SCOREM 2.11: A PROGRAM FOR THE ESTIMATION OF GENERAL STATE-SPACE MODELS WITH THE EM AND SCORING ALGORITHMS

Jacques Raynauld*
Jean-Guy Simonato
Christian Sigouin

Institut d'économie appliquée
École des Hautes Études Commerciales
5255 Decelles, Montréal, H3T 1V6
Phone: 514-340-6312
Fax: 514-340-6469
Internet: RAYNAULD@HEC.CA

August 1993

* We would like to acknowledge the financial support of the Social Sciences and Humanities Research Council of Canada, Les Fonds FCAR of the Québec government, Les Fonds Faribeault of the Université de Montréal and the Center for International Business Studies of l'École des Hautes Études Commerciales. Correspondence should be addressed to the first author.

1. INTRODUCTION

This paper describes how to use a program available on a companion diskette to estimate dynamic models with unobservable variables (also called DYMIMIC models). This program works with GAUSS 2.0 and 3.0. The user is expected to have some basic notions concerning this programming language. We first begin by explaining the estimation methods used by the program. We then show how to set up a model by providing a detailed example. The authors welcome any comments concerning possible improvements or problems with this version of SCOREM.

2. ESTIMATION METHODS

This program is designed to estimate DYMIMIC models using either a Newton scoring algorithm or an expectation-maximization (EM) algorithm. These are two maximum likelihood based estimation methods which rely on the Kalman filter. Both have relative strengths and weaknesses.

The scoring algorithm [described in Engle and Watson (1981)] is fairly accurate and converges relatively rapidly but is highly dependent on the initial values given to the model parameters. Inappropriate initial values lead to many problems such as non-positive definite variance-covariance matrices which make the estimation process very difficult if not impossible. In many cases, finding a good set of initial values could be difficult and an extensive search can be extremely time consuming.

Unlike the scoring algorithm, the EM algorithm [described in Watson and Engle (1983), Shumway and Stoffer (1982), and Shumway (1988)] is not very sensitive to initial values since it uses OLS to find the parameter vector maximizing the likelihood. A few iterations with this algorithm will do a good job finding a parameter vector close to the one maximizing the likelihood function. However, this algorithm is slow to converge and unlike the scoring algorithm, it doesn't provide any standard errors for the estimated parameters.[1]

This program is built in a manner that permits the sequential use of these two algorithms. This way, we can take advantage of both methods' strengths and overcome

---

1 Following Ruud (1991), it is possible to construct an estimate of the Hessian using the score. However, this possibility is not available in the current version of the program.

most of their weaknesses. A few iterations with the EM algorithm can be used to find a "nearly optimal" parameters vector which will not cause any problem if used as initial values for the scoring algorithm. The latter can then be used to find the final estimates and their associated standard errors. However, with the new "squeezing" feature controlling the step-length parameter added to SCOREM 2.11, the user can most of the time bypass the EM algorithm and start right away with the scoring algorithm.

The possibility to use analytical derivatives for the scoring algorithm without much work is also a program's main feature. This permits to speed up the estimation process from two to three times compared to a standard numerical derivatives based method.

3. INSTALLATION

The program should be installed as it is on the diskette, that is, with a main directory named SCOREM (or any other name) and two sub-directories named EM and SCORING. Examples can be found in the sub-directory EXAMPLES. The results from each estimation will also be stored in the main directory.

4. SETTING UP A STATE-SPACE MODEL

The program is designed to work with the state-space formulation of a model [see Harvey (1981, 1989) for further explanations] that is, a measurement equation which links the observed variables (Y_t) to the state variables (α_t):

(1) $$\underset{(nx1)}{Y_t} = \underset{(nxm)}{Z_t}\ \underset{(mx1)}{\alpha_t} + \underset{(nxk1)}{\beta}\ \underset{(k1x1)}{X1_t} + \underset{(nxnm)}{S}\ \underset{(nmx1)}{e_t} \quad 1$$

and a transition equation which describes the process followed by the state variables:

(2) $$\underset{(mx1)}{\alpha_t} = \underset{(mxm)}{T}\ \underset{(mx1)}{\alpha_{t-1}} + \underset{(mxk2)}{\gamma}\ \underset{(k2x1)}{X2_t} + \underset{(mxg)}{R}\ \underset{(gx1)}{n_t} \quad 2$$

$$e_t \_ N(0\ ,\ H\ )\ \text{and}\ n_t \_ N(0\ ,\ Q\ )\ 3$$

where e_t and n_t are random disturbances assumed to follow a normal distribution with variance-covariance matrices H and Q respectively. The dimensions of each matrix are in parenthesis. $X1_t$ and $X2_t$ are vectors of exogenous variables.

A particular model is set up by providing the contents of each matrix in equations

(1) and (2). Actually, this task consists in discriminating between fixed and free parameters (the ones which are to be estimated). Of course, this should always be done in a manner so as to avoid identification problems. Underidentified models will invariably lead to a program failure. A simple rule of thumb is to fix at least one parameter for each state variable. Moreover, practice has shown that it is a good idea to standardize all observable variables (The Ys and the Xs in the measurement equation), at least in the stationary case. This helps the algorithms to converge by putting all variables on the same scale. Furthermore, if for identification purposes a variance is fixed, its value should be in line with the size of the corresponding variables: this will prevent the other estimated variances from becoming negative. Further identification tips are given in the detailed examples of the appendices.

Each different model must be initialized with the <name>.GAU and <name>.PRC files. A simple example illustrates how this can be done.

5. SETTING UP A MODEL: AN EXAMPLE

Suppose we want to estimate a coincident indicator of economic activity in a given country (see Stock and Watson, 1991). It seems reasonable to suppose that variables like employment (EMP), manufacturers' shipments (SHIP) and retail sales (RTS) would be driven by a common (unobservable) variable named economic activity (EA) and by some idiosyncratic components (U1, U2, and U3). As explained before, e1, e2, e3 and n are normally distributed random disturbances. A first difference was applied to all variables which are now stationary (the variables were also standardized). The model to estimate would hence be:

$$\begin{bmatrix} SHIP_t \\ EMP_t \\ RTS_t \end{bmatrix} = \begin{bmatrix} Z_{11} & 1 & 0 & 0 \\ Z_{21} & 0 & 1 & 0 \\ Z_{31} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} EA_t \\ U1_t \\ U2_t \\ U3_t \end{bmatrix} + \begin{bmatrix} \beta_{11} \\ \beta_{21} \\ \beta_{31} \end{bmatrix} [0] \quad 4$$

for the measurement equation and

$$\begin{bmatrix} EA_t \\ U1_t \\ U2_t \\ U3_t \end{bmatrix} = \begin{bmatrix} T_{11} & 0 & 0 & 0 \\ 0 & T_{22} & 0 & 0 \\ 0 & 0 & T_{33} & 0 \\ 0 & 0 & 0 & T_{44} \end{bmatrix} \begin{bmatrix} EA_{t-1} \\ U1_{t-1} \\ U2_{t-1} \\ U3_{t-1} \end{bmatrix} + \begin{bmatrix} \gamma_{11} \\ \gamma_{22} \\ \gamma_{33} \\ \gamma_{44} \end{bmatrix} [0] + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_t \\ e1_t \\ e2_t \\ e3_t \end{bmatrix} 5$$

for the transition equation, with

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & Q_{11} & 0 & 0 \\ 0 & 0 & Q_{22} & 0 \\ 0 & 0 & 0 & Q_{33} \end{bmatrix} 6 .$$

We need to make three remarks . First, our model doesn't contain any exogenous variable. However, we see that the matrices β and γ are still part of it. This is because we filled the matrices X1 and X2 with zeros. It is the way to proceed: in any case, the dimensions of a matrix must never be less than one by one. Matrices which are not used must be filled with zeros.

Second, we must address identification issues. As stated before, there must be a least one tied parameter for each unobservable variable. We have four unobservable variables: we need to fix at least four parameters. Obvious candidates are the idiosyncratic components' coefficients in the measurement equation. We set them to 1. This leaves us with one more parameter to fix. We decided to set the variance of EA to one. Alternatively, we could have set one more parameter to one in matrix Z (for example, Z_{11} which ties the units of measurement of the unobservable variable EA to the observable variable SHIP). Either ways, it leaves us with 10 parameters to estimate. The vector of parameters to estimate (named DELTA in the <name>.GAU file) will then have ten elements.

Third, we can see that there is no measurement errors (no random disturbances in the measurement equation). Normally, this particular case cannot be treated by the EM algorithm because it uses OLS on the measurement equation to find the parameters' value. The program overcomes this problem by automatically adding small random disturbances to the measurement equation when it is needed. This is one more reason to use the scoring algorithm to find the final estimates.

Appendices 1 and 2 show how to program this particular stationary model. The code corresponding to this model can be found in the files DEMO.GAU and DEMO.PRC.

6. OUTPUT FILES

The program will produce two output files: One for the EM algorithm and one for the SCORING algorithm. We are mainly concerned with the latter. Running the previous model will give the following results:

```
ITERATION #   56.00000
----------------------------

COEFFICIENTS VECTOR

ITERATION  ITERATION   GRADIENT   T-TEST
    K        K-1        K-1        K-1

 0.24871   0.24873  -0.00218   2.35892
 0.43910   0.43890   0.00427   3.23036
 0.35303   0.35318  -0.00773   2.74373
 0.53856   0.53872  -0.00249   2.70686
-0.28767  -0.28776   0.00434  -2.49259
-0.45532  -0.45507  -0.00431  -3.25107
-0.16528  -0.16551   0.00720  -1.21610
 0.82197   0.82192   0.00083   5.76538
 0.57326   0.57359  -0.00397   3.43126
 0.78523   0.78501   0.00292   5.05176


LIKELIHOOD

ITERATION  ITERATION   DIFFERENCE
    K        K-1

644.18763 644.18763   0.00000
```

The first part gives the value for the parameters vector at the K^{th} iteration, at the $(K-1)^{th}$ iteration and the standard errors associated with each element. t-statistics are also provided. The second part provides the value of the Gaussian log-likelihood function (excluding the constant) at the K_{th} iteration. It is computed as

$$L = -\frac{1}{2}\sum_{t=1}^{T}\log|F_t| - \frac{1}{2}\sum_{t=1}^{T} v_{t'}\, F_t^{-1}\, v_t\, 7\ .$$

Estimated unobservable variables, variance-covariance matrices, residuals and their autocorrelation as well as smoothed unobservable variables and VCV matrices are all available at the end of the output files.

7. ESTIMATING VARIANCES

Estimating variance terms can sometimes be a bit difficult, particularly in the non-stationary case. If the real variance term is actually very small, some problems may arise due to the computer's precision. In this case, it can push the estimated variance towards a negative value. In some other cases, the gradient associated with a variance term may not tend towards zero. A way to overcome these problems is to estimate the standard error instead of the variance. This is done by using the square of the parameter in the .PRC file. In the first example, we could have used Q[2,2]=DELTA[8,1]^2 instead of Q[2,2]=DELTA[8,1] to estimate the first variance term.

8. OTHER FEATURES

- In the stationary case, the program can compute the initial variance-covariance matrix of the state vector according to the structure of the model postulated.

- By choosing the appropriate options, the program can also used to estimate non stationary models as illustrated in Appendix 3 [see Patry, Raynauld, and Simonato (1989) and Slade (1989) for further illustrations].

- In most of the cases Z_t is not time varying and contains coefficients to be estimated. SCOREM 2.11 can now handle time-varying Z matrix such as in time-varying regression models, Sims' BVAR models, etc. This possibility only works with the scoring algorithm (with both analytic and numerical derivatives).

- The program can also handle linear restrictions within and across equations. This last possibility is not currently documented.

REFERENCES

Engle, R.F. and M. Watson (1981) A one factor multivariate time series model of metropolitain wage rates, *Journal of the American Statistical Association*, **76**, 774-81.

Harvey, A.C. (1981) *Time Series Methods*, Halsted Press, New-York.

Harvey, A.C. (1989) *Forecasting Structural Time Series Models and the Kalman Filter*, Cambridge University Press, Cambridge.

Patry, M., J. Raynauld, and J.G. Simonato (1989) Technical Change as an Unobservable Stochastic Variable: an Application to the U.S. Primary-Metals Industry, Working paper, École des HEC.

Ruud, P.A. (1991) Extensions of estimation methods using the EM algorithm, *Journal of Econometrics,* **49**, 305-341.

Slade, M. (1989) Modeling stochastic and cyclical components of technical change: an application of the kalman filter, *Journal of Econometrics*, **41**, 363-83.

Shumway, R. H. (1988) *Applied Statistical Time Series Analysis*, Prentice Hall, Englewood Cliffs.

Shumway, R.H. and D.S. Stoffer (1982) An Approach to Time Series Smoothing and Forecasting using the EM Algorithm, *Journal of Time Series Analysis*, **3,** 253-264.

Stock, J.H. and M.W. Watson (1991) A probability model of the coincident economic indicators, in Lahiri et G.H. Moore, ed., *Leading Economic Indicators*, Cambridge University Press, Cambridge, 63-89.

Watson, M. and R. Engle (1983), Alternative Algorithms for the Estimation of Dynamic Factor, Mimic and Varying Coefficient Regression Models, *Journal of Econometrics*, 385-400.

APPENDIX 1: DEMO.GAU

```
NEW;

@ State space matrices' dimensions
  -------------------------------- @

  N=3; M=4; NM=1; G=4; K1=1; K2=1; NPAR=10; NOBS=79;



@ Data reading and transformation
  ------------------------------- @

  LOAD Y1[79,1]=EMP.prn;
  LOAD Y2[79,1]=SHIP.prn;
  LOAD Y3[79,1]=RTS.prn;

  Y=Y1~Y2~Y3;

  X1 = zeros(NOBS,1) ;

  X2 = zeros(NOBS,1) ;

  zt = zeros(nobs*N,M); @ time-varying matrix Z (filled with
                          zeros if not used) @

@ Names of the output files
  ------------------------- @

  EMOUT="DEMO_EM.RES"; @ FOR THE EM ALGORITH @
  SCOUT="DEMO_SC.RES"; @ FOR THE SCORING ALGORITHM @

@ Initial values
  -------------- @

  DELTA={

    0.70
    0.30
    0.50
    0.80
    0.30
    0.30
    0.30
    0.50
    0.30
    0.70    };

  DELTA=DELTA';

@ Include file with the CONSTRUC procedure
  ---------------------------------------- @

  #INCLUDE DEMO.PRC;
```

! The variables N, M, NM, G, K1 and K2 correspond to the state-space matrices dimensions defined earlier.
! NPAR is the number of "free" parameters. (number of parameters to estimate)
! NOBS is the number of observations available for the observed variables. (Ys and Xs)

! Each observed variable is loaded and used to form the (NOBS by N) matrix Y.
! Notice that we filled X1 and X2 with zeros because there is no exogenous variables in our model.

! See Appendix 4 for further details on setting up a time-varying Z matrix.

! All results after the final iteration are stored in separate files.
! It's a good idea to change these names for each new model because the program will overwrite these files.

! DELTA is the (NPAR by 1) vector of parameters to estimate.
! It will be used by the <name>.PRC file which build the model's structure.
! We must give initial values for this vector.
! Some models can be very sensitive to initial values. Choosing an "adequate" set can help convergence.

! This tells the program the name of the .PRC file.
! Each new model need a new .PRC file.

```
@ Algorithm choice
 ---------------- @

 CHOICE = 3;  @ =1 to use the EM algorithm ALONE @
          @ =2 to use the SCORING algorithm ALONE @
          @ =3 to use both EM and SCORING algorithm @


@ EM algorithm options
 -------------------- @

 OPTION=ZEROS(15,1);

 OPTION[1,1]=1e-6; @ Convergence criterion @
 OPTION[2,1]=5; @ Maximum number of iterations @
 OPTION[12,1]=0; @ = 1 to skip the convergence check @
 OPTION[4,1]=0;  @ = 1 To print the unobservable variables at the end @
 OPTION[8,1]=0;  @ = 1 To print A0 and P0 at each iteration @
 OPTION[9,1]=0;  @ = 1 To print state space model at each iteration @
 OPTION[10,1]=1; @ = 1 To print the difference in the vector of coef. @
 OPTION[11,1]=0; @ = 1 To print the variances of the unobservable var. @
 OPTION[13,1]=0; @ = 1 To print the smoothed residuals @
 OPTION[3,1]=0;  @ = number of residuals' autocorrelation to compute
              = 0 none @
 OPTION[6,1]=1;  @ = 0 to use the smoothed estimates of a(0) as starting
                value for all iterations but the first
              = 1 to use 0 as a starting value for a(0) @
 OPTION[7,1]=1;  @ = 0 to use the smoothed estimates of P(0) as starting
                value for all the iterations but the first
              = 1 to use the automatic calculation of the matrix P0
                based upon the model's parameters @
 OPTION[14,1]=4; @ = n to compute the standard errors after the n th
                 iteration @
 OPTION[15,1]=9; @ = m to compute the coefficient with a "hill climbing"
                method after the m th iteration @

@ SCORING algorithm options
 ------------------------- @

 OPT=ZEROS(10,1);

 opt[1,1]=1e-6; @ convergence criterion @
 opt[2,1]=100;  @ maximum number of iterations @
 opt[3,1]=0;    @ = 1 if you want to skip the iterations' loop @
 opt[5,1]=1;    @ = 1 to have the smoothed estimates at the end @
 opt[6,1]=1;    @ = 1 to have residuals and autocorrelations @
 opt[4,1]=0;    @ = 0 if the unobservable variable is stationnary
             = 1 if not @
```

```
opt[7,1]=1;    @ = 0 to use the numerical derivatives
               = 1 to use the analytical derivatives @
opt[8,1]=1;    @ = 1 if you want the initial covariance matrix P0
                 to be computed by the program using the
                 model's structure @
opt[9,1]=0; @ =1 if Z is time-varying.  The time-varying matrix has to be
               provided by the matrix ZT @

opt[10,1]=0.01;  @ Convergence criterion on gradient vector (greatest
                gradient tolerated) @
```

! If CHOICE=3, the program will first start with the EM algorithm until the maximum number of iterations is reached or until convergence occured. Then, the estimation will proceed with the scoring algorithm.

! Note that if a VCV matrix become non-positive definite while using the scoring algorithm, the program will go back to the EM algorithm and start the process all over.

! Options for the EM and the scoring algorithm are chosen separately.

! In this case, there will be 5 iterations with the EM algorithm before those with the scoring algorithm.

! Option 14 and 15 are not currently active.

! This option is strongly recommended. For more information, see the .PRC file.

APPENDIX 2: DEMO.PRC

```
PROC (10)=CONSTRUC(DELTA);

 LOCAL Z,BETA,S,H,T,GAM,Q,R,A0,P0,RQR,VP,N,M,K1,K2,NM,G,SZ,I;

@ Initializing matrices
 --------------------- @

 LOAD SZ = SIZE.FMT;
 N=SZ[1,1]; M=SZ[2,1]; K1=SZ[3,1]; K2=SZ[4,1]; NM=SZ[5,1]; G=SZ[6,1];

 A0=ZEROS(M,1); P0=ZEROS(M,M);
 Z=ZEROS(N,M); BETA=ZEROS(N,K1); S=ZEROS(N,NM);
H=ZEROS(NM,NM);
 T=ZEROS(M,M); GAM =ZEROS(M,K2); R=ZEROS(M,G) ;
Q=ZEROS(G,G);

@ Building state-space matrices with the vector DELTA
 -------------------------------------------------- @

 Z[1,1]=DELTA[1,1]; Z[2,1]=DELTA[2,1]; Z[3,1]=DELTA[3,1];
 Z[1,2]=1.0; Z[2,3]=1.0; Z[3,4]=1.0;

 T[1,1]=DELTA[4,1]; T[2,2]=DELTA[5,1]; T[3,3]=DELTA[6,1];
 T[4,4]=DELTA[7,1];

 R=EYE(4);

 Q[1,1]=1.0;
 Q[2,2]=DELTA[8,1]; Q[3,3]=DELTA[9,1]; Q[4,4]=DELTA[10,1];

@ Computing P0 matrix according to the model's structure (see options)
 ------------------------------------------------------------------- @

 IF SZ[9,1] == 1;
   RQR=R*Q*R';
   VP=INV(EYE(M*M)-T.*.T)*VEC(RQR);
   I=1; P0=ZEROS(M,M);
   DO WHILE I <= M ;
```

```
      P0[.,I] = VP[(M*(I-1)+1):(M*I),1];
      I=I+1;
    ENDO ;

  ELSE;
    P0=1000*EYE(M); @ IF NOT @

  ENDIF;

  RETP(A0,P0,Z,BETA,S,H,T,GAM,R,Q) ;

ENDP;
```

! This is the only part of this file that need to be changed for each new model.

! It gives the content of each matrix in the model.

! Each matrix is initialized to have the correct size and is filled with zeros. One only need to assign to each para-meter to estimate, the corresponding element of the vector DELTA and fix the value of any constrained parameter.

! As we can see, the first three elements of vector DELTA the parameters in matrix Z which links observed variables to the common factor EA.

! The four next elements correspond to the autoregressive coefficients of the state variables. (matrix T)

! The 3 last elements of DELTA are the variances associated with each idiosyncratic component's random disturbance.

! The variance of the common factor is set to 1 and so are the parameters linking idiosyncratic components and observed variables in matrix Z.

! Letting the program to compute the P0 matrix is the best way to impose only a diffuse prior. The P0 matrix will then be consistent with the model's structure and the given initial values. This is controled by selecting the corresponding option in the <name>.GAU file.

! If in any case one want to provide his "own" P0 matrix, this should be done on this line.

APPENDIX 3: NON-STATIONNARY EXAMPLES

We simulated the following models:

First model.

$$Y_t = \begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_t \\ U_t \\ U_{t-1} \end{bmatrix} \quad 8$$

$$\begin{bmatrix} V_t \\ U_t \\ U_{t-1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1.6 & -0.64 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} V_{t-1} \\ U_{t-1} \\ U_{t-2} \end{bmatrix} + \begin{bmatrix} 0.21 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_t \\ u_t \end{bmatrix} \quad 9$$

with

$$Q = \begin{bmatrix} 0.25 & 0 \\ 0 & 0.04 \end{bmatrix} \quad 10.$$

Second model.

$$\begin{bmatrix} Y1_t \\ Y2_t \end{bmatrix} = \begin{bmatrix} 1.0 & 0 \\ 0.7 & 0 \end{bmatrix} \begin{bmatrix} C_t \\ C_{t-1} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u1_t \\ u2_t \end{bmatrix} \quad 11$$

$$\begin{bmatrix} C_t \\ C_{t-1} \end{bmatrix} = \begin{bmatrix} 1.26 & -0.3 \\ 1.0 & 0.0 \end{bmatrix} \begin{bmatrix} C_{t-1} \\ C_{t-2} \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} e_t \end{bmatrix} \quad 12 \text{with}$$

$$Q = \begin{bmatrix} 0.01 \end{bmatrix} \qquad H = \begin{bmatrix} 0.063 & 0 \\ 0 & 0.09 \end{bmatrix} \quad 13$$

Estimated parameters are in bold typeface. The first example will be found in the file WAT.GAU and the second in the file NONSTAT.GAU. The results obtained can respectively be found in the files WAT_SC.RES and NSTAT_SC.RES. For each model, the results can be summed up by

First model.

ITERATION #   16.00000000

COEFFICIENTS VECTOR

ITERATION K	K-1	GRADIENT K-1	T-TEST COORD.	PARAM.
1.64235959		0.00001109	17.26940511	T(2,2)
-0.70991797		0.00003498	-7.91137996	T(2,3)
0.22782423		-0.00010292	6.33794386	GAM(1,1)
0.24290041		-0.00004411	8.14277080	Q(1,1)

LIKELIHOOD

ITERATION K	ITERATION K-1	DIFFERENCE
332.56303548	332.56303548	0.00000000

Second model.

ITERATION #   10.00000000

COEFFICIENTS VECTOR

ITERATION K	K-1	GRADIENT K-1	T-TEST COORD.	PARAM.
0.36515882		-0.00003440	2.95977627	Z(1,1)
0.28056720		0.00002926	2.92039778	Z(2,1)
0.07182082		0.00014689	8.96067114	H(1,1)
0.09937344		-0.00006982	9.56661430	H(2,2)
1.77799344		-0.00005512	16.96770507	T(1,1)
-0.79815911		-0.00008542	-7.86118887	T(1,2)

LIKELIHOOD

ITERATION K	ITERATION K-1	DIFFERENCE
229.10345021	229.10345021	0.00000000

Some observations need to be made about the results. First, notice that only four parameters are estimated in the first model. For identification concerns, we need to fix the variance Q(2,2) to 0.04 in the equation for U_t even if Z(1,2) is set to 1.0. This is due to the special nature of the model. There is only one observable variable (Y_t) to identify two distinct sources of random disturbances, namely u_t and v_t. If Q(2,2) is not fixed, the algorithm will have a hard time figuring out whether the randomness of Y_t comes from u_t or v_t. Actually, it will push either the value of Q(1,1) or the value of Q(2,2) towards zero so as to "eliminate" one of the two random disturbances. In the process, one variance may even become negative! This is a common result when an identification problem occurs.

Second, notice that the estimated values for Z(1,1) and Z(2,1) in the second model don't quite match the simulated values. This is because we need to consider these values on a relative basis, that is Z(2,1)/Z(1,1). If we compute this ratio we obtain 0.768, a value much closer to the "simulated ratio" of 0.7. It also worths noting that the sum of the two autoregressive coefficients in the equation for C_t is more significant than the absolute values of these coefficients. In this precise case, the "estimated sum" is about 0.98 and the "simulated sum" is 0.96. For the first example, the simulated sum of coefficient was 0.96 and the estimation gave us a sum of 0.932.

APPENDIX 4: TIME VARYING Z MATRIX

The following model assumes a time varying Z matrix:

$$Y_t = Z_{11,t}\,\alpha_t + B\,X1_t + e_t$$

$$\alpha_t = \alpha_{t-1} + n_t\,.$$

The SCOREM 2.11 timevz.gau file is:

```
@ State space matrices' dimensions
 ---------------------------------- @

 N=1; M=1; NM=1; G=1; K1=3; K2=1; NPAR=4; NOBS=26;

@ Data reading and transformation
 --------------------------------- @

 LOAD Y[26,1]=VOBS.PRN;

 LOAD X1[26,3]=EXO.PRN;

 x2 = zeros(NOBS,1) ;

 load zt[26,1]=lnp.prn;

@ Names of the output files
 -------------------------@

 EMOUT="tmvz_EM.RES"; @ OUTPUT FILE FOR THE EM ALGORITH @
 SCOUT="tmvz_SC.RES"; @ OUTPUT FILE FOR THE SCORING ALGORITHM @

@ Starting values (optionnal if scoring is not used alone)
 ---------------------------------------------------------@

@  DELTA=0.7*ONES(NPAR,1);@ @ IF YOU DON'T WANT TO GIVE ANY SPECIFIC @
                       @ STARTING VALUES @

 LET DELTA[4,1]=
          0.7
          0.3
```

```
        -1.4
         1.5;

@ Include file with the CONSTRUC procedure
  --------------------------------------- @

  #INCLUDE timevz.PRC;   @ THE NAME OF THIS FILE HAS TO BE CHANGED FOR EACH @
                  @ NEW MODEL @

@ Algorithm choice
  --------------- @

  CHOICE = 2;  @ =1 to use the EM algorithm ALONE @
           @ =2 to use the SCORING algorithm ALONE @
           @ =3 to use the integrated EM/SCORING algorithm @

@ SCORING algorithm options
  ------------------------ @

  OPT=ZEROS(10,1);

  opt[1,1]=1e-6;  @ convergence criterion @
  opt[2,1]=100  ;    @ iterations number @
  opt[3,1]=0   ;     @ =1 if you want to skip the iterations' loop @
  opt[5,1]=1   ;     @ =1 to have the smoothed estimates at the end @
  opt[6,1]=1   ;     @ =1 to have residuals and autocorrelations @
  opt[4,1]=0   ;     @ =0 if the unobservable variable is stationnary
                  =1 if not @
  opt[7,1]=1   ;     @ =0 to use the numerical derivatives
                  =1 to use the analytical derivatives @
  opt[8,1]=1   ;     @ =1 if you want the initial covariance matrix P0
                    to be calculated by the program @
  IF OPT[4,1]==1;
    OPT[7,1]=0;  @ the use of analytical derivatives is proscribed if the
               unobservable variable is not stationnary @
  ENDIF;

  opt[9,1]=1; @ =1 if Z is time-varying.  The time-varying matrix has to be
              provided by the matrix ZT @

  opt[10,1]=0.01;  @ Convergence criterion on gradient vector (greatest
               gradient tolerated @
```

The SCOREM 2.11 timevz.prc file is:

```
PROC (10)=CONSTRUC(DELTA) ;

@ Variables locales
```

```
  ---------------- @

  LOCAL Z,BETA,S,H,T,GAM,Q,R,A0,P0,RQR,VP,N,M,K1,K2,NM,G,SZ,I;

@ Former les matrice du modele state-space
  --------------------------------------- @
  LOAD SZ = SIZE.FMT;
  N=SZ[1,1]; M=SZ[2,1]; K1=SZ[3,1]; K2=SZ[4,1]; NM=SZ[5,1]; G=SZ[6,1];

  A0=ZEROS(M,1); P0=ZEROS(M,M);
  a0=-2;

  Z=ZEROS(N,M); BETA=ZEROS(N,K1); S=ZEROS(N,NM); H=ZEROS(NM,NM);
  T=ZEROS(M,M); GAM =ZEROS(M,K2); R=ZEROS(M,G) ; Q=ZEROS(G,G);

  BETA[1,1]=DELTA[1,1]; BETA[1,2]=DELTA[2,1]; BETA[1,3]=delta[3,1];
  H[1,1]=0.01;

  T[1,1]=1.0;

  Q[1,1]=delta[4,1];

  S[1,1]=1.0;
  R[1,1]=1.0;

@ COMPUTES P0 BASED UPON THE MODEL'S PARAMETERS @

  IF SZ[9,1] == 1;
    RQR=R*Q*R';
    VP=INV(EYE(M*M)-T.*.T)*VEC(RQR);
    I=1; P0=ZEROS(M,M);
    DO WHILE I <= M ;
      P0[.,I] = VP[(M*(I-1)+1):(M*I),1];
      I=I+1;
    ENDO ;
  ELSE;
    P0=10*EYE(M); @ IF NOT @

  ENDIF;

  RETP(A0,P0,Z,BETA,S,H,T,GAM,R,Q) ;

ENDP;
```

Note: For identification purposes, the variance of the measurement equation has been fixed to 0.01. This value was estimated after fitting an AR(1) model on the Y variable VOBS.