Econometrics - Problem Set 3

Gualtiero Azzalini

Exercise 1

Part a

First I import the data in MATLAB and I balance the dataset in order to have values for c_t and c_{t+1} (note that this implies missing an observation at the end so we have 249 observations in total). I also define the vector of instruments $z_t = [1, R_t, (1 + c_t)]'$. Note that I take the exp of some variables as the data are in logs.

```
%% Exercise 1 - First specification for z
clear;
% Import the .csv file
M = csvread('hsdata.csv',1,0);
t = csvread('hsdata.csv',1,0,[1,0,250,0])';
c = csvread('hsdata.csv',1,1,[1,1,250,1])';
d = csvread('hsdata.csv',1,2,[1,2,250,2])';
r = csvread('hsdata.csv',1,3,[1,3,250,3])';
% Balance the observations
for i=1:(length(c)-1)
c_1(i)=c(i+1);
r_1(i)=r(i+1);
d_1(i)=d(i+1);
end
r(250)=[];c(250)=[];d(250)=[];
T = length(c);
z=[ones(1,T); exp(r); (1+c)];
```

Then I separately define the objective function that I have to minimize in the program *objective*:

```
%% This program computes the objective function for Euler Equation conditions
%%INPUTS%%
% x = (delta,gamma) parameters to be estimated
% W = weighting matrix (KxK)
% z = observables at t
% T=obs number
% c_1= log(C_t+1/C_t)
% c= log(C_t/C_t-1), r_1= log(R_t+1)
```

```
%%OUTPUTS%%
%Sdf = stochastic discount factor (1xT)
% Euler = set of euler conditions equal to zero (K*T)
% g = sample average of each Euler (Kx1)
% Qn = objective function to minimize (1x1)
% default output is Qn only
function Qn = objective(x, W, z, T, c_1, r_1, r, c);
for i=1:T
sdf(i)=x(1)*exp(-x(2)*c_1(i)+r_1(i))-1;
euler(:,i)=kron(sdf(:,i),z(:,i));
end
g=[sum(euler')/T]';
Qn = 0.5*(g'*W*g);
end
```

In words, the function above computes the moments' condition we have from:

$$\mathbb{E}\left[\left(\delta_0\left(\frac{C_{t+1}}{C_t}\right)^{-\gamma_0}R_{t+1}-1\right)\otimes z_t\right]=0$$

In this case we have three instruments, therefore in total we have the following three moments conditions:

$$\mathbb{E}\left[\left(\delta_0\left(\frac{C_{t+1}}{C_t}\right)^{-\gamma_0}R_{t+1}-1\right)\right]=0$$

$$\mathbb{E}\left[\left(\delta_0\left(\frac{C_{t+1}}{C_t}\right)^{-\gamma_0}R_{t+1}-1\right)R_t\right]=0$$

$$\mathbb{E}\left[\left(\delta_0\left(\frac{C_{t+1}}{C_t}\right)^{-\gamma_0}R_{t+1}-1\right)(1+c_t)\right]=0$$

As our data are in logs, we can use exp and rewrite:

$$\mathbb{E}\left[\left(\delta_0\exp\left\{-\gamma_0 c_{t+1}+r_{t+1}\right\}-1\right)\otimes z_t\right]=0$$

I compute first the term in parenthesis, *sdf* in my code and then compute the Kronecker product with the instruments. After that I get $g_n(\theta)=\frac{1}{n}\sum_{t=1}^{n}g(w_t;\theta)$ (note that in my code $n=T$) and then compute the objective function $Q_n(\theta)=-\frac{1}{2}g_n(\theta)'\hat{W}g_n(\theta)$ (there is no minus in the code as I minimize the inverse of the formula).

Using the above, I compute the GMM estimator using the following code:

```
% Compute GMM - first step
W_1=eye(3);
fun1 = @(x)objective(x, W_1, z, T, c_1, r_1, r, c);
x0=[0.9 0.9];
[theta1_gmm, Qval1] = fminsearch(fun1, x0);
```

First, I define $\hat{W} = I$ as the first-step requires (it's a 3x3 matrix as we have three conditions), then I call the objective function I have separately written in the program *objective*, using as starting values for δ and γ 0.9 and 0.9 (the same average values Hansen gets in the 1982 paper) and finally I minimize the function with respect to the vector of parameters $x = (\delta, \gamma)$.

The first-step GMM estimator I obtain is $\hat{\theta}_{GMM,1} = (\hat{\delta}_{GMM,1}, \hat{\gamma}_{GMM,1}) = (0.99495, 4.4399)$.

Part b

Using the estimator $\hat{\theta}_{GMM,1}$ obtained in (a) I can now compute the two-step efficient estimator in the following way:

```
% Euler conditions at parameters estimated in step 1
eul = ee(theta1_gmm, z, T, c_2, r_2, r_1, c_1);
% 2nd-step weighting matrix - the long-run variance, with lags (b/c dependent data)
W_2=inv(longrunW(eul, T, 12));
fun2 = @(x)objective(x, W_2, z, T, c_2, r_2, r_1, c_1);
[theta2_gmm, Qval2]= fminsearch(fun2, theta1_gmm);
```

First, I compute the moments' conditions at the parameter estimated in (a) using the following separate program called *ee*:

```
%% Euler conditions
% This program computes Euler conditions given parameters
%%INPUTS%%
% x = (delta,gamma) estimated parameters
% z = observables at t
% T=obs number
% c_1= log(C_t+1/C_t)
% c= log(C_t/C_t-1), r_1= log(R_t+1)
%%OUTPUTS%%
% sdf = stochastic discount factor (1xT)
% ee = set of euler conditions equal to zero (K*T) - default only ee
function ee = ee(x, z, T, c_1, r_1, r, c);
for i=1:T
sdf(i)=x(1)*exp(-x(2)*c_1(i)+r_1(i))-1;
ee(:,i)=kron(sdf(:,i),z(:,i));
end
end
```

Then, I use the above result to compute the long run variance as the data may have non zero covariances in order to have an efficient two-step weighting matrix:

$$\hat{W} = \hat{\Gamma}_0 + \sum_{j=1}^{J_n} \left(1 - \frac{j}{J_n + 1}\right) (\hat{\Gamma}_j + \hat{\Gamma}_j')$$

$$\hat{\Gamma}_j = \frac{1}{n} \sum_{t=j+1}^{n} g(w_t; \hat{\theta}) g(w_{t-j}; \hat{\theta})'$$

with the separate program *longrunW* (below), to get the efficient second step estimator, using $J_n = 12$ (results are robust to using the other "rule of thumb" values of lags). In this case the g corresponds to what in the code I call *eul* and $\hat{\theta}$ is the first-step estimate.

```
%% Computation of long-run variance matrix
%%INPUTS%%
% data = matrix of dependent data (KxT)
% T = obs number
% lags = number of lags to include
%%OUTPUTS%%
% W_hat = estimated long-run variance (KxK)
function W_hat = longrunW(data, T, lags);
dataT=data';
W_hat=data*data'/T;
for j=1:lags
gammaj=(data(:,1+j:T)*dataT(1:(T-j),:))/T;
gammaj_sq=gammaj+gammaj';
W_hat=W_hat+(1-j/(lags+1))*gammaj_sq;
end
end
```

After having obtained the two-step efficient matrix I use the same procedure as in (a) to minimize the objective function with the new $\hat{W}$, using the previous estimate as starting value. The two-step GMM estimator I obtain is $\hat{\theta}_{GMM,2} = (\hat{\delta}_{GMM,2}, \hat{\gamma}_{GMM,2}) = (0.97982, 3.4485)$.

Part c

Since I have used the efficient two-step efficient weighting matrix, I can exploit the fact that $\hat{W} \xrightarrow{p} S^{-1}$ to assess that the estimator will reach the efficient asymptotic variance $\left(G'S^{-1}G\right)^{-1}$ and together with the fact that the estimator is asymptotically normal, this implies:

$$\sqrt{n}\left(\hat{\theta} - \theta_0\right) \xrightarrow{d} N\left(0, \left(G'S^{-1}G\right)^{-1}\right)$$

From the results in the notes I can estimate the asymptotic variance consistently with $AV\hat{a}r = \left(\hat{G}'\hat{W}\hat{G}\right)^{-1}$, where $\hat{W}$ is the second-step weigtinh matrix and $\hat{G} = \frac{1}{n}\sum_{t=1}^{n} \frac{\partial g(w_t; \hat{\theta}_{GMM,2})}{\partial \theta'}$.

The matrix of derivatives is:

$$\hat{G} = \frac{1}{n} \begin{bmatrix} \sum_{t=1}^{n} \exp\left(-\hat{\gamma}_{GMM,2} c_{t+1} + r_{t+1}\right) & \sum_{t=1}^{n} -c_{t+1}\hat{\delta}_{GMM,2} \exp\left(-\hat{\gamma}_{GMM,2} c_{t+1} + r_{t+1}\right) \\ \sum_{t=1}^{n} \exp\left(-\hat{\gamma}_{GMM,2} c_{t+1} + r_{t+1} + r_t\right) & \sum_{t=1}^{n} -c_{t+1}\hat{\delta}_{GMM,2} \exp\left(-\hat{\gamma}_{GMM,2} c_{t+1} + r_{t+1} + r_t\right) \\ \sum_{t=1}^{n} \exp\left(-\hat{\gamma}_{GMM,2} c_{t+1} + r_{t+1}\right)(1 + c_t) & \sum_{t=1}^{n} -c_{t+1}\hat{\delta}_{GMM,2} \exp\left(-\hat{\gamma}_{GMM,2} c_{t+1} + r_{t+1}\right)(1 + c_t) \end{bmatrix}$$

Therefore, we get:

$$\frac{\sqrt{n}\left(\hat{\theta}_i-\theta_{0,i}\right)}{\sqrt{A\hat{V}ar_{ii}}}\xrightarrow{d}N\left(0,1\right)$$

Hence, the 95% confidence intervals for γ_0 and δ_0 are computed as follows:

$$\hat{CI}_\alpha\left(\delta_0\right)=\left(\hat{\delta}_{GMM,2}-z_{\alpha/2}\sqrt{\frac{A\hat{V}ar_{11}}{n}};\hat{\delta}_{GMM,2}+z_{\alpha/2}\sqrt{\frac{A\hat{V}ar_{11}}{n}}\right)$$

$$\hat{CI}_\alpha\left(\gamma_0\right)=\left(\hat{\gamma}_{GMM,2}-z_{\alpha/2}\sqrt{\frac{A\hat{V}ar_{22}}{n}};\hat{\gamma}_{GMM,2}+z_{\alpha/2}\sqrt{\frac{A\hat{V}ar_{22}}{n}}\right)$$

where $1-\alpha=0.95$ so that $\alpha=0.05$ and $z_{\alpha/2}$ is the $\alpha/2$ quantile of a standard normal distribution (i.e. 1.96). The 95% confidence intervals are respectively:

$\hat{CI_{0.05}}\left(\delta_0\right)=(0.9346,1.025)$

$\hat{CI_{0.05}}\left(\gamma_0\right)=(0.99396,5.9031)$

Here follows the code for this part:

```
% Compute Asymptotic variance (G'S^(-1)G)^(-1)
% using consistent estimators G_hat and W_hat
G = [sum(exp(-theta2_gmm(2).*c_1+r_1))/T,
sum(-c_1.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_1+r_1))/T;
sum(exp(-theta2_gmm(2).*c_1+r_1+r))/T,
sum(-c_1.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_1+r_1+r))/T;
sum(exp(-theta2_gmm(2).*c_1+r_1).*(1+c))/T,
sum(-c_1.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_1+r_1).*(1+c))/T];
AVar = inv(G'*W_2*G);
delta_CI =
[-1.96*sqrt(AVar(1,1)/(T))+theta2_gmm(1)  1.96*sqrt(AVar(1,1)/(T))+theta2_gmm(1)];
gamma_CI =
[-1.96*sqrt(AVar(2,2)/(T))+theta2_gmm(2)  1.96*sqrt(AVar(2,2)/(T))+theta2_gmm(2)];
```

Part d

The Hansen-Sargan test of over-identifying restrictions tests the null hypothesis $H_0:\ \mathbb{E}\left[g\left(w_t;\theta_0\right)\right]=0$ for some $\theta_0\in\Theta$. The test statistics is:

$$J=-2nQ_n(\hat{\theta})$$

$$J\xrightarrow{d}\chi^2_{k-p}$$

Here, $Q_n(\hat{\theta})$ is the objective function computed using a $\hat{W}$ that is a consistent estimator for S^{-1} (like the two-step weighing matrix) and $\hat{\theta}$ is the estimate previously obtained. As there are 3 moments' conditions and 2 parameters δ and γ, in this case we have $k-p=1$.

Below the code for this part.

```
% Hansen-Sargan Test for over-identification
J = 2*T*Qval2;
pvalue_J = 1-chi2cdf(J,1);
```

Note that in my code I don't have the minus as I minimized the negative of Q_n and $Qval2$ is the value of the objective function in the second-step at $\hat{\theta}_{GMM,2}$. Then I compute the p-value of the test statistics from a chi-square distribution with 1 degree of freedom to get the following result:

J	$p - value$
0.13369	0.71464

The test does not reject the null hypothesis. This means that there is not enough statistical evidence to reject the null hypothesis that the moment conditions hold and, therefore, that there is not enough statistical evidence to reject the hypothesis that the model fits the data.

Part e

The procedure is the same as above, except that now $z_t = [1, R_t, (1 + d_t), (1 + c_t), R_{t-1}, (1 + d_{t-1}), (1 + c_{t-1})]'$ so that we have seven instruments and that here we lose two observations at the end so $T = 248$. I repeat exactly the steps above with this new set of instruments (now the weighting matrix and the matrix of derivatives is going to be 7x7 and $k - p = 7 - 2 = 5$) with the following code:

```
%% Exercise 1 - Second specification for z
clear;
% Import the .csv file
M = csvread('hsdata.csv',1,0);
t = csvread('hsdata.csv',1,0,[1,0,250,0])';
c = csvread('hsdata.csv',1,1,[1,1,250,1])';
d = csvread('hsdata.csv',1,2,[1,2,250,2])';
r = csvread('hsdata.csv',1,3,[1,3,250,3])';
% Balance the observations
for i=1:(length(c)-2)
c_1(i)=c(i+1);
r_1(i)=r(i+1);
d_1(i)=d(i+1);
c_2(i)=c(i+2);
r_2(i)=r(i+2);
d_2(i)=d(i+2);
end
r(250)=[];c(250)=[];d(250)=[];
r(249)=[];c(249)=[];d(249)=[];
T = length(c);
z=[ones(1,T); exp(r_1); (1+d_1); (1+c_1);exp(r); (1+d); (1+c)];
% Compute GMM - first step
W_1=eye(7);
```

```
fun1 = @(x)objective(x, W_1, z, T, c_2, r_2, r_1, c_1);
x0=[0.9 0.9];
[theta1_gmm, Qval1] = fminsearch(fun1, x0);
% Compute GMM - second step
% Euler conditions at parameters estimated in step 1
eul = ee(theta1_gmm, z, T, c_2, r_2, r_1, c_1);
% 2nd-step weighting matrix - the long-run variance, with lags (b/c dependent data)
W_2=inv(longrunW(eul, T, 12));
fun2 = @(x)objective(x, W_2, z, T, c_2, r_2, r_1, c_1);
[theta2_gmm, Qval2]= fminsearch(fun2, theta1_gmm);
% Compute Asymptotic variance (G'S^(-1)G)^(-1)
% using consistent estimators G_hat and W_hat
G = [sum(exp(-theta2_gmm(2).*c_2+r_2))/T,
sum(-c_2.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_2+r_2))/T;
sum(exp(-theta2_gmm(2).*c_2+r_2+r_1))/T,
sum(-c_2.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_2+r_2+r_1))/T;
sum(exp(-theta2_gmm(2).*c_2+r_2).*(1+d_1))/T,
sum(-c_2.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_2+r_2).*(1+d_1))/T;
sum(exp(-theta2_gmm(2).*c_2+r_2).*(1+c_1))/T,
sum(-c_2.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_2+r_2).*(1+c_1))/T;
sum(exp(-theta2_gmm(2).*c_2+r_2+r))/T,
sum(-c_2.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_2+r_2+r))/T;
sum(exp(-theta2_gmm(2).*c_2+r_2).*(1+d))/T,
sum(-c_2.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_2+r_2).*(1+d))/T;
sum(exp(-theta2_gmm(2).*c_2+r_2).*(1+c))/T,
sum(-c_2.*theta2_gmm(1).*exp(-theta2_gmm(2).*c_2+r_2).*(1+c))/T];

AVar = inv(G'*W_2*G);

delta_CI =
[-1.96*sqrt(AVar(1,1)/(T))+theta2_gmm(1) 1.96*sqrt(AVar(1,1)/(T))+theta2_gmm(1)];
gamma_CI =
[-1.96*sqrt(AVar(2,2)/(T))+theta2_gmm(2) 1.96*sqrt(AVar(2,2)/(T))+theta2_gmm(2)];

% Hansen-Sargan Test for over-identification
J = 2*T*Qval2;
pvalue_J = 1-chi2cdf(J,5);
```

I obtain the following results:

$\hat{\theta}_{GMM,1} = (\hat{\delta}_{GMM,1}, \hat{\gamma}_{GMM,1}) = (0.997, 4.5761)$.

$\hat{\theta}_{GMM,2} = (\hat{\delta}_{GMM,2}, \hat{\gamma}_{GMM,2}) = (0.9781, 3.38)$.

$\hat{CI}_{0.05}\,(\delta_0) = (0.94191, 1.0143)$

$\hat{CI}_{0.05}\,(\gamma_0) = (1.4252, 5.3348)$

$J = 0.50219$

$p - value = 0.99204$

Overall, adding 4 instruments leads to almost the same point estimates as in the previous case, but confidence intervals are now narrower. Moreover, the test-statistic is higher and the p-value even higher than before, meaning that the test again fails to reject the null. Overall, results seems to be more accurate with 7 instruments. We expect this as adding more information to the model through additional instruments increases the accuracy of the estimates (provided that the added restrictions are correct - as it seems from the fact that the p-value of the Hansen-Sargan test is higher). It is also worth to notice that the confidence interval for γ_0 is quite large so that might imply less accurate results for this parameter.

Exercise 2

Part a

I define the following matrices:

$$A = \begin{bmatrix} 1 & 0 \\ -\gamma & 1 \end{bmatrix}; \quad B = \begin{bmatrix} \alpha & \beta \\ 0 & 0 \end{bmatrix}; \quad C = \begin{bmatrix} \mu_c \\ \log\delta + \frac{1}{2}\sigma^2(\gamma) \end{bmatrix}; \quad x_t = \begin{bmatrix} c_t \\ r_t \end{bmatrix}$$

(the other martices and expressions are the same as in the text). Then the system becomes:

$$Ax_{t+1} = Bx_t + C + \varepsilon_{t+1}$$

$$\Rightarrow Ax_{t+1} - Bx_t - C = \varepsilon_{t+1}$$

given that $\varepsilon_{t+1} \overset{iid}{\sim} N(0, \Sigma)$, from the above we conclude:

$$(Ax_{t+1} - Bx_t - C) \overset{iid}{\sim} N(0, \Sigma)$$

that is:

$$\begin{bmatrix} 1 & 0 \\ -\gamma & 1 \end{bmatrix} \begin{bmatrix} c_{t+1} \\ r_{t+1} \end{bmatrix} - \begin{bmatrix} \alpha & \beta \\ 0 & 0 \end{bmatrix} \begin{bmatrix} c_t \\ r_t \end{bmatrix} - \begin{bmatrix} \mu_c \\ \log\delta + \frac{1}{2}\sigma^2(\gamma) \end{bmatrix} =$$

$$\begin{bmatrix} c_{t+1} - \alpha c_t - \beta r_t - \mu_c \\ -\gamma c_{t+1} + r_{t+1} - \log\delta - \frac{1}{2}\sigma^2(\gamma) \end{bmatrix} \overset{iid}{\sim} N(0, \Sigma)$$

is distributed as a bivariate normal with zero mean and variance Σ. Therefore, given that the process is *iid* we can use the log-likelihood as the sample criterion function. That is:

$$Q_n(\theta) = \frac{1}{n} \sum_{t=1}^{n} m(w_t; \theta)$$

$$m(w_t; \theta) = \log f((x_t, x_{t+1}); \theta)$$

where $f((x_t, x_{t+1}); \theta)$ is a multivariate (in this case bivariate) normal distribution, so that:
$f((x_t, x_{t+1}); \theta) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left\{-\frac{1}{2}\left(Ax_{t+1} - Bx_t - C\right)' \Sigma^{-1} \left(Ax_{t+1} - Bx_t - C\right)\right\} \Rightarrow \log f((x_t, x_{t+1}); \theta) = -\frac{1}{2}\log\left(|2\pi\Sigma|\right) - \frac{1}{2}\left(Ax_{t+1} - Bx_t - C\right)' \Sigma^{-1} \left(Ax_{t+1} - Bx_t - C\right)$

The code below is for the separate program that computes the sample criterion function we want to maximize (we actually minimize its inverse). First, given that the vector of parameters to be estimated is $\theta = (\delta, \gamma, \alpha, \beta, \mu_c, \sigma_c, \sigma_r, \rho)$, I replace in the expressions above the right component of θ in the code. Then I unstack my data in the right format, I compute the sample criterion function and finally I compute the sample average of the criterion function.

```
%% This program computes the objective function for a bivariate normal
%%OUTPUTS%% % Qn = average log-likelihood for n iid data
 function Qn = like_bivnorm(theta, data);
 sigma=[theta(6)^2,theta(8)*theta(7)*theta(6);theta(8)*theta(7)*theta(6),theta(7)^2];
A=[1,0;-theta(2),1];
B=[theta(3),theta(4);0,0];
s_gamma=[-theta(2),1]*sigma*[-theta(2),1]';
m=[theta(5);log(theta(1))+(1/2)*s_gamma];
y=[data(:,2)';data(:,4)'];
x=[data(:,1)';data(:,3)'];
[row,col]=size(data);
Qn=zeros(row,1);
 for i=1:row
Qn(i,1)=(1/2)*log(det(2*pi*sigma)) +
 (1/2)*((A*y(:,i)-B*x(:,i)-m)'*(sigma^(-1))*(A*y(:,i)-B*x(:,i)-m)); %max the negative
 end
Qn = sum(Qn)/249;
 end
```

Part b

Again, after having imported the data and balanced the dataset (we again lose one observation, hence $T = 249$ - again n in the formulas is this T) we maximize the sample criterion function described in the previous point. Note that $\theta = (\delta, \gamma, \alpha, \beta, \mu_c, \sigma_c, \sigma_r, \rho)$ hence the ML estimator is going to return a vector of 8 parameters, $\hat{\theta} = \left(\hat{\delta}, \hat{\gamma}, \hat{\alpha}, \hat{\beta}, \hat{\mu_c}, \hat{\sigma_c}, \hat{\sigma_r}, \hat{\rho}\right)$. I use *fminunc* where as starting values I use the following: for δ and γ values close to the estimates obtained in the previous exercise, for α and β from the estimated coefficients of a regression of c_{t+1} on c_t and r_t (this follows from the first line of the system in matrix form, i.e. $c_{t+1} = \alpha c_t + \beta r_t + \tilde{\varepsilon}_{t+1}$), for μ_c the sample mean of c_t, for σ_c and σ_r the sample standard deviations of, respectively, c_t and r_t and for ρ the sample correlation between c_t and r_t. Minimization leads to the following estimates:

$$\hat{\theta}_{ML} = (0.99375, 3.5182, -0.20637, 0.00073183, 0.021123, 0.032937, 0.12288, -0.27542)$$

Now, as we know that the ML estimator is asymptotically efficient under correct specification, which we assume in this model (I performed a robustness test, not reported, with the matrix $H^{-1}\Sigma H^{-1}$, which shows that the results are similar so we can use the inverse of the Fisher information), we have that the asymptotic distribution of the estimator is:

$$\sqrt{n}\left(\hat{\theta}_{ML} - \theta_0\right) \xrightarrow{d} N\left(0, \mathbb{I}(\theta_0)^{-1}\right)$$

In this case, $\mathbb{I}(\theta_0)^{-1} = -H^{-1}$. We use the consistent estimator $\mathbb{I}(\hat{\theta_0})^{-1} = -\hat{H}^{-1}$. I used the

MATLAB program HessMp to compute the negative of the Hessian of the criterion function at the ML estimate. Then I take the inverse to get the asymptotic variance. From this I compute the confidence intervals in the following way:

$$\frac{\sqrt{n}\left(\hat{\theta}_i - \theta_{0,i}\right)}{\sqrt{\mathbb{I}(\hat{\theta_0})^{-1}{}_{ii}}} \xrightarrow{d} N(0,1)$$

Hence, the 95% confidence intervals for γ_0 and δ_0 are computed as follows:

$$\hat{CI}_\alpha(\delta_0) = \left(\hat{\delta}_{ML} - z_{\alpha/2}\sqrt{\frac{\mathbb{I}(\hat{\theta_0})^{-1}_{11}}{n}}; \hat{\delta}_{ML} + z_{\alpha/2}\sqrt{\frac{\mathbb{I}(\hat{\theta_0})^{-1}_{11}}{n}}\right)$$

$$\hat{CI}_\alpha(\gamma_0) = \left(\hat{\gamma}_{ML} - z_{\alpha/2}\sqrt{\frac{\mathbb{I}(\hat{\theta_0})^{-1}_{22}}{n}}; \hat{\gamma}_{ML} + z_{\alpha/2}\sqrt{\frac{\mathbb{I}(\hat{\hat{\theta_0}})^{-1}_{22}}{n}}\right)$$

where $1-\alpha = 0.95$ so that $\alpha = 0.05$ and $z_{\alpha/2}$ is the $\alpha/2$ quantile of a standard normal distribution (i.e. 1.96). The 95% confidence intervals are respectively:

$\hat{CI}_{0.05}(\delta_0) = (0.92959, 1.0579)$

$\hat{CI}_{0.05}(\gamma_0) = (1.2356, 5.8008)$

As before, the confidence interval for γ_0 is quite large, meaning that the results for this parameter may be less accurate.

```
%% Exercise 2
clear;
% Import the .csv file
M = csvread('hsdata.csv',1,0);
t = csvread('hsdata.csv',1,0,[1,0,250,0])';
c = csvread('hsdata.csv',1,1,[1,1,250,1])';
d = csvread('hsdata.csv',1,2,[1,2,250,2])';
r = csvread('hsdata.csv',1,3,[1,3,250,3])';
% Balance the observations
for i=1:(length(c)-1)
c_1(i)=c(i+1);
r_1(i)=r(i+1);
end
r(250)=[];c(250)=[];d(250)=[];
T = length(c);
% Min the criterion function
%theta=(delta,gamma,alpha,beta,mu_c,sigma_c,sigma_r,rho)
data=[c' c_1' r' r_1'];
b=regress(c_1',[c' r']); %find alpha and beta starting values
x0=[0.9 3 b(1) b(2) mean(c) std(c) std(r) corr(c',r')];
options = optimset;
[theta_ml] = fminunc(@like_bivnorm,x0,options,data);
delta=theta_ml(1);gamma=theta_ml(2);
```

```
alpha=theta_ml(3);beta=theta_ml(4); mu_c=theta_ml(5);sigma_c=theta_ml(6);
sigma_r=theta_ml(7);rho=theta_ml(8);
% Compute Hessian
H = HessMp(@like_bivnorm,theta_ml,data) % this returns the negative of the Hessian
% Asymptotic variance
AVar = inv(H);
% Confidence intervals
delta_CI =
[theta_ml(1,1)-1.96*sqrt(AVar(1,1)/(T)) theta_ml(1,1)+1.96*sqrt(AVar(1,1)/(T))];
gamma_CI =
[theta_ml(1,2)-1.96*sqrt(AVar(2,2)/(T)) theta_ml(1,2)+1.96*sqrt(AVar(2,2)/(T))];
```

Part c

The following table sums up the results:

	$\hat{\delta}$	$\hat{\gamma}$	$\hat{CI}_{0.05}(\delta_0)$	$\hat{CI}_{0.05}(\gamma_0)$
GMM-two-step, 3 instruments	0.97982	3.4485	$(0.9346, 1.025)$	$(0.99396, 5.9031)$
GMM-two-step, 7 instruments	0.9781	3.38	$(0.94191, 1.0143)$	$(1.4252, 5.3348)$
ML	0.99375	3.5182	$(0.92959, 1.0579)$	$(1.2356, 5.8008)$

Point estimates for the parameters are really similar in all the three specifications. As pointed out before, the GMM estimates with 7 instruments seems more accurate regarding the confidence intervals. The ML estimates, on the other hand, have larger confidence intervals, especially for γ. It seems therefore that the GMM estimates seems more credible. In particular, it is probable that the ML is not really accurate because the model is not well specified (this VAR assumes that the innovations are iid but as shown in part (a) they are function of the difference in consumption - at least the first line - which may not be iid). Moreover, the normality assumption on the shocks may not be an accurate way of modelling the expressions obtained in the 2x1 vector in part (a). In conclusion, while the ML method provides more structure on the model (pdf is needed) it may lead to better estimates than the GMM if the model is correctly specified, as it would provide “more” information with its structure but in turn this may lead to bad accuracy if the model is misspecified, in which case we may prefer GMM (as in this case).