MATLAB: Session 1

Brian Higgins   Ciaran Rogers

Outline

1. Recap functions from Preliminary Problem Set
2. Solve Endowment Economy Numerically
3. Introduce Finite Horizon Dynamic Programming

Example 1: Fibonacci

$$f(1) = 1; f(2) = 1$$
$$f(n) = f(n-1) + f(n-2)$$

Write code that returns the nth element

1. Recursion
2. Loop
3. Comparison of Performance
   - Write code that returns first n elements (in loop and recursion)
   - Use tic/toc to time recursion and loop functions; plot the differences up to n=30
   - Question: Why is one faster than the other?

Example 2: Numerical Integration/Pareto Distribution

$$F_X(x) = \begin{bmatrix} 1 - \frac{x_m}{x}^{\alpha} & if & x \geq x_m \\ 0 & & ifx < x_m \end{bmatrix}$$

$$f_X(x) = \begin{bmatrix} \frac{\alpha x_m^{\alpha}}{x^{\alpha+1}} & if & x \geq x_m \\ 0 & & ifx < x_m \end{bmatrix}$$

EXERCISES

1. Write a function for the Pareto PDF ($\alpha = 2$; $x_m = 1$)
2. Write a function for the Pareto CDF
3. Plot the Pareto PDF
4. Define a grid x that covers (3,4), with step size d
5. Evaluate Pareto density on x, store as fx
6. Using the grid from 4., and fx, figure out a way to approximate the likelihood of the Pareto distributed random variable to lie within [34]

Market Equilibrium Without FOCs

Core Problem: Perfectly Competitive Exchange Economy

- Two consumers: A,B
- Two goods: x_1, x_2
- Utility: $[x_1^r + x_2^r]^{\frac{1}{r}}$
- Endowments: $\{(y_1^A, y_2^A), (y_1^B, y_2^B)\} = \{(10, 5), (5, 40)\}$

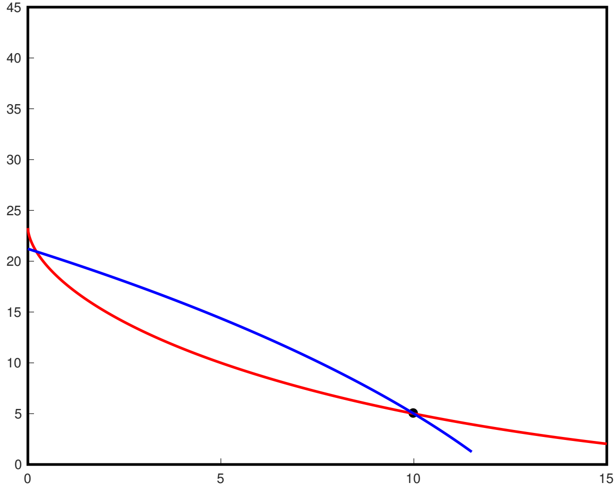
GOAL: Find equilibrium prices and allocation

TOOLS: Dynamic Programming

Outline of Process

1. Construct Edgeworth Box
2. Draw Indifference Curves at endowment
3. Calculate individual demands given prices
4. Draw individual demand points on Edgeworth Box
5. Calculate Total Demand (fixed supply) given prices
6. Find equilibrium prices and allocations

Step 1: Draw Edgeworth Box



Step 2: Construct Indifference Curves

- cesutility.m
- cesindiff.m
- Make two new functions with r as given, not as argument
- Find utility of Agent A at endowment point
- Draw indifference curve through the endowment point

Practice 1: Plot indifference curve for B

HINT: What does a point in the box represent?

Practice 2: Plot budget constraint of Agent A, assuming $p_1 = 1$ and $p_2 = 0.5$

Step 3: Derive Individual Demands

$$\max_{x_1,x_2} \left[x_1^r + x_2^r\right]^{\frac{1}{r}}$$

$$s.t. x_1 p_1 + x_2 p_2 \leq y_1 p_1 + y_2 p_2 \equiv m$$

Useful matlab function: fmincon

fmincon(fun,x0,A,b)

- What is the objective?
- What is A? What is b?
- What is a good guess?

Step 4: Demand in Box

Practice 1: Make a new box, with the endowment point, the budget constraint, and two demand points

Practice 2: Add indifference curve at each of individual demand points.

Step 5: Total Demand

Construct a function that calculates total demand for both goods.

Practice 1: Draw the demand curve on a graph

Practice 2: Add the supply curve to the graph

Step 6: Find Equilibrium Prices and Allocation

Step 6: Find Equilibrium Prices and Allocation

Practice 1: Draw equilibrium point and the Indifference Curves

Practice 2: Run code for $r = 0.2$ and $r = 0.9$. What is the difference? What is the intuition for it?

Introduction to Dynamic Programming

- Very general and useful method in economics
- Heavily featured in Labour, IO, Macro, Finance, etc
- In programming camp - informal introduction through 3 examples:
  - Cake-eating problem
  - Patent Renewal
  - PCF Championship

Note: No proofs, only methods!

Finite Horizon Problem

Consider the following general problem:

$$V(y_0, 0) = \max_{\{x_t\}_{t=0}^{T-1}} \sum_{t=0}^{T-1} \beta^t u(x_t, y_t)$$

s.t. $y_{t+1} = f(x_t, y_t)$

where y_t is the state variable and x_t is the control variable.

Solution Method: Backward Induction

Backward Induction

$$
\begin{aligned}
V(y_0, 0) &= \max_{\{x_t\}_{t=0}^{T-1}} \sum_{t=0}^{T-1} \beta^t u(x_t, y_t) \\
&= \max_{\{x_t\}_{t=0}^{T-1}} u(x_0, y_0) + \beta \sum_{t=0}^{T-2} \beta^t u(x_{t+1}, y_{t+1}) \\
&= \max_{\{x_t\}_{t=0}^{T-1}} u(x_0, y_0) + \beta V(y_1, 1)
\end{aligned}
$$

There, we can solve the problem, if and only if we knew $V(., 1)$!

BUT, we know that $V(., T) = 0$!! $\rightarrow$ solve for $V(., T-1)$...

Cake-Eating Problem

$$V(y_0) = \max_{\{x_t\}_{t=0}^{T}} \sum_{t=0}^{T} \beta^t \log(x_t)$$

subject to

- $y_{t+1} = y_t - x_t$
- $x_t = 0;\ y_{t+1} = 0$
- $y_0 = A$

Backward Induction: Solution Method

Steps

1. Set up V-grid that defines the value function over T states
2. Define state variable grid on which you will evaluate $V(., t)$
3. Set up the loop over backward induction
    - at each t
        - Construct a spline function for $V(., t+1)$
        - Set up grid of potential x_t choices (and satisfies constraints !!)
        - Find optimal choices, saving it as well as the value function