HW2-ECON-482

Kungang Zhang

November 14, 2016

1 Prob 1:

(1) The SVAR model is:

$$\begin{aligned} A_0 y_t =& A_0 + A_1 y_{t-1} + \cdots + A_p y_{t-p} + \varepsilon \qquad (1)\\ y_t =& B_0 + B_1 y_{t-1} + \cdots + B_p y_{t-p} + A_0^{-1}\varepsilon \qquad (2) \end{aligned}$$

Use the posterior form to the the MAP for β and A_0. The MAP for β is just:

$$\begin{aligned} \hat{B} =& (X^T X + \Omega^{-1})^{-1}(X^T Y + \Omega^{-1}\hat{b})\\ =& (\lambda^2 X^T X + \Phi^{-1})^{-1}(\lambda^2 X^T Y + \Phi^{-1}\hat{b}) \qquad (3) \end{aligned}$$

where I used the notation, $\Omega = \lambda^2\Phi$, and $\hat{\beta} = vec(\hat{B})$. The definition of Φ can be found on the last homework. To get posterior mode of A_0, I need to maximize:

$$p(A_0|Y) \propto |A_0|^{T+M} exp\left\{-\frac{1}{2} tr\left(\left(\hat{S} + \left(\hat{B} - \hat{b}\right)^T \Omega^{-1}\left(\hat{B} - \hat{b}\right)\right) A_0^T A_0\right)\right\} \qquad (4)$$

To maximize it, first take a log and then use global maximization toolbox in Matlab. Denote $\hat{\hat{S}} = \hat{S} + \left(\hat{B} - \hat{b}\right)^T \Omega^{-1}\left(\hat{B} - \hat{b}\right)$. Se the initial guess for a_0 as the upper Cholesky decomposition of $\hat{\hat{S}}^T\hat{\hat{S}}$ with only free variables being chosen so that they can be nonzero. Use multi-starting points to achieve somewhat "global" optimal. By experiments, I found that the impulse response functions w.r.t. the monetary policy shock can be different for 200 and 1000 multi-starts. The optimal A_0 is as shown below (use 2500 multi-starts).

0	1	2	3	4	5
236.59	0	0	0	0	0
3.4657	-808.23	0	0	0	0
-54.187	-19.423	-662.14	0	0	0
-5.6246	-12.024	55.141	54.821	18.665	-23.984
3.5117	142.47	0	0	484.68	29.806
0	0	0	0	-2.4762	200.09

The mode of coefficient B is as shown below.

0	1	2	3	4	5
0.0024995	0.00037705	0.0030223	0.0011985	0.0017546	-0.00029039
1.0432	-0.0031481	-0.092717	0.3159	0.066138	0.03291
-0.12654	1.0749	0.00061063	-0.23366	0.035172	0.26388
-0.18982	-0.010569	0.80563	-0.81992	0.021933	-0.5486
-0.0063155	0.00029585	-0.0090512	1.3156	-0.0068237	0.033789
0.092677	0.019188	-0.028114	-0.008927	1.2851	0.13065
0.0048814	0.011311	-0.0038128	-0.19803	-0.096011	1.1499
-0.056275	-0.0040837	0.048057	-0.16877	-0.071547	0.021358
0.13625	0.034751	0.013832	0.41073	-0.070913	-0.12965
-0.12998	-0.014194	0.16816	0.93464	0.047025	0.3393
0.014615	0.0012958	0.0066163	-0.28352	0.0066023	-0.012538
-0.075994	-0.01108	0.02656	-0.026497	-0.12228	-0.10318
0.0074856	-0.0073815	0.0039201	-0.12667	0.046865	-0.2758
-0.041562	-0.0053412	0.016779	0.04669	-0.00041426	-0.0067541
-0.030863	0.03165	-0.0008748	-0.17896	-0.0059094	-0.081325
0.16551	-0.015384	0.011579	0.26123	-0.03838	0.010615
-0.00064908	0.0013074	0.00094177	-0.036111	0.0021111	-0.0068439
-0.020789	-0.0081374	0.0007174	-0.025154	-0.043231	-0.070443
-0.013694	0.0025127	-0.0050628	0.19096	0.025779	-0.0049773
0.0039848	0.0032588	-0.0085818	-0.10587	0.014897	-0.015206
-0.039694	-0.022046	0.018316	0.097144	0.024261	-0.031873
0.095941	0.01913	0.024149	-0.2112	-0.037107	0.035127
-0.0054951	2.3663e-05	0.0012998	0.03784	-0.0015856	-0.0025607
0.021194	0.0006758	9.9617e-05	0.10944	-0.060999	-0.011888
-0.033608	0.0025541	0.010941	0.060938	0.0076353	-0.0048363
0.02669	-0.0014923	-0.00016873	0.033491	-0.0012554	-0.001907
0.01686	-0.02931	-0.013391	0.011569	0.02389	0.013008
0.066987	0.0040055	-0.024373	-0.06267	0.0048707	0.058487
-0.0019254	6.8794e-05	-0.00026886	0.0080306	0.00068816	-0.0019128
0.012927	0.0025736	0.00039225	0.1049	-0.0090995	0.025653
-0.015289	-0.0012048	0.003849	-0.00051988	0.00048701	0.021305

0.028628	0.00070748	0.0077604	-0.12207	0.0048435	-0.011151
0.021128	-0.015086	-0.017527	0.03596	0.028162	0.0091092
-0.023059	0.006399	-0.0030975	-0.076519	-0.012422	0.026243
-0.0043803	-0.00053571	3.7016e-05	0.0063542	0.00057166	-0.0055767
-0.0020683	0.0038224	0.0057177	0.00045367	-0.013959	0.026865
0.011583	-0.001273	-0.0031394	0.0089924	0.00062275	0.015473
0.0057398	-0.00069748	0.0038036	0.021321	0.0039677	0.0084347
0.022258	-0.021719	-0.0067442	0.024271	0.012037	-0.0032183
0.04671	-0.0012065	-0.010751	0.25872	0.0011429	-0.0085655
0.00093087	-0.00078428	-0.00045688	-0.013151	0.0012252	-0.0017298
-0.0078229	0.00054927	0.003699	-0.080875	-0.014366	-0.018213
0.0038125	0.0013456	0.00056348	0.073796	0.0048307	0.014824
-0.010559	-0.00012706	0.0040818	-0.0083816	-0.0029892	-0.022115
-0.00045657	-0.01232	0.0046271	0.018066	-0.0072089	0.00028188
-0.0013129	0.004909	0.0036231	-0.028976	-0.0040495	-0.041927
-0.00041602	-0.00017436	-0.00026998	-0.011602	-0.00029902	0.0025248
-0.011427	-0.0016969	-0.0049524	-0.026471	0.0029556	4.1121e-05
0.0065581	-0.0010373	-0.001695	-0.0064076	-0.0016373	0.041303
-0.0086535	0.0013565	0.0030704	0.0073006	-0.0058006	-0.010657
0.017364	-0.0077827	-0.0024327	-0.06358	-0.0036475	0.0051014
0.0048232	-0.00095742	0.00081146	-0.08878	0.0051755	-0.039723
0.00047016	-4.499e-05	0.00013497	-0.0069114	-0.0013482	0.00075189
-5.0282e-05	-0.00055836	0.00036769	0.015494	-0.0080114	0.010095
0.012493	0.00075276	0.0016641	-0.003832	0.0016634	0.014683
-0.0012476	0.0038358	0.0041524	0.023466	-0.0041873	0.00065236
-0.0040752	-0.0091117	-0.0017417	-0.0077325	-0.0067299	-0.0045606
0.0122	0.0035822	-0.0062262	-0.04439	0.008934	0.023723
0.0020412	-7.6841e-05	0.00061742	-0.0029142	-0.00091519	-0.00095576
-0.00010134	-0.003147	-0.00033095	0.0028643	-0.0096499	0.0059585
0.0012729	0.00070003	0.0022674	-0.020793	0.0054785	-0.012239
0.0073024	0.0022788	0.003623	0.00031349	-0.0021881	-0.0081993
-0.0011023	-0.0090368	0.00012188	-0.031834	-0.010227	-0.01258
0.0032327	-7.529e-05	0.0068466	-0.040144	-0.0029634	0.030225
0.0011865	-0.00027568	0.00052631	-0.0038476	0.00038754	-0.0019966
-0.0037077	-0.0026218	-0.0021245	-0.012128	-0.0022872	0.0023097
-0.002891	0.0001392	0.00032933	-0.018292	0.0023756	-0.0062968
0.0016856	0.00076718	0.0046278	-0.023929	0.00068119	0.00086905
-0.0063349	-0.0068268	0.0023338	-0.043819	-0.0084584	-0.01439
0.0024146	-0.0017194	-0.0048095	-0.017461	-0.0032487	0.025841
4.4788e-05	-0.00042231	-6.7738e-05	-0.0071023	-9.225e-05	-0.0020327
-0.0025974	-0.00087197	-0.0015475	-0.010874	-0.0028442	-0.00049228

-0.003287	-0.00024567	-0.0018072	-0.024455	0.00083882	0.010157
0.00099732	0.0026816	0.0054982	-0.019258	-0.0020769	0.011633
-0.0051501	-0.008376	0.002803	-0.038774	-0.010529	-0.013392
-0.01288	-3.0418e-05	0.0073077	0.01124	0.00080731	0.0091526
0.00025061	-0.00042657	-0.00015234	-0.0067499	-0.00065513	0.00012151
-0.0025304	0.0013592	-0.00058297	-0.039046	-0.001377	0.0020105
-0.00019859	-0.00075354	-0.001412	-0.027386	-0.0019404	0.019953

(2) Using companion form to construct impulse response functions.

$$Z_t = \Psi + \Phi Z_{t-1} + G\xi \tag{5}$$

where

$$\begin{aligned} Z_t =& [y_t, y_{t-1}, \cdots, y_{t-p+1}] & (6) \\ \Psi =& [B_0; 0_{(p-1)M\times 1}] & (7) \\ \Phi =& \begin{bmatrix} B_1 & B_2 & B_3 & \cdots & B_{p-1} & B_p \\ I_M & 0 & 0 & \cdots & 0 & 0 \\ 0 & I_M & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & I_M & 0 \end{bmatrix} & (8) \\ G =& \begin{bmatrix} A_0^{-1} & 0_{M\times(p-1)M} \\ 0_{(p-1)M\times M} & 0_{(p-1)M} \end{bmatrix} & (9) \\ \xi =& [\varepsilon; 0_{(p-1)M\times 1}] & (10) \end{aligned}$$

(3) Here, we use Metropolis algorithm to sample the marginal posterior of A_0, and then for each sample we sample a coefficient B. In this way, we sample from full joint posterior of A_0 and B. The 90% and 68% error bands are given in Figure 1. Figure 2 are trace plots of this MCMC ($C = 0.5$ and $NumSim = 1,000,000$; the accepting rate is around 0.2).

(4) The monetary policy shock has been well identified because the these impulse response functions have tendency eventually going to zero. During simulation, we set a sign constraint to ensure that a positive monetary policy shock will result in immediate positive response of the federal funds rate.

(5) The variance decomposition is like following:

$$VD_{i,j}(k) = \frac{\left[\sum_{s=0}^{s=k-1} \Phi^s G\tilde{D}G^T\Phi^{sT}\right]_{i,i}}{\left[\sum_{s=0}^{s=k-1} \Phi^s GDG^T\Phi^{sT}\right]_{i,i}} \tag{11}$$

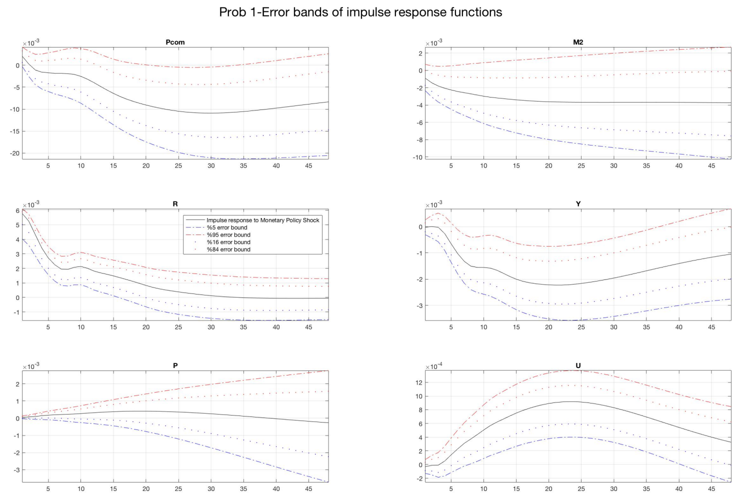


Figure 1: The impulse response functions under monetary policy shock for projection horizon 48. The comparison with figures in [1] are very close.

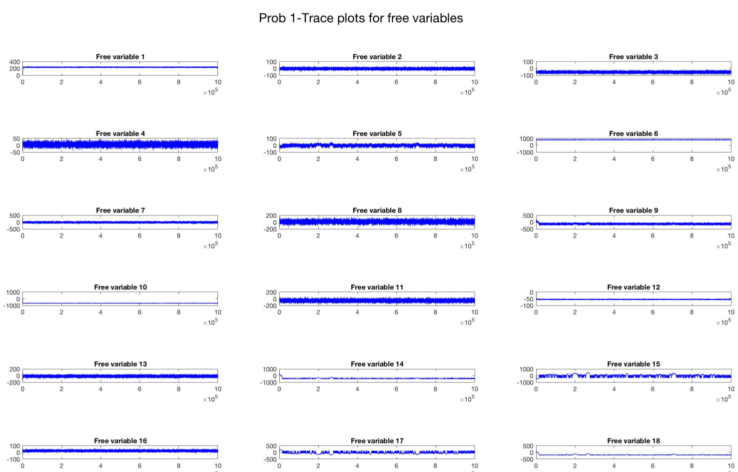


Figure 2: Trace plots for free variables in A_0.

where $\tilde{D}$ has the only nonzero entry (j, j) equal to 1 and D is M dimensional identity matrix. The portion of variance due to monetary policy shock is 0.133948456672977.

2 Prob 2:

(1) Replace *the real GDP* with *the employment-population ratio*. The procedure of calculating mode and error bands is repeated (Figure 3). Results are approximately robust

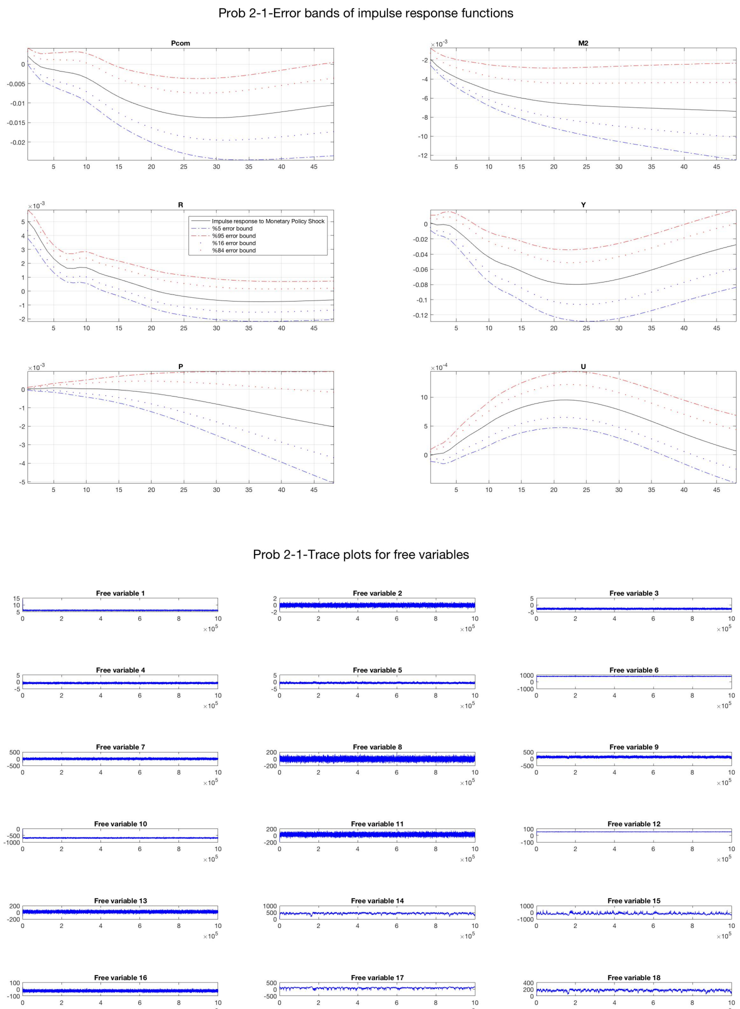


Figure 3: IMF with error bands with *the real GDP* replaced by *the employment-population ratio*.

to these changes.

(2) Replace *the real GDP* with *the industrial production*. The procedure of calculating

mode and error bands is repeated (Figure 4). Results are not approximately robust to

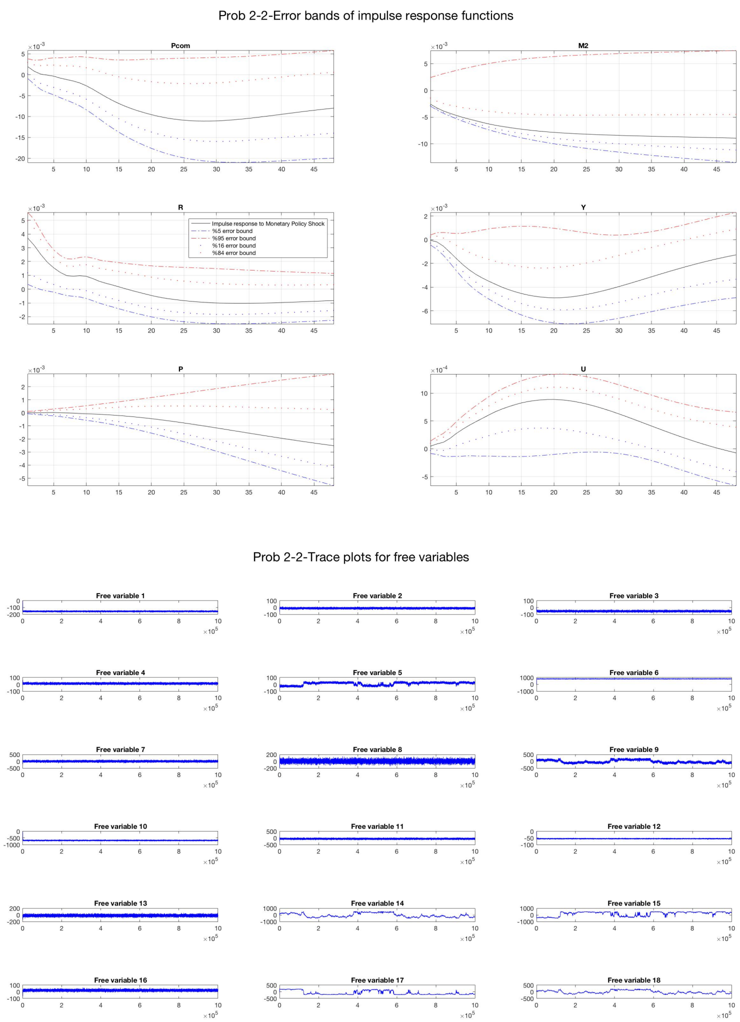


Figure 4: IMF with error bands with *the real GDP* replaced by *the employment-population ratio*.

these changes.

(3) Replace *the real GDP* with *the industrial production*, and *the M2 divisia monetary index* with *the M2 money stock*. The procedure of calculating mode and error bands is repeated (Figure 5). Results are approximately robust to these changes.

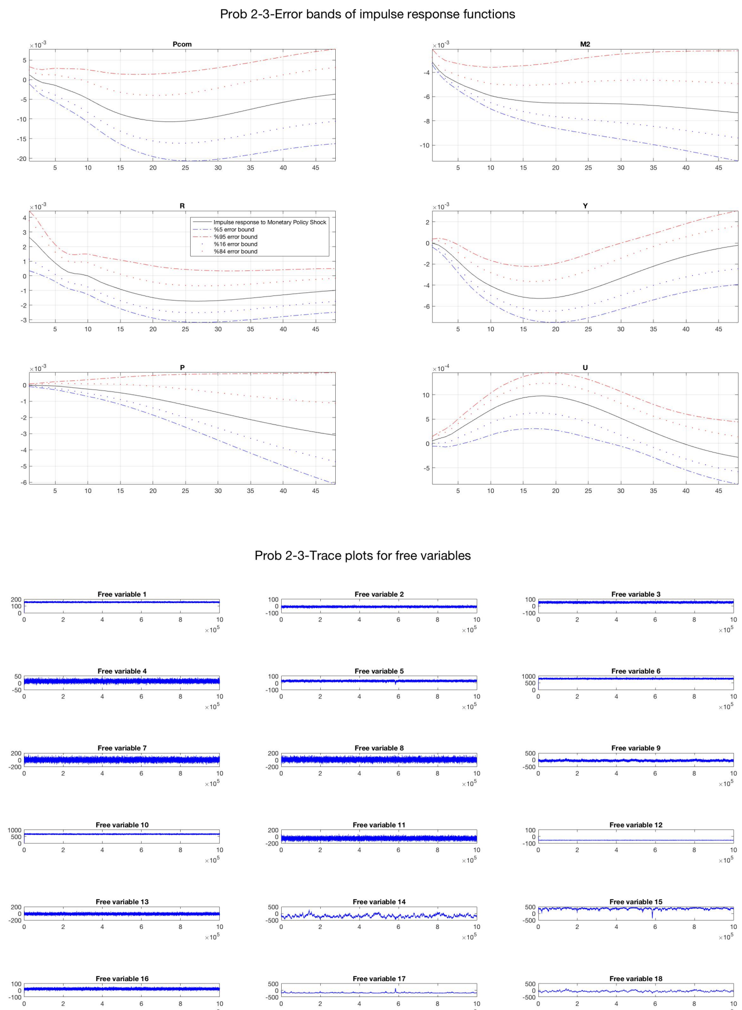


Figure 5: IMF with error bands with *the real GDP* replaced by *the employment-population ratio*, and *the M2 divisia monetary index* replaced by *the M2 money stock*.

3 Prob 3:

(1) Use extended periods of data from 1959 : 1 to 2008 : 12 excluding ZLB period. Repeat the previous procedure (Figure 6). Results are not approximately robust to these changes.

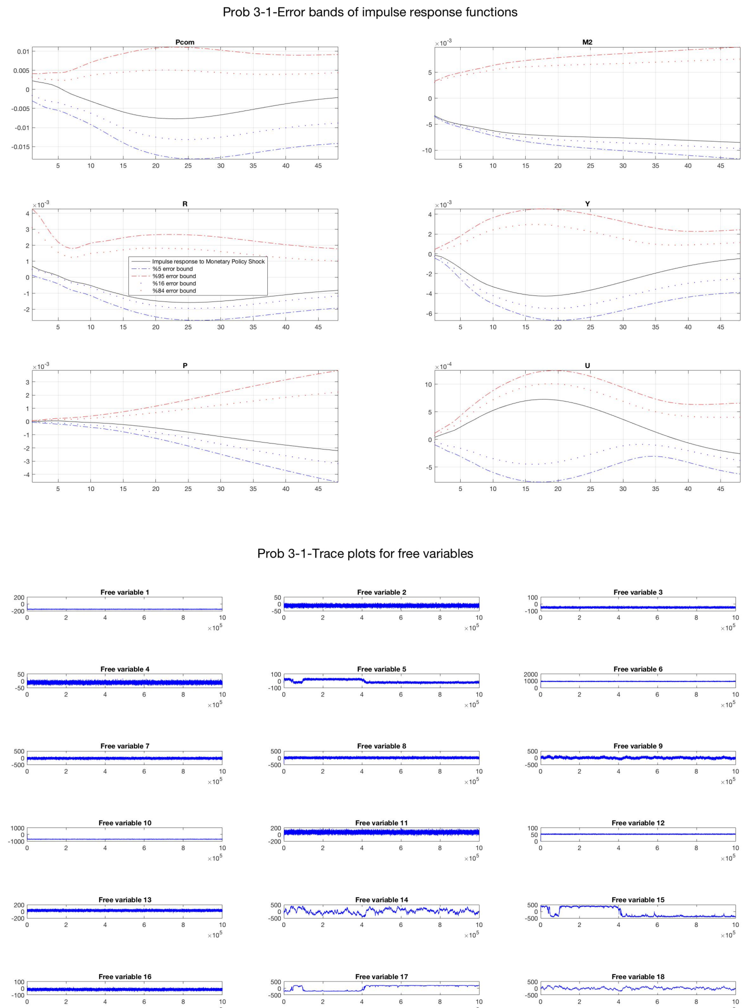
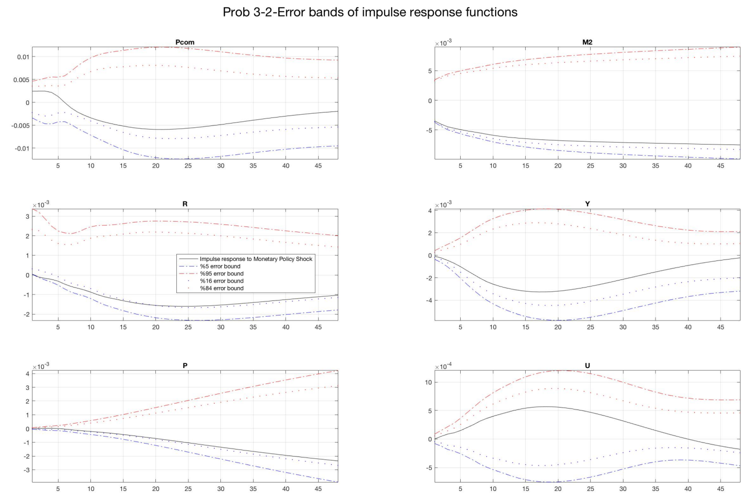


Figure 6: Use extended periods of data from 1959 : 1 to 2008 : 12 excluding ZLB period.

(2) Use the entire extended periods of data from 1959 : 1 to 2014 : 12. Repeat the previous procedure (Figure 7).



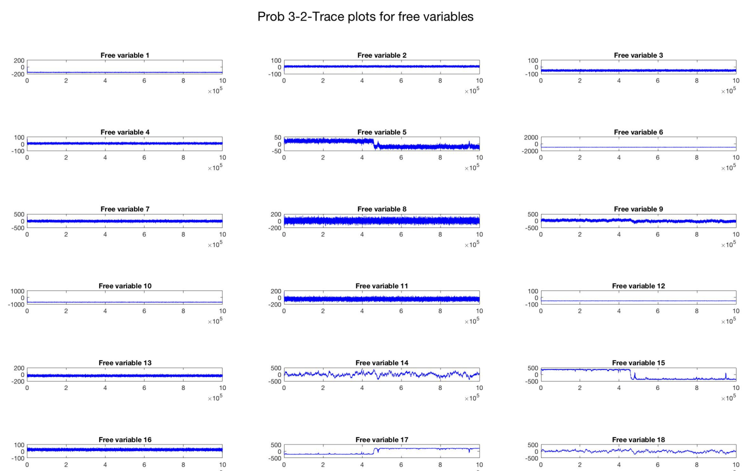


Figure 7: Use the entire extended periods of data from $1959:1$ to $2014:12$.

Results are not approximately robust to these changes.

References

[1] C. A. Sims and T. Zha, Were there regime switches in US monetary policy?, The American Economic Review **96**(1), 54–81 (2006).

Appendices

```
clear();
%Load the data set
Dat = xlsread('SZdata.xlsx');
Dat = Dat(:,2:7);
%Set constants
p = 13;%Maximum lag
lam = 0.2;%Hyperparameter
M = 6;%Dimension of vector
T = size(Dat,1)-p;%Time for forcasting
K = M*p+1;
vc = 10^6;%First element in Minnesota prior
mu = 1;
Num = 500000;%Number of MCMC simulation
C = 0.5;%Scaling parameter for the proposal distribution variation
ir_t = 48;%Impulse response function horizon
mp_shock = zeros(M,1);%Monetary policy shock
mp_shock(M) = 1;
multistarts = 2500;%Number of starts
t_proj = 36;%Variance decomposition
text = 'Prob 1-';
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t ,mp_shock,multistarts,t_proj,text);
```

Listing 1: *prob1.m* solves problem 1.

```
clear();
%Load the data set
Dat_all = xlsread('SZdata.xlsx');
Dat = Dat_all(:,[8,3:7]);
%Set constants
p = 13;%Maximum lag
lam = 0.2;%Hyperparameter
M = 6;%Dimension of vector
T = size(Dat,1)-p;%Time for forcasting
K = M*p+1;
vc = 10^6;%First element in Minnesota prior
mu = 1;
Num = 500000;%Number of MCMC simulation
C = 0.5;%Scaling parameter for the proposal distribution variation
ir_t = 48;%Impulse response function horizon
```

```
mp_shock = zeros(M,1);%Monetary policy shock
mp_shock(M) = 1;
multistarts = 2500;%Number of starts
t_proj = 36;%Variance decomposition
text = 'Prob 2-1-';
% Replace GDP with employment-population ratio
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
% Replace GDP with industrial production
Dat = Dat_all(:,[9,3:7]);
text = 'Prob 2-2-';
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
% Replace GDP with industrial produciton, and M2 divisia monetary index
% with M2 money stock
Dat = Dat_all(:,[9,3:5,10,7]);
text = 'Prob 2-3-';
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
```

Listing 2: *prob2.m*

```
clear();
%Load the data set
Dat_all = xlsread('SZdataExtended.xlsx');
Dat = Dat_all(:,2:7);
%Set constants
p = 13;%Maximum lag
lam = 0.2;%Hyperparameter
M = 6;%Dimension of vector
T = size(Dat,1)-p;%Time for forcasting
K = M*p+1;
vc = 10^6;%First element in Minnesota prior
mu = 1;
Num = 500000;%Number of MCMC simulation
C = 0.5;%Scaling parameter for the proposal distribution variation
ir_t = 48;%Impulse response function horizon
mp_shock = zeros(M,1);%Monetary policy shock
mp_shock(M) = 1;
multistarts = 2500;%Number of starts
t_proj = 36;%Variance decomposition
text = 'Prob 3-2-';
% Use the entire data set
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
% Excluding ZLB period
Dat = Dat_all(1:600,2:7);
text = 'Prob 3-1-';
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
```

Listing 3: *prob3.m*

```
clear();
%Load the data set
Dat = xlsread('SZdata.xlsx');
```

```
Dat = Dat(:,[8,3:7]);
%Set constants
p = 13;%Maximum lag
lam = 0.2;%Hyperparameter
M = 6;%Dimension of vector
T = size(Dat,1)-p;%Time for forcasting
K = M*p+1;
vc = 10^6;%First element in Minnesota prior
mu = 1;
Num = 500000;%Number of MCMC simulation
C = 0.5;%Scaling parameter for the proposal distribution variation
ir_t = 48;%Impulse response function horizon
mp_shock = zeros(M,1);%Monetary policy shock
mp_shock(M) = 1;
multistarts = 2500;%Number of starts
t_proj = 36;%Variance decomposition
text = 'Prob 2-1-';
% Replace GDP with employment-population ratio
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
% Replace GDP with industrial production
Dat = Dat(:,[9,3:7]);
text = 'Prob 2-2-';
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
% Replace GDP with industrial produciton, and M2 divisia monetary index
% with M2 money stock
Dat = Dat(:,[9,3:5,10,7]);
text = 'Prob 2-3-';
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
```

Listing 4: *prob2.m*

```
clear();
%Load the data set
Dat = xlsread('SZdataExtended.xlsx');
Dat = Dat(:,2:7);
%Set constants
p = 13;%Maximum lag
lam = 0.2;%Hyperparameter
M = 6;%Dimension of vector
T = size(Dat,1)-p;%Time for forcasting
K = M*p+1;
vc = 10^6;%First element in Minnesota prior
mu = 1;
Num = 500000;%Number of MCMC simulation
C = 0.5;%Scaling parameter for the proposal distribution variation
ir_t = 48;%Impulse response function horizon
mp_shock = zeros(M,1);%Monetary policy shock
mp_shock(M) = 1;
multistarts = 2500;%Number of starts
t_proj = 36;%Variance decomposition
text = 'Prob 3-2-';
```

```
% Use the entire data set
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
% Excluding ZLB period
Dat = Dat(1:600,2:7);
text = 'Prob 3-1-';
IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,t_proj,text);
```

Listing 5: *prob3.m*

```
function IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,
    t_proj,text)

%clear();
%Load the data set
% Dat = xlsread('SZdata.xlsx');
% Dat = Dat(:,2:7);
% %Set constants
% p = 13;
% lam = 0.2;
% M = 6;
% T = size(Dat,1)-p;
% K = M*p+1;
% vc = 10^6;
% mu = 1;
% Num = 100000;%Number of MCMC simulation
% C = 0.6;%Scaling parameter for the proposal distribution variation
% ir_t = 48;
% mp_shock = zeros(M,1);
% mp_shock(6) = 1;
% multistarts = 2500;
% t_proj = 36;
%Create matrix of X and Y combining observations and dummy observations
[X,Y] = createXY(Dat,p,T,M,K,mu);

%Estimate elements of Omega
omega_d = createOmega(Dat,p,T,M,K,vc,lam);

%Estimate the MAP for B and constant for b_hat in prior
b_hat = [zeros(M,1),eye(M,M*p)]';
B_hat = ((X'*X)+diag(omega_d.^(-1)))\(X'*Y + diag(omega_d.^(-1))*b_hat);
csvwrite([text,'B_hat_mine.csv'],B_hat);

%% Optimize posterior for A_0
func = @(a0)-logPostA0(a0,X,Y,omega_d/lam^2,b_hat,B_hat,M,T,lam);
solver = 'fminunc';
options = optimoptions(@fminunc,'Algorithm','quasi-newton','Display', 'off');
%Solve once and obtain hessian matrix for the proposal distribution in
%Metropolis alg
a0_0 = initPoint(X,Y,B_hat,b_hat,omega_d);%a0_0 is a column vector
[a0_init,post_val_init,exitflag,output_init,grad_init,Hessian]=...
    fminunc(func,a0_0,options);
```

```matlab

%Use global optimization toolbox to find "global" optimial for the
%posterior of A_0
tic;
filename = 'p1-1.csv';
[a0_opt,manymin] = findGlobalOpt(func,solver,options,multistarts,a0_0,
    filename);
A0_opt = vec2mat(a0_opt,M);
csvwrite([text,'A0_opt_mine.csv'],A0_opt);
toc;

%% Construct Impulse Response Function
% Monetary policy shock
time_range = [1,ir_t];
tic;
%Sign constraint, so that the one positive monetary shock will result in
%immediate decrease of the federal funds rate (R)
imd_resp = A0_opt\mp_shock;
sign_con = sign((imd_resp(M)));
imf = sign_con*IMF(ir_t,M,p,A0_opt,B_hat,mp_shock,time_range);
toc;
h_errbd = figure();
hold on;
box on;
annotation('textbox', [0 0.9 1 0.1], ...
    'String', [text, 'Error bands of impulse response functions'], ...
    'EdgeColor', 'none', ...
    'HorizontalAlignment', 'center',...
    'FontSize',20);
legendinfo{1}=['Impulse response to Monetary Policy Shock'];
handle(1) = plot_all_TS(imf,'-k',[time_range,-inf,inf]);
%Check the results and they are close

%% Metropolis algorithm and error bands
%acceptance rate
eps_hat = Y - X*B_hat;
S_dhat = eps_hat'*eps_hat+(B_hat-b_hat)'/diag(omega_d)*(B_hat-b_hat);
tic;
[ar,trace_dat] = Metropolis_alg(Num,a0_0,C,Hessian,S_dhat,T,M,text);%Draw A0
    from its marginal posterior distribution
toc;
imf_MCMC = zeros(ir_t,M,Num);%Store the IMF for each MCMC draw
tic;
for i = 1:Num
    A0_temp = vec2mat(trace_dat(i,:)',M);
    inv_A0_temp = inv(A0_temp);
    imd_resp = inv_A0_temp*mp_shock;%Sign constraint as before
    sign_con = sign(imd_resp(M));
    V_temp = kron(inv_A0_temp*inv_A0_temp',inv(X'*X+diag(omega_d.^(-1))));
    B_new = reshape(mvnrnd(reshape(B_hat,K*M,1),V_temp),K,M);%Draw new B
```

```
    based on each draw of A0
    imf_temp = sign_con*IMF(ir_t ,M,p,A0_temp,B_new,mp_shock,time_range);% Calculate IMF for this sample of A0 and B
    imf_MCMC(:,:,i) = imf_temp(:,2:end);
end
err_bd = quantile(imf_MCMC,[0.05,0.95,0.16,0.84],3);%Empirical quantile for each IMF
legendinfo{2} = ['%5 error bound'];
handle(2) = plot_all_TS([(1:ir_t)',err_bd(:,:,1)],'b-.',[time_range,-inf,inf]);
legendinfo{3} = ['%95 error bound'];
handle(3) = plot_all_TS([(1:ir_t)',err_bd(:,:,2)],'r-.',[time_range,-inf,inf]);
legendinfo{4} = ['%16 error bound'];
handle(4) = plot_all_TS([(1:ir_t)',err_bd(:,:,3)],'b.',[time_range,-inf,inf]);
legendinfo{5} = ['%84 error bound'];
handle(5) = plot_all_TS([(1:ir_t)',err_bd(:,:,4)],'r.',[time_range,-inf,inf]);
legend(handle,legendinfo,'Location','northeast');
toc;
saveas(gca,[text,'Error_Bands'],'jpg');
saveas(gca,[text,'Error_Bands'],'fig');
close(gcf);

%Trace plot
tic;
h_trace_plot = figure();
hold on;
box on;
annotation('textbox', [0 0.9 1 0.1], ...
    'String', [text,'Trace plots for free variables'], ...
    'EdgeColor', 'none', ...
    'HorizontalAlignment', 'center',...
    'FontSize',20);
handle1 = plot_trace(trace_dat,6,3,'b-');
toc;
saveas(gca,[text,'Trace_plots'],'jpg');
saveas(gca,[text,'Trace_plots'],'fig');
close(gcf);

%% Variance Decomposition
var_imf_mp = imf_var(M,p,A0_opt,B_hat,mp_shock,t_proj);
var_imf_all = imf_var(M,p,A0_opt,B_hat,ones(M,1),t_proj);
display([text,'The portion of variance of GDP due to monetary policy shock is:',10]);
display(var_imf_mp(1,1)/var_imf_all(1,1));
```

```
end
```

Listing 6: *IMF_Errbd.m* function calculates the mode of posteriors for A_0 and B. Then, use them calculate impulse response functions with error bands by empirical quantile from MCMC simulation.

```
function omega_d = createOmega(Dat,p,T,M,K,vc,lam)
%% The first observation is given and the rest of observation up to the
% end of current window are for prediction as stated in the problem
% statement
% Follow Hamilton 1994 to get the MLE for variance
sum1 = sum(Dat(p:p+T-1,:),1);%sum_1^T y_(t-1)
sum2 = sum(Dat(p+1:p+T,:),1);%sum_1^T y_(t)
sum3 = sum(Dat(p:p+T-1,:).^2,1);%sum_1^T y_(t-1)^2
sum4 = sum(Dat(p+1:p+T,:).^2,1);%sum_1^T y_(t)^2
sum5 = sum(Dat(p+1:p+T,:).*Dat(p:p+T-1,:),1);%sum_1^T y_(t)y_(t-1)
sigma2_hat = zeros(1,M);
for m = 1:M
     c_phi_hat = [T,sum1(m);sum1(m),sum3(m)]\[sum2(m);sum5(m)];% c_phi_hat = [
    c_hat,phi_hat]
    %temp_sum = 0;
     sigma2_hat(m) = 1.0/(T)*(sum4(m)+c_phi_hat(2)^2*sum3(m)+2*c_phi_hat(1)*
    c_phi_hat(2)*sum1(m)...
                        -2*c_phi_hat(1)*sum2(m)-2*c_phi_hat(2)*sum5(m))+c_phi_hat
    (1)^2;
end
omega_d = zeros(1,K);
omega_d(1) = [vc/lam^2];
for j = 1:p
    omega_d(1+(j-1)*M+1:1+j*M) = (sigma2_hat*j^2).^(-1);
end
omega_d = lam^2 * omega_d;
end
```

Listing 7: *createOmega.m* creates Ω for Minnesota prior.

```
function [X,Y] = createXY(Dat,p,T,M,K,mu)
%Assemble the ordinary Y
Y = Dat((p+1:p+T),:);
%Assemble the ordinary X
X = zeros(T,K);% a T*K matrix
for j = 1:T
    X(j,:) = [1,reshape(Dat((j+p-1:-1:j),:)',1,K-1)];% have to transpose
    because the reshape function operate in column
end

%Add sum-of-coefficient
Y_bar = diag(mu*mean(Dat(1:p,:),1));
Y = [Y;Y_bar];
Xplus = [zeros(M,1),repmat(diag(mu*mean(Dat(1:p,:),1)),[1 p])];
```

```
X= [X;Xplus]; % stack x at the bottom
end
```

Listing 8: *createXY.m* constructs matrix X and Y with sum-of-coefficient prior with dummy observations.

```
function [a0_opt, manymin] = findGlobalOpt(objFunc,solver,options,multistarts,a0_0,filename)
%% Use Global Optimization Toolbox in Matlab to find a "global" optimal of
%the objective function.
%Input:
    %objFunc: The objFunc has to have defined the variables and parameters
%==========================================
    %Input:
        %lam: lambda;
        %A0: The A_0 matrix (M-by-M);
        %X: X matrix ((T+M)-by-K);
        %Y: Y matrix ((T+M)-by-M);
        %d_phi: Diagonal entries for square root of Phi matrix (K entries)
        %b_hat: Matrix form for the mean of prior for beta (which is a vector form of B);
        %M: Dimension of vector variables;
        %T: Time periods for forcasting;
        %p: The maximum lag;
    %Output:
        %logfval: The log posterior function value;
%==========================================
    %solver: The specified solver. Here use 'fminunc';
    %options: Options for the solver (optimoptions(@fminunc,'Algorithm','quasi-newton','Display', 'off')).
    %          Notice that here we are necessarily requiring gradient of objFunc
    %multistarts: Number of multistart;
    %a0_0: The initial guess for a_0;
    %filename: The file name for writing the resutls
%Output:
    %a0_opt: The optimal a0 with the optimal function value.
%% ===========================================================================
%Create an optimization problem
prob = createOptimProblem(solver,'objective',objFunc,'x0',a0_0,'options',options);
%Number of multistart
MS = MultiStart;
%Run multistarts times of this problem
[x,f,~,~,manymin] = run(MS,prob,multistarts);
%Write to a file and get the optimal value for free variables
csvwrite(filename,x);
a0_opt = x;
```

```
end
```

Listing 9: *findGlobalOpt.m* uses multi-starts in global optimization toolbox to find somewhat "global" optimal value for matrix A_0 by maximizing the marginal posterior 4

```
function imf = IMF(T,M,p,A0,B_hat,mp_shock,time_range)
%% Use companion form to get the impulse response function interms of time.
%Input:
%   T: The time horizon to investigate the impulse response;
%   M: Dimension of vector variable;
%   p: The maximum time lag;
%   A0: The MAP of A0 matrix (M-by-M);
%   B_hat: The MAP of B matrix (K-by-M);
%   mp_shock: To specify which shock is in interest (M-by-1);
%Output:
%   imf: Impulse response function in terms of time periods (T-by-M);

% Phi is a K-by-K matrix
Phi = sparse([B_hat(2:end,:)';eye((p-1)*M,p*M)]);
% Big shock column: K-by-1
MP_shock = sparse([mp_shock;zeros((p-1)*M,1)]);
% Big matrix pre-multiplied on shocks
G = sparse(zeros(p*M,p*M));
G(1:M,1:M) = A0\eye(M);

imf = zeros(T,M);
temp = G*MP_shock;
for i = 1:T
    temp = Phi*temp;
    imf(i,:) = temp(1:M)';
end
imf = [(time_range(1):(time_range(2)-time_range(1))/(T-1):time_range(2))',imf
    ];
end
```

Listing 10: *IMF.m* constructs impulse response functions using companion form.

```
function a0_0 = initPoint(X,Y,B_hat,b_hat,omega_d)
%Residual;
eps_hat = Y - X*B_hat;
%Another part;
temp1 = (diag(omega_d.^0.5)\(B_hat-b_hat));
%S_hat
S_hat = eps_hat'*eps_hat+temp1'*temp1;
%Cholesky decomposition on S_hat'*S_hat to get the initial A0 and then
%return the vector version of it.
A0_0 = chol(S_hat'*S_hat,'upper');
a0_0 = mat2vec(A0_0);
```

```
end
```

Listing 11: *initPoint.m* returns initial guess for global optimization executed by *findGlobalOpt.m*.

```
function logfval = logPostA0(a0,X,Y,d_phi,b_hat,B_hat,M,T,lam)
%% Calculate the log posterior for A_0 and return the B_hat
%Input:
%lam: lambda;
%A0: The A_0 matrix (M-by-M);
%X: X matrix ((T+M)-by-K);
%Y: Y matrix ((T+M)-by-M);
%d_phi: Diagonal entries for square root of Phi matrix (K entries)
%b_hat: Matrix form for the mean of prior for beta (which is a vector form of
    B);
%B_hat: Matrix of coefficient (K-by-M);
%M: Dimension of vector variables;
%T: Time periods for forcasting;
%p: The maximum lag;
%Output:
%logfval: The log posterior function value;

%% ================================================================================
%MAP for B;
% B_hat = (lam^2*(X'*X)+diag(d_phi.^(-1)))\(lam^2*X'*Y + diag(d_phi.^(-1))*
    b_hat);
%Residual;
eps_hat = Y - X*B_hat;
%Assemble the A0 from a0
A0 = vec2mat(a0,M);
%A simple way to calculate the arguments inside exponant;
% temp1 = eps_hat*A0';
% temp2 = lam^(-1)*(diag(d_phi.^0.5)\(B_hat-b_hat))*A0';
%The trace of a matrix times its transpose is just the Frobenius norm of
%that matrix;
%logfval = (T+M)*log(det(A0))-0.5*(sum(sum(temp1.^2))+sum(sum(temp2.^2))); %
    equivalent form
logfval = (T+M)*log(det(A0))-0.5*trace((eps_hat'*eps_hat+lam^(-2)*(B_hat-
    b_hat)'/diag(d_phi)*(B_hat-b_hat))*(A0'*A0));
end
```

Listing 12: *logPostA0.m* calculates log-posterior of A_0.

```
function a = mat2vec(A)
[n,m] = size(A);
A = reshape(A,n*m,1);
a = A([1,2,3,4,5,8,9,10,11,15,16,22,28,29,30,34,35,36]);
end
```

Listing 13: *mat2vec.m* transfers a vector a_0 of free variables in matrix A_0 into a matrix A_0.

```
function [ar,trace_dat] = Metropolis_alg(Num,a0_0,C,Hessian,S_dhat,T,M,text)

if (issymmetric(Hessian)~=1)
    Hessian = (Hessian+Hessian')/2;
end
delta = min(eig(Hessian));
if (delta < 0)
    Hessian = Hessian -2*delta*eye(size(Hessian,1));
end

V = C^2*inv(Hessian);
trace_dat = zeros(Num,length(a0_0));
accept = 0;
for i = 1:Num
    A0_0 = vec2mat(a0_0,M);
    a0_new = mvnrnd(a0_0,V);
    A0_new = vec2mat(a0_new,M);
    lgp_0 = (T+M)*log(det(A0_0))-0.5*trace(S_dhat*(A0_0'*A0_0));
    lgp_new = (T+M)*log(det(A0_new))-0.5*trace(S_dhat*(A0_new'*A0_new));
    rate = exp(lgp_new-lgp_0);
    if rate >= 1
        a0_0 = a0_new;
        accept = accept + 1;
    else
        u = rand;
        if u <= rate
            a0_0 = a0_new;
            accept = accept + 1;
        end
    end
    trace_dat(i,:) = a0_0';
end
ar = accept / Num;
csvwrite([text,'trace_plot.csv'],trace_dat);
end
```

Listing 14: *Metropolis_alg.m* implements MCMC using Metropolis algorithm.

```
function h=plot_all_TS(TSdat,patten,axlmt)
    subplot(3,2,4);
    hold on;
    box on;
    grid on;
    plot(abs(TSdat(:,1)),TSdat(:,2),patten);
    axis(axlmt)
    % legend('Observed Data','DC Prediction')
    %legend(h,text)
    title('Y')

    subplot(3,2,5);
    hold on;
```

```
    box on;
    grid on;
    plot(abs(TSdat(:,1)),TSdat(:,3),patten);
    axis(axlmt)
    % legend('Observed Data','DC Prediction')
    title('P')

    subplot(3,2,6);
    hold on;
    box on;
    grid on;
    plot(abs(TSdat(:,1)),TSdat(:,4),patten);%,xt,rt2.predgdp,'-.m',xt,rt3.
    predgdp,'--c',xt,rt4.predgdp,':r')
    axis(axlmt)
    % legend('Observed Data','DC Prediction')
    title('U')

    subplot(3,2,1);
    hold on;
    box on;
    grid on;
    plot(abs(TSdat(:,1)),TSdat(:,5),patten);%,xt,rt2.predgdp,'-.m',xt,rt3.
    predgdp,'--c',xt,rt4.predgdp,':r')
    axis(axlmt)
    % legend('Observed Data','DC Prediction')
    title('Pcom')

    subplot(3,2,2);
    hold on;
    box on;
    grid on;
    plot(abs(TSdat(:,1)),TSdat(:,6),patten);%,xt,rt2.predgdp,'-.m',xt,rt3.
    predgdp,'--c',xt,rt4.predgdp,':r')
    axis(axlmt)
    % legend('Observed Data','DC Prediction')
    title('M2')

    subplot(3,2,3);
    hold on;
    box on;
    grid on;
    h=plot(abs(TSdat(:,1)),TSdat(:,7),patten);%,xt,rt2.predgdp,'-.m',xt,rt3.
    predgdp,'--c',xt,rt4.predgdp,':r')
    axis(axlmt)
    % legend('Observed Data','DC Prediction')
    title('R')
end
```

Listing 15: *plot_all_TS* plots impulse response functions (w.r.t t) into subplots.

```
function h=plot_trace(trace_dat,n,m,patten)
```

```

[Num,~] = size(trace_dat);

for i = 1:(n*m)

subplot(n,m,i);
hold on;
box on;
h=plot(1:Num,trace_dat(:,i),patten);
% legend('Observed Data','DC Prediction')
%legend(h,text)
title(['Free variable ',num2str(i)]);
end
end
```

Listing 16: *plot_trace.m* plots trace plots of free variables in vector a_0.

```
function A = vec2mat(a,M)
% A = zeros(M^2,1);
% A([1,7,13,19,25,8,14,20,26,15,21,22,23,29,35,24,30,36]) = a;
% A = reshape(A,M,M);
% A = A';

A = zeros(M^2,1);
A([1,2,3,4,5,8,9,10,11,15,16,22,28,29,30,34,35,36]) = a;
A = reshape(A,M,M);
end
```

Listing 17: *vec2mat.m* transfers the free variables in A_0 into a vector a_0.

```
function IMF_Errbd(Dat,p,lam,M,T,K,vc,mu,Num,C,ir_t,mp_shock,multistarts,
    t_proj,text)

%clear();
%Load the data set
% Dat = xlsread('SZdata.xlsx');
% Dat = Dat(:,2:7);
% %Set constants
% p = 13;
% lam = 0.2;
% M = 6;
% T = size(Dat,1)-p;
% K = M*p+1;
% vc = 10^6;
% mu = 1;
% Num = 100000;%Number of MCMC simulation
% C = 0.6;%Scaling parameter for the proposal distribution variation
% ir_t = 48;
% mp_shock = zeros(M,1);
% mp_shock(6) = 1;
% multistarts = 2500;
```

```
% t_proj = 36;
%Create matrix of X and Y combining observations and dummy observations
[X,Y] = createXY(Dat,p,T,M,K,mu);

%Estimate elements of Omega
omega_d = createOmega(Dat,p,T,M,K,vc,lam);

%Estimate the MAP for B and constant for b_hat in prior
b_hat = [zeros(M,1),eye(M,M*p)]';
B_hat = ((X'*X)+diag(omega_d.^(-1)))\(X'*Y + diag(omega_d.^(-1))*b_hat);
csvwrite([text,'B_hat_mine.csv'],B_hat);

%% Optimize posterior for A_0
func = @(a0)-logPostA0(a0,X,Y,omega_d/lam^2,b_hat,B_hat,M,T,lam);
solver = 'fminunc';
options = optimoptions(@fminunc,'Algorithm','quasi-newton','Display', 'off');
%Solve once and obtain hessian matrix for the proposal distribution in
%Metropolis alg
a0_0 = initPoint(X,Y,B_hat,b_hat,omega_d);%a0_0 is a column vector
[a0_init,post_val_init,exitflag,output_init,grad_init,Hessian]=...
    fminunc(func,a0_0,options);

%Use global optimization toolbox to find "global" optimial for the
%posterior of A_0
tic;
filename = 'p1-1.csv';
[a0_opt,manymin] = findGlobalOpt(func,solver,options,multistarts,a0_0,
    filename);
A0_opt = vec2mat(a0_opt,M);
csvwrite([text,'A0_opt_mine.csv'],A0_opt);
toc;

%% Construct Impulse Response Function
% Monetary policy shock
time_range = [1,ir_t];
tic;
%Sign constraint, so that the one positive monetary shock will result in
%immediate decrease of the federal funds rate (R)
imd_resp = A0_opt\mp_shock;
sign_con = sign((imd_resp(M)));
imf = sign_con*IMF(ir_t,M,p,A0_opt,B_hat,mp_shock,time_range);
toc;
h_errbd = figure(1);
hold on;
box on;
annotation('textbox', [0 0.9 1 0.1], ...
    'String', [text, 'Error bands of impulse response functions'], ...
    'EdgeColor', 'none', ...
    'HorizontalAlignment', 'center',...
    'FontSize',20);
```

```
legendinfo{1}=['Impulse response to Monetary Policy Shock'];
handle(1) = plot_all_TS(imf,'-k',[time_range,-inf,inf]);
%Check the results and they are close

%% Metropolis algorithm and error bands
%acceptance rate
eps_hat = Y - X*B_hat;
S_dhat = eps_hat'*eps_hat+(B_hat-b_hat)'/diag(omega_d)*(B_hat-b_hat);
tic;
[ar,trace_dat] = Metropolis_alg(Num,a0_0,C,Hessian,S_dhat,T,M);%Draw A0 from its marginal posterior distribution
toc;
imf_MCMC = zeros(ir_t,M,Num);%Store the IMF for each MCMC draw
tic;
for i = 1:Num
    A0_temp = vec2mat(trace_dat(i,:)',M);
    inv_A0_temp = inv(A0_temp);
    imd_resp = inv_A0_temp*mp_shock;%Sign constraint as before
    sign_con = sign(imd_resp(M));
    V_temp = kron(inv_A0_temp*inv_A0_temp',inv(X'*X+diag(omega_d.^(-1))));
    B_new = reshape(mvnrnd(reshape(B_hat,K*M,1),V_temp),K,M);%Draw new B based on each draw of A0
    imf_temp = sign_con*IMF(ir_t,M,p,A0_temp,B_new,mp_shock,time_range);%Calculate IMF for this sample of A0 and B
    imf_MCMC(:,:,i) = imf_temp(:,2:end);
end
err_bd = quantile(imf_MCMC,[0.05,0.95,0.16,0.84],3);%Empirical quantile for each IMF
legendinfo{2} = ['%5 error bound'];
handle(2) = plot_all_TS([(1:ir_t)',err_bd(:,:,1)],'b-.',[time_range,-inf,inf]);
legendinfo{3} = ['%95 error bound'];
handle(3) = plot_all_TS([(1:ir_t)',err_bd(:,:,2)],'r-.',[time_range,-inf,inf]);
legendinfo{4} = ['%16 error bound'];
handle(4) = plot_all_TS([(1:ir_t)',err_bd(:,:,3)],'b.',[time_range,-inf,inf]);
legendinfo{5} = ['%84 error bound'];
handle(5) = plot_all_TS([(1:ir_t)',err_bd(:,:,4)],'r.',[time_range,-inf,inf]);
legend(handle,legendinfo,'Location','northeast');
toc;
saveas(h_errbd,[text,'Error_Bands'],'jpg');
saveas(h_errbd,[text,'Error_Bands'],'fig');

%Trace plot
tic;
h_trace_plot = figure(2);
hold on;
box on;
```

```
annotation('textbox', [0 0.9 1 0.1], ...
    'String', [text,'Trace plots for free variables'], ...
    'EdgeColor', 'none', ...
    'HorizontalAlignment', 'center',...
    'FontSize',20);
handle1 = plot_trace(trace_dat,6,3,'b-');
toc;
saveas(h_trace_plot,[text,'Trace_plots'],'jpg');
saveas(h_trace_plot,[text,'Trace_plots'],'fig');

%% Variance Decomposition
var_imf_mp = imf_var(M,p,A0_opt,B_hat,mp_shock,t_proj);
var_imf_all = imf_var(M,p,A0_opt,B_hat,ones(M,1),t_proj);
display([text,'The portion of variance of GDP due to monetary policy shock is
    :',10]);
display(var_imf_mp(1,1)/var_imf_all(1,1));
end
```

Listing 18: *imf_var.m* calculates the IMF variance based on shock given.