NAG Library Routine Document

G05PJF

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

G05PJF generates a realization of a multivariate time series from a vector autoregressive moving average (VARMA) model. The realization may be continued or a new realization generated at subsequent calls to G05PJF.

2 Specification

```
      SUBROUTINE G05PJF(MODE, N, K, XMEAN, IP, PHI, IQ, THETA, VAR, LDVAR, R,
     1                  LR, STATE, X, LDX, IFAIL)
      INTEGER           MODE, N, K, IP, IQ, LDVAR, LR, STATE(lstate), LDX,
     1                  IFAIL
      double precision  XMEAN(K), PHI(IP*K*K), THETA(IQ*K*K), VAR(LDVAR,K),
     1                  R(LR), X(LDX,N)
```

3 Description

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

$$\begin{aligned} X_t - \mu = \quad & \phi_1(X_{t-1} - \mu) + \phi_2(X_{t-2} - \mu) + \cdots + \phi_p(X_{t-p} - \mu) + \\ & \epsilon_t - \theta_1\epsilon_{t-1} - \theta_2\epsilon_{t-2} - \cdots - \theta_q\epsilon_{t-q} \end{aligned} \tag{1}$$

where $\epsilon_t = (\epsilon_{1t}, \epsilon_{2t}, \ldots, \epsilon_{kt})^{\mathrm{T}}$, is a vector of k residual series assumed to be Normally distributed with zero mean and covariance matrix Σ. The components of ϵ_t are assumed to be uncorrelated at non-simultaneous lags. The ϕ_i's and θ_j's 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

$$A(\phi) = \begin{bmatrix} \phi_1 & I & 0 & . & . & . & 0 \\ \phi_2 & 0 & I & 0 & . & . & 0 \\ . & & & . & & & \\ . & & & & . & & \\ . & & & & & . & \\ \phi_{p-1} & 0 & . & . & . & 0 & I \\ \phi_p & 0 & . & . & . & 0 & 0 \end{bmatrix}_{pk \times pk} \quad \text{and} \quad B(\theta) = \begin{bmatrix} \theta_1 & I & 0 & . & . & . & 0 \\ \theta_2 & 0 & I & 0 & . & . & 0 \\ . & & & . & & & \\ . & & & & . & & \\ . & & & & & . & \\ \theta_{q-1} & 0 & . & . & . & 0 & I \\ \theta_q & 0 & . & . & . & 0 & 0 \end{bmatrix}_{qk \times qk}$$

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 realization of the state vector at time zero computed. For all other cases the routine computes a realization of the pre-observed vectors $X_0, X_{-1}, \ldots, X_{1-p}$, $\epsilon_0, \epsilon_{-1}, \ldots, \epsilon_{1-q}$, from (1), see Shea (1988). This realization 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 your request a new realization of the time series may be generated more efficiently using the information in a reference vector created during a previous call to G05PJF. See the description of the parameter MODE in Section 5 for details.

The routine returns a realization of $X_1, X_2, \ldots, X_n$. On a successful exit, the recent history is updated and saved in the array R so that G05PJF may be called again to generate a realization of $X_{n+1}, X_{n+2}, \ldots$,etc. See the description of the parameter MODE in Section 5 for details.

Further computational details are given in Shea (1988). Note, however, that G05PJF uses a spectral decomposition rather than a Cholesky factorization to generate the multivariate Normals. Although this method involves more multiplications than the Cholesky factorization 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).

One of the initialization routines G05KFF (for a repeatable sequence if computed sequentially) or G05KGF (for a non-repeatable sequence) must be called prior to the first call to G05PJF.

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 – INTEGER *Input*

On entry: a code for selecting the operation to be performed by the routine.

MODE $= 0$
Set up reference vector and compute a realization of the recent history.

MODE $= 1$
Generate terms in the time series using reference vector set up in a prior call to G05PJF.

MODE $= 2$
Combine the operations of MODE $= 0$ and 1.

MODE $= 3$
A new realization of the recent history is computed using information stored in the reference vector, and the following sequence of time series values are generated.

If MODE $= 1$ or 3, then you must ensure that the reference vector R and the values of K, IP, IQ, XMEAN, PHI, THETA, VAR and LDVAR have not been changed between calls to G05PJF.

Constraint: MODE $= 0$, 1, 2 or 3.

2: N – INTEGER *Input*

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

Constraint: N ≥ 0.

3: K – INTEGER *Input*

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

Constraint: K ≥ 1.

4: XMEAN(K) – ***double precision*** array *Input*

On entry: μ, the vector of means of the multivariate time series.

5: IP – INTEGER *Input*

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

Constraint: IP ≥ 0.

6: PHI(IP $\times$ K $\times$ K) – ***double precision*** array *Input*

On entry: must contain the elements of the IP $\times$ K $\times$ K autoregressive parameter matrices of the model, $\phi_1, \phi_2, \ldots, \phi_p$. If PHI is considered as a three-dimensional array, dimensioned as PHI(K, K, IP), then the (i,j)th element of ϕ_l would be stored in PHI(i,j,l); that is, PHI$((l-1) \times k \times k + (j-1) \times k + i)$ must be set equal to the (i,j)th element of ϕ_l, for $l = 1, 2, \ldots, p$ and $i, j = 1, 2, \ldots, k$.

Constraint: the elements of PHI must satisfy the stationarity condition.

7: IQ – INTEGER *Input*

On entry: q, the number of moving average parameter matrices.

Constraint: IQ ≥ 0.

8: THETA(IQ $\times$ K $\times$ K) – ***double precision*** array *Input*

On entry: must contain the elements of the IQ $\times$ K $\times$ K moving average parameter matrices of the model, $\theta_1, \theta_2, \ldots, \theta_q$. If THETA is considered as a three-dimensional array, dimensioned as THETA(K,K,IQ), then the (i,j)th element of θ_l would be stored in THETA(i,j,l); that is, THETA$((l-1) \times k \times k + (j-1) \times k + i)$ must be set equal to the (i,j)th element of θ_l, for $l = 1, 2, \ldots, q$ and $i, j = 1, 2, \ldots, k$.

Constraint: the elements of THETA must be within the invertibility region.

9: VAR(LDVAR,K) – ***double precision*** array *Input*

On entry: VAR(i,j) must contain the (i,j)th element of Σ, for $i, j = 1, 2, \ldots,$ K. Only the lower triangle is required.

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

10: LDVAR – INTEGER *Input*

On entry: the first dimension of the arrays VAR and X as declared in the (sub)program from which G05PJF is called.

Constraint: LDVAR $\geq$ K.

11: R(LR) – ***double precision*** array *Communication Array*

On entry: if MODE $= 1$ or 3, the array R as output from the previous call to G05PJF must be input without any change.

If MODE $= 0$ or 2, the contents of R need not be set.

On exit: information required for any subsequent calls to the routine with MODE $= 1$ or 3. See Section 8.

12: LR – INTEGER *Input*

On entry: the dimension of the array R as declared in the (sub)program from which G05PJF is called.

Constraints:

$$\text{if K} \geq 6,\ \text{LR} \geq \left(5r^2 + 1\right) \times \text{K}^2 + (4r + 3) \times \text{K} + 4;$$
$$\text{if K} < 6,\ \text{LR} \geq \left((\text{IP} + \text{IQ})^2 + 1\right) \times \text{K}^2 +$$
$$(4 \times (\text{IP} + \text{IQ}) + 3) \times \text{K} + \max\left\{\text{K}r(\text{K}r + 2), \text{K}^2(\text{IP} + \text{IQ})^2 + l(l + 3) + \text{K}^2(\text{IQ} + 1)\right\} + 4.$$

Where $r = \max(\text{IP}, \text{IQ})$ and if IP $= 0$, $l = \text{K}(\text{K} + 1)/2$, or if IP ≥ 1, $l = \text{K}(\text{K} + 1)/2 + (\text{IP} - 1)\text{K}^2$.

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

13: STATE($lstate$) – INTEGER array *Communication Array*

Note: $lstate = \text{LSTATE}$, the dimension of STATE as supplied to the initialization routine G05KFF or G05KGF.

On entry: contains information on the selected base generator and its current state.

On exit: contains updated information on the state of the generator.

14: X(LDX,N) – ***double precision*** array *Output*

On exit: $\text{X}(i,t)$ will contain a realization of the ith component of X_t, for $i = 1, 2, \ldots, k$ and $t = 1, 2, \ldots, n$.

15: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G05PJF is called.

Constraint: $\text{LDX} \geq \text{K}$.

16: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

On exit: $\text{IFAIL} = 0$ 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, if you are 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 $\text{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\text{IFAIL} = 1$

On entry, $\text{MODE} \neq 0$, 1, 2 or 3.

$\text{IFAIL} = 2$

On entry, $\text{N} < 0$.

$\text{IFAIL} = 3$

On entry, $\text{K} < 1$.

$\text{IFAIL} = 5$

On entry, $\text{IP} < 0$.

$\text{IFAIL} = 6$

The autoregressive parameter matrices, stored in PHI, are such that the model is non-stationary.

$\text{IFAIL} = 7$

On entry, $\text{IQ} < 0$.

IFAIL $= 8$

On entry, the moving average parameter matrices, stored in THETA, are such that the model is non-invertible.

IFAIL $= 9$

The covariance matrix Σ, stored in VAR, is not positive-definite.

IFAIL $= 10$

On entry, LDVAR $<$ K.

IFAIL $= 11$

Either R has been corrupted or the value of K is not the same as when R was set up in a previous call to G05PJF with MODE $= 0$ or 2.

IFAIL $= 12$

On entry, LR is too small.

IFAIL $= 13$

On entry, STATE vector was not initialized or has been corrupted.

IFAIL $= 15$

On entry, LDX $<$ K.

IFAIL $= 20$

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

IFAIL $= 21$

G05PJF has not been able to calculate all the required elements of the array R. This is likely to be because the AR parameters are very close to the boundary of the stationarity region.

IFAIL $= 22$

This is an unlikely exit brought about by an excessive number of iterations being needed by the NAG Library routine used to evaluate the eigenvalues of the covariance matrix.

IFAIL $= 23$

G05PJF has not been able to calculate all the required elements of the array R. This is an unlikely exit brought about by an excessive number of iterations being needed by the NAG Library routine used to evaluate eigenvalues to be stored in the array R. If this error occurs please contact NAG.

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 $= 8$, G05PJF will permit moving average parameters on the boundary of the invertibility region.

The elements of R 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 R, specified by the parameter LR, for $k = 1, 2$ or 3, and for various values of p and q.

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

Note that 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 realizations, each of length 48, from the bivariate AR(1) model

$$X_t - \mu = \phi_1(X_{t-1} - \mu) + \epsilon_t$$

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

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

The pseudorandom number generator is initialized by a call to G05KFF. Then, in the first call to G05PJF, MODE $= 2$ in order to set up the reference vector before generating the first realization. In the subsequent call MODE $= 3$ and a new recent history is generated and used to generate the second realization.

9.1 Program Text

```
*     G05PJF Example Program Text
*
*     NAG COPYRIGHT 2008.
*
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MSEED, MSTATE
      PARAMETER        (MSEED=1,MSTATE=633)
```

```
      INTEGER          KMAX, LDV, IPMAX, IQMAX, NMAX, LR, LDX
      PARAMETER        (KMAX=3,LDV=KMAX,IPMAX=2,IQMAX=2,NMAX=100,LR=600,
     +                 LDX=KMAX)
*     .. Local Scalars ..
      INTEGER          GENID, I, IFAIL, II, IP, IQ, J, K, L, LSEED,
     +                 LSTATE, N, SUBID
*     .. Local Arrays ..
      DOUBLE PRECISION PHI(KMAX*KMAX*IPMAX), R(LR),
     +                 THETA(KMAX*KMAX*IQMAX), VAR(LDV,KMAX),
     +                 X(LDX,NMAX), XMEAN(KMAX)
      INTEGER          SEED(MSEED), STATE(MSTATE)
*     .. External Subroutines ..
      EXTERNAL         G05KFF, G05PJF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G05PJF Example Program Results'
      WRITE (NOUT,*)
*     Read data from a file
*     Skip heading
      READ (NIN,*)
*     Read in initial parameters
      READ (NIN,*) K, IP, IQ, N
*     Check the specified array limits
      IF (K.LE.0 .OR. K.GT.KMAX .OR. IP.LT.0 .OR. IP.GT.IPMAX .OR.
     +    IQ.LT.0 .OR. IQ.GT.IQMAX .OR. N.LE.0 .OR. N.GT.NMAX) THEN
         WRITE (NOUT,99995)
         GO TO 160
      END IF
*     Read in the AR parameters
      DO 40 L = 1, IP
         DO 20 I = 1, K
            II = (L-1)*K*K + I
            READ (NIN,*) (PHI(II+K*(J-1)),J=1,K)
   20    CONTINUE
   40 CONTINUE
*     Read in the MA parameters
      DO 80 L = 1, IQ
         DO 60 I = 1, K
            II = (L-1)*K*K + I
            READ (NIN,*) (THETA(II+K*(J-1)),J=1,K)
   60    CONTINUE
   80 CONTINUE
*     Read in the means
      READ (NIN,*) (XMEAN(I),I=1,K)
*     Read in the variance / covariance matrix
      DO 100 I = 1, K
         READ (NIN,*) (VAR(I,J),J=1,I)
  100 CONTINUE
*     Initialize the seed
      SEED(1) = 1762543
*     GENID and SUBID identify the base generator
      GENID = 1
      SUBID = 1
*     Initialize the generator to a repeatable sequence
      LSTATE = MSTATE
      LSEED = MSEED
      IFAIL = 1
      CALL G05KFF(GENID,SUBID,SEED,LSEED,STATE,LSTATE,IFAIL)
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99997) IFAIL
         GO TO 160
      END IF
*     Generate the first realisation
*     Use MODE = 2 to set up R and generate values
      IFAIL = 1
      CALL G05PJF(2,N,K,XMEAN,IP,PHI,IQ,THETA,VAR,LDV,R,LR,STATE,X,LDX,
     +            IFAIL)
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99996) IFAIL
         GO TO 160
      END IF
*     Display the results
```

```
      WRITE (NOUT,*)
      WRITE (NOUT,*) ' Realisation Number 1'
      DO 120 I = 1, K
         WRITE (NOUT,99999) '  Series number ', I
         WRITE (NOUT,*) '  -------------'
         WRITE (NOUT,*)
         WRITE (NOUT,99998) (X(I,J),J=1,N)
  120 CONTINUE
*     Generate a second realisation
*     Use MODE = 3 to reset the series and generate values
      IFAIL = 1
      CALL G05PJF(3,N,K,XMEAN,IP,PHI,IQ,THETA,VAR,LDV,R,LR,STATE,X,LDX,
     +            IFAIL)
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99996) IFAIL
         GO TO 160
      END IF
*     Display the results
      WRITE (NOUT,*)
      WRITE (NOUT,*) ' Realisation Number 2'
      DO 140 I = 1, K
         WRITE (NOUT,99999) '  Series number ', I
         WRITE (NOUT,*) '  -------------'
         WRITE (NOUT,*)
         WRITE (NOUT,99998) (X(I,J),J=1,N)
  140 CONTINUE
*
  160 CONTINUE
*
99999 FORMAT (/1X,A,I3)
99998 FORMAT (8(2X,F8.3))
99997 FORMAT (1X,' ** G05KFF returned with IFAIL = ',I5)
99996 FORMAT (1X,' ** G05PJF returned with IFAIL = ',I5)
99995 FORMAT (1X,' ** Problem size too large, increase array limits')
      END
```

9.2 Program Data

```
G05PJF Example Program Data
 2 1 0 48                   : K, IP, IQ, N
 0.80 0.07
 0.00 0.58                  : End of PHI(,,1)
 5.00 9.00                  : XMEAN
 2.97
 0.64 5.38                  : End of matrix VAR
```

9.3 Program Results

```
 G05PJF Example Program Results


  Realisation Number 1

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

     4.833     2.813     3.224     3.825     1.023     1.415     2.184     3.005
     5.547     4.832     4.705     5.484     9.407    10.335     8.495     7.478
     6.373     6.692     6.698     6.976     6.200     4.458     2.520     3.517
     3.054     5.439     5.699     7.136     5.750     8.497     9.563    11.604
     9.020    10.063     7.976     5.927     4.992     4.222     3.982     7.107
     3.554     7.045     7.025     4.106     5.106     5.954     8.026     7.212

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

     8.458     9.140    10.866    10.975     9.245     5.054     5.023    12.486
    10.534    10.590    11.376     8.793    14.445    13.237    11.030     8.405
     7.187     8.291     5.920     9.390    10.055     6.222     7.751    10.604
    12.441    10.664    10.960     8.022    10.073    12.870    12.665    14.064
```

```
   11.867    12.894    10.546    12.754     8.594     9.042    12.029    12.557
    9.746     5.487     5.500     8.629     9.723     8.632     6.383    12.484

 Realisation Number 2

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

    5.396     4.811     2.685     5.824     2.449     3.563     5.663     6.209
    3.130     4.308     4.333     4.903     1.770     1.278     1.340    -0.527
    1.745     3.211     4.478     5.170     5.365     4.852     6.080     6.464
    2.765     2.148     6.641     7.224    10.316     7.102     5.604     3.934
    4.839     3.698     5.210     5.384     7.652     7.315     7.332     7.561
    7.537     7.788     6.868     7.575     6.108     6.188     8.132    10.310

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

   11.345    10.070    13.654    12.409    11.329    13.054    12.465     9.867
   10.263    13.394    10.553    10.331     7.814     8.747    10.025    11.167
   10.626     9.366     9.607     9.662    10.492    10.766    11.512    10.813
   10.799     8.780     9.221    14.245    11.575    10.620     8.282     5.447
    9.935     9.386    11.627    10.066    11.394     7.951     7.907    12.616
   15.246     9.962    13.216    11.350    11.227     6.021     6.968    12.428
```