Solving HJBE with Finite Differences

Jesse Perla
UBC

November 21, 2017

Here, we expand on details for how to discretize HJBE with a control of the drift.[1] In particular, this set of notes will focus on solving the neoclassical growth model first with a deterministic and then a stochastic TFP.

1 Neoclassical Growth

This solves the simple deterministic neoclassical growth model.

1.1 Value Function with Capital Reversibility

Take a standard neoclassical growth model with capital k, consumption c, production $f(k)$, utility $u(c)$, depreciation rate δ, and discount rate ρ.[2] The law of motion for capital in this setup is,

$$\boldsymbol{\partial}_t k(t) = f(k) - \delta k - c \tag{1}$$

With this, the standard HJBE for the value of capital k is,

$$\rho v(k) = \max_c \left\{u(c) + (f(k) - \delta k - c)\, v'(k)\right\} \tag{2}$$

Assume an interior c and envelope conditions, then taking the FOC of the choice is,

$$u'(c) = v'(k) \tag{3}$$

Assume a functional form $u(c) = \frac{c^{1-\gamma}}{1-\gamma}$ for the utility and $f(k) = Ak^{\alpha}$ for production. Then the FOC can be inverted such that

$$c = \left(v'(k)\right)^{-\frac{1}{\gamma}} \tag{4}$$

And,

$$u(c) = \frac{\left(v'(k)\right)^{-\frac{1-\gamma}{\gamma}}}{1-\gamma} \tag{5}$$

[1]See `operator_discretization_finite_differences.pdf` for more details on the discretization of a linear diffusion operator with finite-differences, and general notation (with equation numbers in that document prefaced by *FD*). Thanks to Sari Wang for superb Research Assistance.

[2] This builds on `http://www.princeton.edu/~moll/HACTproject/HACT_Additional_Codes.pdf` Section 2.2, with code in `http://www.princeton.edu/~moll/HACTproject/HJB_NGM_implicit.m`

If (4) and (5) were substituted back into (2), we would have a nonlinear ODE in just k. Define the following, which will be important in the finite difference scheme

$$\mu(k) \equiv f(k) - \delta k - c \tag{6}$$

Then with (4) this can be defined as a function of the $v(\cdot)$ function,

$$\mu(k; v) \equiv f(k) - \delta k - \left(v'(k)\right)^{-\frac{1}{\gamma}} \tag{7}$$

Then with (2), (5) and (7)

$$\rho v(k) = \frac{\left(v'(k)\right)^{-\frac{1-\gamma}{\gamma}}}{1-\gamma} + \mu(k; v) v'(k) \tag{8}$$

The HJBE (8) is a nonlinear ODE in $v(k)$.

Steady State The goal is for the finite-difference scheme to find the steady-state on its own. However, in this example we have an equation to verify the solution:

$$k^* = \left(\frac{\alpha A}{\rho + \delta}\right)^{\frac{1}{1-\alpha}} \tag{9}$$

$$c^* = A\left(k^*\right)^{\alpha} - \delta\left(k^*\right)^{\alpha} \tag{10}$$

1.2 Finite Differences and Discretization

In order to solve this problem, we will need to use the appropriate "upwind" direction for the finite differences given the sign of the drift in (7).

Two basic approaches:

- Fix the nonlinear $v'(k)$ function so that the operator becomes linear, solve it as a sparse linear system, and then iterate on the $v(k)$ solution.
- Solve it directly as a nonlinear system of equations.

Setup

- Define a uniform grid with I discrete points $\{k_i\}_{i=1}^I$, for some small $k_1 \in (0, k^*)$ and large $k_I > k^*$, with distance between grid points $\Delta \equiv k_i - k_{i-1}$ for all i. After discretizing, we will denote the grid with the variable name, i.e. $k \equiv \{k_i\}_{i=1}^I$. Further define the notations $\underline{k} \equiv k_1$, $\overline{k} \equiv k_I$ and $v_i \equiv v(k_i)$ for simplicity.
- When we discretize a function, use the function name without arguments to denote the vector. i.e. $v(k)$ discretized on a grid $\{k_i\}_{i=1}^I$ is $v \equiv \{v(k_i)\}_{i=1}^I \in \mathbb{R}^I$.
- When referring to a variable μ, define the notation $\mu^- \equiv \min\{\mu, 0\}$ and $\mu^+ \equiv \max\{\mu, 0\}$. This can apply to vectors as well. For example, $\mu_i^- = \mu_i$ if $\mu_i < 0$ and 0 if $\mu_i > 0$, and $\mu^- \equiv \left\{\mu_i^-\right\}_{i=1}^I$.

- To discretize the derivative at k_i, consider both forwards and backwards differences,

$$v'_F(k_i) \approx \frac{v_{i+1} - v_i}{\Delta} \tag{11}$$

$$v'_B(k_i) \approx \frac{v_i - v_{i-1}}{\Delta} \tag{12}$$

subject to the state constraints

$$v'_F(\overline{k}) = (f(\overline{k}) - \delta\overline{k})^{-\gamma} \tag{13}$$

$$v'_B(\underline{k}) = (f(\underline{k}) - \delta\underline{k})^{-\gamma} \tag{14}$$

Our finite difference approximation to the HJB equation (8) is then

$$\rho v(k_i) = \frac{(v'(k_i))^{-\frac{1-\gamma}{\gamma}}}{1-\gamma} + \mu(k_i; v)v'(k_i) \tag{15}$$

Following the "upwind scheme", we will use the forwards approximation whenever there is a positive drift above the steady state level of capital k^*, and the backwards approximation when below k^*.

1.3 Iterating on a Linear System of Equations

This section solves the problem through a series of iterations on a linear system of equations.

Iterative Method Pick an h, which is used in the iterative process.[3] We first make an initial guess $v^0 = (v_1^0, \dots v_I^0)$. Then for each consecutive iteration $n = 0, 1, 2, ...$, update v^n using the following equation

$$\frac{v_i^{n+1} - v_i^n}{h} + \rho v_i^{n+1} = u(c_i^n) + (v_i^{n+1})' \underbrace{[f(k_i) - \delta k_i - c_i^n]}_{\equiv \mu_i} \tag{16}$$

Using both the forwards and backwards difference approximations (11) and (12), find consumption levels c_F^n and c_B^n as given by (4) and compute savings,

$$\mu_{i,F}^n = f(k_i) - \delta k_i - \left(v_F^n{}'(k_i)\right)^{-\frac{1}{\gamma}} \tag{17}$$

$$\mu_{i,B}^n = f(k_i) - \delta k_i - \left(v_B^n{}'(k_i)\right)^{-\frac{1}{\gamma}} \tag{18}$$

Depending upon the sign of the drift, a choice is made between using forward or backward differences. Let $\mathbf{1}_{\{\cdot\}}$ be an indicator function[4] and $\bar{v_i}'$ be the derivative at the steady state, given in this example by $\bar{v_i}' = (f(k_i) - \delta k_i)^{-\gamma}$. Then the approximation of the derivative is

$$v_i^{n\prime} = v_{i,F}^n{}'\mathbf{1}_{\{\mu_{i,F}>0\}} + v_{i,B}^n \mathbf{1}_{\{\mu_{i,B}<0\}} + \bar{v}_i^{n\prime}\mathbf{1}_{\{\mu_{i,F}<0<\mu_{i,B}\}} \tag{19}$$

Define the vectors $X, Y, Z \in \mathbb{R}^I$ such that

$$X = -\frac{\mu_B^-}{\Delta} \tag{20}$$

$$Y = -\frac{\mu_F^+}{\Delta} + \frac{\mu_B^-}{\Delta} \tag{21}$$

$$Z = \frac{\mu_F^+}{\Delta} \tag{22}$$

[3] **TODO:** I believe this is the Howard algorithm. In effect, I think that this is like using an explicit time step in a PDE, where we are forward iterating fixing the policy? Because of this, there is a high likelihood that it is not unconditionally stable.

[4] For now, assume that if the case $\mu_F > 0$ and $\mu_B < 0$ should arise, take $\mathbf{1}_{\{\mu_{i,F}>0\}}$ to be 1 and $\mathbf{1}_{\{\mu_{i,B}<0\}}$ to be 0.

With these, construct the sparse matrix A^n

$$A^n \equiv \begin{bmatrix} Y_1 & Z_1 & 0 & \cdots & \cdots & \cdots & 0 \\ X_2 & Y_2 & Z_2 & 0 & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & X_{I-1} & Y_{I-1} & Z_{I-1} \\ 0 & \cdots & \cdots & \cdots & 0 & X_I & Y_I \end{bmatrix} \in \mathbb{R}^{I \times I} \tag{23}$$

By substituting the approximation $v^{n\prime}$ found in (19) into equation (4), define the utility vector

$$u(c^n) \equiv u\left(\left(v^{n\prime}\right)^{-\frac{1}{\gamma}}\right) \tag{24}$$

We can then write as a system of equations in matrix form and solve for v^{n+1}

$$B^n \equiv \left(\rho + \frac{1}{h}\right) I - A^n \tag{25}$$

$$b^n \equiv u(c^n) + \frac{v^n}{h} \tag{26}$$

$$B^n v^{n+1} = b^n \tag{27}$$

When v^{n+1} is sufficiently close in value to v^n, the algorithm is complete. Otherwise, update the value of v^n and repeat the previous steps for the next iteration. Keep in mind that the solution may not be unconditionally stable for an arbitrary h.

1.4 Nonlinear System of Equations

This section will focus on solving the problem directly as a nonlinear system of equations. Returning to equation (8), we have

$$\rho v_i = \frac{(v_i')^{-\frac{1-\gamma}{\gamma}}}{1-\gamma} + \mu(k_i; v) v_i' \tag{28}$$

for every k_i, $i = 1, 2, ...I$.

References