Some Notes on Solving a RBC Model

• We want to solve and estimate model of the type:

$$\max E_0 \sum_{t=0}^{\infty} \beta^t (u(c_t) + v(1 - h_t))$$

subject to

$$c_t + i_t = \lambda_t k_t^\theta h_t^{1-\theta}$$
$$k_{t+1} = (1 - \delta)k_t + i_t$$
$$\lambda_{t+1} = (1 - \rho) + \rho \lambda_t + \epsilon_{t+1}$$

• Let’s assume $u(c_t) = \log c_t$, $v(1 - h_t) = A \log(1 - h_t)$. 
• We are going to follow Kydland and Prescott (1982) by replacing this problem with a linear quadratic approximation of this model.

• We are going to take a Taylor-approximation around the non-stochastic steady-state of this model.

• So we are do this in three steps

  1. Solve the model for a steady state.
  2. Substitute the constraint into the objective function.
  3. Take a Taylor approximation of the objective function
Step 1: Solve for a steady state

- Set of the non-stochastic Lagrangian

\[
\mathcal{L} = \sum_{t=0}^{\infty} \beta^t \left[ \log c_t + A \log(1 - h_t) + \mu_t (\lambda_t k_t^{\theta} h_t^{1-\theta} + (1 - \delta) k_t - c_t - k_{t+1}) \right]
\]

- Taking first-order conditions yields

\[
\frac{\partial \mathcal{L}}{\partial c_t} = \frac{1}{c_t} - \mu_t = 0 \\
\frac{\partial \mathcal{L}}{\partial k_{t+1}} = -\mu_t + \beta \mu_{t+1} \left[ (1 - \delta) + \lambda_{t+1} \theta k_{t+1}^{\theta - 1} h_{t+1}^{1-\theta} \right] = 0 \\
\frac{\partial \mathcal{L}}{\partial h_t} = -\frac{A}{1 - h_t} + \mu_t (1 - \theta) \lambda_t k_t^{\theta} h_t^{-\theta} = 0 \\
\frac{\partial \mathcal{L}}{\partial \mu_t} = \lambda_t k_t^{\theta} h_t^{1-\theta} + (1 - \delta) k_t - c_t - k_{t+1} = 0
\]
• Impose the steady-state

\[
\frac{1}{c} = \mu \\
\mu = \beta \mu \left[(1 - \delta) + \lambda \theta k^{\theta - 1} h^{1 - \theta}\right] \\
\frac{A}{1 - h_t} = \mu (1 - \theta) \lambda k^\theta h^{-\theta} \\
\lambda k^\theta h^{1 - \theta} + (1 - \delta) k - c - k = 0
\]

• Do some algebra and you get

\[
k = \left[\frac{1 - \beta (1 - \delta)}{\beta \lambda \theta}\right]^{1/\theta - 1} h \\
k = \phi^{1/\theta - 1} h \\
c = \phi^{1/\theta - 1} \left[\lambda \phi^\theta - \delta\right] h \\
h = \frac{(1 - \theta) \lambda \phi}{\lambda \phi^\theta - \delta + (1 - \theta) \lambda \theta / A} \frac{(1-\theta)\lambda \phi}{\lambda \phi^\theta - \delta + (1 - \theta) \lambda \theta / A}
\]
Step 2: Substitute the constraint into the objective function

So the problem becomes:

$$\max E_0 \sum_{t=0}^{\infty} \beta^t \left[ \log \left( \lambda_t k_t h_t \right) - i_t \right] + A \log(1 - h_t)$$

subject to:

$$k_{t+1} = (1 - \delta) k_t + i_t$$

$$\lambda_{t+1} = (1 - \rho) + \rho \lambda_t + \epsilon_{t+1}$$
So now the problem has the following form

\[
\max_{u_t} E_0 \sum_{t=0}^{\infty} \beta^t r_t(z_t)
\]

subject to:

\[
x_{t+1} = Ax_t + Bu_t + Cw_{t+1}
\]

where

\[
x_t \quad \text{is the vector of state variables}
\]
\[
u_t \quad \text{is the vector of control variables}
\]
\[
w_{t+1} \quad \text{is a vector of shocks}
\]
\[
z_t = [x_t' \ ; \ u_t']
\]

and \(A, B,\) and \(C\) are matrices.
In particular \(w_{t+1}\) is a martingale difference sequence with \(E w_{t+1} w_{t+1}' = I.\)
In our case

\[
\begin{align*}
    x_t &= \begin{bmatrix} \lambda_t \\ k_t \end{bmatrix}
    \quad \Rightarrow \\
    u_t &= \begin{bmatrix} i_t \\ h_t \end{bmatrix}
    \quad \Rightarrow \\
    z_t &= \begin{bmatrix} \lambda_t \\ k_t \\ i_t \\ h_t \end{bmatrix}
    \quad \Rightarrow
\end{align*}
\]

Writing out the law of motion for the state variable.

\[
\begin{bmatrix}
    x_{t+1} \\
    \lambda_{t+1} \\
    k_{t+1}
\end{bmatrix} =
\begin{bmatrix}
    1 & 1 & 0 & 0 \\
    1 & 1 & -\rho & 0 \\
    0 & 0 & 1 & -\delta
\end{bmatrix}
\begin{bmatrix}
    x_t \\
    \lambda_t \\
    k_t
\end{bmatrix}
+ \begin{bmatrix}
    0 \\
    0 \\
    1 \\
\end{bmatrix}
\sigma_e
+ \begin{bmatrix}
    0 \\
    0 \\
    w_{t+1}
\end{bmatrix}
\]
Replace the $r(z_t)$ function with a Taylor approximation:

$$r(z_t) \approx \hat{r}(z_t) = r(z_{ss}) + r(z - z_{ss}) \frac{\partial r}{\partial z} + \frac{1}{2} (z - z_{ss}) \frac{\partial^2 r}{\partial z \partial z} (z - z_{ss})$$

In matrix form

$$\hat{r}(z_t) = z'Mz$$

where $M = \left( \begin{array}{c} 1 \\ \frac{\partial r(z_{ss})}{\partial z_{ss}} \\ \frac{\partial^2 r(z_{ss})}{\partial z_{ss}^2} \end{array} \right) \left( \begin{array}{c} 1 \\ \frac{\partial r(z_{ss})}{\partial z_{ss}} \\ \frac{\partial^2 r(z_{ss})}{\partial z_{ss}^2} \end{array} \right)$

and $e$ is a vector of zeros with 1 in the element corresponding to the constant term in $x_t$. 

$M$ is
So we are going to generate an approximate solution to original problem by solving the following linear quadratic problem

$$\max_{u_t} E_0 \sum_{t=0}^{\infty} \beta^t z_t' M z_t$$

subject to:

$$x_{t+1} = Ax_t + Bu_t + Cw_{t+1}$$

Let’s map this problem into an optimal linear regulator formulation

$$\hat{r}(z) = z' M z$$

(5)

$$= [x \ u] \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

(6)

$$= [x \ u] \begin{bmatrix} Q & W \\ W' & R \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

(7)
We can write the problem as:

$$\max_{u_t} E_0 \sum_{t=0}^{\infty} \beta^t [x_t'Qx_t + u_t'Ru_t + 2x_t'Wu_t]$$

subject to:

$$x_{t+1} = Ax_t + Bu_t + Cw_{t+1}$$

The most straightforward way to solve this problem is via dynamic programming. Let $V(x)$ be the optimal value associated with the program starting from an initial state vector $x_0 = x$.

$$V_{j+1}(x_t) = \max_{u_t} \{x_t'Qx_t + u_t'Ru_t + 2x_t'Wu_t + \beta E_t V(x_{t+1})\}$$

subject to:

$$x_{t+1} = Ax_t + Bu_t + Cw_{t+1}$$
Can solve via backward recursions. Initialize $V_0(x) = 0$ This will yield the following quadratic form.

$$V_{j+1}(x_t) = x_t'P_jx_t + \rho_j$$

where

$$P_{j+1} = Q + \beta A'P_jA - (\beta A'P_jB + W)(R + \beta B'P_jB + W)(R + \beta B'P_jB)^{-1}(\beta B'P_jA + W')$$

$$\rho_{j+1} = \beta \rho_j + \beta \text{trace}P_jCC'$$

This first equation is known as a matrix Riccati equation.

By backward recursions, the value function will converge to:

$$V(x_t) = x_t'P_{x_t} + \rho$$

If we substitute $P_j$ into the value function and take the derivative with respect to $u$ we can solve the decision rule

$$u_t = -F_jx_t$$

where

$$F_j = (R + \beta B'P_jB)^{-1}(\beta B'P_jA + W')$$
If we iterate on $P$ we will get a fixed point, so

$$F = (R + \beta B'PB)^{-1}(\beta B'PA + W')$$

Notice the decision rule does not depend on $C$. Certainty Equivalence! But the value function depends on $C$.

Return to our law of motion for the state variables

$$x_{t+1} = Ax_t + Bu_t + Cw_{t+1}$$

Substitute

$$x_{t+1} = Ax_t + B(-Fx_t) + Cw_{t+1}$$

$$= (A - BF)x_t + Cw_{t+1}$$

Call $A_o = (A - BF)$.

Now we have

$$x_{t+1} = A_o x_t + Cw_{t+1}$$

We can augment this system with a second matrix equation

$$y_t = Gx_t$$

where $y$ is a vector of variables we observe and care about.