NAG Fortran Library Routine Document

G05HDF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G05HDF generates a realisation of a multivariate time series from a vector autoregressive moving average (VARMA) model. The realisation may be continued or a new realisation generated at subsequent calls to G05HDF.

2 Specification

```
      SUBROUTINE G05HDF(MODE, K, IP, IQ, MEAN, PAR, LPAR, QQ, IK, N, W, REF,
     1                  LREF, IWORK, LIWORK, IFAIL)
      INTEGER           K, IP, IQ, LPAR, IK, N, LREF, IWORK(LIWORK), LIWORK,
     1                  IFAIL
      real              PAR(LPAR), QQ(IK,K), W(IK,N), REF(LREF)
      CHARACTER*1       MODE, MEAN
```

3 Description

Let the vector $W_t = (w_{1t}, w_{2t}, \ldots, w_{kt})^T$ denote a k-dimensional time series which is assumed to follow a vector autoregressive moving average (VARMA) model of the form:

$$\begin{aligned} W_t - \mu = {} & \phi_1(W_{t-1} - \mu) + \phi_2(W_{t-2} - \mu) + \ldots + \phi_p(W_{t-p} - \mu) \\ & + \epsilon_t - \theta_1\epsilon_{t-1} - \theta_2\epsilon_{t-2} - \ldots - \theta_q\epsilon_{t-q} \end{aligned} \qquad (1)$$

where $\epsilon_t = (\epsilon_{1t}, \epsilon_{2t}, \ldots, \epsilon_{kt})^T$ is a vector of k residual series assumed to be Normally distributed with zero mean and positive-definite covariance matrix Σ. The components of ϵ_t are assumed to be uncorrelated at non-simultaneous lags. The ϕ_is and θ_js are k by k matrices of parameters. $\{\phi_i\}$, for $i = 1, 2, \ldots, p$, are called the autoregressive (AR) parameter matrices, and $\{\theta_j\}$, for $j = 1, 2, \ldots, q$, the moving average (MA) parameter matrices. The parameters in the model are thus the p k by k ϕ-matrices, the q k by k θ-matrices, the mean vector μ and the residual error covariance matrix Σ. Let

$$A(\phi) = \begin{bmatrix} \phi_1 & I & 0 & . & . & . & 0 \\ \phi_2 & 0 & I & 0 & . & . & 0 \\ . & & & . & & & \\ . & & & & . & & \\ . & & & & & . & \\ \phi_{p-1} & 0 & . & . & . & 0 & I \\ \phi_p & 0 & . & . & . & 0 & 0 \end{bmatrix}_{pk\times pk}$$

and

$$B(\theta) = \begin{bmatrix} \theta_1 & I & 0 & . & . & . & 0 \\ \theta_2 & 0 & I & 0 & . & . & 0 \\ . & & & . & & & \\ . & & & & . & & \\ . & & & & & . & \\ \theta_{q-1} & 0 & . & . & . & 0 & I \\ \theta_q & 0 & . & . & . & 0 & 0 \end{bmatrix}_{qk\times qk}$$

where I denotes the k by k identity matrix.

The model (1) must be both stationary and invertible. The model is said to be stationary if the eigenvalues of $A(\phi)$ lie inside the unit circle and invertible if the eigenvalues of $B(\theta)$ lie inside the unit circle.

For $k \geq 6$ the VARMA model (1) is recast into state space form and a realisation of the state vector at time zero computed. For all other cases the routine computes a realisation of the pre-observed vectors $W_0, W_{-1}, \ldots, W_{1-p}$, $\epsilon_0, \epsilon_{-1}, \ldots, \epsilon_{1-q}$, from equation (1), see Shea (1988). This realisation is then used to generate a sequence of successive time series observations. Note that special action is taken for pure MA models, that is for $p = 0$.

At the user's request a new realisation of the time series may be generated with less computation using only the information saved in a reference vector from a previous call to G05HDF. See the description of the parameter MODE in Section 5 for details.

The routine returns a realisation of $W_1, W_2, \ldots, W_n$. On a successful exit, the recent history is updated and saved in the array REF so that G05HDF may be called again to generate a realisation of $W_{n+1}, W_{n+2}, \ldots$, etc. See the description of the parameter MODE in Section 5 for details.

Further computational details are given in Shea (1988). Note however that this routine uses a spectral decomposition rather than a Cholesky factorisation to generate the multivariate Normals. Although this method involves more multiplications than the Cholesky factorisation method and is thus slightly slower it is more stable when faced with ill-conditioned covariance matrices. A method of assigning the AR and MA coefficient matrices so that the stationarity and invertibility conditions are satisfied is described in Barone (1987).

4 References

Barone P (1987) A method for generating independent realisations of a multivariate normal stationary and invertible ARMA(p, q) process *J. Time Ser. Anal.* **8** 125–130

Shea B L (1988) A note on the generation of independent realisations of a vector autoregressive moving average process *J. Time Ser. Anal.* **9** 403–410

5 Parameters

1: MODE – CHARACTER*1 *Input*

On entry: must be set as follows:

if MODE = 'S' (**S**tart), the routine is being called for the first time; a realisation of the recent history is computed, and the sequence of time series values from the VARMA model is then generated;

if MODE = 'R' (**R**estart), the routine must have been called before with the same VARMA model; a new realisation of the recent history is computed using information stored in the reference vector, followed by the sequence of time series values;

if MODE = 'C' (**C**ontinue), the routine must have been called before with the same VARMA model; a new sequence is generated, from the point at which the last sequence ended, using a realisation of the recent history which was updated and stored by the previous call to the routine;

if MODE = 'R' or 'C', then the user must ensure that the reference vector REF and the values of K, IP, IQ, MEAN, PAR, QQ and IK have not been changed between calls to G05HDF.

Constraint: MODE = 'S', 'R' or 'C'.

2: K – INTEGER *Input*

On entry: the dimension k of the multivariate time series.

Constraint: K ≥ 1.

3: IP – INTEGER *Input*

On entry: the number of AR parameter matrices, p.

Constraint: IP ≥ 0.

4: IQ – INTEGER *Input*

On entry: the number of MA parameter matrices, q.

Constraint: IQ ≥ 0.

5: MEAN – CHARACTER*1 *Input*

On entry: indicates whether or not all elements of μ are to be supplied by the user or to be taken as zero.

If MEAN = 'M', the values of μ, are supplied in the array PAR.

If MEAN = 'Z', all elements of μ are to be taken as zero.

Constraint: MEAN = 'M' or 'Z'.

6: PAR(LPAR) – ***real*** array *Input*

On entry: the parameter values read in row by row in the order $\phi_1, \phi_2, \ldots, \phi_p$, $\theta_1, \theta_2, \ldots, \theta_q$, μ.

If IP > 0, then PAR$((l-1) \times k \times k + (i-1) \times k + j)$ must be set equal to the (i,j)th element of ϕ_l, for $l = 1, 2, \ldots, p$; $i, j = 1, 2, \ldots, k$.

If IQ > 0, then PAR$(p \times k \times k + (l-1) \times k \times k + (i-1) \times k + j)$ must be set equal to the (i,j)th element of θ_l, for $l = 1, 2, \ldots, q$; $i, j = 1, 2, \ldots, k$.

If MEAN = 'M', then PAR$((p+q) \times k \times k + i)$ must be set equal to the ith component of the mean vector μ, for $i = 1, 2, \ldots, k$.

Constraint: the first IP $\times$ K $\times$ K elements of PAR must satisfy the stationarity condition and the next IQ $\times$ K $\times$ K elements of PAR must satisfy the invertibility condition.

7: LPAR – INTEGER *Input*

On entry: the dimension of the array PAR as declared in the (sub)program from which G05HDF is called.

Constraints:

LPAR $\geq$ max$(1, npar)$ where

npar = (IP + IQ) $\times$ K $\times$ K if MEAN = 'Z' and
npar = (IP + IQ) $\times$ K $\times$ K + K if MEAN = 'M'.

8: QQ(IK,K) – ***real*** array *Input/Output*

On entry: QQ(i,j) must contain the (i,j)th element of Σ. Only the lower triangle is required.

On exit: used as internal workspace prior to being restored and hence is unchanged.

Constraint: the elements of QQ must be such that Σ is positive-definite.

9: IK – INTEGER *Input*

On entry: the first dimension of the arrays QQ and W as declared in the (sub)program from which G05HDF is called.

Constraint: IK $\geq$ K.

10: N – INTEGER *Input*

On entry: the number of observations to be generated, n.

Constraint: N ≥ 1.

11: W(IK,N) – ***real*** array *Output*

On exit: W(i,t) will contain a realisation of the ith component of W_t, for $i = 1, 2, \ldots, k$; $t = 1, 2, \ldots, n$.

12: REF(LREF) – ***real*** array *Input/Output*

On entry: if MODE = 'R' or 'C', then the array REF as output from the previous call to G05HDF must be input without any change to the first $m + (k+1)(k+2) + (m+1)(m+2)$ elements where $m = k \times \max(p, q)$ if $k \geq 6$ and $k(p+q)$ if $k < 6$.

If MODE = 'S', then the contents of REF need not be set.

On exit: the first $m + (k+1)(k+2) + (m+1)(m+2)$ elements of the array REF contain information required for any subsequent calls to the routine with MODE = 'R' or 'C'; the rest of the array is used as workspace. See Section 8.

13: LREF – INTEGER *Input*

On entry: the dimension of the array REF as declared in the (sub)program from which G05HDF is called.

Constraints:

Let $r = \max(\mathrm{IP}, \mathrm{IQ})$

and $l = \mathrm{K}(\mathrm{K}+1)/2$ if $\mathrm{IP} = 0$,
$l = \mathrm{K}(\mathrm{K}+1)/2 + (\mathrm{IP}-1)\mathrm{K}^2$ if $\mathrm{IP} \geq 1$.

If $\mathrm{K} \geq 6$, then $\mathrm{LREF} \geq (5r^2+1)\mathrm{K}^2 + (4r+3)\mathrm{K} + 4$.

If $\mathrm{K} < 6$, then $\mathrm{LREF} \geq ((\mathrm{IP}+\mathrm{IQ})^2+1)\mathrm{K}^2 + (4(\mathrm{IP}+\mathrm{IQ})+3)\mathrm{K} + \max\{\mathrm{K}r(\mathrm{K}r+2), \mathrm{K}^2(\mathrm{IP}+\mathrm{IQ})^2 + l(l+3) + \mathrm{K}^2(\mathrm{IQ}+1)\} + 4$.

See Section 8 for some examples of the required size of the array REF.

14: IWORK(LIWORK) – INTEGER array *Workspace*
15: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which G05HDF is called.

Constraint: $\mathrm{LIWORK} \geq \mathrm{K} \times \max(\mathrm{IP}, \mathrm{IQ})$.

16: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, MODE ≠ 'S', 'R' or 'C',
or K < 1,
or IP < 0,
or IQ < 0,
or MEAN ≠ 'M' or 'Z',
or $\mathrm{LPAR} < \max((\mathrm{IP}+\mathrm{IQ}) \times \mathrm{K} \times \mathrm{K} + \mathrm{K}, 1)$ and MEAN = 'M',

or LPAR $<$ max((IP + IQ) $\times$ K $\times$ K, 1) and MEAN = 'Z',
or IK $<$ K,
or N $<$ 1,
or LREF is too small,
or LIWORK is too small.

IFAIL = 2

On entry, either the value of Σ is not positive-definite, or the AR parameters are such that the model is non-stationary, or the MA parameters are such that the model is non-invertible. To proceed, the user must try different parameter values.

IFAIL = 3

This is an unlikely exit brought about by an excessive number of iterations being needed by the routine used to evaluate the eigenvalues of $A(\phi)$ or $B(\theta)$. If this error occurs please contact NAG.

IFAIL = 4

G05HDF has not been able to calculate all the required elements of the array REF. This is an unlikely exit brought about by an excessive number of iterations being needed by the routine to evaluate eigenvalues to be stored in the array REF. If this error occurs please contact NAG.

IFAIL = 5

G05HDF has not been able to calculate all the required elements of the array REF. This is likely to be because the AR parameters are very close to the boundary of the stationarity region.

IFAIL = 6

The reference vector REF has been corrupted, when MODE is set to 'R' or 'C'. To proceed, the user should set MODE to 'S'.

7 Accuracy

The accuracy is limited by the matrix computations performed, and this is dependent on the condition of the parameter and covariance matrices.

8 Further Comments

Note that, in reference to IFAIL = 2, G05HDF will permit MA parameters on the boundary of the invertibility region.

The elements of REF contain amongst other information details of the spectral decompositions which are used to generate future multivariate Normals. Note that these eigenvectors may not be unique on different machines. For example the eigenvectors corresponding to multiple eigenvalues may be permuted. Although an effort is made to ensure that the eigenvectors have the same sign on all machines, differences in the signs may theoretically still occur.

The following table gives some examples of the required size of the array REF, specified by the parameter LREF, for $k = 1, 2, 3$, and for various values of p and q.

		q			
		0	1	2	3
p	0	13 36 85	20 56 124	31 92 199	46 144 310
	1	19 52 115	30 88 190	45 140 301	64 208 448
	2	35 136 397	50 188 508	69 256 655	92 340 838
	3	57 268 877	76 336 1024	99 420 1207	126 520 1426

Note that the routine G13DXF may be used to check whether a VARMA model is stationary and invertible.

The time taken depends on the values of p, q and especially n and k.

9 Example

This program generates two realisations, each of length 48, from the bivariate AR(1) model

$$W_t - \mu = \phi_1(W_{t-1} - \mu) + \epsilon_t$$

with

$$\phi_1 = \begin{bmatrix} 0.80 & 0.07 \\ 0.00 & 0.58 \end{bmatrix},$$

$$\mu = \begin{bmatrix} 5.00 \\ 9.00 \end{bmatrix},$$

and

$$\Sigma = \begin{bmatrix} 2.97 & 0 \\ 0.64 & 5.38 \end{bmatrix}.$$

In the first call MODE is set to 'S' in order to set up the reference vector before generating the first realisation. In the subsequent call MODE is set to 'R' and a new recent history is generated and used to generate the second realisation.

The generator mechanism used is selected by an initial call to G05ZAF.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     G05HDF Example Program Text
*     Mark 20 Revised. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          KMAX, IK, IPMAX, IQMAX, LPARMX, NMAX, LREF,
     +                 LIWORK
      PARAMETER        (KMAX=3,IK=KMAX,IPMAX=2,IQMAX=2,
     +                 LPARMX=(IPMAX+IQMAX)*KMAX*KMAX+KMAX,NMAX=100,
```

```
     +                  LREF=554,LIWORK=10)
*     .. Local Scalars ..
      INTEGER           I, IFAIL, IP, IQ, J, K, N, NPAR
      CHARACTER         MEAN
*     .. Local Arrays ..
      real              PAR(LPARMX), QQ(IK,KMAX), REF(LREF), W(IK,NMAX)
      INTEGER           IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL          G05CBF, G05HDF, G05ZAF
*     .. Executable Statements ..
      CALL G05ZAF('O')
      WRITE (NOUT,*) 'G05HDF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) K, IP, IQ, N, MEAN
*
      IF (K.GT.0 .AND. K.LE.KMAX .AND. IP.GE.0 .AND. IP.LE.IPMAX .AND.
     +    IQ.GE.0 .AND. IQ.LE.IQMAX) THEN
         NPAR = (IP+IQ)*K*K
         IF (MEAN.EQ.'M' .OR. MEAN.EQ.'m') NPAR = NPAR + K
         IF (N.GT.0 .AND. N.LE.NMAX) THEN
            READ (NIN,*) (PAR(I),I=1,NPAR)
            DO 20 I = 1, K
               READ (NIN,*) (QQ(I,J),J=1,I)
   20       CONTINUE
*
            CALL G05CBF(0)
*
            IFAIL = 0
*
            CALL G05HDF('Start',K,IP,IQ,MEAN,PAR,NPAR,QQ,IK,N,W,REF,
     +                  LREF,IWORK,LIWORK,IFAIL)
*
            WRITE (NOUT,*)
            WRITE (NOUT,*) ' Realisation Number 1'
*
            DO 40 I = 1, K
               WRITE (NOUT,99999) '  Series number ', I
               WRITE (NOUT,*) '  -------------'
               WRITE (NOUT,*)
               WRITE (NOUT,99998) (W(I,J),J=1,N)
   40       CONTINUE
*
            IFAIL = 0
*
            CALL G05HDF('Restart',K,IP,IQ,MEAN,PAR,NPAR,QQ,IK,N,W,REF,
     +                  LREF,IWORK,LIWORK,IFAIL)
*
            WRITE (NOUT,*)
            WRITE (NOUT,*)
            WRITE (NOUT,*) ' Realisation Number 2'
*
            DO 60 I = 1, K
               WRITE (NOUT,99999) '  Series number ', I
               WRITE (NOUT,*) '  -------------'
               WRITE (NOUT,*)
               WRITE (NOUT,99998) (W(I,J),J=1,N)
   60       CONTINUE
*
         END IF
      END IF
      STOP
*
99999 FORMAT (/1X,A,I3)
99998 FORMAT (8(2X,F8.3))
      END
```

9.2 Program Data

```
G05HDF Example Program Data
```

```
2 1 0 48 'M'                      : K, IP, IQ, N, MEAN
0.80 0.07 0.00 0.58 5.00 9.00     : PAR
2.97                              : QQ
0.64 5.38
```

9.3 Program Results

```
G05HDF Example Program Results

 Realisation Number 1

  Series number   1
  -------------

    4.722     6.101     3.707     2.501     2.757     7.143     4.752     5.624
    5.197     3.596     4.752     4.441     4.733     4.970     5.585     4.820
    4.847     1.414     0.548     1.212     0.203    -1.066    -1.992    -1.765
   -0.948    -0.311     5.809     2.649     6.345     3.522     3.982     2.394
   -0.972    -1.839    -2.293    -1.304    -2.571    -0.447     2.301    -1.910
   -2.155    -0.375     1.737     3.194     2.236     3.504     4.163     5.562

  Series number   2
  -------------

    1.434    -0.767     0.403    -3.162     1.674     1.241     0.596    -2.187
   -0.681    -3.079    -0.786     4.618     3.477     2.454     4.775     0.451
    3.902    -0.017    -3.620    -1.489    -4.478    -5.614    -5.265     0.275
   -1.325    -5.113    -1.932    -1.989    -2.075    -3.710    -3.205    -5.205
   -4.857    -0.731     1.350    -2.720    -0.110    -0.161     1.944     2.219
    0.387     2.266     2.049     0.214     0.638    -0.026    -0.822     1.735


 Realisation Number 2

  Series number   1
  -------------

    1.289     2.151     0.168     2.621     3.190     0.488     1.254    -0.500
   -2.296    -4.731    -4.677    -2.975    -0.964     0.694    -1.225    -0.809
   -3.127    -1.230    -2.990    -5.861    -3.854    -6.995    -5.921    -4.316
   -5.010    -6.696    -6.756    -5.418    -5.828    -4.878    -4.853    -4.411
   -2.006    -1.415    -2.847    -3.532    -3.479    -0.209     0.034     0.565
    0.962     2.168     2.168     3.531     2.702     1.524    -2.152     0.718

  Series number   2
  -------------

   -0.521    -2.962    -3.000    -0.633    -2.936    -5.076    -2.939    -1.469
   -3.834    -1.155    -4.113    -3.726    -0.830    -0.403    -4.221    -3.672
    2.775     0.979    -0.732    -0.063     1.502    -3.152    -6.403    -5.306
   -3.899    -5.066    -2.451    -0.843    -1.178    -4.910    -5.041    -1.291
    0.818     1.957    -2.845     2.020     0.847    -2.144    -1.480    -1.286
    1.859    -6.834    -3.248    -1.032    -1.977    -1.491    -0.719    -0.196
```