Computation of a Baby-step Growth Model

H. Cetin *

March 14, 2023

Abstract

We compute a baby-step stochastic growth model with one factor of production. We first define the problem. Then we show three different ways of solving this baby-step problem: (i) Value Function Iteration (VFI); (ii) Linear-Quadratic Model (LQ Method); (iii) Vaughan's Method. The code can be found here.

*University of Minnesota, E-mail: cetin019@umn.edu

1 The Model

$$
\begin{aligned}
&\max_{\{c_t, k_{t+1}\}} E \sum_{t=0}^{\infty} \beta^t \log(c_t) \\
&\text{subj. to } c_t + k_{t+1} = z_t k_t^{\theta} \\
&\qquad \log z_t = \rho \log z_{t-1} + \epsilon_t, \quad \epsilon \sim N\left(0, \sigma_{\epsilon}^2\right)
\end{aligned} \tag{1}
$$

2 Bellmann Operator

Let z_t be stochastic productivity shock. The Bellmann Operator is:

$$(TV)(y) = \max_{0 \le c \le y} u(c) + \beta \int V(f(y-c)z))\phi(dz)$$

where:

$$f(y_t - c_t) = z_t(y_t - c_t)^{\alpha}$$

$$[log(z_{t+1}) = \rho_0 + \rho log(z_t) + \epsilon_{t+1}] \equiv [z_{t+1} = z_t^{\rho} + exp(\rho_0 + \epsilon_{t+1})], \quad \epsilon \sim N(0, \sigma_{\epsilon}^2)$$

Here, $\psi_{t+1} = exp(\rho_0 + \epsilon_{t+1})$ can be considered as the shock.

Note that $z_{t+1} = z_t^{\rho} + exp(\rho_0 + \epsilon_{t+1})$ is an AR(1) process. Two possibilities:

- $\rho = 0$: Then it is simple to solve since defining a shock vector is enough to solve the problem. Because the process does not depend on the previous factor z_t.
- $\rho \neq 0$: It is more problematic since it is AR(1) process and we need to define Markov transition matrix Q to solve the problem.

2.1 $\rho = 0$ case

Let's determine what the parameters of the problem are:

- β, Time discounting factor
- ρ_0, ρ, Factor production parameters

- $\mu = 0$, σ_ϵ^2, Distribution parameters of the shock
- utility function u, production function f
- grid size of values $grid\_size$
- grid size of shocks $shock\_size$

Now, before defining our class, note that we are going to use Monte Carlo simulation to compute the integral.

$$\int V(f(y-c)z)\phi(dz) \approx \frac{1}{n}\sum_{i=1}^{n} V(f(y-c)z)\psi_i$$

where $\psi_i \equiv exp(\rho_0 + \epsilon_i)$ are IID draws.

Note also that we have finite grid points but $f(y-c)z$ might not be one of those grid points. What should we do?

Answer: Use interpolation! We'll use linear interpolation for simplicity. We'll use "scipy.interpolate.interp1d" method. ".RHS_Bellman" in our class will compute the RHS of the bellman equation. We need to maximize it as well. To do this, we are going to use "scipy.optimize.minimize_scalar". To make our class definition shorter and to be able to use it in the general case, we'll define maximize function outside of the class definition.

2.2 $\rho \neq 0$ case

If we look at our AR(1) process, it is:

$$z_{t+1} = z_t^\rho + exp(\epsilon_{t+1})$$

But as you can see, it is a non-linear AR(1) process. What should we do?

Answer: Instead of tracking z's, track $\hat{z} = log(z)$'s! Then, when we want to use $\hat{z}$ in our production function, just take exponentially of $\hat{z}$. So, we have:

$$\hat{z}_{t+1} = \rho\hat{z}_t + \epsilon_{t+1} \quad \epsilon \sim N(0, \sigma_\epsilon^2)$$

Now, we need to find a **stationary Markov transition matrix** $Q((z,k),z')$ which takes current period productivity and capital as input and gives the probability distribution of the next period's productivity

Now, let N be the grid size of z and M be the grid size of shocks. Then Q would be the MxNxN matrix. But for simplicity, take M = N to have NxNxN transition matrix Q.

Another thing is that, when we make a grid of shocks and productivity z's, then the next period z' will lie in the same grid that we used for z's. This means we can use the 2d NxN matrix for the transition matrix. We are going to use "qe.markov" to construct this Stationary Markov Process.

$Q((z,s),z')$: Probability of the next period's factor is z', given z,s today.

We'll use "qe.markov.approximation.rouwenhorst" method to **discretize Gaussian linear AR(1) processes** in the form $y_t = \bar{y} + \rho y_{t-1} + \varepsilon_t$ via Rouwenhorst's method.

Note that now, the value function will be an NxN matrix as well.

3 LQ Method

Note: To be able to solve the LQ model, I used Marimon and Scott's "Computational Methods for the Study of Dynamic Economies" book.

LQ approximation is reasonable when the following conditions are met:

- The deterministic version of the model converges to a stable steady state.
- Linear Law of motion of the state variables.

A typical LQ problem can be written as:

$$V(z,s) = \max_d \{r(z,s,d) + \beta\, \mathbb{E}[V(z',s')|z]\}$$

$$s.t. \quad s' = A(z,s,d), \;\; z' = L(z) + \epsilon', \;\; \epsilon \sim N(0,\sigma)$$

where:

- z: n_zx1 vector of **exogenous state variables**
- s: n_sx1 vector of **endogenous state variables**
- d: n_dx1 vector of **control variables**
- ϵ: $n_\epsilon x1$ vector of **shocks** with zero mean and finite variance
- r: return function
- A, L: Linear law of motions of state variables

A general solution algorithm for LQ problems is:

1. Choose a point about which to expand the return function. In most cases, it is the steady state point of the deterministic version of the model: $(\bar{z}, \bar{s}, \bar{d})$, that we obtain when we substitute the random variables with their unconditional means.
2. Construct a quadratic approximation of $r(z, s, d)$ about $(\bar{z}, \bar{s}, \bar{d})$.
3. Compute optimal value function $V^*(z, s)$ by successive iterations on Bellman operator:

$$V_{n+1}(z, s) = T[V_n(z, s)] = \max_d \{r(z, s, d) + \beta\, \mathbb{E}[V_n(z', s')|z]\}$$

3.1 Step 1: Computing $(\bar{z}, \bar{s}, \bar{d})$

In our problem:

- z: z (technology)
- s: k (current capital stock)
- d: x (investment) [or k' since there is full depreciation]
- r(z,k,x): $log(e^z k^\alpha - x)$
- A(z,k,x): x

- L(z): ρz so that the law of motion of technology: $\rho z + \epsilon' = z'$

First, we are going to take the log of z, k, and x and denote them with hats:

$$\hat{z} = log(z), \ \ \hat{k} = log(k), \ \ \hat{x} = log(x)$$

The reason we are doing this is to make the law of motion linear.

To compute the deterministic steady state, we substitute shocks by their unconditional mean: $\epsilon_{t+1} = 0$. So, since $\rho < 1$, in limit, $\lim_{t->\infty} \hat{z}_t = 0$. So in steady state, it should be:

$$\bar{\hat{z}} = 0 \rightarrow \bar{z} = 1$$

Now let's write the deterministic version of the problem to find its steady state:

$$\max_{c_t, k_{t+1}} \sum \beta^t log(c_t)$$

$$s.t. \quad c_t + x_t = k_t^\alpha, \quad k_{t+1} = x_t \ \ c_t, k_{t+1} \geq 0, \ \ k_0 \ given$$

From its FOC we have:

$$\frac{c_{t+1}}{c_t} = \beta[\alpha k_{t+1}^{\alpha-1}]$$

In steady state, $\frac{c_{t+1}}{c_t} = 1$. So we have:

$$\bar{k} = [\alpha\beta]^{\frac{1}{1-\alpha}}$$

$$\bar{x} = \bar{k}$$

So, our steady state that we are going to expand the return function is:

$$(\bar{\hat{z}}, \bar{\hat{k}}, \bar{\hat{x}}) = (0, log([\alpha\beta]^{\frac{1}{1-\alpha}}), log([\alpha\beta]^{\frac{1}{1-\alpha}}))$$

3.2 Step 2: Constructing the quadratic approximation of the return function

We will use **2nd order Taylor approximation** to make r(z,s,d) our return function quadratic:

$$r(z,s,d) \simeq \bar{R} + (W - \bar{W})^T \bar{J} + \frac{1}{2}(W - \bar{W})^T \bar{H}(W - \bar{W})$$

where:

- W: $[z, s, d]^T$, vector of ordered state and control variables
- $\bar{W}$: $[\bar{z}, \bar{s}, \bar{d}]$, steady-state values
- $\bar{R}$: $r(\bar{z}, \bar{s}, \bar{d})$ (i.e. return value evaluated at the steady state)
- $\bar{J}$: $[\bar{J}_z, \bar{J}_s, \bar{J}_d]^T$(i.e. **Jacobian** evaluated at the steady state)
- $\bar{H}$: **Hessian** evaluated at the steady state:

$$\bar{H} = \begin{bmatrix} \bar{H}_{\bar{z}\bar{z}} & \bar{H}_{\bar{z}\bar{s}} & \bar{H}_{\bar{z}\bar{d}} \\ \bar{H}_{\bar{s}\bar{z}} & \bar{H}_{\bar{s}\bar{s}} & \bar{H}_{\bar{s}\bar{d}} \\ \bar{H}_{\bar{d}\bar{z}} & \bar{H}_{\bar{d}\bar{s}} & \bar{H}_{\bar{d}\bar{d}} \end{bmatrix}$$

The Taylor approximation can be written as:

$$r(z,s,d) \simeq (\bar{R} - \bar{W}^T \bar{J} + \frac{1}{2}\bar{W}^T \bar{H} \bar{W}) + W^T(\bar{J} - \bar{H}\bar{W}) + \frac{1}{2} W^T \bar{H} W$$

And this equation can be written in **quadratic form** as:

$$r(z,s,d) \simeq \begin{bmatrix} 1 & W^T \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12}^T \\ Q_{12} & Q_{22} \end{bmatrix} \begin{bmatrix} 1 \\ W \end{bmatrix} \quad \equiv \quad \begin{bmatrix} 1 & W^T \end{bmatrix} Q \begin{bmatrix} 1 \\ W \end{bmatrix}$$

where:

- $Q_{11} = \bar{R} - \bar{W}^T \bar{J} + \frac{1}{2}\bar{W}^T \bar{H} \bar{W}$
- $Q_{12} = \frac{1}{2}(\bar{J} - \bar{H}\bar{W})$
- $Q_{22} = \frac{1}{2}\bar{H}$

Here, the role of 1 is to select the constant term of the quadratic expression.

Dimension of Q is $n_z + n_s + n_d + 1$

3.3 Step 3: Computing the Optimal Value Function

Bellman operator is:

$$V_{n+1}(z,s) = \max_d \{[1\, W^T]\, Q\, [1\, W]^T + \beta\, \mathbb{E}[V_n(z', s')|z]\}$$

$$s.t. \quad s' = A(z,s,d), \quad z' = L(z) + \epsilon'$$

The initial guess should be a quadratic and concave function $V_n = F^T P_n F$ where $F = [1, z, s]^T$, P_n is symmetric and negative semidefinite matrix with dimension $1 + n_z + n_s$

Note that any square matrix with dimension $(1 + n_z + n_s)$x$(1 + n_z + n_s)$ with very small negative numbers on the diagonal and zeros anywhere else would satisfy P_n's necessary conditions.

We will use **Certainty Equivalence Principle** because in general operating with expectation is a hard thing to do. Note that the certainty equivalence principle holds only when the objective function is quadratic and the constraints are linear.

By certainty equivalence, the covariance matrix of the vector of random variables $\Sigma = CC' = 0$, so our Bellmann operator becomes:

$$V_{n+1} = \max_{d}\{[1\, W^T]\, Q\, [1\, W]^T + \beta\, (F')^T P_n (F')\}$$

$$s.t. \quad s' = A(z, s, d), \quad z' = L(z)$$

The next step is to transform this Bellmann operator as a quadratic function of $[1W]^T$. To do this, we use constraints A, and L to substitute the forwarded values of the states out of our Bellmann operator.

Specifically, we need to find a rectangular matrix B (a matrix of linear constraints) of dimension $(1 + n_z + n_s)$x$(1 + n_z + n_s + n_d)$ which satisfies:

$$F' = B[1W]^T$$

(i.e. $B[1\ W]^T$ is the law of motion of state variable $F = [1, z, s]^T$)

In our model, the B is:

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \rho & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

After finding B, we substitute F' in the Bellmann operator with $B[1W]^T$ and we get

$$V_{n+1} = \max_{d}\{[1\, W^T]\, Q\, [1\, W]^T + \beta\, [1W^T]B^T P_n B[1W^T]^T\}$$

Notice that we get rid of the constraints by introducing B. (we incorporated them into the objective function)

As you can see the objective function is now a quadratic function of $[1W]^T$:

$$V_{n+1} = \max_d \{[1\, W^T]\,(Q + \beta\, B^T P_n B)[1W^T]^T\}$$

The next step is to differentiate this new Bellmann operator to obtain decision rules $d_n(z, s)$. (i.e. solve the maximization problem on RHS)

Note that since the objective function is quadratic, the FOC's will be linear in (z, s).

To make the calculations easier we are going to separate state variable and control variable calculations by making partitions. First, define a partition of Q:

$$Q = \begin{bmatrix} \hat{Q}_{FF} & \hat{Q}_{Fd}^T \\ \hat{Q}_{Fd} & \hat{Q}_{dd} \end{bmatrix}$$

and define:

$$M_n = B^T P_n B \qquad M_n = \begin{bmatrix} M_{FF}^n & (M_{Fd}^n)^T \\ M_{Fd}^n & M_{dd}^n \end{bmatrix}$$

Note that Q_{FF}... is not necessarily equal to Q_{11}... We define this new partition just to simplify the algebra. However, in our model, they are going to coincide.

Here:

- Q_{FF} and M_{FF}^n: Symmetric matrices of dimension $(1+n_z+n_s)$x$(1+n_z+n_s)$
- Q_{Fd} and M_{Fd}^n: Symmetric matrices of dimension n_dxn_d
- Q_{FF} and M_{FF}^n: Rectangular matrices of dimension n_dx$(1 + n_z + n_s)$

Substituting this partitions into our Bellmann operator, it becomes: (remember $[1W]^T = [Fd]^T$)

$$V_{n+1} = \max_d \begin{bmatrix} F^T & d^T \end{bmatrix} \begin{bmatrix} Q_{FF} + \beta M_{FF}^n & Q_{Fd}^T + \beta (M_{Fd}^n)^T \\ Q_{Fd} + \beta M_{Fd}^n & Q_{dd} + \beta M_{dd}^n \end{bmatrix} \begin{bmatrix} F \\ d \end{bmatrix}$$

This can be rewritten as:

$$V_{n+1} = \max_d F^T[Q_{FF} + \beta M_{FF}^n]F + 2d^T[Q_{Fd} + \beta M_{Fd}^n]F + d^T[Q_{dd} + \beta M_{dd}^n]d$$

The RHS is a maximization problem with respect to d. So differentiate RHS with respect to d^T (use matrix differentiation tactics):

$$2[Q_{Fd} + \beta M_{Fd}^n]F + 2[Q_{dd} + \beta M_{dd}^n]d = 0$$

Thus, the optimal policy function is:

$$d_n(z, s) = -(Q_{dd} + \beta M_{dd}^n)^{-1}(Q_{Fd} + \beta M_{Fd}^n)F \quad \equiv \quad G_n^T F$$

Define:

$$P_{n+1} = Q_{FF} + \beta M_{FF}^n - (Q_{Fd} + \beta M_{Fd}^n)^T(Q_{dd} + \beta M_{dd}^n)^{-1}(Q_{Fd} + \beta M_{Fd}^n)$$

Then:

$$V_{n+1} = F^T P_{n+1} F$$

We will do the value function iteration until $P_n = P_{n+1}$

4 Vaughan's Method

Now, we used the following quadratic return equation above:

$$r(z,s,d) \simeq \begin{bmatrix} 1 & W^T \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12}^T \\ Q_{12} & Q_{22} \end{bmatrix} \begin{bmatrix} 1 \\ W \end{bmatrix} \quad \equiv \quad \begin{bmatrix} 1 & W^T \end{bmatrix} Q \begin{bmatrix} 1 \\ W \end{bmatrix}$$

But this can be written in McGrattan (1990) way as follows:

$$r(z,s,d) \equiv \quad \begin{bmatrix} F^T \end{bmatrix} \begin{bmatrix} Q_{FF} \end{bmatrix} \begin{bmatrix} F \end{bmatrix} + d \begin{bmatrix} Q_{dd} \end{bmatrix} d + 2 \begin{bmatrix} F^T \end{bmatrix} \begin{bmatrix} Q_{Fd}^T \end{bmatrix} d$$

(Remember that F = [1,z,s], W=[z,s,d])

And we also know that the law of motion is the following:

$$\begin{bmatrix} 1 \\ \hat{z}_{t+1} \\ \hat{k}_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ \hat{z}_t \\ \hat{k}_t \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \hat{x}_t + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \epsilon_{t+1}$$

Here:

- $\mathbb{Q} = Q_{FF} = Q[0:3, 0:3]$ 3x3 matrix
- $\mathbb{R} = Q_{dd} = Q[3,3]$ 1x1 scalar
- $\mathbb{W} = Q_{Fd}^T = Q[3, 0:3]^T$ 3x1 vector
- $\mathbb{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & 0 \end{bmatrix}$
- $\mathbb{B} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$

- $\mathbb{C} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$

Now, since we know the A, B, C, R, Q, and W; we can solve Vaughan's method easily.

First, let's normalize everything to get rid of the discount factor and cross-product of state and control variables:

$$\tilde{X_t} = \beta^{\frac{t}{2}} X_t$$

$$\tilde{u_t} = \beta^{\frac{t}{2}}(u_t + R^{-1}W'X_t)$$

$$\tilde{A} = \sqrt{\beta}(A - BR^{-1}W')$$

$$\tilde{B} = \sqrt{\beta}B$$

$$\tilde{Q} = Q - WR^{-1}W'$$

Let's define now the following Hamiltonian Matrix:

$$\mathbb{H} = \begin{bmatrix} \tilde{A^{-1}} & \tilde{A^{-1}}\tilde{B}R^{-1}\tilde{B}' \\ \tilde{Q}\tilde{A^{-1}} & \tilde{Q}\tilde{A^{-1}}\tilde{B}R^{-1}\tilde{B}' + \tilde{A}' \end{bmatrix}$$

and decompose it then order the eigenvalues such that the first half of the eigenvalues exceed the unit root, and the second half is the reverse of the first half.

Then, Vaughan says that $P = V_{21}V_{11}^{-1}$ and $F = \dfrac{1}{(R + (\tilde{B}'P\tilde{B}))}(\tilde{B}'P\tilde{A}) + R^{-1}W'$

References

[1] McGrattan, E. R. (1990). Solving the Stochastic Growth Model by Linear-Quadratic Approximation. Journal of Business & Economic Statistics, 8(1), 41–44. https://doi.org/10.2307/1391750

[2] McGrattan, E. R. (1994). A note on computing competitive equilibria in linear models. Journal of Economic Dynamics and Control, 18(1), 149-160. https://doi.org/10.1016/0165-1889(94)90073-6