Numerical Methods for New Keynesian ZLB Models: Part II

Takeki Sunakawa

February 7, 2018

What is covered

- Equilibrium with a Taylor rule with interest rate inertia
    - Linear Interporation
    - Solving nonlinear equations: Golden section search vs. Newton method
- Equilibrium under the optimal commitment policy (Adam and Billi, 2006)

Equilibrium with Taylor rule with interest rate inertia

- Equilibrium conditions are

$$
\begin{aligned}
y_t &= E_t y_{t+1} - \left(i_t - E_t \pi_{t+1} - s_t\right), \\
\pi_t &= \kappa y_t + \beta E_t \pi_{t+1}, \\
i_t^* &= \rho_r i_{t-1}^* + (1 - \rho_r)\left(r^* + \phi_\pi E_t \pi_{t+1}\right),
\end{aligned}
$$

  and the zero lower bound

$$
i_t = \max\left\{0, i_t^*\right\}.
$$

- s_t follows a Markov chain.

Solving the model

- The solution has a form of

$$y = y(i^*_{-1}, s_k), \quad \pi = \pi(i^*_{-1}, s_k), \quad i = i(i^*_{-1}, s_k).$$

- Now the solution is a function of continuous object (i.e., $i^*_{-1} \in \mathbb{R}$) indexed by the state of the Markov chain $s_k \in \{s_1, s_2, ..., s_{N_s}\}$.

- How to deal with such a continuous object?

Setting up grid points

- We set grid points for i^*_{-1}:

$$i^*_{m,-1} \in \mathcal{I} = \{i^*_{1,-1}, i^*_{2,-1}, ..., i^*_{N,-1}\},$$

where m is an index for the set of grid points $\mathcal{I}$.

Policy function iteration: Initial guess

- A guess of the policy functions

$$y = y^{(0)}(i^*_{-1}, s_k), \quad \pi = \pi^{(0)}(i^*_{-1}, s_k), \quad i = i^{(0)}(i^*_{-1}, s_k).$$

- We know the values of the functions only at each *grid point*, e.g.,

$$\begin{aligned} y^{(0)}(i^*_{m,-1}, s_k) &\in y^{(0)}(i^*_{-1}, s_k) = [y_{1k}, y_{2k}, ..., y_{Nk}]', \\ \pi^{(0)}(i^*_{m,-1}, s_k) &\in \pi^{(0)}(i^*_{-1}, s_k) = [\pi_{1k}, \pi_{2k}, ..., \pi_{Nk}]', \\ i^{(0)}(i^*_{m,-1}, s_k) &\in i^{(0)}(i^*_{-1}, s_k) = [i_{1k}, i_{2k}, ..., i_{Nk}]'. \end{aligned}$$

for $m = 1, ..., N$ and $k = 1, ..., N_s$.

Two dimensional grid points

- Now we have two dimensional state space with grid points indexed by (m, k). One is for $i^*_{m,-1}$, the other is for s_k. For example, the set of grid points are represented by $N \times N_s$ matrix:

$$y^{(0)}(i^*_{m,-1}, s_k) \in y^{(0)}(i^*_{-1}, s) = \begin{bmatrix} y_{11} & \cdots & \cdots & y_{1N_s} \\ y_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ y_{N1} & \cdots & \cdots & y_{NN_s} \end{bmatrix}.$$

Policy function iteration: Solving at each grid

- At each grid point (m, k), having the values of $(i^*_{m,-1}, s_k)$ at hand, we solve

$$
\begin{aligned}
y_{mk} &= y^e_k(i^*_{mk}) - \left(i_{mk} - \pi^e_k(i^*_{mk}) - s_k\right), \\
\pi_{mk} &= \kappa y_{mk} + \beta \pi^e_k(i^*_{mk}), \\
i^*_{mk} &= \rho_r i^*_{m,-1} + (1-\rho_r)\left(r^* + \phi_\pi \pi^e_k(i^*_{mk})\right), \\
i_{mk} &= \max\left\{0, i^*_{mk}\right\},
\end{aligned}
$$

for $(y_{mk}, \pi_{mk}, i^*_{mk}, i_{mk})$, where

$$
\begin{aligned}
y^e_k(i^*_{mk}) &= \sum_{l=1}^{N_s} p(k,l) y^{(0)}(i^*_{mk}, s_l), \\
\pi^e_k(i^*_{mk}) &= \sum_{l=1}^{N_s} p(k,l) \pi^{(0)}(i^*_{mk}, s_l).
\end{aligned}
$$

and $p(k, l)$ is the (k, l) element of the transition matrix.

Policy function iteration: Updating and convergence

- Once this is done for all the grid points, we update

$$y^{(1)}(i^*_{-1}, s_k) = [y_{1k}, y_{2k}, ..., y_{Nk}]',$$
$$\pi^{(1)}(i^*_{-1}, s_k) = [\pi_{1k}, \pi_{2k}, ..., \pi_{Nk}]',$$
$$i^{(1)}(i^*_{-1}, s_k) = [i_{1k}, i_{2k}, ..., i_{Nk}]',$$

  for $k = 1, ..., N_s$.

- We repeat the procedure until the policy functions converge, i.e., $\left\| x^{(j)}(i^*_{-1}, s_k) - x^{(j-1)}(i^*_{-1}, s_k) \right\| < \epsilon$ for $x \in \{y, \pi, i\}$.

Solving nonlinear equation

- We need to solve the following nonlinear equation:

$$i^* = \rho_r i^*_{m,-1} + (1-\rho_r)\left(r^* + \phi_\pi \pi^e_k(i^*)\right)$$

  for i^*.

- Also, we need to evaluate the function $\pi^e_k(i^*)$ at $i^* \notin \mathcal{I}$.

- More genereally, we want to:
  - solve $f(x) = 0$ for x,
  - when we know only the values of $f(x_k)$ at $x_k \in \{x_1, ..., x_N\}$.

Linear interpolation

- Suppose we have the values of $f(x_k)$ at grid points $x_k \in \{x_1, ..., x_N\}$.
- Then, the value of $f(x)$ at $x \in [x_k, x_{k+1}]$ is approximated by

$$\hat{f}(x) = w(x)f(x_k) + (1 - w(x))f(x_{k+1}),$$

  where $w(x) = \frac{x_{k+1}-x}{x_{k+1}-x_k}$.
- Note that $\hat{f}(x)$ is not differentiable at the grid points.
- The command `interp1` in Matlab does the linear interpolation (but slow).

Minimization

- Suppose $f(x)$ is a continuous (but not necessarily differentiable) function. We will find $x^* \in [a, b]$ such that $f(x) \geq f(x^*)$ for any $x \in [x^* - \varepsilon, x^* + \varepsilon]$,
$$x^* = \text{arg} \min_{x \in [a,b]} f(x).$$
- Note that the method can be used even when f is not differentiable. However, it may find local minima when f is not quasi-convex.
- The command `fminbnd` is a minimization function based on the golden section search.
- Root-finding = Minimization problem of $|f(x)|$.

Golden section search

- We search iteratively for x, at each step tightening the bounds which bracket it. Set r to the golden ratio, $(3-\sqrt{5})/2$ and let

$$\begin{aligned} c &= (1-r)a + rb. \\ d &= ra + (1-r)b. \end{aligned}$$

1. If $f(c) \geq f(d)$ then we know the minumum will be in $[c, b]$.
2. If $f(c) < f(d)$ then we know the minumum will be in $[a, d]$.

In case 1, shift the new bounds $[a', b']$ onto $[c, b]$. In case 2, shift the new bounds $[a', b']$ onto $[a, d]$.

Golden section search

- [Demonstrate the golden section search]

Newton-Raphson method

- We solve an equation $f(x) = 0$ for x.
- First-order Taylor expansion: $f(x) \approx f(x_0) + f'(x_0)(x - x_0) = 0$.
- The Newton-Raphson method updates x_k

$$x_{k+1} = x_k - f'(x_k)^{-1} f(x_k),$$

until $\|x_{k+1} - x_k\| \leq \epsilon$.

Optimal Commitment Policy

- The policymaker chooses $\{\pi_t, y_t, i_t\}$ so as to maximize

$$V_0 \equiv -E_0 \sum_{t=0}^{\infty} \beta^t \left(\pi_t^2 + \lambda y_t^2\right)$$

subject to

$$\begin{aligned} y_t &= E_t y_{t+1} - (i_t - E_t \pi_{t+1}) + g_t, \\ \pi_t &= \kappa y_t + \beta E_t \pi_{t+1} + u_t, \\ i_t &\geq 0, \end{aligned}$$

Shock processes

- Exogenous shocks are given by

$$
\begin{aligned}
g_t &= (1-\rho_g)g + \rho_g g_{t-1} + \varepsilon_{g,t}, \\
u_t &= \rho_u u_{t-1} + \varepsilon_{u,t},
\end{aligned}
$$

  where $\varepsilon_{g,t} \sim N(0, \sigma_g^2)$ and $\varepsilon_{u,t} \sim N(0, \sigma_u^2)$.

- Note that $g = r^*$.

Optimal Commitment Policy: Lagrangean

- Lagrangean is

$$\mathcal{L} \equiv E_0 \sum \beta^t \left(\pi_t^2 + \lambda y_t^2\right) + 2\phi_{PC,t} \left(-\pi_t + \kappa y_t + \beta E_t \pi_{t+1} + u_t\right)$$
$$+ 2\phi_{EE,t} \left(-y_t - i_t + E_t y_{t+1} + E_t \pi_{t+1} + g_t\right) + 2\phi_{ZLB,t} i_t.$$

- First-order necessary conditions are

$$\begin{aligned}
\partial \pi_t :& \pi_t - \phi_{PC,t} + \phi_{PC,t-1} + \beta^{-1} \phi_{EE,t-1} = 0, \\
\partial y_t :& \lambda y_t + \kappa \phi_{PC,t} - \phi_{EE,t} + \beta^{-1} \phi_{EE,t-1} = 0, \\
\partial i_t :& - \phi_{EE,t} + \phi_{ZLB,t} = 0.
\end{aligned}$$

Complementary slackness

- Complementary slackness condition:

$$\phi_{ZLB,t} > 0 \perp i_t > 0.$$

- When $i_t > 0$, $\phi_{ZLB,t} = 0$. Equilibrium conitions are

$$\begin{aligned}
i_t &= -y_t + E_t y_{t+1} + E_t \pi_{t+1} + g_t, \\
\pi_t &= \kappa y_t + \beta E_t \pi_{t+1} + u_t, \\
\pi_t &- \phi_{PC,t} + \phi_{PC,t-1} + \beta^{-1} \phi_{EE,t-1} = 0, \\
\lambda y_t &+ \kappa \phi_{PC,t} + \beta^{-1} \phi_{EE,t-1} = 0.
\end{aligned}$$

- When $i_t = 0$, $\phi_{ZLB,t} > 0$. Equilibrium conitions are

$$\begin{aligned}
0 &= -y_t + E_t y_{t+1} + E_t \pi_{t+1} + g_t, \\
\pi_t &= \kappa y_t + \beta E_t \pi_{t+1} + u_t, \\
\pi_t &- \phi_{PC,t} + \phi_{PC,t-1} + \beta^{-1} \phi_{EE,t-1} = 0, \\
\lambda y_t &+ \kappa \phi_{PC,t} - \phi_{EE,t} + \beta^{-1} \phi_{EE,t-1} = 0.
\end{aligned}$$

Complementary slackness

- Note that $E_t y_{t+1} = y^e(\phi_{PC,t}, \phi_{EE,t})$ and $E_t \pi_{t+1} = \pi^e(\phi_{PC,t}, \phi_{EE,t})$.
- When $i_t > 0$, we solve

$$\left(-\phi_{PC,t} + \phi_{PC,t-1} + \beta^{-1}\phi_{EE,t-1}\right) - \kappa\lambda^{-1}\left(\kappa\phi_{PC,t} + \beta^{-1}\phi_{EE,t-1}\right) + \beta\pi^e(\phi_{PC,t}, 0) + u_t = 0$$

  for $\phi_{PC,t}$.

- When $i_t = 0$, we solve

$$\left(-\phi_{PC,t} + \phi_{PC,t-1} + \beta^{-1}\phi_{EE,t-1}\right) - \kappa\lambda^{-1}\left(\kappa\phi_{PC,t} - \phi_{EE,t} + \beta^{-1}\phi_{EE,t-1}\right) + \beta\pi^e(\phi_{PC,t}, \phi_{EE,t}) + u_t = 0,$$

$$-\lambda^{-1}\left(\kappa\phi_{PC,t} - \phi_{EE,t} + \beta^{-1}\phi_{EE,t-1}\right) + y^e(\phi_{PC,t}, \phi_{EE,t}) + \pi^e(\phi_{PC,t}, \phi_{EE,t}) + g_t = 0,$$

  for $(\phi_{PC,t}, \phi_{EE,t})$.

Bi-linear interpolation

- Suppose we have the values of $f(x_k, y_l)$ at grid points $(x_k, y_l) \in X \times Y$, where $X = \{x_1, ..., x_N\}$ and $Y = \{y_1, ..., y_N\}$.

- Then, the value of $f(x, y)$ at $x \in [x_k, x_{k+1}]$ and $y \in [y_k, y_{k+1}]$ is approximated by

$$\begin{aligned}\hat{f}(x,y) &= w(y)\left[w(x)f(x_k, y_l) + (1 - w(x))f(x_{k+1}, y_l)\right], \\ &\quad + (1 - w(y))\left[w(x)f(x_k, y_{l+1}) + (1 - w(x))f(x_{k+1}, y_{l+1})\right],\end{aligned}$$

  where $w(x) = \frac{x_{k+1} - x}{x_{k+1} - x_k}$ and $w(y) = \frac{y_{l+1} - y}{y_{l+1} - y_l}$.

- The command `interp2` in Matlab does the linear interpolation (but slow).