High School Students' DSGE Modelling With Dynare

Weijie Chen[1]

Department of Political and Economic Studies

University of Helsinki

April 1, 2012

[1]Any questions, please send email to weijie.chen@helsinki.fi.

Abstract

This the simplest DSGE model ever, even a high school student can understand what this is. The purpose of this note is to illustrate a two-equation linear rational expectation model and how to employ Dynare to play with it. If you do not have faith on climbing the mountain of DSGE yet, this shall be your first step. We will work through solution and Bayesian estimation in this note.

1 Simplest DSGE Model

We only have two endogenous variables in this linear DSGE (LRE) model, y and x. And we will start with a single exogenous variable first, u.

We specify an agent behaviour rule,

$$y_t = \beta E_t y_{t+1} + x_t \tag{1}$$

and an exogenous Markov process to describe the behaviour of nature,

$$x_t = \rho x_{t-1} + \varepsilon_t \qquad \varepsilon_t \sim iid(0, \sigma^2), \quad |\rho| < 1 \tag{2}$$

y_t is forward-looking variable and x_t backward-looking variable. *The basic idea of solving* (1) *is to find a difference equation to describe the dynamic of* y_t *by its own history and its shocks* ε.[1] x_t is already fully described by its history and nature, namely x_{t-1} and ε, thus we don't need to worry about x_t.

We solve y_t forwards, move one period forwards,

$$y_{t+1} = \beta E_{t+1} y_{t+2} + x_{t+1}$$

Take condition expectation,

$$E_t y_{t+1} = \beta E_t[E_{t+1} y_{t+2}] + E_t x_{t+1} = \beta E_t y_{t+2} + E_t x_{t+1} \tag{3}$$

where the second equation makes use of iterated law of expectation. Substitute back to (1),

$$y_t = \beta(\beta E_t y_{t+2} + E_t x_{t+1}) + x_t = \beta^2 E_t y_{t+2} + \beta E_t x_{t+1} + x_t \tag{4}$$

Then move (1) forwards two periods,

$$y_{t+2} = \beta E_{t+2} y_{t+3} + x_{t+2}$$

[1] This is the core idea of solving DSGE model.

Take conditional expectation,

$$E_t y_{t+2} = \beta E_t[E_{t+2} y_{t+3}] + E_t x_{t+2} = \beta E_t y_{t+3} + E_t x_{t+2}$$

Substitute into (3),

$$\begin{aligned} E_t y_{t+1} &= \beta^2(\beta E_t y_{t+3} + E_t x_{t+2}) + \beta E_t x_{t+1} + x_t \\ &= \beta^3 E_t y_{t+3} + \beta^2 E_t x_{t+2} + \beta E_t x_{t+1} + x_t \end{aligned}$$

Perform recursive substitution infinite times, we get

$$y_t = E_t \sum_{i=0}^{\infty} \beta^i x_{t+i} \tag{5}$$

where we have used the transversality condition

$$\lim_{i\to\infty} E_t y_{t+i} = 0$$

We find the time path of y_t described by x, we need to take a deeper look into x_t, move (2) one period forwards and take conditional expectation,

$$E_t x_{t+1} = E_t[\rho x_t + \varepsilon_t] = \rho x_t$$

Move two periods and take conditional expectation,

$$E_t x_{t+2} = E_t[\rho x_{t+1} + \varepsilon_{t+2}] = \rho E_t x_{t+1} = \rho^2 x_t$$

Then we can deduct that

$$E_t x_{t+i} = \rho^i x_t$$

Substitute to (5),

$$y_t = \sum_{i=0}^{\infty} \beta^i E_t x_{t+i} = \sum_{i=0}^{\infty} \beta^i \rho^i x_t = x_t \sum_{i=0}^{\infty} \beta^i \rho^i$$

where the third equation holds because x_t has no i anymore. If we make a key assumption here $|\beta| < 1$, we can use geometric series rewrite the last equation,

$$y_t = x_t \sum_{i=0}^{\infty} \beta^i \rho^i = \frac{x_t}{1 - \beta\rho} \tag{6}$$

However, we still need to replace x_t by ε's, we come back to (**??**),

$$
\begin{aligned}
x_t - \rho x_{t-1} &= \varepsilon_t \\
(1 - \rho L) x_t &= \varepsilon_t \\
x_t &= \frac{\varepsilon_t}{1 - \rho L}
\end{aligned} \tag{7}
$$

Substitute (7) back to (6),

$$
\begin{aligned}
y_t &= \frac{\varepsilon_t}{1 - \rho L} \frac{1}{1 - \beta \rho} \\
y_t - \rho y_{t-1} &= \frac{1}{1 - \beta \rho} \varepsilon_t \\
y_t &= \rho y_{t-1} + \frac{1}{1 - \beta \rho} \varepsilon_t
\end{aligned} \tag{8}
$$

(8) is the final solution of first equation. Together with (2),

$$
\begin{aligned}
y_t &= \rho y_{t-1} + \frac{1}{1 - \beta \rho} \varepsilon_t \\
x_t &= \rho x_{t-1} + \varepsilon_t
\end{aligned}
$$

In vector form,

$$
\begin{bmatrix} y_t \\ x_t \end{bmatrix} = \rho \begin{bmatrix} y_{t-1} \\ x_{t-1} \end{bmatrix} + \varepsilon_t \begin{bmatrix} \frac{1}{1-\beta\rho} \\ 1 \end{bmatrix} \tag{9}
$$

(9) is the so-called *policy and transition function*, which describes the dynamics of the system on a saddle-path.

However, since the system can be collapsed into one equation, the policy function can simply be (8) too.

2 Dynare Code and Results

Here is the Dynare code for the simplest DSGE model

```
var x y;
varexo e;
```

```
parameters rho beta;
beta=0.9;
rho=0.9;

model(linear);
y=beta*y(+1)+x;
x=rho*x(-1)+e;
end;

initval;
x = 0;
y = 0;
end;

steady;
check;
resid;

shocks;
var e;  stderr 0.1;
end;

stoch_simul(periods=900,order=1,irf=40,drop=400,aim_solver) y x;
```

And here is impulse response function with $\rho = 0.9$ and $\beta = 0.9$ 1,

Recalibrate with $\rho = 0.2$ and $\beta = 0.2$ 2 we can see that parameters determines the persistence and amplification of the model economy, the parameters summerise the core behavioural features of economy. From the first graph, we can see the 1% increse in e will push x up 10% and y by 52%.

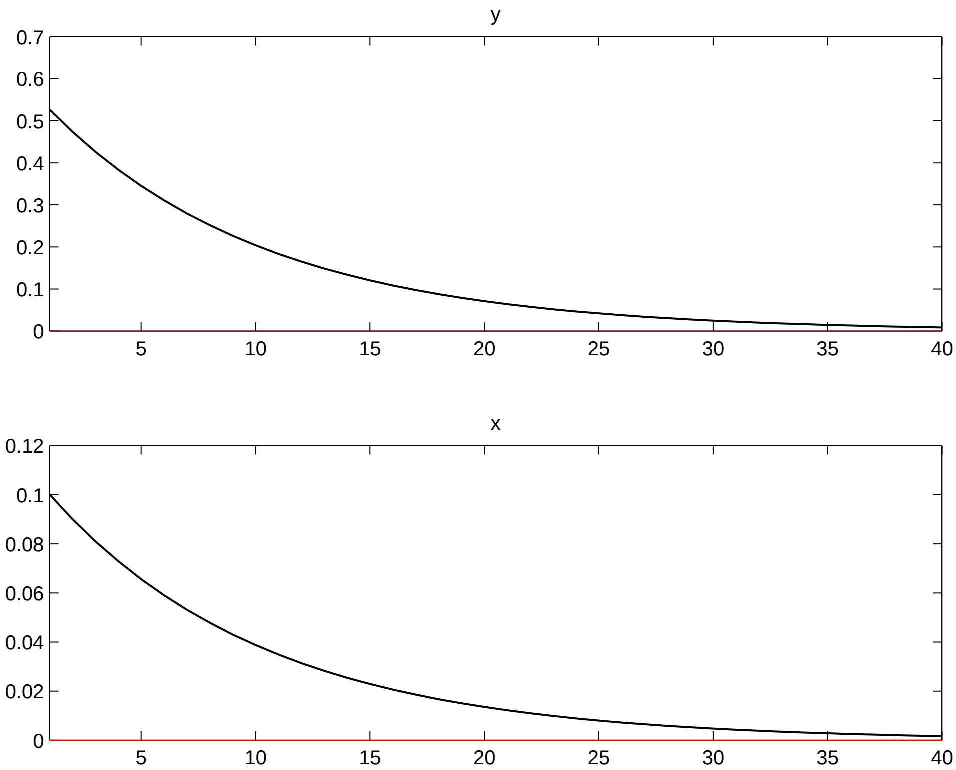

Figure 1: Impulse response of e

We come back to transition and policy function, the impulse response function at period t will take the form as following, first the state variable x_t

$$x_t = \rho \cdot 0 + 0.1 = 0.1$$

then to endogenous variable y_t,

$$y_t = \rho \cdot 0 + 0.1 \times \frac{0.1}{1 - 0.9 \times 0.9} = 0.5263$$

The impulse response function for $t + 1$, state variable

$$x_{t+1} = \rho x_t + 0 = 0.9 \times 0.1 = 0.09$$

and endogenous variable y_{t+1}

$$y_{t+1} = \rho y_t + 0 = 0.9 \times 0.5263 = 0.4737$$

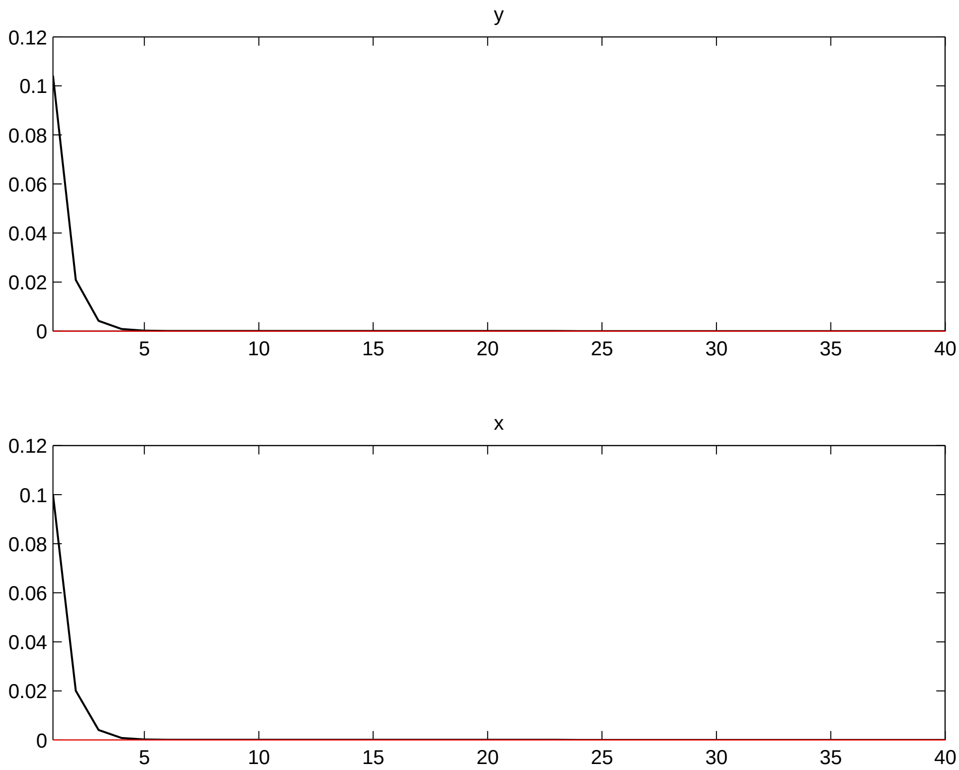

Figure 2: Impulse response of e

Following this pattern you can see that IRF can be represented by

$$x_{t+i} = \rho^i x_t$$

$$y_{t+i} = \rho^i y_t$$

And we can also plot the simulated data 3.

3 Maximum Likelihood Estimation

MLE is a full information estimation method, but we won't talk about any theoretical ideas here. First we need to add another shock u to the system in order to initialise the Kalman filter which will simulate the likelihood

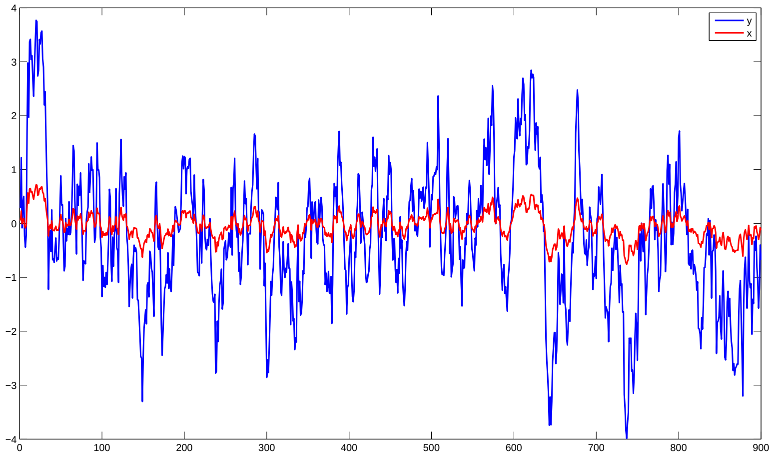

Figure 3: Simulated data

function. Thus our system will be

$$y_t = \beta E_t y_{t+1} + x_t + u_t \qquad u_t \sim N(0, \sigma_u^2)$$
$$x_t = \rho x_{t-1} + e_t \qquad \varepsilon_t \sim N(0, \sigma_\varepsilon^2)$$

Then we add these codes behind

```
save Simul_data y x;


varobs y x;


estimated_params;
beta, 0.8, 0.01, 0.99;
rho, 0.8, 0.01, 0.99;
end;
```

which specifies the parameters we intend to estimate, namely ρ and β.[2] 0.8 is our initial value setting for estimated parameters, rest of two figures are lower and upper bound.

[2] For code details refer to Dynare User Guide and Dynare Manual.

Then we estimate with code

```
estimation(datafile=Simul_data, mode_check,
           prefilter=1, order=1, mode_compute=1) y x;
```

The estimation machanism is beyond the scope of this note, for interested readers, please study through *Structural Macroeconometrics* by Dejong.

Here is the log-likelihood function and its mode 4, as you can see from

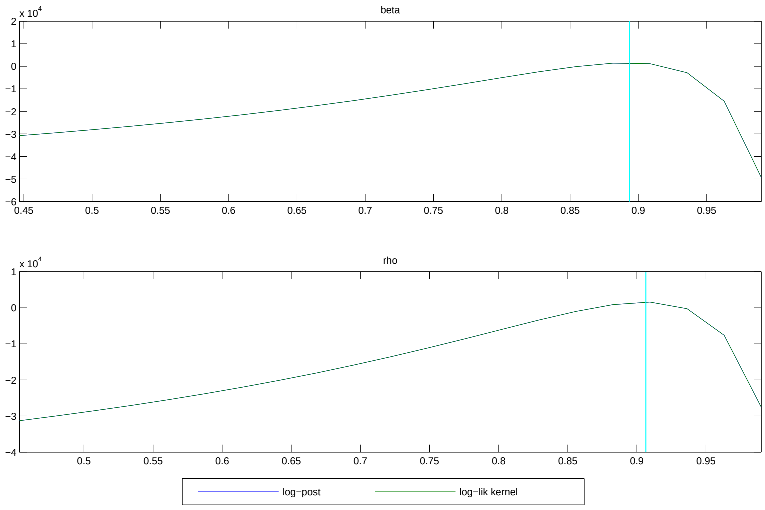

Figure 4: Log-likelihood function

both plots, the estimated mode is very close to 0.9 as we previous set, which completely reflects the basic idea of Monte Carlo experiment we are implementing here. This MLE is a just-identified case, next we will add more parameters to render the underidentificiation.

We modify the code block with two more parameters to estiamte σ_ε and σ_u

```
estimated_params;
beta, 0.8, 0.01, 0.99;
rho, 0.8, 0.01, 0.99;
stderr e, 0.1;
```

```
stderr u, 0.1;
end;
```

Next we can see that Matlab is optimising the log-likelihood function by using *Newton-Raphson method* iteratively, eighty-six times.

```
                                    Max     Line search  Directional  First-order
 Iter F-count        f(x)   constraint   steplength   derivative   optimality Procedure
    0      5      14016.7        -0.19
    1     14      13635.3      -0.1781       0.0625    -3.12e+005    7.79e+004
    2     19      11180.6   -8.674e-018           1        -29.1          440
    3     39      10009.7   -8.674e-018   3.05e-005         -440    2.86e+004  Hessian modified
    4     75      9861.87   -8.674e-018   4.66e-010    -1.41e+003   6.81e+010  Hessian modified twice
    5    113       9770.5   -8.674e-018   1.16e-010    -1.47e+003   5.18e+012  Hessian modified twice
    6    118       9753.7            0            1         -641    7.34e+003
    7    125      9685.34    -0.004278         0.25         -617          803
    8    130      9657.78      -0.0227            1         -511          920
    9    136      9533.74     -0.04105          0.5         -613    1.12e+004
   10    141      9101.15     -0.03298            1     -2.9e+003   1.15e+003
   11    146      9026.68            0            1         -968      9.2e+003
   12    151      8949.85     -0.02912            1    -2.14e+003   1.04e+003
   13    156      8925.06     -0.02855            1         -490          865
   14    161      8918.35     -0.02853            1         -265         98.6
   15    166      8917.58     -0.02867            1        -96.2          146
   16    171      8917.54     -0.02877            1        -17.1         65.5
   17    176      8917.52     -0.02888            1         -4.5         1.14
   18    181      8917.52     -0.02888            1       -0.119        0.961
   19    186      8917.52     -0.02889            1      -0.0489         1.01  Hessian modified
   20    191      8917.52     -0.02892            1      -0.0385         2.05  Hessian modified
   21    196      8917.52     -0.02899            1       -0.034         4.23  Hessian modified
   22    201      8917.51     -0.02917            1      -0.0317         7.74  Hessian modified
   23    206       8917.5     -0.02964            1      -0.0304         13.4
   24    211      8917.47     -0.03086            1      -0.0297         22.7
   25    216      8917.37     -0.03406            1      -0.0292         37.5
   26    221      8917.14     -0.04246            1       -0.029         61.7
   27    226      8916.51     -0.06445            1      -0.0288          101
   28    231      8914.86      -0.1155            1      -0.0287          164
   29    236      8910.51       -0.136            1      -0.0286          266
   30    241       8898.8      -0.1712            1      -0.0284          432
   31    246      8883.37   -8.674e-018           1       -0.042          448
   32    251      8869.43            0            1        -6.08    1.03e+003  Hessian modified
   33    256      8865.71            0            1        -3.49       1e+003
```

34	261	8862.36	0	1	-3.65	538	
35	266	8861.54	0	1	-4.76	165	
36	271	8861.34	0	1	-1.31	60.5	
37	276	8861.31	0	1	-0.0602	60.8	
38	281	8861.04	0	1	-0.0502	61	
39	286	8860.53	0	1	-0.0382	61	
40	291	8858.99	0	1	-0.0354	122	
41	296	8855.14	0	1	-0.033	214	
42	301	8844.69	0	1	-0.0318	365	
43	306	8815.44	0	1	-0.0308	602	
44	311	8699.09	0	1	-0.0297	984	
45	316	8648.74	0	1	-0.0388	1.6e+003	Hessian modified
46	321	8494.19	0	1	-0.183	935	
47	326	8112.33	0	1	-0.071	944	
48	333	6318.63	0	0.25	-0.147	6.62e+003	Hessian modified
49	340	6225.19	0	0.25	-1.51	530	
50	346	5539.79	-0.01626	0.5	-2.07	8.87e+003	
51	353	5048.06	-0.01219	0.25	-1.83	3.27e+004	
52	367	4215.27	-0.01217	0.00195	-3.44	5.7e+004	Hessian modified
53	379	3268.6	-0.01973	0.00781	-18.1	7e+004	Hessian modified
54	385	2870.01	-0.09381	0.5	-87.6	5.93e+004	
55	393	2807.01	-0.2046	0.125	-752	5.15e+004	
56	408	2209.46	-0.2044	0.000977	-111	4.46e+004	
57	413	1311.22	0	1	-159	3.15e+004	
58	420	675.729	-0.1431	0.25	-389	2.99e+004	
59	429	550.541	-0.1535	0.0625	-524	2.87e+004	Hessian modified
60	434	-59.13	0	1	-2.41e+003	1.81e+004	
61	441	-182.125	0	0.25	-4.1e+003	1.44e+004	
62	446	-536.142	0	1	-1.73e+003	1.04e+004	
63	457	-1038.48	-0.01531	0.0156	-2.17e+003	1.87e+004	Hessian modified
64	464	-1220.03	-0.0683	0.25	-2.23e+003	2.28e+004	
65	473	-1245.34	-0.06403	0.0625	-3.46e+003	3.1e+004	
66	480	-1362.74	-0.08475	0.25	-1.29e+004	2.35e+004	
67	486	-1472.13	-0.04603	0.5	-1.25e+004	9.55e+003	
68	492	-1492.98	-0.06875	0.5	-1.09e+003	3.1e+003	
69	502	-1538.59	-0.06726	0.0313	-3.23e+003	2.48e+004	
70	507	-1559.46	-0.07776	1	-3.63e+003	2.72e+003	
71	512	-1572.43	-0.08666	1	-1.12e+003	2.63e+003	
72	517	-1574.41	-0.07284	1	-445	3.47e+003	
73	523	-1576.46	-0.08842	0.5	-256	2.34e+003	
74	528	-1577.56	-0.0881	1	-899	2.34e+003	
75	533	-1579.4	-0.08907	1	-808	2.22e+003	

```
76     538     -1581.09     -0.08968             1          -766    1.95e+003
77     543     -1583.72     -0.08807             1    -1.01e+003    1.43e+003
78     548     -1587.46     -0.08661             1    -1.27e+003    1.27e+003
79     553     -1589.02     -0.08655             1          -835          724
80     558      -1589.2      -0.0861             1          -270          290
81     563     -1589.23     -0.08537             1         -33.6          113
82     568     -1589.23     -0.08442             1         -10.7         16.3
83     573     -1589.24     -0.08371             1         -4.43         9.63
84     578     -1589.24     -0.08345             1         -2.16         2.52
85     583     -1589.24     -0.08343             1         -1.98        0.543  Hessian modified
86     588     -1589.24     -0.08344             1        -0.182        0.139  Hessian modified
```

It is not a fancy technique once you have studied the theory behind it. However, we get it a notice from Matlab

```
POSTERIOR KERNEL OPTIMIZATION PROBLEM!
 (minus) the hessian matrix at the "mode" is not positive definite!
=> posterior variance of the estimated parameters are not positive.
You should  try  to change the initial values of the parameters using
the estimated_params_init block, or use another optimization routine.
Warning: The results below are most likely wrong!
```

Just pay attention to last sentence, we know what we are doing above might be completely wrong, even if we can get a nice graph 5 (you will see the strangeness of this graph soon). This is the famous *indentification problem*, which renders a negative definite Hessian matrix.

The most ridiculous part comes, here we are

```
RESULTS FROM MAXIMUM LIKELIHOOD
parameters
    Estimate     s.d. t-stat
beta    0.8935  0.0000  0.0000
rho     0.9066  0.0000  0.0000


standard deviation of shocks
```

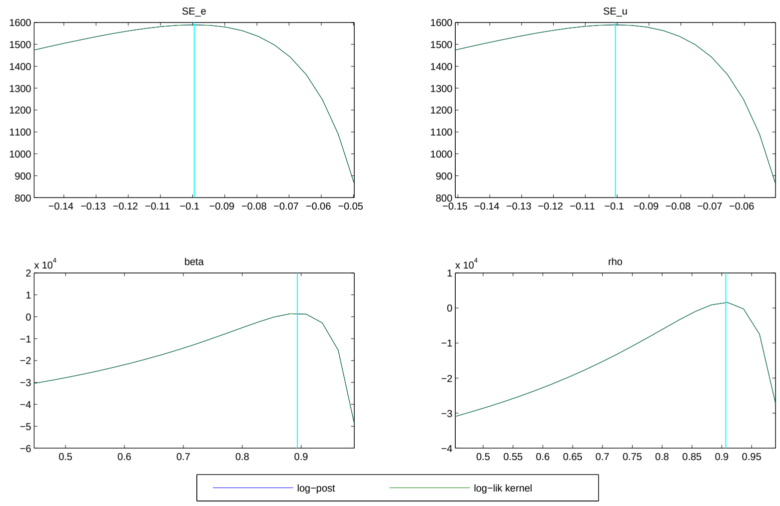

Figure 5: Mode Plot Check

```
     Estimate    s.d. t-stat
e  -0.0995  0.0023 42.4270
u  -0.1005  0.0024 42.4266
```

For parameters β and ρ, we have quite precise estimated value 0.89 and 0.91, however the t-statistic seems very strange, very close to 0. And for standard deviations of shocks σ_ε and σ_u are *negative* numbers, which is impossible. And more funny is that they even have highly significant t-statistics. However, you can notice easily, the estimated standard errors of shocks are simply reversed sign.

4 Bayesian Estimation

Now come to the Bayesian estimation, again, no theories will be discussed here. In order not to waste time on computer, we only estimate one parameter with very short Markov chain.

```
estimated_params;
rho,  normal_pdf, 0.7, 0.2;
```

```
end;

estimation(datafile=Simul_data,prefilter=1,
           mh_replic=500,mh_nblocks=2,mh_jscale=0.4,mode_compute=1,
           order=1,bayesian_irf,irf=30,mode_check) y x;
```

The details of the codes can be seen on the reference manual. Here we simply demean the data, use two parallel Markov chain with 500 replication, and scale the jumping distribution down to 40%. This is the estimation result

```
ESTIMATION RESULTS
Log data density is 1582.649230.
parameters
      prior mean post. mean   conf. interval  prior     pstdev


rho      0.700     0.9000     0.8990   0.9007  norm      0.2000
```

Posterior mean is exactly 0.9. Since the model is using simulated data, the posterior will look like a spike 6 the gray line on the bottom is normal distribution which is collapse nearly into a straight line, and green dashed line is posterior mode. And here is the Bayesian IRFs 7, Because using simulated model, DGP is exactly we are using here, so the confidence bounds collapse together with IRF makes it look thick line.

Note that here we only provide a parsimonious example, so the results might seen very inappropriate there are lots of functions we have not shown here, reading reference manual would be a good choice.

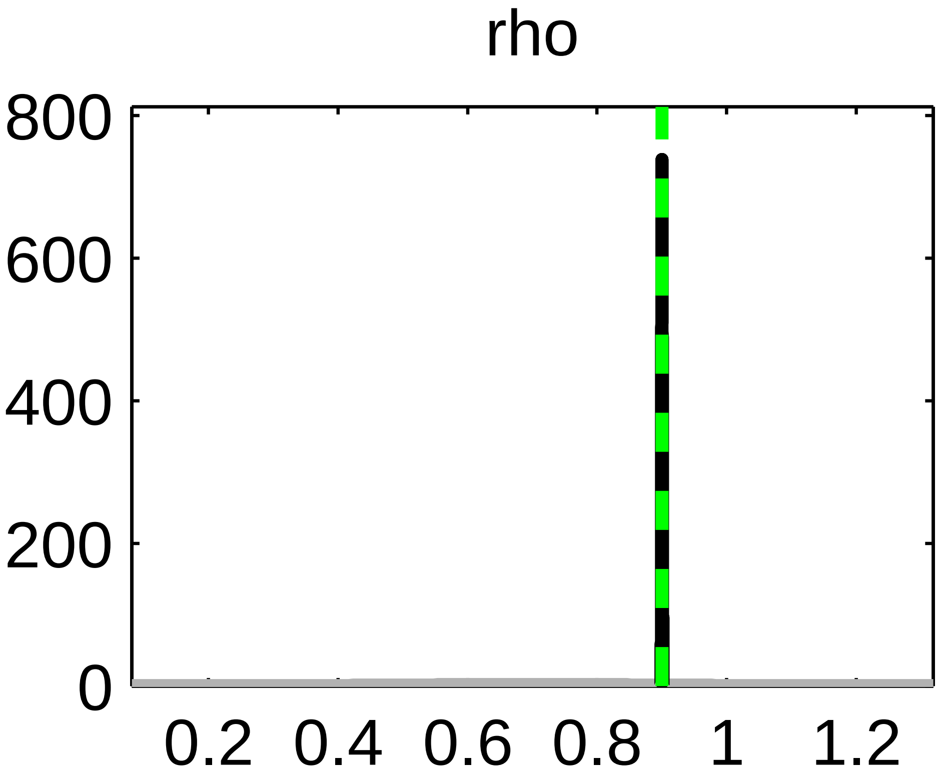

Figure 6: Prior and posterior

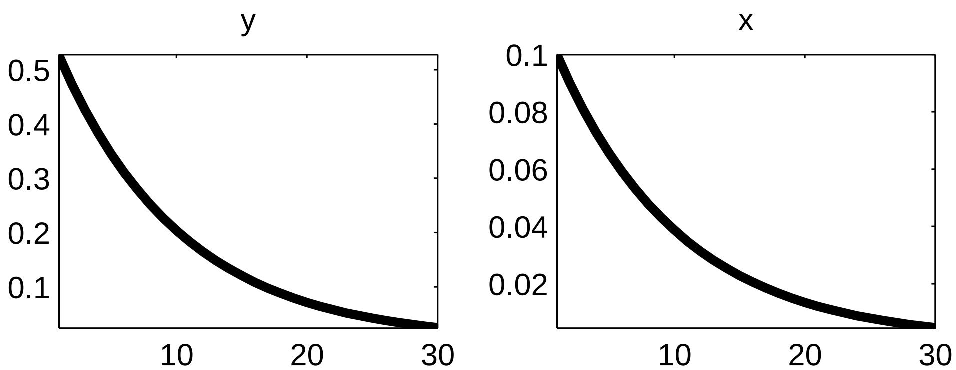

Figure 7: Bayesian IRFs