Time Iteration with Euler equation

Takeki Sunakawa

Quantitative Methods for Monetary Economics

University of Mannheim

May 2018

Introduction

- Quantitative methods are required more than before to look at interesting economic problems.
- See also `quantecon.org` by Sargent and Stachurski (2018).

Neoclassical growth model

- We consider an example of neoclassical growth model. An individual maximizes life-time utility

$$\sum_{t=0}^{\infty} \beta^t u(c_t)$$

subject to

$$c_t + k_{t+1} \leq f(k_t).$$

where $u(\cdot)$ and $f(\cdot)$ satisfy standard conditions.

- The first-order necessary condition is given by

$$u_c(c_t) = \beta u_c(c_{t+1}) f_k(k_{t+1})$$

where $u_c(\cdot)$ denotes the derivative of u wrt c and $f_k(\cdot)$ denotes the derivative of f wrt k.

Collman operator

- There is a mapping $\sigma = K\sigma$ that solves

$$u_c(c) = \beta u_c\left(\sigma(f(k) - c)\right) f_k(f(k) - c)$$

for $c = \sigma(k)$. σ is called policy function.

  - Note that $k' = f(k) - c$ and $c' = \sigma(k') = \sigma(f(k) - c)$.
  - Bellman operator

$$V(k) = \max_{c \in (0, f(k)]} \left\{u(c) + \beta V\left(f(k) - c\right)\right\}.$$

    is to solve the Bellman equation, whereas the Collman operator is helpful to solve the Euler equation.

Collman operator, cont'd

- Collman (1990) proves the existence of the fixed point of K in a stochastic neoclassical growth model with distortionary tax.
    - Greenwood and Huffman (1995) extend it to several cases. Also see Richter, Throckmorton and Walker (2014) and Sargent and Stachurski (2018).

Time iteration: Algorithm

- Time iteration is a method to solve the Collman operator.
- The time iteration method takes the following steps:
  1. Make an initial guess for the policy function $\sigma^{(0)}$.
  2. For $i = 1, 2, ...$ (i is an index for the number of iterations), given the policy function previously obtained $\sigma^{(i-1)}$, solve

$$u_c(c) = \beta u_c \left(\sigma^{(i-1)}(f(k) - c) \right) f_k(f(k) - c)$$

     for c.
  3. Update the policy function by setting $c = \sigma^{(i)}(k)$.
  4. Repeat 2-3 until $\left\| \sigma^{(i)} - \sigma^{(i-1)} \right\|$ is small enough.
- How to solve this problem? We will come back later.

Time iteration to solve New Keynesian models

- Time iteration is a popular method to solve nonlinear New Keynesian models.
  - We have to look at the decentralized economy, as the second welfare theorem fails to hold.

Plan

- In today's lecture, we will cover...
  - Equilibrium with a simple Taylor rule and ZLB
  - Tauchen's method approximating AR(1) process
  - Equilibrium under the optimal discretionary policy with ZLB (Adam and Billi, 2007)

Equilibrium with Taylor rule

- Equilibrium conditions are

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

  and the zero lower bound

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

Two-state shock process

- Exogenous shocks take only $N_s = 2$ values, $s_t \in \{s_H, s_L\}$. The stochastic process follows a Markov chain with the transition matrix:

$$\begin{bmatrix} 1 - p_H & p_H \\ 1 - p_L & p_L \end{bmatrix}.$$

- p_H is the frequency of crisis and p_L is the duration of crisis.

Two routes

- The solution has a form of (we omit time subscripts for the policy function)

$$y = \sigma_y(s), \quad \pi = \sigma_\pi(s), \quad r^n = \sigma_{r^n}(s).$$

- We are solving the same model with two different methods.
  - Analytical solution.
  - Numerical solution, using the policy function iteration (time iteration).

Analytical solution

- We know that the functions have only two values, i.e.,

$$y = \begin{cases} y_H, \\ y_L, \end{cases} \qquad \pi = \begin{cases} \pi_H, \\ \pi_L, \end{cases} \qquad r^n = \begin{cases} r^n_H, \\ r^n_L. \end{cases}$$

Analytical solution, cont'd

- We assume that $r_H^n > 0$ and $r_L^n = 0$. Then we have

$$
\begin{aligned}
y_H &= (1-p_H)y_H + p_H y_L - (r_H^n - [(1-p_H)\pi_H + p_H\pi_L] - s_H)\,, \\
\pi_H &= \kappa y_H + \beta\left[(1-p_H)\pi_H + p_H\pi_L\right], \\
r_H^n &= r^* + \phi_\pi\left[(1-p_H)\pi_H + p_H\pi_L\right], \\
y_L &= (1-p_L)y_H + p_L y_L - (0 - [(1-p_L)\pi_H + p_L\pi_L] - s_L)\,, \\
\pi_L &= \kappa y_L + \beta\left[(1-p_L)\pi_H + p_L\pi_L\right], \\
r_L^n &= 0.
\end{aligned}
$$

- There are 6 equations and 6 unknowns, so we can solve for the unknowns.

Collman operator

- We can solve the same model with the time iteration method. The mapping $\sigma = K\sigma$ solves

$$y = \sum_{s'} p_{s'|s}\sigma_y(s') - \left(r^n - \sum_{s'} p_{s'|s}\sigma_\pi(s') - s \right),$$

$$\pi = \kappa y + \beta \sum_{s'} p_{s'|s}\pi(s'),$$

$$r^n = \max\left\{ 0, \bar{r} + \phi_\pi \sum_{s'} p_{s'|s}\sigma_\pi(s') \right\},$$

for $y = \sigma_y(s)$ and $\pi = \sigma_\pi(s)$.

Time iteration

- Recap: The time iteration method takes the following steps
  1. Make an initial guess for the policy function $\sigma^{(0)}$.
  2. For $i = 1, 2, ...$ (i is an index for the number of iterations), given the policy function previously obtained $\sigma^{(i-1)}$, solve for the endogenous variables y and π at each grid.
  3. Update the policy function by setting $y = \sigma_y^{(i)}(s)$ and $\pi = \sigma_\pi^{(i)}(s)$.
  4. Repeat 2-3 until $\left\| \sigma^{(i)} - \sigma^{(i-1)} \right\|$ is small enough.

Time iteration: Initial guess

- A guess of the policy functions

$$y = \sigma_y^{(0)}(s), \quad \pi = \sigma_\pi^{(0)}(s).$$

- Consider the general case of $N_s \geq 2$. We know the values of the functions only at each *grid point*, i.e.,

$$\sigma_\pi^{(0)}(s) = [y_1, y_2, ..., y_{N_s}]',$$
$$\sigma_\pi^{(0)}(s) = [\pi_1, \pi_2, ..., \pi_{N_s}]'.$$

Time iteration: Solving

- Given the policy function previously obtained $\sigma^{(i-1)}$, at each grid point $j = 1, ..., N_s$, we solve

$$
\begin{aligned}
y_j &= y^e - \left(r_j^n - \pi^e - s_j\right), \\
\pi_j &= \kappa y_j + \beta \pi^e, \\
r_j^n &= \max\left\{0, \bar{r} + \phi_\pi \pi^e\right\},
\end{aligned}
$$

for (y_j, π_j, r_j^n), where

$$
\begin{aligned}
y^e &= \sum_{l=1}^{N_s} p(j,k) \sigma_y^{(i-1)}(s_k), \\
\pi^e &= \sum_{l=1}^{N_s} p(j,k) \sigma_\pi^{(i-1)}(s_k).
\end{aligned}
$$

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

Time iteration: Updating

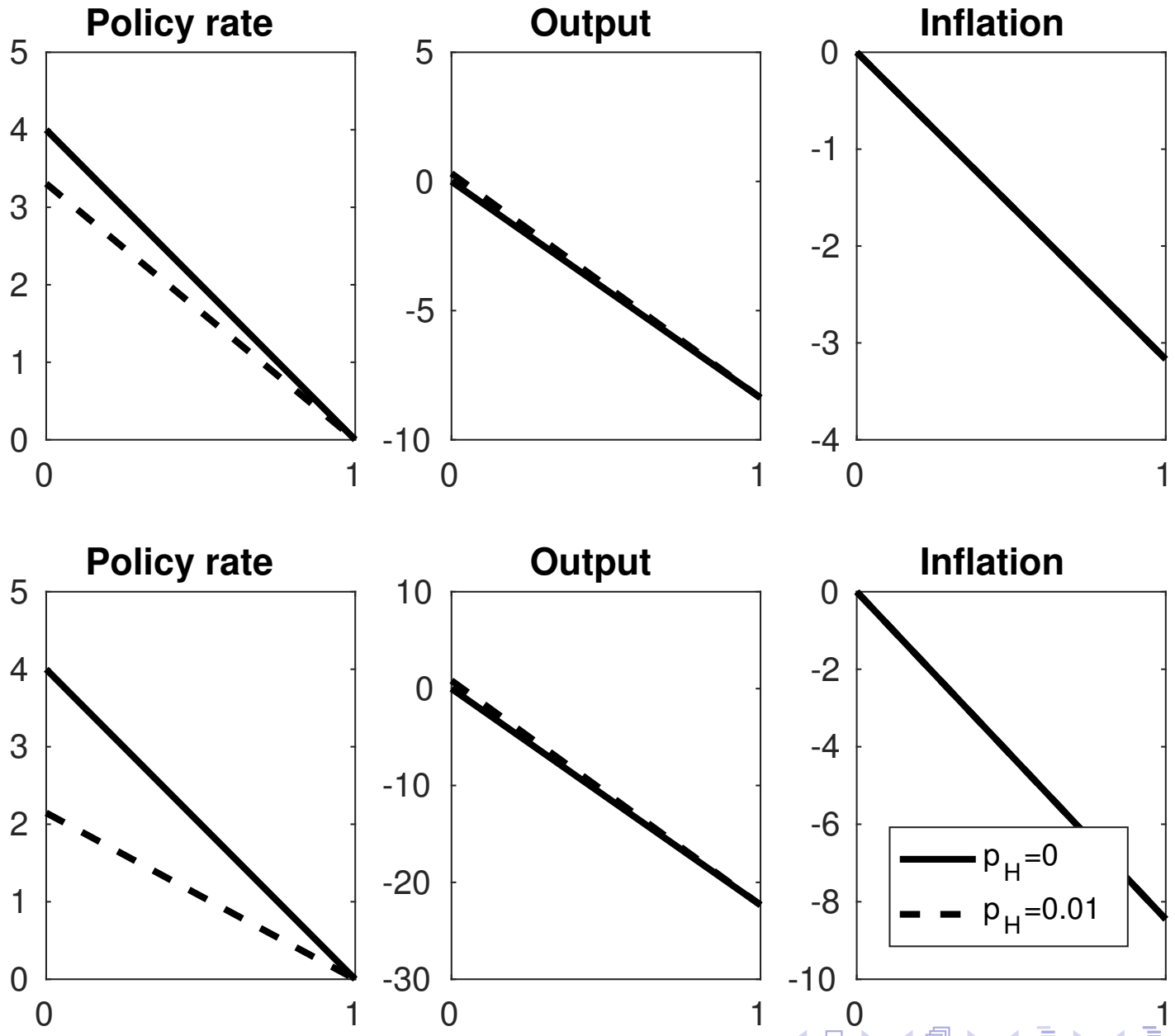
- Once this is done for all the grid points, we update

$$\sigma_y^{(i)}(s) = [y_1, y_2, ..., y_{N_s}]',$$
$$\sigma_\pi^{(i)}(s) = [\pi_1, \pi_2, ..., \pi_{N_s}]'.$$

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

Numerical examples

- Upper panels are for $s_L = \bar{r} - 2.5$ and lower panels are for $s_L = \bar{r} - 5.0$.



Tauchen's method

- Tauchen (1986) developed a method for approximating AR(1) stochastic process by using Markov chain.
- We have the following AR(1) stochastic process

$$x' = c + \rho x + \varepsilon', \varepsilon' \sim N(0, \sigma_\varepsilon^2).$$

- We want to approximate the stochastic process by a Markov chain $x_k \in \{x_1, x_2, ..., x_N\}$.

Tauchen's method: Grid points

- We set the grid points for x:

$$x_k \in \mathcal{I} = \{x_1, x_2, ..., x_N\},$$

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

- For example, we set $x_1 = \frac{-m\sigma_\varepsilon}{\sqrt{1-\rho^2}}$, $x_N = \frac{m\sigma_\varepsilon}{\sqrt{1-\rho^2}}$ and $x_k = x_{k-1} + w$ for $k = 2, ..., N-1$, where $w = \frac{x_N - x_1}{N-1}$.

Tauchen's method: Transition matrix

- Given the grid points for x. What is the probablity of moving from one point x_j to another x_k?
- We know

$$\varepsilon' = x' - c - \rho x_j \sim N(0, \sigma_\varepsilon^2).$$

Tauchen's method: Transition matrix, cont'd

- Then, the probability of $x' \in \left[x_k - \frac{w}{2}, x_k + \frac{w}{2}\right]$ can be used as approximation. That is,

$$p_{kl} = \Phi\left(x_k + \frac{w}{2} - c - \rho x_j\right) - \Phi\left(x_k - \frac{w}{2} - c - \rho x_j\right),$$

  where $\Phi(\cdot)$ is the cdf of $N(0, \sigma_\varepsilon^2)$. Be careful at the boundary points.

- Once this is done for all j, k, we have the transition matrix

$$P = \begin{bmatrix} p_{11} & \cdots & \cdots & p_{1N} \\ p_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ p_{N1} & \cdots & \cdots & p_{NN} \end{bmatrix}.$$

Optimal Discretionary Policy

- The policymaker chooses $\{\pi_t, y_t, r_t^n\}$ 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

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

taking $E_t y_{t+1}$ and $E_t \pi_{t+1}$ as given.

Shock processes

- Exogenous shocks are given by

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

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

- Note that $g = r^*$.

Lagrangean

- We know that Markov-perfect equilibrium has only natural state variables.
- Lagrangean is

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

- First-order necessary conditions are

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

Complementary slackness

- Complementary slackness condition:

$$\phi_{ZLB,t} > 0 \perp r_t^n > 0.$$

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

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

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

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

Solving the model

- The solution has a form of

$$y = y(g, u), \quad \pi = \pi(g, u), \quad r^n = r^n(g, u).$$

- Now consider the case in which there are only two-state g shocks. We know that the functions have only two values, i.e.,

$$y = \begin{cases} y_H, \\ y_L, \end{cases} \qquad \pi = \begin{cases} \pi_H, \\ \pi_L, \end{cases} \qquad r^n = \begin{cases} r^n_H, \\ r^n_L. \end{cases}$$

- Again, we will look at analytical solution and numerical solution.

Analytical solution

- We assume that $r_H^n > 0$ and $r_L^n = 0$. Then we have

$$
\begin{aligned}
y_H &= (1-p_H)y_H + p_H y_L - (r_H^n - [(1-p_H)\pi_H + p_H \pi_L]) + g_H, \\
\pi_H &= \kappa y_H + \beta \left[(1-p_H)\pi_H + p_H \pi_L\right], \\
0 &= \lambda y_H + \kappa \pi_H, \\
y_L &= (1-p_L)y_H + p_L y_L - (0 - [(1-p_L)\pi_H + p_L \pi_L]) + g_L, \\
\pi_L &= \kappa y_L + \beta \left[(1-p_L)\pi_H + p_L \pi_L\right], \\
\phi_L &= \lambda y_L + \kappa \pi_L.
\end{aligned}
$$

- There are 6 equations and 6 unknowns, so we can solve for the unknowns.

Joint shock process

- Let's get back to the general case. The shock processes are appoximated by Markov chains. That is,

$$g_m \in \{g_1, g_2, ..., g_{N_g}\},$$
$$u_n \in \{u_1, u_2, ..., u_{N_u}\},$$

and

$$P^g = \begin{bmatrix} p^g_{11} & \cdots & \cdots & p^g_{1N_g} \\ p^g_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ p^g_{N_g 1} & \cdots & \cdots & p^g_{N_g N_g} \end{bmatrix}, \quad P^u = \begin{bmatrix} p^u_{11} & \cdots & \cdots & p^u_{1N_g} \\ p^u_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ p^u_{N_g 1} & \cdots & \cdots & p^u_{N_g N_g} \end{bmatrix}.$$

Joint shock process, cont'd

- 2D grid coordinate with g and u represents the joint shock process.
  - For example, when $N_g = N_u = 2$

$$
\begin{aligned}
s_1 &= (g_1, u_1), \\
s_2 &= (g_1, u_2), \\
s_3 &= (g_2, u_1), \\
s_4 &= (g_2, u_2).
\end{aligned}
$$

  - Note that each index points to a pair of shocks, $s_j = (g_{m(j)}, u_{n(j)})$.
- A kronecker product of the transition matrices $P^s = P^g \otimes P^u$ is the transition matrix of the joint shock process.

Policy function iteration: Initial guess

- A guess of the policy functions

$$y = y^{(0)}(s), \quad \pi = \pi^{(0)}(s).$$

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

$$y^{(0)}(s) = [y_1, y_2, ..., y_{N_s}]',$$
$$\pi^{(0)}(s) = [\pi_1, \pi_2, ..., \pi_{N_s}]'.$$

Policy function iteration: Solving

- Given the policy function previously obtained $\sigma^{(i-1)}$, at each grid point $j = 1, ..., N$, we solve

$$\pi_j = \kappa y_j + \beta \pi^e + u_{n(j)},$$
$$0 = \lambda y_j + \kappa \pi_j,$$

for (y_j, π_j), where

$$y^e = \sum_{k=1}^{N_s} P^s(j,k) y^{(i-1)}(s_k),$$
$$\pi^e = \sum_{k=1}^{N_s} P^s(j,k) \pi^{(i-1)}(s_k).$$

Policy function iteration: Solving, cont'd

- Check $r_j^n = -y_j + y^e + \pi^e + g_{m(j)} \geq 0$. If not, we solve instead

$$0 = -y_j + y^e + \pi^e + g_{m(j)},$$
$$\pi_j = \kappa y_j + \beta \pi^e + u_{n(j)},$$

  for (y_j, π_j), and set $r_j^n = 0$.

Policy function iteration: Updating

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

$$y^{(i)}(s) = [y_1, y_2, ..., y_{N_s}]',$$
$$\pi^{(i)}(s) = [\pi_1, \pi_2, ..., \pi_{N_s}]'.$$

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

Figure 4 in Adam and Billi (2007)

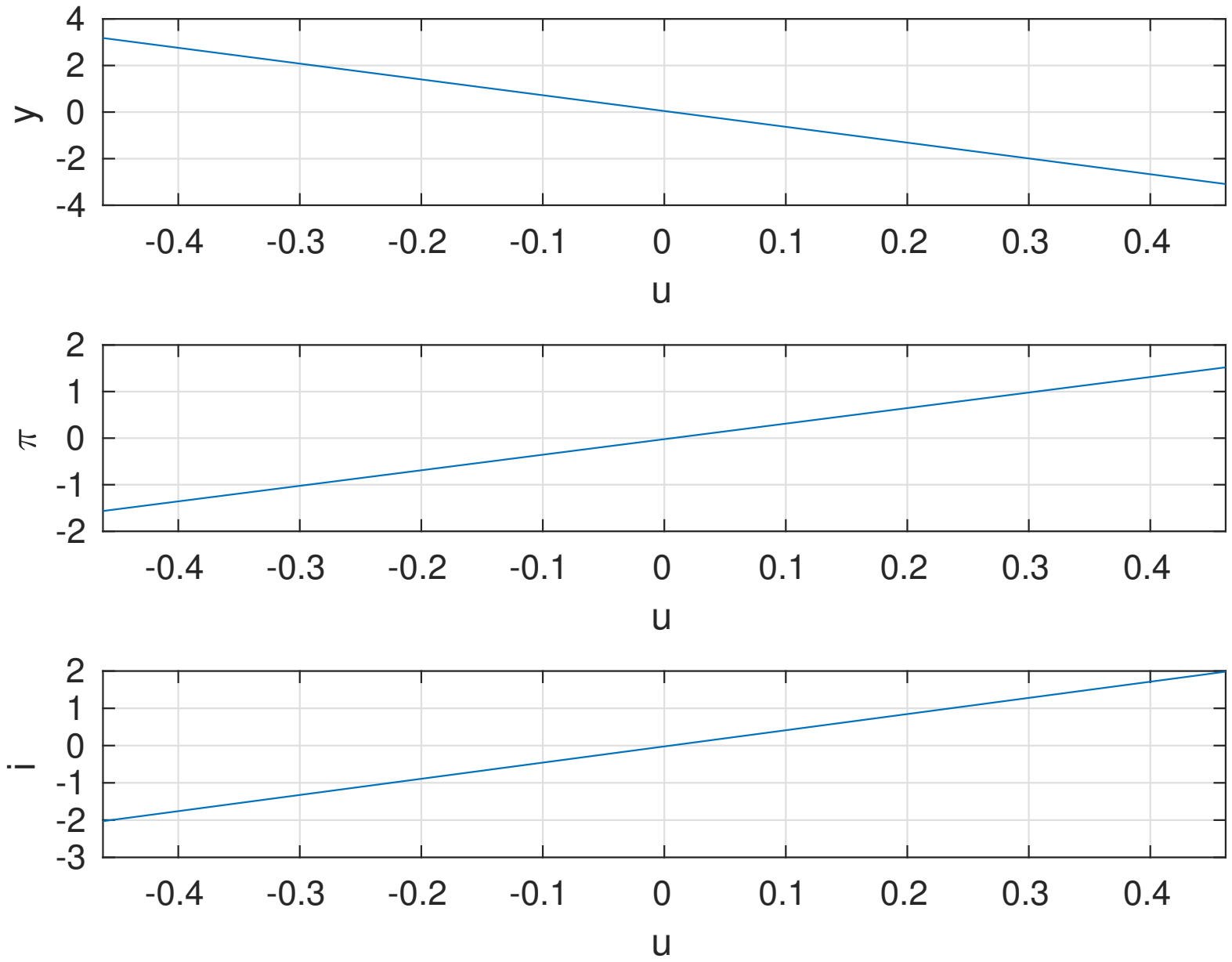


Figure 5 in Adam and Billi (2007)