NAG Fortran Library Routine Document

G05PCF

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

G05PCF 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 G05PCF.

2 Specification

```
      SUBROUTINE G05PCF(MODE, K, XMEAN, IP, PHI, IQ, THETA, VAR, LDV, N, X,
     1                  IGEN, ISEED, R, NR, IWORK, LIWORK, IFAIL)
      INTEGER           MODE, K, IP, IQ, LDV, N, IGEN, ISEED(4), NR,
     1                  IWORK(LIWORK), LIWORK, IFAIL
      real              XMEAN(K), PHI(*), THETA(*), VAR(LDV,K), X(LDV,*),
     1                  R(NR)
```

3 Description

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

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

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_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

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

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 $X_0, X_{-1}, \ldots, X_{1-p}$, $\epsilon_0, \epsilon_{-1}, \ldots, \epsilon_{1-q}$, 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 G05PCF. See the description of the parameter MODE in Section 5 for details.

The routine returns a realisation 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 G05PCF may be called again to generate a realisation 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 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).

One of the initialisation routines G05KBF (for a repeatable sequence if computed sequentially) or G05KCF (for a non-repeatable sequence) must be called prior to the first call to G05PCF.

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 (start)

Set up reference vector and compute a realisation of the recent history.

MODE = 1 (continue)

Generate terms in the time series using reference vector set up in a prior call to G05PCF.

MODE = 2 (start and generate)

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

MODE = 3 (restart and generate)

A new realisation 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 the user must ensure that the reference vector R and the values of K, IP, IQ, XMEAN, PHI, THETA, VAR and LDV have not been changed between calls to G05PCF.

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

2: K – INTEGER *Input*

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

Constraint: K $\geq$ 1.

3: XMEAN(K) – ***real*** array *Input*

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

4: IP – INTEGER *Input*

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

Constraint: IP ≥ 0.

5: PHI($*$) – ***real*** array *Input*

Note: the dimension of the array PHI must be at least $\max(1, \mathrm{IP} \times \mathrm{K} \times \mathrm{K})$.

On entry: contains the elements of the IPK × K autoregressive parameter matrices of the model, ϕ_1, $\phi_2, \ldots$, ϕ_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 $l = 1, 2, \ldots, p$; $i, j = 1, 2, \ldots, k$.

Constraint: the first IP × K × K elements of PHI must satisfy the stationarity condition.

6: IQ – INTEGER *Input*

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

Constraint: IQ ≥ 0.

7: THETA($*$) – ***real*** array *Input*

Note: the dimension of the array THETA must be at least $\max(1, \mathrm{IQ} \times \mathrm{K} \times \mathrm{K})$.

On entry: contains the elements of the IQK × K moving average parameter matrices of the model, θ_1, $\theta_2, \ldots$, θ_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 $l = 1, 2, \ldots, q$; $i, j = 1, 2, \ldots, k$.

8: VAR(LDV,K) – ***real*** array *Input*

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

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

9: LDV – INTEGER *Input*

On entry: the first dimension of the array VAR X as declared in the (sub)program from which G05PCF is called.

Constraint: LDV $\geq$ K.

10: N – INTEGER *Input*

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

Constraint: N ≥ 0.

11: X(LDV,*) – ***real*** array *Output*

Note: the second dimension of the array X must be at least $\max(1, \mathrm{N})$.

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

12: IGEN – INTEGER *Input*

On entry: must contain the identification number for the generator to be used to return a pseudo-random number and should remain unchanged following initialisation by a prior call to one of the routines G05KBF or G05KCF.

13: ISEED(4) – INTEGER array *Input/Output*

On entry: contains values which define the current state of the selected generator.

On exit: contains updated values defining the new state of the selected generator.

14: R(NR) – ***real*** array *Input/Output*

On entry: if MODE $= 1$ or 2, then the array R as output from the previous call to G05PCF 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 $= 0$ or 2, then the contents of R need not be set.

On exit: the first $m+(k+1)(k+2)+(m+1)(m+2)$ elements of the array R contain information required for any subsequent calls to the routine with MODE $= 1$ or 3; the rest of the array is used as workspace. See Section 8.

15: NR – INTEGER *Input*

On entry: the dimension of the array R as declared in the (sub)program from which G05PCF is called.

Constraints:

Let $r = \max(\mathrm{IP}, \mathrm{IQ})$ and $l = \mathrm{K}(\mathrm{K}+1)/2$ if IP $= 0$, $l = \mathrm{K}(\mathrm{K}+1)/2 + (\mathrm{IP}-1)\mathrm{K}^2$ if IP ≥ 1.
If K ≥ 6, then NR $\geq (5r^2+1)\mathrm{K}^2 + (4r+3)\mathrm{K} + 4$.
If K < 6, then

$$\begin{aligned}\mathrm{NR} \geq \quad & ((\mathrm{IP}+\mathrm{IQ})^2+1)\mathrm{K}^2 + (4(\mathrm{IP}+\mathrm{IQ})+3)\mathrm{K} \\ & + \max\{\mathrm{K}r(\mathrm{K}r+2), \mathrm{K}^2(\mathrm{IP}+\mathrm{IQ})^2 + l(l+3) + \mathrm{K}^2(\mathrm{IQ}+1)\} + 4.\end{aligned}$$

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

16: IWORK(LIWORK) – INTEGER array *Workspace*
17: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which G05PCF is called.

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

18: 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, N < 0.

IFAIL = 2

On entry, K < 1.

IFAIL = 3

On entry, IP < 0.

IFAIL = 4

On entry, IQ < 0.

IFAIL = 5

On entry, LDV < K.

IFAIL = 6

On entry, MODE < 0 or MODE > 3.

IFAIL = 7

On entry, LIWORK < K × $\max(\mathrm{IP}, \mathrm{IQ})$.

IFAIL = 8

On entry, NR is too small.

IFAIL = 9

On entry, the covariance matrix Σ, as stored in VAR, is not positive-definite.

IFAIL = 10

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

IFAIL = 11

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

IFAIL = 12

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

IFAIL = 13

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

IFAIL = 14

G05PCF 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 = 15

G05PCF 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 Fortran Library routine used to evaluate eigenvalues to be stored in the array R. If this error occurs please contact NAG.

IFAIL $= 16$

Either R has been corrupted or the value of IP or IQ is not the same as when R was set up in a previous call with MODE $= 0$ or 2. To proceed, the user should set MODE to 0 or 2.

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 $= 12$, G05PCF 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 NR, for $k = 1, 2, 3$, and for various values of p and q.

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

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

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 pseudo-random number generator is initialised by a call to G05KBF. Then, in the first call to G05PCF, MODE is set to 2 in order to set up the reference vector before generating the first realisation. In the subsequent call MODE is set to 3 and a new recent history is generated and used to generate the second realisation.

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.

```
*     G05PCF Example Program Text
*     Mark 20 Release. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          KMAX, LDV, IPMAX, IQMAX, NMAX, NR, LIWORK
      PARAMETER        (KMAX=3,LDV=KMAX,IPMAX=2,IQMAX=2,NMAX=100,NR=600,
     +                 LIWORK=10)
*     .. Local Scalars ..
      INTEGER          I, IFAIL, IGEN, II, IP, IQ, J, K, L, N
*     .. Local Arrays ..
      real             PHI(KMAX*KMAX*IPMAX), R(NR),
     +                 THETA(KMAX*KMAX*IQMAX), VAR(LDV,KMAX),
     +                 X(LDV,NMAX), XMEAN(KMAX)
      INTEGER          ISEED(4), IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         G05KBF, G05PCF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G05PCF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) K, IP, IQ, N
*
      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
         IF (N.GT.0 .AND. N.LE.NMAX) THEN
            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
            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 (NIN,*) (XMEAN(I),I=1,K)
            DO 100 I = 1, K
               READ (NIN,*) (VAR(I,J),J=1,I)
  100       CONTINUE
*           Initialise the seed to a repeatable sequence
            ISEED(1) = 1762543
            ISEED(2) = 9324783
            ISEED(3) = 4234401
            ISEED(4) = 742355
*           IGEN identifies the stream.
            IGEN = 1
            CALL G05KBF(IGEN,ISEED)
*
            IFAIL = 0
*
            CALL G05PCF(2,K,XMEAN,IP,PHI,IQ,THETA,VAR,LDV,N,X,IGEN,
     +                  ISEED,R,NR,IWORK,LIWORK,IFAIL)
*
            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
*
            IFAIL = 0
*
            CALL G05PCF(3,K,XMEAN,IP,PHI,IQ,THETA,VAR,LDV,N,X,IGEN,
     +                  ISEED,R,NR,IWORK,LIWORK,IFAIL)
*
            WRITE (NOUT,*)
            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
*
         END IF
      END IF
      STOP
*
99999 FORMAT (/1X,A,I3)
99998 FORMAT (8(2X,F8.3))
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 G05PCF Example Program Results

  Realisation Number 1

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

     0.765    -1.017    -4.504    -5.754    -6.718    -7.753    -5.568    -3.281
    -2.620    -1.680    -1.850    -2.949    -3.696    -2.921    -2.060    -1.461
    -1.124    -1.821     2.230     0.791     1.297     1.508     2.993     2.041
     0.568     1.404     1.283    -1.770    -4.011    -5.397    -5.331    -2.474
    -3.371    -2.307    -1.907    -2.033     0.162     1.331     3.169     5.801
     4.174     0.627     2.696     0.844    -3.461    -2.623    -3.665    -5.144

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

     5.749     4.246    -0.174    -0.386    -0.221    -1.690    -3.039    -4.406
    -3.688    -2.646    -2.157    -1.864     1.653     0.416     1.083     2.012
     1.745    -0.193    -4.682    -2.491    -1.480     0.205     0.877    -0.918
    -0.592     2.218     0.648     0.260    -2.369     1.407    -1.216    -2.491
    -3.826    -0.406    -1.100    -0.425    -0.693     0.726    -3.697     0.989
     0.924     4.597     7.134     6.464     2.816    -0.572     2.221    -1.131
 Realisation Number 2

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

    1.253     4.454     3.619     2.403     1.684     2.998     1.669     2.011
    2.899     3.984     2.873     3.161     1.462     0.396    -2.457    -1.896
   -0.715    -1.163    -3.891    -2.628     0.804    -3.071     1.479     0.964
    1.594     3.835     4.217     3.546     2.011     2.797     3.386     3.290
    3.227    -0.837    -0.768     0.161     1.794     2.188     1.552     2.101
    0.061    -1.036     1.517     1.315    -1.011    -0.448    -1.921    -0.861

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

    2.114     4.425     1.316     0.063    -2.676    -5.327    -3.432    -1.897
   -0.439     0.097     2.745     1.028     3.138     0.973    -0.253     1.753
    6.455     1.861     5.161     0.624     3.976     1.141    -1.069    -0.711
    0.520     1.412    -0.752    -4.771    -5.166    -2.160    -0.633     3.120
    4.373     5.411     0.508     3.724     2.858     3.463     5.742     3.301
    5.039     3.476     4.437     2.757     2.972     0.273     0.496    -2.606
```