GMM: A User Guide

Ogaki, Masao

Working Paper No. 348
April 1993

GMM: A USER GUIDE

Masao Ogaki

University of Rochester

Working Paper No. 348

April 1993

This is a user's guide for Hansen/Heaton/Ogaki GMM package written in GAUSS. Hansen's work for the package was supported by NSF SES-8512371 and Ogaki's work was supported by NSF SES-9213930.

Table of Contents

1. Introduction

This is a user's guide for Hansen/Heaton/Ogaki GMM package written in GAUSS. This package contains programs to implement Hansen's (1982) GMM and its extensions described in the companion paper, Ogaki (1993).

2. What the User Should Know about GAUSS

This section covers essential features of GAUSS you must know to run the GMM programs. Read the GAUSS manual for more advanced features of GAUSS. I assume that GAUSS has already been properly installed. A square bracket indicates a key to be pressed. For example, press the Enter key for the notation [Enter]. If two keys are indicated in a bracket as [Alt-F2], press [Alt] and [F2] keys simultaneously.

2.A. Starting GAUSS

From the DOS prompt, type GAUSSI [Enter] to start GAUSS (for earlier versions, you may have to type GAUSS or GAUSS386 instead of GAUSSI). You will be in the COMMAND mode of GAUSS and see the GAUSS prompt, >>. In the COMMAND mode, you can execute screen-resident programs.

2.B. Exiting GAUSS

To exit GAUSS, press [Esc]. Then type Y at the prompt.

2.C. Running a Program Stored in a File from the COMMAND Mode

From the GAUSS prompt, >>, type RUN FILENAME.EXP [F4][F2] to run a file named FILENAME.EXP for example.

2.D. Editing a File

From the GAUSS prompt, type EDIT FILENAME.EXP [F4][F2] to edit a file named FILENAME.EXP for example. You will be in the EDIT mode of GAUSS. You can edit the file with a full screen editor.

You can move around the file using arrow keys and [Pg Dn] [Pg Up] keys that are usually at the right of the keyboard. You can edit the file by deleting letters using [Del] key and typing in letters.

2.E. Rules of Syntax

This section lists some of the general rules of syntax for GAUSS programs.

2.E.1. Statements

A GAUSS program consists of a series of statements. A statement is a complete expression or command. Statements in GAUSS ends with a semicolon.

2.E.2. Case

GAUSS does not distinguish between upper and lower case except inside double quotes.

2.E.3. Comments

Comments can be placed inside /* and */, which can nest other comments or inside @ and @, which cannot nest other comments.

2.F. Reading and Storing Data

```
LOAD X[n,m]=FILENAME.DAT;
```

reads in data stored in a ASCII file named FILENAME.DAT for example. This data file should contain data separated by spaces in the form of n x m matrix.

If X is a matrix of numbers in GAUSS,

```
SAVE XFILE=X;
```

stores X into a file named XFILE.FMT. Then you can read in the data again by

```
LOAD X=XFILE;
```

2.G. Operators

2.G.1. Operators for Matrix Manipulations

Assignment operator:

Assignments are done with one equal sign.

```
Y=3;
```

assigns the value 3 to 1 x 1 matrix Y.

Indexing operator:

Brackets [] are used to index matrices.

```
Y=X[3,3];
```

assigns 3-3 element of X to Y. Commas are used to separate row indices from column indices. A vector can take one argument.

Period:

Dots are used in brackets to signify "all rows" or "all columns".

```
Y=X[.,3];
```

assigns the third column of X to Y.

Colon:

A colon is used within brackets to create a continuous range of indices.

```
Y=X[1:5,.];
```

Transpose operator:

' transposes matrices.

Vertical Concatenation:

| is used to concatenate two matrices vertically.

```
Z=X|Y;
```

Horizontal Concatenation:

~ is used to concatenate two matrices horizontally.

```
Z=X~Y;
```

2.G.2. Numeric Operators

Usual Operators:

Usual operators in GAUSS work according to standard rules of matrix algebra. For example, * is the operator for matrix multiplication, and

```
Y=X*Z;
```

performs matrix multiplication when X and Z are conformable in the sense of matrix algebra.

Element by Element Operators:

In some applications, X is a m x n matrix, and Y is a m x 1 vector, and it is convenient to multiply each row of Y with each of the n elements in each row of X. The Element by Element Operators allow you to perform such operations. For example, .* is the element by element multiplication operator, and

```
Z=X.*Y.
```

performs the operation described above. Other element by element operators are the following:

./ Element by element division:

```
Y=X./Z;
```

^ Element by element exponetiation:

```
Y=X^Z;
```

+ Addition:

```
Y=X+Z;
```

performs element by element addition.

- Subtraction:

```
Y=X-Y;
```

performs element by element subtraction.

2.H. GAUSS Commands

2.H.1. Printing

```
PRINT X Y;
```

will print matrices X and Y to the screen. Instead of matrices, you can print words inside double quotes:

```
PRINT "This will be printed";
```

You can use ? instead of PRINT:

```
? X Y;
```

2.H.2. Preparing an Output File

```
OUTPUT FILENAME.OUT RESET;
```

allows you to write the output of PRINT statements to a file named FILENAME.OUT, for example. To print out or edit the output file, you have to close the output file by

```
OUTPUT OFF;
```

command. Most of the programs in the GMM package contains this statement toward the end of them. However, if your program does not reach its end because of errors, you have to issue this command from the COMMAND mode to close the file to check the output file.

3. Which Program Should be Used?

The GMM package contains many programs for different purposes, though

GMM.SET is a general nonlinear GMM program that can be used for most applications of GMM. This section provides brief descriptions of the programs in the GMM package. The documentation about how to use programs are either in *.EXP files or in the programs themselves.

3.A. GMM.SET

GMM.SET is a general nonlinear GMM program that implements GMM estimation explained in Section 2 of Ogaki (1993) with options of the truncated kernel estimator or the QS kernel estimator (both nonprewhitened and VAR prewhitened) for estimation of Ω explained in Ogaki (1993, Section 6). This program can be used for linear GMM models in Ogaki (1993, Subsections 3.1 and 3.2), too. When this program is used for linear models, it is much faster to set hflag=2 (see the description of MINQUAD.SET in the next section for an explanation of this variable). It is even faster to use one of LGMM*.* programs for linear models. Another advantage of using one of the LGMM*.* programs is that some of them allow you to impose conditional homoskedasticity on the estimator for Ω while the GMM.SET program always uses conditional heteroskedasticity consistent estimator (see Ogaki (1993, Section 6)).

The GMM.SET program can be used to implement the Minimum Distance Estimation explained in Ogaki (1993, Section 4.4)). For the linear version of the minimum distance estimation, LMDE.SET can be used, too.

GMM.SET uses MINQUAD.SET for nonlinear search. GMM.EXP is an example file and the user can modify this example file for his or her own problem. This example uses data in GMMQ.DAT. The example in GMM.EXP is a single return model of Hansen and Singleton (1982). GMMHF.EXP is another example file for GMM.SET for a more complicated asset pricing model based on habit formations estimated by Ferson and Constantinides (1991). This file is a modification of the program used by Cooley and Ogaki (1993) for multiple return.

3.B. GMMQ.SET

GMMQ.SET is the same as GMM.SET except that it allows the user to use different estimation methods of Ω. Specifically, the GMMQ.SET allows the user to specify different orders of MA for different disturbances when the truncated kernel estimator or the modified Durbin's method is used. In contrast, the user must use the same order of MA for all disturbances for these estimators. On the other hand, the user can use the QS kernel estimator with the GMM.SET but not with the GMMQ.SET. GMMQ.EXP is an example file and

3.C. MINQUAD.SET

MINQUAD.SET defines a program that minimizes a nonlinear function of quadratic form. This program is used by GMM.SET and GMMQ.SET and can be used for other purposes when a quadratic form nonlinear function is minimized.

Section 5.A describes an example of how to control MINQUAD nonlinear search by hitting keys during the search with the GMMHF.EXP program.

3.D. GMM2S.SET

This program defines a procedure for sequential (or two step) GMM estimation explained in Ogaki (1993, Section 4.1).

4. How to Use Programs

The GMM.EXP file printed as Appendix A below is an example file and contains documentation for GMM.SET. It is probably the easiest to modify the file for your particular applications. After modifying GMM.SET, you can change the name of the file by hitting Alt-O (Alt and O keys together).

When GMM.EXP is run, an output file GMMQ.OUT is created. This output file is printed in Appendix B. In some applications, GMM estimation takes many iterations of nonlinear search and you may wish to print out only part of the output file. You can do this by marking the part of the output file by hitting Alt-L and then Alt-P.

Other programs work in similar ways and instructions for a program is either in an example file with a name *.EXP or in the program file.

5. Hints

5.A. A Priori Information about Parameter Values: An Example from a Habit Formation Model

In some applications, the econometrician has a priori information about parameters. An extreme case is that some parameter values are not admissible. Another example is that certain parameter values are not very plausible. In the GMM program, it is possible (and sometimes necessary) to incorporate such information.

The GMM estimation is based on moment restrictions of the form (in the following, Ogaki's (1993) notation is used).

(1) $$E(f(X_t, \beta_0)) = 0.$$

Let $\phi(\beta)$ be a real valued function. Then one can define a new function $f^*(X_t, \beta)=f(X_t, \beta)\phi(\beta)$, and we still have moment restrictions

(2) $$E(f^*(X_t, \beta_0)) = 0.$$

One can apply GMM to moment restrictions (2) instead of (1).

In many applications, unconditional moment restrictions (1) come from conditional moment restrictions as explained in Ogaki (1993, Section 3.3). In such applications, even random variables can be used to normalize the GMM disturbance function. Let $g(x_t,\beta)$ be a k-dimensional vector of functions and $e_t=g(x_t,\beta_0)$. Suppose that the econometrician has conditional moment restrictions, $E[e_t|I_t]=0$. If y_t is a vector of random variables in I_t and if $\phi(y_t,\beta)$ is a (measurable) real valued function, then one can define a new function $g^*(x_t,\beta)=g(x_t,\beta)\phi(y_t,\beta)$. Because $\phi(y_t,\beta)$ is in I_t, $g^*(x_t,\beta_0)$ still satisfies conditional moment conditions $E[g^*(x_t,\beta)|I_t]=0$. Thus the GMM estimation can be applied to $g^*(x_t,\beta)$.

These ideas are now illustrated for Ferson and Constantinides's (1991) asset pricing model with habit formation described in Ogaki (1993, Section 8.1.3). In their model, the Euler equation implies $E(e_t|I_t)=0$, where $e_t=\{\delta(S_{t+1}^{-\alpha}+\delta a_1 S_{t+2}^{-\alpha})R_{t+1}-(S_t^{-\alpha}+\delta a_1 S_{t+1}^{-\alpha})\}/\{S_t^{-\alpha}[1+\delta a_1]\}$.[1] In the GMM package, the GMMHF.EXP file (Appendix C), which is a minor modification of a program used by Cooley and Ogaki (1993), is included to give an example of programing.

When you run GMMHF.EXP, the initial GMM estimsyion with the identity distance matrix does not converge easily. When Step Size becomes very small, it is recommended to hit the mumber key "1" on the key board to try the DFP update instead of the outer product method (see MINQUAD.SET file for explanations). In the example output file preinted in Appendix D, the number key "1" was hit after the nonlinear search iteration number 28. After this, the program converged at the iteration number of 60. It is also possible to hit "C" to force a termination of the nonlinear search when the

[1]Here, the disturbance is normalized by $\{S_t^{-\alpha}[1+\delta a_1]\}$ to avoid clear violations of the stationarity and identification assumptions.

convergence criteria are too strict for a particular application. Because the outer product mehtod is usually better, it is recommended to go back to it for another iteration. In the example output file, the number key "2" was hit after 6 nonlinear search iterations in the second GMM iteration.

One problem that researchers have encountered in these applications is that $C_t+a_1C_{t-1}$ may be negative when a_1 is close to minus one. The values of a_1 which makes $C_t+a_1C_{t-1}$ negative are not admissible. In order to incorporate this a priori information, one can program the *hu* procedure in GMM.EXP, so that very large numbers are returned as the values of *hu* when a_1 falls to the nonadmissible region. It is necessary to modify the numerical derivative procedure GRAD2.PRC in order to prevent these fictitious large numbers are used to calculate numerical derivatives. An example is in GRADQG.PRC, which is used by GMMHF.EXP.

When a_1 is positive and is greater than one, we can obtain a well behaved utility function. However, one may argue that it is not plausible for a_1 to be greater than one: why is the previous period's consumption more important than this period's consumption for this period's utility level? One way to incorporate this type of a priori information is to use a Bayesian method. Another way is to penalize such implausible parameter values in defining the GMM disturbance. Let $f(X_t,\beta)$ be the original GMM disturbance function, where $\beta=(\delta,a_1,\alpha)$. Then define

$$(3) \qquad \phi(\beta) = \begin{cases} 1 & \text{if } a_1 \leq 1 \\ (a_1-1)^2+1 & \text{if } a_1 \leq 1 \end{cases}$$

and $f^*(X_t,\ \beta)=f(X_t,\ \beta)\phi(\beta)$. Now $f^*(X_t,\ \beta)$ can be used as the new GMM disturbance. Here the function ϕ is designed so that it is differentiable at $a_1=1$ and it does not affect the function f when $a_1\leq 1$. The latter feature is important because of small sample considerations. Even though any other differentiable function can be used as the normalization function ϕ without disturbing the consistency of the GMM estimator, the small sample properties of GMM estimator will be affected by normalizations. It is thus desirable not to disturb the GMM function for the parameter region where we do not have any a priori information.

One might argue that the curvature parameter α, which will be close to the Relative Risk Aversion parameter as explained by Ferson and Constantinides under some circumstances, is not likely to be greater than ten (see Mehra and Prescott (1985)). This restriction can be incorporated by a normalization similar to (3) as in the GMMHF.EXP program.

5.B. Minimum Distance Estimation

In order to implement minimum distance estimation (MDE) described in Ogaki (1993, Section 4.4), set tend=1, w0flag=2, and kgm=0 (note that kgm is only used to make a small sample degree of freedom adjustment for the estimation of Ω, which is not necessary for MDE). Assuming that you have an estimate of Ω in a GAUSS matrix file named omegamde.fmt, include a line

```
load w0=omegamde;
```

before the GMM procedure is called in the program. If GMM.EXP is modified, the third line from the bottom needs to be modified: the fifth argument in gmm($\cdot$), rows(bgm) needs to be changed into kgm.

5.C. Test Statistics

Wald test statistic can be constructed from BGM.FMT (containing GMM estimate) and VAR.FMT (containing the estimated covariance matrix for the GMM estimator). For nonlinear restrictions, Wald tests are not recommended (see Ogaki 1993, Section 7).

In order to calculate likelihood ratio type test statistic (see Ogaki 1993, Section 7), you first perform restricted estimation by programing the *hu* procedure that defines the GMM disturbance with the restrictions imposed. If there are *p* parameters and *s* restrictions in the econometric model, you will have *p-s* parameters in your *bgm* after imposing the restrictions. In the GMM.EXP file, Hansen's *J* statistic, TJ_T is returned as a global variable *chi*. Save this variable. Second, you perform unrestricted estimation by modifying the *hu* procedure used in the first step to estimate all *p* parameters. In this restricted estimation, make sure that you set w0flag=2 and include a statement "load w0;" anywhere before the *gmm* procedure defined by GMM.SET is called. For initial values of *bgm*, it is a very good idea to use the parameter values implied by the restricted estimates. Alternatively, you can perform the restricted estimations first and then the unrestricted estimation.

References

Cooley, T.F. and M. Ogaki (1993): "A Time Series Analysis of Real Wages, Consumption, and Asset Returns: A Cointegration-Euler Equation Approach," Rochester Center for Economic Research Working Paper No. 285R, University of Rochester.

Ferson, W.E. and G.M. Constantinides (1991): "Habit Persistence and Durability in Aggregate Consumption," *Journal of Financial Economics* 29, 199-240.

Hansen, L.P. (1982): "Large Sample Properties of Generalized Method of Moments Estimators," *Econometrica* 50, 1029-1054.

Hansen, L.P. and K.J. Singleton (1982): "Generalized Instrumental Variables Estimation of Nonlinear Rational Expectations Models," *Econometrica*, 50, 1269-1286.

Mehra, R. and E.C. Prescott (1985): "The Equity Premium: A Puzzle," *Journal of Monetary Economics* 15, 145-161.

Ogaki, M. (1993): "Generalized Method of Moments: Econometric Applications," in G.S. Maddala, C.R. Rao, and H.D. Vinod, eds., *Handbook of Statistics, Vol. 11*. Elsevier Science, in press.

APPENDIX A GMM.EXP FILE

```
@ GMM.EXP @
@------------------------------------------------------------------------------
Prepared by Lars P. Hansen, John C. Heaton, and Masao Ogaki
    Financial support from the National Science Foundation
         Grant numbers: SES-8512371 and SES-9213930

 Last Revision:  04/07/93
------------------------------------------------------------------------------@
/*
This program has been used and seems to be free of errors. However,
we (Lars P. Hansen, John C. Heaton, and Masao Ogaki) do not assume
responsibility for any remaining errors.  In no event shall we be liable to
for any damages whatsoever arising out of the use of or inability to use
this program.

This program is for the generalized method of moments (GMM) procedure.
See "Generalized Instrumental Variable Estimation of Nonlinear Rational
Expectations Models," by L.P. Hansen and K.J. Singleton (Econometrica
1982, Vol.50, 1269-1286).  This example file is for Hansen and Singleton
model, but uses different data than theirs.

   @@ To run this example file, type
        RUN MINQUAD.SET [F4][F2]
        RUN GMM.SET [F4][F2]
        RUN GMM.EXP [F4][F2]

      The programs MINQUAD.SET and GMM.SET define necessary procedures.
      To use these programs for the user's problem, the user can
      modify this example file for the user's problem.
      Usually the user does not have to modify MINQUAD.SET and GMM.SET
      because all the paramters for these two programs are controlled
      by this example file.

   @@ The user should go through Step 1-4 and modify this example file for
      the user's own problem.

Model:   E[h(t,b0)]=0:  nmr moments restrictions, h(t,b0) is nmr by 1

Algorithm:
   This program calculates and iterates the following:
     itital W0 given.

  1. g(b)=C*[h(1,b)+,...,+h(T,b)]
     b=argmin{g(t,b)'W0*g(t,b)}

  2. Rzw(j)=(1/T)[h(1,b)h(1,b)'+,..,+h(T,b)h(T-j,b)']
                             j=0,1,..,mas
     W1=Rzw(0)+{Rzw(1)+Rzw(1)'}+,..,+{Rzw(mas+1)+Rzw(mas+1)'}
     W0=inv(W1)
     (W1 can be calculated by other methods depending on the value of
      calwflag.  See below for more explanations.)

  3. go back to 1.

   Here C is a scaling multiplier, and taken to be 'const' or 'const2'
   depending on whether or not W0 is I.

OUTPUT:
        chi: chi-square test
              statistic for the overidentifying restrictioins
    The following values in GAUSS matrix files:
        a. Current parameter values in MINQUAD.SET search: crparv.fmt
        b. Current inverse hessian matrix in MINQUAD.SET search: crhess.fmt
        c. GMM estimates which uses W0=I: bgmi.fmt
        d. GMM estimates of the last GMM iteration: bgm.fmt
        e. W0 calculated from the last GMM iteration results: w0.fmt
        f. Covariance matrix for bgm: varb.fmt
*/
@ ======================== User Definition Area ============================ @
@ ********************************************************************** @
@ ---------- Step 1: Prepare Output File -------------------------@
@ ********************************************************************** @
output file=gmmq.out reset; @ Specify the name of the output file. @

? "
************************* GMM Results *********************************";

    @ Prepare the following message which will be printed
      at the top of the output file @
?"
GMMQ.OUT";
? "Hansen and Singleton Model (Single return, VWR) (Econometrica '82)";
datestr(0);
timestr(0);
? "Parameter Names  'beta,gamma'";

@ See GMMQ.EXP for an expamle with multiple returns.@

@ ********************************************************************** @
@----------- Step 2:  Define Global Variables  ----------------------- @
@ ********************************************************************** @

@----------------------------@
   @ Section A. User must specify variables in this section. @
@----------------------------@

tend=334;   @ # of observations=T; scalar @

bgm=1|1;     @ bgm=b(1)|..|b(kgm); kgm by 1 vector
                      Initial values of the coefficients @

nmr=3;           @ scalar: the number of moment restrictions@

mas=0;           @ The order of MA of the disturbance @

@---------------------------------@
     @ Section B.  User can leave the variables in this section as they are,
                  and go to STEP 3.

                  Only advanced users should modify this section.@

@---------------------------------@

gradname=&GRAD2; @ Specify the name of proc that calculates
                   the gradient dgT(b)/db. @

const=1/sqrt(tend);    @ Scaling Multiplier when W0=eye(L) @
const2=1/sqrt(tend);   @ Scaing Multiplier when W0 is not eye(L) @
```

```
                        @ These scaling multipliers should be set so that
                          the value of function (vof) in nonlinear search of
                          MINQUAD.SET is near 1.  If vof is too close to 0,
                          the search will not work properly.

                          const2=1/sqrt(tend) will generally be good. @
 w0flag=0;    @ scalar;
                 If w0flag=0, W0=I is used as the initial weighing matrix W0.

                 If w0flag=1, initial bgm is used to calculate initial W0.

                 If w0flag=2, W0 in the memory is used as initial W0.

                 If w0flag=3, W0 and bgm in the memory are used to give
                      the first GMM result.                        @

 maxitegm=5; @ Sets maximum # of iteration over weighting matrix, W0.  Set
                  w0flag=0 and maxitegm=2 to execute usual 2-Stage GMM. @

 zero=1E-2;     @ Iteration over W0 continues until the maximum
                  difference of the current and the previous W0 in
                  absolute value becomes less than 'zero', or
                  the # of iteration exceeds maxitegm. @

 calwflag=0;  @ This variable is used to choose the method to calculate
                the distance matrics, W0.

                If calwflag=0, Durbin's method will be used when W0
                    is singular.

                If calwflag=1, the QS kernel estimator (nonprewhitened
          or prewhitened) will be used when W0 is singular.

                If calwflag=3, Durbin's method will be used.

                If calwflag=4, the QS kernel estimator (nonprewhitened
                 or prewhitened) will be used.
                     @

ordard=mas+3; @ Order of AR representation for
                 Durbin's method @

   @The following four variables control the QS kernel estimator.@

st=0;    @ A scalar to control the bandwidth parameter for the QS kernel.
          if st=0, then an automatic bandwidth estimator is used.
              if st/=0, then st is used as the bandwidth parameter.@

wav=ones(nmr,1); @nmr by 1 vector; weights given to the
      a-th element of z(t)w(t) for the automatic bandwidth estimator
          (used only if st=0). @

maxd=0;  @  scalar:
               if maxd=0, then nonprewhitened HAC with the QS kernel is used.
          if maxd>0, then the elements of DeltaLS with the absolute
                 value greater than maxd is replaced by maxd.  See Andrews
                 and Monahan's (1990) footnote 4. @

bst=10e+5;  @ scalar:
     When automatic bandwidth parameter is calculated to be bigger than
     bst, bst is used.  @

    @ See MINQUAD.SET for the following globals @
hflag=2;
dfpflag=1;
sstol=1e-25;
    @ See MAXMUM.DOC on MODULE9 of GAUSS for the following globals @
gradtol=1e-5;
btol=1e-5;
typf=1;
typb=1;

@ ************************************************************** @
@ ----------- Step 3: READING IN DATA ------------------------- @
@ ************************************************************** @

load x[335,3]=gmmq.dat;
c=x[1:tend+1,1];     @ c(t)/c(t-1) @
re=x[1:tend+1,2];    @ vwr @
rf=x[1:tend+1,3];    @ T-Bill rate @
clear x;

@ ******************************************************************* @
/* -------------------------------------------------------------------
          Step 4: Define the proc hu(b) that returns tend by L matrix
                            |[z(1)w(1)]'       |
                            |     |            |
                            |[z(tend)w(tend)]'|

                     where [z(t)w(t)]'=[w1(t)z1(t)',...,wnw(t)znw(t)']
---------------------------------------------------------------------- */
@ ********************************************************************* @

@ The proc must have the name "hu", and should take only one argument. @

@ Note that the proc "hu" will be called many times in GMM procedure.
  If some calculations can be done outside the proc, that will speed up
  the program. For example, here vectors of instrumental variables are
  constructed outside the proc. @

  @ Constructing instrumental variables @

  zp=ones(tend,1)~c[1:tend,.]~re[1:tend,.];

  @ Now define the proc "hu" @

proc hu(b);
 local ps,w1,w2,zw1,zw2;

   ps=b[1].*(c[2:tend+1,.]^(-b[2]));
   w1=ps.*re[2:tend+1,.]-1;
   zw1=zp.*w1;

 retp(zw1);
endp;

@========== The User does not have to change the code below. ============= @

@-------------------- Print results of estimation-------------------@
proc (0)=prntrslt(x,s);
```

```
? "
-------------- Minimization Results --------------------
Step size: " s "
Value of the objective function: " vof "
Minimiser is"  x' "
-------------------------------------------------------";
endp;

@ ------------------------------------------------------------------------- @
@ ************************************************************************ @
@ THE PROGRAM STARTS @

chi=gmm(gradname,tend,nmr,mas,rows(bgm),zero,maxitegm,st,wav,maxd,bst);
output off;

@ ---------------------- END OF PROGRAM ------------------------------- @
```

APPENDIX B   GMM.EXP OUTPUT

```
************************** GMM Results *********************************

GMMQ.OUT
Hansen and Singleton Model (Single return, VWR) (Econometrica '82)
 4/15/93
 9:48:25
Parameter Names  'beta,gamma'
Initial values of the coefficients=      1.0000000        1.0000000
mas=       0.0000000
Initial W0=I
Scaling Constant used for first iteration=     0.054717566

initial function value is     0.0043337597
------------- Iteration number        1.0000000  ---------------
Step Size        1.0000000
Value of objective function    2.0669745e-05
Current parameter values:        1.0095215        5.3628787
Current relative gradients:        11.225721      0.029825644
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------- Iteration number        2.0000000  ---------------
Step Size        1.0000000
Value of objective function    1.7380173e-07
Current parameter values:        1.0093484        5.3516703
Current relative gradients:       0.76170597     0.0019775884
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------- Iteration number        3.0000000  ---------------
Step Size        1.0000000
Value of objective function    1.7380172e-07
Current parameter values:        1.0093483        5.3516375
Current relative gradients:    1.7633947e-05    4.5783138e-08
Outer product used for Hessian: DFP=1, Outer Prodct=2

--------------- Minimization Results ---------------------
Step size:      0.031250000
Value of the objective function:    1.7380172e-07
Minimiser is       1.0093483        5.3516375
---------------------------------------------------------------
------- For next GMM iteration
  All the zero restirictions are successfully imposed on W0 -------
   max(|difference|)=       27873294.

initial function value is        8.0807047
------------- Iteration number        1.0000000  ---------------
Step Size        1.0000000
Value of objective function       0.34342802
Current parameter values:       0.99401969      -0.36071855
Current relative gradients:        107.42284        1.0667538
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------- Iteration number        2.0000000  ---------------
Step Size        1.0000000
Value of objective function       0.33576389
Current parameter values:       0.99376077      -0.38387275
Current relative gradients:        38.210142      0.096282015
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------- Iteration number        3.0000000  ---------------
Step Size        1.0000000
Value of objective function       0.33576389
Current parameter values:       0.99376073      -0.38389084
Current relative gradients:     0.0017423840    8.8393871e-06
Outer product used for Hessian: DFP=1, Outer Prodct=2

--------------- Minimization Results ---------------------
Step size:       0.50000000
Value of the objective function:       0.33576389
Minimiser is      0.99376073      -0.38389085
--------------------------------------------------------------

================ GMM ITERATION        2.0000000 ===================
   b=      0.99376073      -0.38389085
   s.e.=    0.0059686757         2.0745479
   chi square=      0.33576389 (      0.56228465 )
   d.f.=        1.0000000
   difference in prob. value from last GMM iteration=       99.437715
(1/sqrt(T))g(b)=   -0.0094112041    -0.0093943874    -0.0081874349
          s.e.=     0.016241577      0.016212555      0.014129632
   scaling const used is     0.054717566
------- For next GMM iteration
  All the zero restirictions are successfully imposed on W0 -------
   max(|difference|)=       3718351.4

initial function value is       0.35607207
------------- Iteration number        1.0000000  ---------------
Step Size        1.0000000
Value of objective function       0.35552069
Current parameter values:       0.99370634      -0.41818200
Current relative gradients:        7.1697351      0.027446903
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------- Iteration number        2.0000000  ---------------
Step Size        1.0000000
Value of objective function       0.35552069
Current parameter values:       0.99370628      -0.41819993
Current relative gradients:     0.0021704309    9.8736663e-07
Outer product used for Hessian: DFP=1, Outer Prodct=2

--------------- Minimization Results ---------------------
Step size:        1.0000000
Value of the objective function:       0.35552069
Minimiser is      0.99370628      -0.41819994
--------------------------------------------------------------

================ GMM ITERATION        3.0000000 ===================
   b=      0.99370628      -0.41819994
   s.e.=    0.0055478643         1.9663534
   chi square=      0.35552069 (      0.55100451 )
   d.f.=        1.0000000
   difference in prob. value from last GMM iteration=     0.011280136
(1/sqrt(T))g(b)=   -0.0087304499    -0.0087148367    -0.0074986111
          s.e.=     0.014642127      0.014615942      0.012576170
   scaling const used is     0.054717566
------- For next GMM iteration
  All the zero restirictions are successfully imposed on W0 -------
   max(|difference|)=       61299.012

initial function value is       0.35491763
------------- Iteration number        1.0000000  ---------------
Step Size        1.0000000
Value of objective function       0.35491736
Current parameter values:       0.99370727      -0.41825751
Current relative gradients:      0.23662556    0.00062813948
Outer product used for Hessian: DFP=1, Outer Prodct=2

--------------- Minimization Results ---------------------
Step size:        1.0000000
Value of the objective function:       0.35491736
Minimiser is      0.99370727      -0.41825754
-----------------------------------------------------------

================ GMM ITERATION        4.0000000 ====================
   b=      0.99370727      -0.41825754
   s.e.=    0.0055533656        1.9682546
   chi square=      0.35491736 (      0.55134264 )
   d.f.=       1.0000000
   difference in prob. value from last GMM iteration=   0.00033812976
(1/sqrt(T))g(b)=   -0.0087094179    -0.0086937540    -0.0074774679
         s.e.=     0.014619264      0.014592971      0.012551364
   scaling const used is     0.054717566
------- For next GMM iteration
  All the zero restirictions are successfully imposed on W0 -------
   max(|difference|)=       208.43319

initial function value is       0.35491568
------------- Iteration number        1.0000000  ---------------
Step Size        1.0000000
Value of objective function       0.35491568
Current parameter values:       0.99370727      -0.41825803
Current relative gradients:    0.00084259978    2.3096390e-06
Outer product used for Hessian: DFP=1, Outer Prodct=2

--------------- Minimization Results ---------------------
Step size:        1.0000000
Value of the objective function:       0.35491568
Minimiser is      0.99370727      -0.41825802
-----------------------------------------------------------

================ GMM ITERATION        5.0000000 ====================
   b=      0.99370727      -0.41825802
   s.e.=    0.0055533842        1.9682613
   chi square=      0.35491568 (      0.55134359 )
   d.f.=       1.0000000
   difference in prob. value from last GMM iteration=   9.4322780e-07
(1/sqrt(T))g(b)=   -0.0087093427    -0.0086936786    -0.0074773922
         s.e.=     0.014619172      0.014592879      0.012551267
   scaling const used is     0.054717566
```

```
@ GMMMHF.EXP @  @ Another example file for GMM.SET to illustrate a more
complicated application.  See GMM.EXP for a simpler example.@

@ This is a minor modification of a program used by
Cooley and Ogaki (1993), "A Time Series Analysis of Real Wages, Consumption,
and Asset Returns: A Cointegration-Euler Equation Approach," Rocheseter
Center for Economic Research Working Paper No. 285R, University of Rochester,
for a asset pricing model with habit formation.  See "GMM: A User's Guide"
Section 5.A.  @
   @ To run this example file, type
        RUN MINQUAD.SET [F4][F2]
        RUN GMM.SET [F4][F2]
        RUN GMMHF.EXP [F4][F2]

  When you run this program with the default hflag value of 2 (see
MINQUAD.SET for explanations of hflag), the initial
idendity weighing matrix estimation will not converge.  After about 30
iterations, hit 1 on your key board.  Then the MINQUAD program will change
the hflag value to 1 to use the DFP update method, and you will get a
convergence.  After that, hit 2 on your key board to set hflag to 2 again
and use the outer products method, which is preferable for most cases.
 @


#include "gradqg.prc";

@ ======================= User Definition Area ========================== @
@ ******************************************************************** @
@ ----------- Step 1: Prepare Output File ----------------------------@
@ ******************************************************************** @
output file=gmmq.out reset; @ Specify the name of the output file. @

datestr(0);  @ print date using a procedure in TIME.ARC @
timestr(0);  @ print time using a procedure in TIME.ARC @

?"GMMHF.OUT
  Another example for GMM.SET
NDS,  VWR and T-Bill,
Parameter Names are  'delta a1 alpha";
@ ********************************************************************* @
@----------- Step 2:  Define Global Variables  ---------------------- @
@ ********************************************************************* @

@--------------------------------
    Specify the following globals
 ------------------------------- @

nob=176;  @nob=176 47:1-90:4@
izlag=1;
taubeg=1+izlag;
tauend=nob-2;
? "izlag,taubeg,tauend=" izlag~taubeg~tauend;
tend=tauend-taubeg+1;
bgm=0.99|0|1;  @ bgm=b(1)|..|b(kgm); Initial values of the coefficients @
kgm=rows(bgm);          @ # of parameters @
ndbt=2;          @ # of disturbance terms in w(t) @
nmr=ndbt*(1+2*izlag);
```

```
mas=1;              @ The order of MA of the disturbance @

@-------------------------------@
     @ Section B.  User can leave the variables in this section as they are,
                   and go to STEP 3.

                   Only advanced users should modify this section.@

@-------------------------------@

gradname=&GRADQG; @ Specify the name of proc that calculates
                    the gradient dgT(b)/db. @

const=1/sqrt(tend);      @ Scaling Multiplier when W0=eye(L) @
const2=1/sqrt(tend);     @ Scaing Multiplier when W0 is not eye(L) @
                         @ These scaling multipliers should be set so that
                           the value of function (vof) in nonlinear search of
                           MINQUAD.SET is near 1.  If vof is too close to 0,
                           the search will not work properly.

                           const2=1/sqrt(tend) will generally be good. @
w0flag=0;      @ scalar;
                  If w0flag=0, W0=I is used as the initial weighing matrix W0.

                  If w0flag=1, initial bgm is used to calculate initial W0.

                  If w0flag=2, W0 in the memory is used as initial W0.

                  If w0flag=3, W0 and bgm in the memory are used to give
                       the first GMM result.                           @

maxitegm=5; @ Sets maximum # of iteration over weighting matrix, W0.  Set
                w0flag=0 and maxitegm=2 to execute usual 2-Stage GMM. @

zero=1E-2;     @ Iteration over W0 continues until the maximum
                  difference of the current and the previous W0 in
                  absolute value becomes less than 'zero', or
                  the # of iteration exceeds maxitegm. @

calwflag=0;  @ This variable is used to choose the method to calculate
               the distance matrics, W0.

               If calwflag=0, Durbin's method will be used when W0
                  is singular.

               If calwflag=1, the QS kernel estimator (nonprewhitened
          or prewhitened) will be used when W0 is singular.

               If calwflag=3, Durbin's method will be used.

               If calwflag=4, the QS kernel estimator (nonprewhitened
                or prewhitened) will be used.
                   @

ordard=mas+3; @ Order of AR representation for
                 Durbin's method @

   @The following four variables control the QS kernel estimator.@

st=0;   @ A scalar to control the bandwidth parameter for the QS kernel.
```

```
          if st=0, then an automatic bandwidth estimator is used.
              if st/=0, then st is used as the bandwidth parameter.@

 wav=ones(nmr,1); @nmr by 1 vector; weights given to the
       a-th element of z(t)w(t) for the automatic bandwidth estimator
           (used only if st=0). @

 maxd=0;  @  scalar:
              if maxd=0, then nonprewhitened HAC with the QS kernel is used.
          if maxd>0, then the elements of DeltaLS with the absolute
                 value greater than maxd is replaced by maxd.  See Andrews
                 and Monahan's (1990) footnote 4. @

 bst=10e+5;  @ scalar:
      When automatic bandwidth parameter is calculated to be bigger than
      bst, bst is used.  @

    @ See MINQUAD.SET for the following globals @
 hflag=2;
 dfpflag=1;
 sstol=1e-25;
    @ See MAXMUM.DOC on MODULE9 of GAUSS for the following globals @
 gradtol=1e-5;
 btol=1e-5;
 typf=1;
 typb=1;

 @ *************************************************************** @
 @ ----------- Step 3: READING IN DATA ------------------------- @
 @ *************************************************************** @
 load nw[178,1]=qwages92.dat;
 load cln[176,2]=qnrnd91.dat;
 load cls[176,2]=qnrs91.dat;
 load mpopl[541,2]=mpop92.dat;
 load mret[780,6]=mret.dat;
 load nrfr[781,11]=rskfr91.dat;
 nrfr=exp(nrfr[251:781,8].*nrfr[251:781.,10]/36500);
           @exp(r*NDM/365), gross return@
 nrfr=reshape(nrfr,177,3);
 nrfr=nrfr[.,3];
 vwrv=mret[253:780,2];  @vwrv ex post monthly retrun, 253 is 1947 Jan.@
 vwrv=1+vwrv;  @gross retrun@
 vwrv=reshape(vwrv,176,3);
 vwr=vwrv[.,1].*vwrv[.,2].*vwrv[.,3];  @gross quarterly return@

 mpopl=mpopl/1000000;
 clv=cln+cls; @ NDS@

 @ 2. Transform the data if necessary @
 @ real per capita consumption @
 mpopl=mpopl[.,2];   @mpopl[.,2]; 16+@
 popul=reshape(mpopl[1:nob*3],nob,3);
 popul=meanc(popul');  @ave. over each quarter@

 cl=clv[1:nob,2]./popul[1:nob];
 pl=clv[1:nob,1]./clv[1:nob,2];

 clstr=cl[2:nob,.]./cl[1:nob-1,.];
 c2str=cl[3:nob,.]./cl[1:nob-2,.];
 cmlstr=cl[1:nob-1,.]./cl[2:nob,.];
```

```
vwr=vwr[2:nob,.].*pl[1:nob-1,.]./pl[2:nob,.];
rfr=nrfr[2:nob,.].*pl[1:nob-1,.]./pl[2:nob,.];

clear popul;

 @Conventional Instruments@
 zpv=ones(tend,1)~clstr[taubeg-1:tauend-1,.]~vwr[taubeg-1:tauend-1,.];
 i=1;
 do until i>(izlag-1);
  zpv=zpv~clstr[(taubeg-1-i):(tauend-1-i),.]~
       vwr[(taubeg-1-i):(tauend-1-i),.];
  i=i+1;
 endo;

 zpr=ones(tend,1)~clstr[taubeg-1:tauend-1,.]~rfr[taubeg-1:tauend-1,.];
 i=1;
 do until i>(izlag-1);
  zpr=zpr~clstr[(taubeg-1-i):(tauend-1-i),.]~
       rfr[(taubeg-1-i):(tauend-1-i),.];
  i=i+1;
 endo;

rfr=rfr[taubeg:tauend,.];
vwr=vwr[taubeg:tauend,.];

maxcr=maxc(-cl[taubeg:tauend+2]./cl[taubeg-1:tauend+1]);
lpm=(1e+20)*ones(tend,nmr);

proc hu(b);                  @ scale by 1+b[1]*b[2] for identification@
 local wv,wr,s1str,s2str,hr,bca;
 bca=b[3];
 if b[2]<=maxcr;
  hr=lpm;
  goto lhue;
 endif;
  s1str=((cl[taubeg+1:tauend+1]+b[2]*cl[taubeg:tauend])./(cl[taubeg:tauend]+
   b[2]*cl[taubeg-1:tauend-1]))^(-bca);
  s2str=((cl[taubeg+2:tauend+2]+b[2]*cl[taubeg+1:tauend+1])./(
        cl[taubeg:tauend]+b[2]*cl[taubeg-1:tauend-1]))^(-bca);
  wv=(b[1]*(s1str+b[1]*b[2]*s2str).*vwr-1-b[1]*b[2]*s1str)/(1+b[1]*b[2]);
  wr=(b[1]*(s1str+b[1]*b[2]*s2str).*rfr-1-b[1]*b[2]*s1str)/(1+b[1]*b[2]);
  hr=((zpv.*wv)~(zpr.*wr));
  if b[2]>1;
   hr=hr*((b[2]-1)^2+1);
  endif;
  if b[3]>10;
   hr=hr*((b[3]-10)^2+1);
  endif;
 lhue:
 retp(hr);
endp;
@========== The User does not have to change the code below. ============= @

@------------------- Print results of estimation--------------------@
proc (0)=prntrslt(x,s);

? "
-------------- Minimization Results ---------------------
Step size: " s "
Value of the objective function: " vof "
```

```
Minimiser is"  x' "
-------------------------------------------------------------";
endp;

@ ************************************************************************** @
@ THE PROGRAM STARTS @

chi=gmm(gradname,tend,nmr,mas,rows(bgm),zero,maxitegm,st,wav,maxd,bst);

output off;
@ ------------------------ End of Program ------------------------------ @
```

APPENDIX D   GMMHF.EXP OUTPUT

```
 4/15/93
10:43:58
GMMHF.OUT
  Another example for GMM.SET
NDS,  VWR and T-Bill,
Parameter Names are  'delta al alpha
izlag,taubeg,tauend=       1.0000000        2.0000000        174.00000
Initial values of the coefficients=      0.99000000        0.0000000
       1.0000000
mas=       1.0000000
Initial W0=I
Scaling Constant used for first iteration=     0.076028592

initial function value is      0.090980311
------------ Iteration number        1.0000000  --------------
Step Size     0.0078125000
Value of objective function      0.089432048
Current parameter values:       0.99267138       0.95315187       1.6545734
Current relative gradients:        4.5712001    0.00013778674      0.017144551
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------ Iteration number        30.000000  --------------
Step Size    5.9604645e-08
Value of objective function      0.071751996
Current parameter values:         1.0247420       1.0028508       9.3120601
Current relative gradients:        5.4346460     0.0017315558      0.014564150
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------ Iteration number        31.000000  --------------
Step Size    1.1920929e-07
Value of objective function      0.071751992
Current parameter values:         1.0247419       1.0031844       9.3120548
Current relative gradients:        5.4346452     0.0019591130      0.014564275
DFP update used for inverse hessian: DFP=1, Outer Product=2
------------ Iteration number        60.000000  --------------
Step Size        1.0000000
Value of objective function      0.070449701
Current parameter values:         1.0280224       1.0032910       10.004849
Current relative gradients:    2.0644055e-05    1.0345819e-08    6.5463404e-08
DFP update used for inverse hessian: DFP=1, Outer Product=2

-------------- Minimization Results --------------------
Step size:    0.00024414063
Value of the objective function:      0.070449701
Minimiser is        1.0280224        1.0032910        10.004849
-------------------------------------------------------
------- For next GMM iteration
  All the zero restirictions are successfully imposed on W0 -------
   max(|difference|)=        12454519.

initial function value is        26.469544
------------ Iteration number        1.0000000  --------------
Step Size     0.00097656250
Value of objective function        26.456596
Current parameter values:         1.0302650       1.0032955       10.004749
Current relative gradients:        22.986583     0.045668719       1.0230552
DFP update used for inverse hessian: DFP=1, Outer Product=2
------------ Iteration number        7.0000000  --------------
Step Size        1.0000000
Value of objective function        21.203747
```

```
Current parameter values:        1.0086573        1.0245790        7.1094594
Current relative gradients:        161.14324        0.24397764        1.1150644
DFP update used for inverse hessian: DFP=1, Outer Product=2
------------ Iteration number        8.0000000  --------------
Step Size        1.0000000
Value of objective function        20.189136
Current parameter values:        1.0028921        1.0245696        5.9687827
Current relative gradients:        159.20589        0.21201631        0.99962075
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------ Iteration number        9.0000000  --------------
Step Size     0.031250000
Value of objective function        18.419779
Current parameter values:        1.0027720      0.29787395        5.9259664
Current relative gradients:        202.68132        0.23829092        1.0986074
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------ Iteration number        15.000000  --------------
Step Size        1.0000000
Value of objective function        6.6430220
Current parameter values:        1.0070075     -0.093629141        3.8547589
Current relative gradients:    8.5847885e-06    4.9228143e-05    7.3148572e-07
Outer product used for Hessian: DFP=1, Outer Prodct=2

-------------- Minimization Results --------------------
Step size:        1.0000000
Value of the objective function:        6.6430220
Minimiser is        1.0070075     -0.093629127        3.8547589
---------------------------------------------------------

================ GMM ITERATION        2.0000000 ===================
   b=       1.0070075     -0.093629127        3.8547589
   s.e.=    0.0099678404        0.15214721        2.3389047
   chi square=        6.6430220 (      0.084189543 )
   d.f.=       3.0000000
   difference in prob. value from last GMM iteration=        99.915810
(1/sqrt(T))g(b)=      0.15359365        0.15380531        0.16005698
    -0.076226805     -0.076684594     -0.076148060
          s.e.=     0.069957386       0.069878801       0.072419149
     0.034360428      0.034594806       0.034315846
   scaling const used is      0.076028592
------- For next GMM iteration
  All the zero restirictions are successfully imposed on W0 -------
   max(|difference|)=        55715343.

initial function value is        19.268267
------------ Iteration number        1.0000000  --------------
Step Size        1.0000000
Value of objective function        9.1171447
Current parameter values:        1.0156587      -0.21836827        5.1801459
Current relative gradients:        899.82199        21.076705        2.6768330
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------ Iteration number        5.0000000  --------------
Step Size        1.0000000
Value of objective function        7.1443271
Current parameter values:        1.0155299      -0.17128018        5.1644763
Current relative gradients:    0.00031960792     0.00019761956    1.1660495e-05
Outer product used for Hessian: DFP=1, Outer Prodct=2

-------------- Minimization Results --------------------
Step size:        0.25000000
Value of the objective function:        7.1443271
```

```
Minimiser is       1.0155299      -0.17128018        5.1644762
------------------------------------------------------------

================= GMM ITERATION        3.0000000 ====================
   b=        1.0155299       -0.17128018         5.1644762
   s.e.=    0.0065164995       0.057433479        1.7169752
   chi square=       7.1443271 (      0.067437113 )
   d.f.=       3.0000000
   difference in prob. value from last GMM iteration=     0.016752430
(1/sqrt(T))g(b)=      0.19998023        0.20057936        0.20518189
    -0.029286731     -0.029350813      -0.029289354
          s.e.=     0.081711267       0.081825990       0.084504590
     0.016054372      0.016167746       0.016116591
   scaling const used is     0.076028592
OMEGA is not positive definite when
 calculated with zero restrictions.
------ For next GMM iteration Durbin's Method used for W0 -------
Order of AR used=       4.0000000
Order of MA used=       1.0000000
   max(|difference|)=       46683901.

initial function value is        6.8646894
------------- Iteration number        1.0000000  ---------------
Step Size       1.0000000
Value of objective function        6.4038977
Current parameter values:        1.0103916      -0.19037374        3.9982885
Current relative gradients:        94.003057        2.2208629       0.13164446
Outer product used for Hessian: DFP=1, Outer Prodct=2
================= GMM ITERATION        4.0000000 ====================
   b=        1.0103165       -0.19479256        3.9810898
   s.e.=    0.0083642287       0.11324665        2.0986082
   chi square=       6.4018232 (      0.093615814 )
   d.f.=       3.0000000
   difference in prob. value from last GMM iteration=     0.026178701
(1/sqrt(T))g(b)=      0.19270069        0.19332983        0.19949788
    -0.038004028     -0.038048013      -0.037941580
          s.e.=     0.082426710       0.082538468       0.085191484
     0.020085773      0.020184989       0.020109717
   scaling const used is     0.076028592
OMEGA is not positive definite when
 calculated with zero restrictions.
------ For next GMM iteration Durbin's Method used for W0 -------
Order of AR used=       4.0000000
Order of MA used=       1.0000000
   max(|difference|)=       3550828.6

initial function value is        6.4307647
------------- Iteration number        1.0000000  ---------------
Step Size       1.0000000
Value of objective function        6.4053565
Current parameter values:        1.0110239      -0.18831134        4.0851279
Current relative gradients:        47.457550      0.057880021       0.16338760
Outer product used for Hessian: DFP=1, Outer Prodct=2
------------- Iteration number        3.0000000  ---------------
Step Size       1.0000000
Value of objective function        6.4053553
Current parameter values:        1.0110256      -0.18837702        4.0855692
Current relative gradients:    5.2255106e-05    4.7735722e-05    3.2083291e-06
Outer product used for Hessian: DFP=1, Outer Prodct=2
```

```
-------------- Minimization Results ---------------------
Step size:        1.0000000
Value of the objective function:        6.4053553
Minimiser is       1.0110256      -0.18837702       4.0855694
-----------------------------------------------------------

================ GMM ITERATION        5.0000000 ===================
   b=       1.0110256       -0.18837702        4.0855694
   s.e.=    0.0080947569       0.095868927        2.0544765
   chi square=        6.4053553 (     0.093470727 )
   d.f.=        3.0000000
   difference in prob. value from last GMM iteration=   0.00014508730
(1/sqrt(T))g(b)=      0.19652630       0.19715297       0.20326612
    -0.034090395     -0.034136036     -0.034022080
          s.e.=     0.083760512      0.083893233      0.086772461
     0.016996128      0.017062567      0.016981345
   scaling const used is      0.076028592
```