

Materials 12e - PQ solution
- works for the default model
See Notes 6 Jan 2020

Laura Gáti

January 6, 2020

1 Model equations and goal

$$x_t = -\sigma i_t + \hat{\mathbb{E}}_t \sum_{T=t}^{\infty} \beta^{T-t} ((1-\beta)x_{T+1} - \sigma(\beta i_{T+1} - \pi_{T+1}) + \sigma r_T^n) \quad (1)$$

$$\pi_t = \kappa x_t + \hat{\mathbb{E}}_t \sum_{T=t}^{\infty} (\alpha\beta)^{T-t} (\kappa\alpha\beta x_{T+1} + (1-\alpha)\beta\pi_{T+1} + u_T) \quad (2)$$

$$i_t = \psi_\pi \pi_t + \psi_x x_t + \bar{i}_t \quad (3)$$

Goal: obtain endogenous stuff as a function of expectations and states:

$$z_t = \begin{bmatrix} \pi_t \\ x_t \\ i_t \end{bmatrix} = A_a f_a + A_b f_b + A_s s_t \quad (4)$$

where I already have expectations f_a, f_b and the state vector can vary by model, but in this default case with $\rho = 0$ it is

$$s_t = \begin{bmatrix} r_t^n \\ \bar{i}_t \\ u_t \end{bmatrix} \quad (5)$$

That is, we want the matrices A_a, A_b and A_s .

2 Method 1 - MN method (old method) - this is done to check the PQ method

This follows the notation I use in Mathematica (`materials12e.nb`)

$$\underbrace{\begin{bmatrix} \sigma\psi_\pi & 1 + \sigma\psi_x \\ 1 & -\kappa \end{bmatrix}}_{\equiv M} \begin{bmatrix} \pi_t \\ x_t \end{bmatrix} = \underbrace{\begin{bmatrix} \left[\sigma(1 - \beta\psi_\pi), & 1 - \beta - \sigma\beta\psi_x, & 0 \right] f_b + d_{x,s} s_t \\ \left[(1 - \alpha)\beta, & \kappa\alpha\beta, & 0 \right] f_a + d_{\pi,s} s_t \end{bmatrix}}_{\equiv N} \quad (6)$$

where

$$d_{x,s} = -\sigma \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} InxBhx \quad InxBhx \equiv (I_{nx} - \beta h_x)^{-1} \quad (7)$$

$$d_{\pi,s} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} InxABhx \quad InxABhx \equiv (I_{nx} - \alpha\beta h_x)^{-1} \quad (8)$$

$$d_{i,s} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \quad (9)$$

where I only specify $InxABhx, InxBhx$ in Matlab, leaving them parametric in Mathematica. Then Mathematica solves for x and π as

$$\begin{bmatrix} \pi_t^* \\ x_t^* \end{bmatrix} = M^{-1}N \quad (10)$$

Then the solution for the interest rate will just be

$$i_t = \psi_\pi \pi_t^* + \psi_x x_t^* + d_{i,s} s_t \quad (11)$$

but I don't solve for it here since I only compare π_{MN}^* to π_{PQ}^* .

3 Method 2 - PQ method (new method)

Instead of subbing out the interest rate in the original equations, write the system as:

$$\underbrace{\begin{bmatrix} 0 & 1 & \sigma \\ 1 & -\kappa & 0 \\ -\psi_\pi & -\psi_x & 1 \end{bmatrix}}_{\equiv P} \begin{bmatrix} \pi_t \\ x_t \\ i_t \end{bmatrix} = \underbrace{\begin{bmatrix} \left[\sigma, & 1 - \beta, & \beta(-\sigma) \right] f_b + c_{x,s} s_t \\ \left[(1 - \alpha)\beta, & \alpha\beta\kappa, & 0 \right] f_a + c_{\pi,s} s_t \\ c_{i,s} s_t \end{bmatrix}}_{\equiv Q} \quad (12)$$

where

$$c_{x,s} = \sigma \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \cdot \text{InxBhx}; \quad (13)$$

$$c_{\pi,s} = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} \cdot \text{InxABhx} \quad (14)$$

$$c_{i,s} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix} = d_{i,s} \quad (15)$$

where InxABhx and InxBhx are the same as before. Then (10) is replaced by

$$\begin{bmatrix} \pi_t^* \\ x_t^* \\ i_t^* \end{bmatrix} = P^{-1}Q \quad (16)$$

where you also have to impose the relation

$$f_b(3) = \psi_\pi f_b(1) + \psi_x f_b(2) + \frac{1}{\beta} \{ \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} (I_{nx} - \beta h_x)^{-1} s_t - \begin{bmatrix} -1 & 1 & 0 \end{bmatrix} s_t \} \quad (*)$$

so that Mathematica recognizes that the interest rate expectations are just a function of those of π and x .

The last step is to gather the matrices $g_{i,j}$, the coefficients of i on j , $i = x, \pi, i, j = f_a, f_b, s$. Mathematica will output these g -matrices and stack them appropriately to give the A -matrices:

$$\underbrace{A_a}_{ny \times ny} = \begin{pmatrix} g_{\pi,a} \\ g_{x,a} \\ g_{i,a} \end{pmatrix} \quad \underbrace{A_b}_{ny \times ny} = \begin{pmatrix} g_{\pi,b} \\ g_{x,b} \\ g_{i,b} \end{pmatrix} \quad \underbrace{A_s}_{ny \times nx} = \begin{pmatrix} g_{\pi,s} \\ g_{x,s} \\ g_{i,s} \end{pmatrix} \quad (17)$$

Now you can be copy this directly into Matlab (for the default model `matrices_A3.m`), specifying only `InxABhx`, `InxBhx` in Matlab.