Macro II

Professor Griffy

UAlbany

Spring 2024

Announcements

- Today: continue solutions methods: value function iteration.
- Using:
  1. Grid search;
  2. Interpolation (grid search with functions filling in between nodes).
- Go through examples with neoclassical growth model.
- Homework assignment: do same with RBC model (HW5).
- Midterm grades by end of the week (my bad).

Solving a Model

- When we say "solve a model" what do we mean?
  1. Find the equilibrium of the model.
  2. Generally, determine the policy functions.
  3. Determine the transition equations given the individual and aggregate state.
  4. i.e., aggregate up the policy functions and determine prices given distributions.
- Generically, this is hard: many states, non-linear decision rules, etc.

Solving a Model

- Generically, this is hard: many states, non-linear decision rules, etc.
- Much of quantitative macro is about finding "shortcuts" without sacrificing accuracy of solution (some we have seen):
    1. Planner's problem: use welfare theorems to remove prices from problem.
    2. Rational expectations & complete markets: Aggregate worker decision rules by assuming they make same predictions about future prices, and face same consumption risk.
    3. Exogenous wage distribution/prices: agents do not respond to decisions of other agents.
    4. Block Recursive Equilibrium: agents face an equilibrium with individual prices, i.e., no need to know distribution.
- Linearization: assume the economy is close enough to steady-state that transition equations (i.e., policy functions) are close to linear within small deviations.
- Value function iteration: discretize state space and solve model at "nodes" in state space.

Neoclassical Growth Model

- Problem:

$$V(k) = \max_{k'} u(c) + \beta V(k') \tag{1}$$

$$c + k' = F(k) + (1-\delta)k \tag{2}$$

- Assume power utility: $u(c) = \frac{c^{1-\sigma}-1}{1-\sigma}$
- Cobb-Douglas Production: $F(k) = k^{\alpha}$

Value Function Iteration

- Basic approach to value function iteration:
  1. Create grid of points for each dimension in state-space.
  2. Specify terminal condition V_t for $t = T$ at each point in state-space.
  3. Solve problem of agent in period $T - 1$: $V_t(y) = max_x u(c(x)) + \beta E[V_{t+1}(x)]$.
  4. x is policy function, which yields the largest value from $\{x_1, ..., x_N\}$, where N is the number of grid points.
  5. Check to see if function has converged, i.e., $|V_t - V_{t+1}| < errtol$
  6. Update $V_{t+1} = V_t$
- Interpolation: same idea, but functions used to fill in between grid points.

Parameter Values

- Before we can solve the model (or write down grids) we need parameter values.
- Pick reasonable ones from the literature:
  - $\alpha = 0.3$ (roughly capital share)
  - $\sigma = 2$ (standard risk aversion)
  - $\delta = 0.1$ (annual depreciation 10%)
  - $\beta = 0.96$ (annual interest rate $\approx$ 4.2%)
- If we were estimating this model: we would evaluate the performance of the model given these parameters.
- i.e., how does it fit the data if we use this set of parameters.

Grids

- Want: smallest grids reasonable.
- Find k^*, pick grids around this.
- Euler Equation

$$u'(c) = \beta[\alpha k^{\alpha-1} + (1-\delta)]u'(c') \tag{3}$$

- In steady-state, $c = c' = c^*$

$$\rightarrow u'(c^*) = \beta[\alpha k^{*\alpha-1} + (1-\delta)]u'(c^*) \tag{4}$$

$$1 = \beta[\alpha k^{*\alpha-1} + (1-\delta)] \tag{5}$$

$$(\frac{1}{\alpha\beta} - \frac{1-\delta}{\alpha})^{\frac{1}{\alpha-1}} = k^* \tag{6}$$

$$\tag{7}$$

- For our parameter values, $k^* = 2.92$.
- Pick grids st $k, k' \in [0.66 \times k^*, 1.5 \times k^*]$
- Arbitrary, probably larger than needed.

Neoclassical Growth Model

- Problem:

$$V(k) = \max_{k'} u(c) + \beta V(k') \tag{8}$$

$$c + k' = F(k) + (1-\delta)k \tag{9}$$

- Assume power utility: $u(c) = \frac{c^{1-\sigma}-1}{1-\sigma}$
- Cobb-Douglas Production: $F(k) = k^{\alpha}$
- $k, k' \in \{k_1, ..., k_N\}$
- $V_0 =$? Safest bet to set it to zero at all k.

Value Function First Iteration

- Intuitively, take as given capital today ($\bar{k}$), choose capital in the future that maximizes value.
- Problem:

$$V(\bar{k}) = \max_{k' \in k_1, \dots, k_N} u(c) + \beta \cancel{V(k')} \tag{10}$$

$$c + k' = F(\bar{k}) + (1 - \delta)\bar{k} \tag{11}$$

- That is, policy function is k_i where i is the index of the optimal policy from the following:

$$u(F(\bar{k}) + (1 - \delta)\bar{k} - k_1) + \beta \times 0 \tag{12}$$

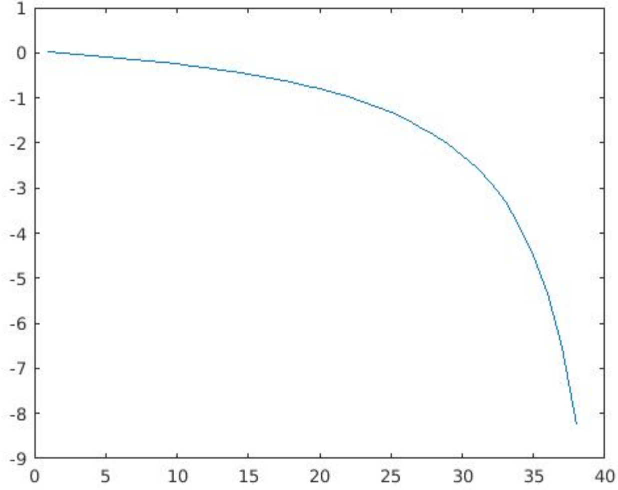
$$u(F(\bar{k}) + (1 - \delta)\bar{k} - k_2) + \beta \times 0 \tag{13}$$

$$\dots \tag{14}$$

$$u(F(\bar{k}) + (1 - \delta)\bar{k} - k_N) + \beta \times 0 \tag{15}$$

Value Function First Iteration

- Value of $V_{t+1}(k')$ given $k = \bar{k}$ (x-axis is num. of grid pts.):



- What is optimal choice?

Value Function First Iteration

- Now, check if problem has converged.
- What does this mean?
- The value in the current state is not changing over time.
- i.e., $V_t(k) \approx V_{t+1}(k)$.
- First iteration: it won't be.
- What do we do now?
- Update the continuation value:
- $V_{t+1} = V_t$ for all k
- Solve same problem again.

Value Function Second Iteration

- Solved for $V(k')$ in previous iteration.
- Again, faced with maximization problem given capital $\bar{k}$ today:

$$V(\bar{k}) = \max_{k' \in k_1, \ldots, k_N} u(c) + \beta V(k') \tag{16}$$

$$c + k' = F(\bar{k}) + (1-\delta)\bar{k} \tag{17}$$

- Note that the continuation value is **not** zero!

$$u(F(\bar{k}) + (1-\delta)\bar{k} - k_1) + \beta V(k_1) \tag{18}$$

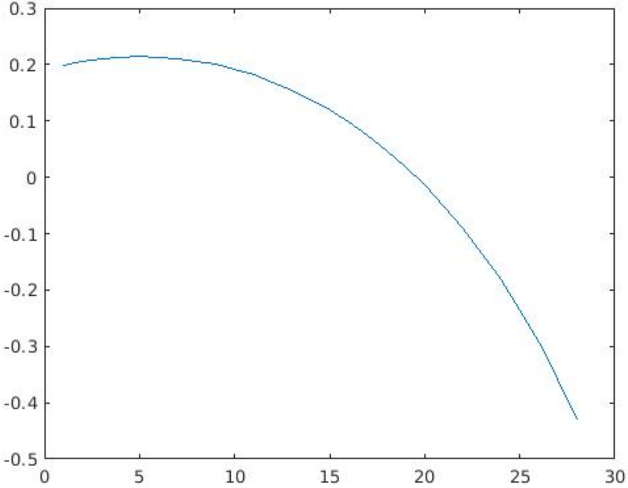
$$u(F(\bar{k}) + (1-\delta)\bar{k} - k_2) + \beta V(k_2) \tag{19}$$

$$\ldots \tag{20}$$

$$u(F(\bar{k}) + (1-\delta)\bar{k} - k_N) + \beta V(k_N) \tag{21}$$

Value Function Second Iteration

- Value of $V_{t+1}(k')$ given $k = \bar{k}$ (x-axis is num. of grid pts.):



- What is optimal choice?

Value Function Second Iteration

- We check again to see if it has converged.
- is $V_t(k) \approx V_{t+1}(k)$.
- What do we do now?
- Update the continuation value:
- $V_{t+1} = V_t$ for all k
- Solve same problem again.
- Keep doing this until the difference is very small.

Great, we're done!

- Not so fast: this isn't very accurate.
- Very slow if we have large numbers of states & grid points (scales exponentially).

Fundamental Problem

- The reason we need to use a computer to solve this problem is that we *don't know* the function $V(k)$.

$$V(k) = \max_{k'} u(c) + \beta V(k') \tag{22}$$

$$c + k' = F(k) + (1-\delta)k \tag{23}$$

- What is we *approximate* $V(k)$ with other functions?
- Some useful properties we can pick these functions to have:
  - Continuous
  - Differentiable
- If our approximation is accurate enough, we can drop some grid points!

Interpolation

- Again, take capital today as given $k = \bar{k}$. Grid search:

$$V(\bar{k}) = \max_{k' \in k_1, \ldots, k_N} u(c) + \beta V(k') \tag{24}$$

$$c + k' = F(\bar{k}) + (1-\delta)\bar{k} \tag{25}$$

- Optimal policy is the index largest of:

$$u(F(\bar{k}) + (1-\delta)\bar{k} - k_1) + \beta V(k_1) \tag{26}$$

$$\ldots \tag{27}$$

$$u(F(\bar{k}) + (1-\delta)\bar{k} - k_N) + \beta V(k_N) \tag{28}$$

- Call interpolated function $\hat{V}(k)$. Then,

$$V(\bar{k}) = \max_{k'} u(c) + \beta \hat{V}(k') \tag{29}$$

$$c + k' = F(\bar{k}) + (1-\delta)\bar{k} \tag{30}$$

- Where k' solves

$$u'(F'(\bar{k}) + (1-\delta)\bar{k} - k') = \beta \frac{\partial \hat{V}}{\partial k'} \tag{31}$$

Updating

- We do *exactly the same thing as before*:

$$V(\bar{k}) = u(c(k^{'*})) + \beta V(k^{'*}) \tag{32}$$

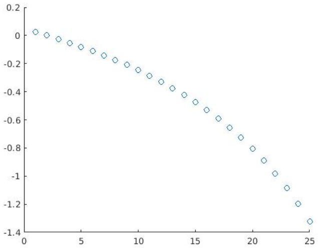
- For each $\bar{k}$. Then, we check the convergence criteria:

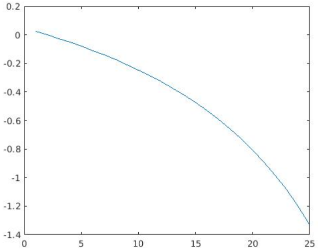
$$|V_t - V_{t+1}| < errtol \tag{33}$$

- How do we create the function $\hat{V}(k)$?
- "Connect the dots" of $V_t(k)$ between all capital levels in order.

Interpolation

- Left is value function for grid search. Right is for (linearly) interpolated function:



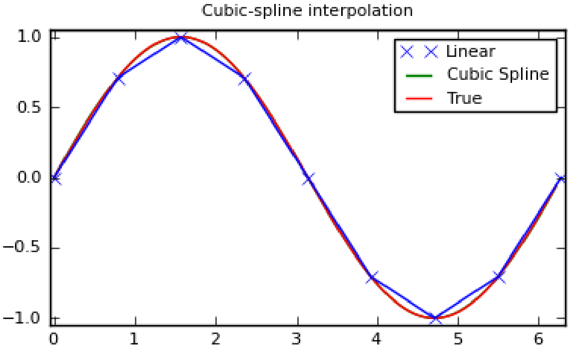


Interpolation

- In constructing our function $\hat{V}(k)$, we need to choose an interpolation scheme.
- Roughly, what *order* function do we believe will be accurate enough to mimick the value function:
  - First-order (linear)
  - Third-order (cubic)
  - Fifth-order (quintic)
- Some other useful interpolation routines:
  - PCHIP (piecewise cubic hermite interpolating polynomial): shape-preserving (not "wiggly") continuous 3rd order spline with continuous first derivative.

Interpolation

- Choice DOES matter:



Polynomial Interpolation

- Suppose we have a function $y = f(x)$ for which we know the values of y at $\{x_1, ..., x_n\}$.
- Then, the nth-order polynomial approximation to this function f is given by

$$f(x) \approx P_n(x) = a_n x^n + a_{n-1} x^{n-1} + ... + a_1 x + a_0 \tag{34}$$

- Then, we have a linear system with n coefficients.
- We could write this as $y = X\beta$. Look familiar?

Polynomial Interpolation

- We solve

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ \vdots \\ y_n \end{bmatrix} \tag{35}$$

- For $a_0, ..., a_n$
- What's the example we are all familiar with? Linear regression: $y = \alpha + X\beta$.
- In practice, this is computationally expensive, but this is the intuition.

Great, we're done!

- Not so fast: how do we handle expected values?
- Depends on expectation.
- Need an accurate way to perform numerical integration.

Stochastic Neoclassical Growth Model

- Problem:

$$V(z,k) = \max_{k'} u(c) + \beta E[V(z',k')] \tag{36}$$
$$c + k' = e^z F(k) + (1-\delta)k \tag{37}$$
$$z' = \rho z + \epsilon \tag{38}$$
$$\epsilon \sim N(0, \sigma_\epsilon) \tag{39}$$

- Assume power utility: $u(c) = \frac{c^{1-\sigma}-1}{1-\sigma}$
- Cobb-Douglas Production: $F(k) = k^\alpha$
- Make sure your process for z stays non-negative.

Expectations with AR(1) Process

- Approximate a continuous AR(1) process with a markov process:
- Create grid of potential z values $\{z_1, ..., z_N\}$, approximate AR(1) process through transition probabilities.

$$E[z_t] = E[\rho z_{t-1} + \epsilon_t] = 0 \tag{40}$$

$$V[z_t] = V[\rho z_{t-1} + \epsilon_t] = \rho^2 \sigma_z^2 + \sigma_\epsilon^2 \tag{41}$$

$$\rightarrow (1 - \rho^2)\sigma_z^2 = \sigma_\epsilon^2 \tag{42}$$

- Define this process $G(\bar{\epsilon})$
- Tauchen (1986):

$$z_N = m(\frac{\sigma_\epsilon^2}{1 - \rho^2}) \tag{43}$$

$$z_1 = -z_N \tag{44}$$

$$z_2, ..., z_{N-1} \text{ equidistant} \tag{45}$$

Expectations with AR(1) Process

- Tauchen (1986):

$$z_N = m(\frac{\sigma_\epsilon^2}{1-\rho^2}) \tag{46}$$

$$z_1 = -z_N \tag{47}$$

$$z_2, ..., z_{N-1} \text{ equidistant} \tag{48}$$

- Create an *nxn* transition matrix Π using probabilities

$$\pi_{ij} = G(z_j + d/2 - \rho z_i) - G(z_j - d/2 - \rho z_i) \tag{49}$$

$$\pi_{i1} = G(z_1 + d/2 - \rho z_i) \tag{50}$$

$$\pi_{iN} = 1 - G(z_N + d/2 - \rho z_i) \tag{51}$$

Expectations Generally

- Expected values also need to be calculated carefully.
- Continuation value from before:

$$E[V(z, k')] \tag{52}$$

- If *not* an AR(1)/markov process, need to approximate integral.
- Generically, pick function f and weights w_i

$$E[V(z, k')] = \int_a^b f(x)dx \approx \sum_{i=1}^{N} w_i f(x_i) \tag{53}$$

- x_i may be known or picked optimally.
- We will return to this in the future.

Next Time

- Calibration and RBC extensions.