the maximum number of iterations (maxiter) to 300 - for the case when the method does not converge.
- Step 2. Compute the steady state value of capital k_{ss} as a function of model parameters: use the formula (3) you had derived above.
- Step 3. Discretize the state space by constructing a grid for capital: $K = [k_1, k_2, ..., k_n]$ on the interval between 0 and $5k_{ss}$ where $n = 1000$. In MATLAB use the function linspace

$$K = \mathsf{linspace}\,(0, 5k_{ss}, n)\,;$$

- Step 4. Set up return matrix $\mathbf{R}$ and consumption matrix $\mathbf{C}$ which are both of size $(n \times n)$:
  - first, do their declaration in MATLAB:

$$\begin{aligned} \mathbf{R} &= \mathsf{zeros}\,(n, n)\,; \\ \mathbf{C} &= \mathsf{zeros}\,(n, n)\,; \end{aligned}$$

  - then, create two loops through which you will compute the consumption and the utility for all possible combinations of the current values of capital, k, and its next period values, k^{+}:

$$\begin{aligned} \mathsf{for}\ i &= 1 : n, \\ k_i &= K\,(i)\,; \end{aligned}$$

$$\begin{aligned} \text{for } h &= 1:n, \\ k_h^+ &= K(h); \\ c_{ih} &= Ak_i^\alpha + (1-\delta)k_i - k_h^+; \\ \mathbf{C}(i,h) &= \max(0, c_{ih}); \\ & \text{end } \% \text{ end of loop h} \\ & \text{end } \% \text{ end of loop i} \end{aligned}$$

What is the role of the 6^{th} line above? Finally[2], compute the return matrix:

$$\mathbf{R} = 1/(1-\sigma)\,\mathbf{C}^{1-\sigma};$$

- Step 5. Value function iteration loop:
  - Initialize the counter of iterations and set and initial error at a high value:

$$\begin{aligned} \text{iter} &= 0; \\ \text{error} &= 100; \end{aligned}$$

  - Set initial value function V^0 equal to zero, i.e. on the grid it will be a vector of size $(n \times 1)$

$$\begin{aligned} V^0 &= \text{zeros}(n,1); \\ V &= V^0; \end{aligned}$$

  - Create a while loop for iterating value function with the continuation rule that error is larger than ε and the number of iterations is smaller than maximum of number of iterations:

$$\text{while } (\text{error} > \varepsilon)\,\&\,(\text{iter} < \text{maxiter}), \quad \% \text{ continuation conditions}$$

[2] By $\mathbf{C}^{1-\sigma}$ we mean here

$$\begin{pmatrix} c_{11}^{1-\sigma} & \dots & c_{1n}^{1-\sigma} \\ \dots & \dots & \dots \\ c_{n1}^{1-\sigma} & & c_{nn}^{1-\sigma} \end{pmatrix}$$

which can be performed in MATLAB by element-by-element power operation, '.^', i.e.

$$C.\hat{}\,(1-\text{sigma}).$$

– Compute the maximization step of the value function iteration, i.e. given current state k_i find the maximal value of $TV(i)$ across all possible next period capitals, i.e. the sum of the i-th row of the return matrix and the discounted row of the next-period value function

$$TV(i) = \max_{\text{across row}} \left(\mathbf{R}(i,\cdot) + \beta V'(\cdot)\right).$$

In order to perform this in one step for all values of k_i's at once, use the identity vector $\mathbf{1}_n = [1, 1, ..., 1]'$ of size $(n \times 1)$. Then

$$TV = \max_{\text{across rows}} \left(\mathbf{R} + \beta \ \cdot \mathbf{1}_n \ \cdot V'\right)$$

In MATLAB[3] it is

$$TV = \max\left(\left(\mathbf{R} + \beta * \text{ones}(n,1) * V'\right)'\right); \qquad \text{\% Bellman operation}$$

– the LHS is a row vector so

$$TV = TV';$$

– the difference between two successive iterations of value function is computed by the use of the maximum norm, $\|X\| = \max(|X_1|, |X_2|, ..., |X_n|)$, i.e. in MATLAB

$$\text{error} = \max(\text{abs}(V - TV)); \qquad \text{\% max difference between two successive steps}$$

– further update the value function, increase the counter of iterations by one and finish the while loop

$$\begin{aligned} V &= TV; && \text{\% update of value function} \\ \text{iter} &= \text{iter} + 1; && \text{\% update of the iteration counter} \\ & \text{end} && \text{\% end of loop} \end{aligned}$$

- Step 6. Read the stopwatch timer

$$\text{time} = \text{toc};$$

[3]The argument on the righ-hand side needs to be transposed since function max in MATLAB looks for maxima across columns.

- Step 7. Check whether the program stopped due to a proper convergence and report the time elapsed

  if (error $< \varepsilon$) & (iter $<$ maxiter)

  display (['The program successfully converged to the fixed point'])

  display (['Time taken = ', num2str (time) , ' seconds'])

  end

- Step 9. Compute policy function $k^{+} = k^{+}(k)$ - noted KP in MATLAB - by repeating the last Bellman step and using the function max which returns the indices of the maximum values in vector TI

$$\begin{aligned} [TV, TI] &= \max\left((\mathbf{R} + \beta * \text{ones}(n,1) * V')'\right); \\ KP &= K(TI()); \end{aligned}$$

- Step 9. Plot the graph of the obtained value function $V(k)$. Plot the graph of policy function $k^{+}(k)$ and a 45-degree line. What is the meaning of the cross-section of policy function with 45-degree line?

3. Modify the code to get a numerical solution for the following parameter values: $\beta = 0.95$, $\alpha = 0.5$, $A = 5$, alternatively $\delta = 1$, 0.5, and 0, and $\sigma = 1$, and 2. Print the plots. Report and compare the changing number of iterations and elapsed times for alternative parameter values.

4. For the case 3. use different grids for the capital stock of size 200, 50, and 20, respectively, around the steady state and compare the results above. Print the plots.

5. Modify the code you have created using the algorithm above in order to compute the return matrix using a compact matrix formulation instead of using two for loops :

$$\begin{aligned} \mathbf{C} &= (AK^{\alpha} + (1-\delta)K) \cdot \mathbf{1}'_n - \mathbf{1}_n \cdot K'; \\ \mathbf{C} &= \max(\mathbf{0}_n, \mathbf{C}) \\ \mathbf{R} &= 1/(1-\sigma)\,\mathbf{C}^{1-\sigma} \end{aligned}$$

where $\mathbf{0}_n$ is $(n \times n)$ zero matrix[4]. Solve the model at original parameter values and compare the times needed to solve the model with more compact matrix formulation and without it. Which one seems to be faster?

[4] In MATLAB you use the command zeros(n, n) to create it.