NAG Library Routine Document

G13DPF

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

G13DPF calculates the sample partial autoregression matrices of a multivariate time series. A set of likelihood ratio statistics and their significance levels are also returned. These quantities are useful for determining whether the series follows an autoregressive model and, if so, of what order.

2 Specification

```
SUBROUTINE G13DPF (K, N, Z, KMAX, M, MAXLAG, PARLAG, SE, QQ, X, PVALUE,        &
                   LOGLHD, WORK, LWORK, IWORK, IFAIL)

INTEGER            K, N, KMAX, M, MAXLAG, LWORK, IWORK(K*M), IFAIL
REAL (KIND=nag_wp) Z(KMAX,N), PARLAG(KMAX,KMAX,M), SE(KMAX,KMAX,M),            &
                   QQ(KMAX,KMAX,M), X(M), PVALUE(M), LOGLHD(M),                &
                   WORK(LWORK)
```

3 Description

Let $W_t = (w_{1t}, w_{2t}, \ldots, w_{kt})^{\mathrm{T}}$, for $t = 1, 2, \ldots, n$, denote a vector of k time series. The partial autoregression matrix at lag l, P_l, is defined to be the last matrix coefficient when a vector autoregressive model of order l is fitted to the series. P_l has the property that if W_t follows a vector autoregressive model of order p then $P_l = 0$ for $l > p$.

Sample estimates of the partial autoregression matrices may be obtained by fitting autoregressive models of successively higher orders by multivariate least squares; see Tiao and Box (1981) and Wei (1990). These models are fitted using a QR algorithm based on the routines G02DCF and G02DFF. They are calculated up to lag m, which is usually taken to be at most $n/4$.

The routine also returns the asymptotic standard errors of the elements of $\hat{P}_l$ and an estimate of the residual variance-covariance matrix $\hat{\Sigma}_l$, for $l = 1, 2, \ldots, m$. If S_l denotes the residual sum of squares and cross-products matrix after fitting an AR(l) model to the series then under the null hypothesis $H_0 : P_l = 0$ the test statistic

$$X_l = -\left((n - m - 1) - \tfrac{1}{2} - lk\right) \log\left(\frac{|S_l|}{|S_{l-1}|}\right)$$

is asymptotically distributed as χ^2 with k^2 degrees of freedom. X_l provides a useful diagnostic aid in determining the order of an autoregressive model. (Note that $\hat{\Sigma}_l = S_l/(n - l)$.) The routine also returns an estimate of the maximum of the log-likelihood function for each AR model that has been fitted.

4 References

Tiao G C and Box G E P (1981) Modelling multiple time series with applications *J. Am. Stat. Assoc.* **76** 802–816

Wei W W S (1990) *Time Series Analysis: Univariate and Multivariate Methods* Addison–Wesley

5 Parameters

1: K – INTEGER *Input*

On entry: k, the number of time series.

Constraint: $\mathrm{K} \geq 1$.

2: N – INTEGER *Input*

On entry: n, the number of observations in the time series.

Constraint: $\mathrm{N} \geq 4$.

3: Z(KMAX,N) – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{Z}(i,t)$ must contain the observation w_{it}, for $i = 1,2,\ldots,k$ and $t = 1,2,\ldots,n$.

4: KMAX – INTEGER *Input*

On entry: the first dimension of the arrays Z, PARLAG, SE and QQ and the second dimension of the arrays PARLAG, SE and QQ as declared in the (sub)program from which G13DPF is called.

Constraint: $\mathrm{KMAX} \geq \mathrm{K}$.

5: M – INTEGER *Input*

On entry: m, the number of partial autoregression matrices to be computed. If in doubt set $\mathrm{M} = 10$.

Constraint: $\mathrm{M} \geq 1$ and $\mathrm{N} - \mathrm{M} - (\mathrm{K} \times \mathrm{M} + 1) \geq \mathrm{K}$.

6: MAXLAG – INTEGER *Output*

On exit: the maximum lag up to which partial autoregression matrices (along with their likelihood ratio statistics and their significance levels) have been successfully computed. On a successful exit MAXLAG will equal M. If $\mathrm{IFAIL} = 2$ on exit then MAXLAG will be less than M.

7: PARLAG(KMAX,KMAX,M) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{PARLAG}(i,j,l)$ contains an estimate of the (i,j)th element of the partial autoregression matrix at lag l, $\hat{P}_l(ij)$, for $l = 1,2,\ldots,\mathrm{MAXLAG}$, $i = 1,2,\ldots,k$ and $j = 1,2,\ldots,k$.

8: SE(KMAX,KMAX,M) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{SE}(i,j,l)$ contains an estimate of the standard error of the corresponding element in the array PARLAG.

9: QQ(KMAX,KMAX,M) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{QQ}(i,j,l)$ contains an estimate of the (i,j)th element of the corresponding variance-covariance matrix $\hat{\Sigma}_l$, for $l = 1,2,\ldots,\mathrm{MAXLAG}$, $i = 1,2,\ldots,k$ and $j = 1,2,\ldots,k$.

10: X(M) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{X}(l)$ contains X_l, the likelihood ratio statistic at lag l, for $l = 1,2,\ldots,\mathrm{MAXLAG}$.

11: PVALUE(M) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{PVALUE}(l)$ contains the significance level of the statistic in the corresponding element of X.

12: LOGLHD(M) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{LOGLHD}(l)$ contains an estimate of the maximum of the log-likelihood function when an $\mathrm{AR}(l)$ model has been fitted to the series, for $l = 1,2,\ldots,\mathrm{MAXLAG}$.

13: WORK(LWORK) – REAL (KIND=nag_wp) array *Workspace*

14: LWORK – INTEGER *Input*

On entry: the dimension of the array WORK as declared in the (sub)program from which G13DPF is called.

Constraint: LWORK $\geq (k+1)k + l(4+k) + 2l^2$, where $l = mk + 1$.

15: IWORK(K × M) – INTEGER array *Workspace*

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.

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.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

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, K < 1,
or N < 4,
or KMAX $<$ K,
or M < 1,
or N $-$ M $-$ (K × M $+ 1) <$ K,
or LWORK is too small.

IFAIL $= 2$

The recursive equations used to compute the sample partial autoregression matrices have broken down at lag MAXLAG $+ 1$. This exit could occur if the regression model is overparameterised. For your settings of k and n the value returned by MAXLAG is the largest permissible value of m for which the model is not overparameterised. All output quantities in the arrays PARLAG, SE, QQ, X, PVALUE and LOGLHD up to and including lag MAXLAG will be correct.

7 Accuracy

The computations are believed to be stable.

8 Further Comments

The time taken is roughly proportional to nmk.

For each order of autoregressive model that has been estimated, G13DPF returns the maximum of the log-likelihood function. An alternative means of choosing the order of a vector AR process is to choose the order for which Akaike's information criterion is smallest. That is, choose the value of l for which $-2 \times \mathrm{LOGLHD}(l) + 2lk^2$ is smallest. You should be warned that this does not always lead to the same choice of l as indicated by the sample partial autoregression matrices and the likelihood ratio statistics.

9 Example

This example computes the sample partial autoregression matrices of two time series of length 48 up to lag 10.

9.1 Program Text

```
!   G13DPF Example Program Text
!   Mark 24 Release. NAG Copyright 2012.

    Module g13dpfe_mod

!     G13DPF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                   :: nin = 5, nout = 6
    Contains
      Subroutine zprint(k,m,ldpar,maxlag,parlag,se,qq,x,pvalue,nout,ifail)

!       .. Scalar Arguments ..
        Integer, Intent (In)               :: ifail, k, ldpar, m, maxlag, nout
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In)    :: parlag(ldpar,ldpar,m),         &
                                              pvalue(m), qq(ldpar,ldpar,m),  &
                                              se(ldpar,ldpar,m), x(m)
!       .. Local Scalars ..
        Real (Kind=nag_wp)                 :: sum
        Integer                            :: i, i2, j, l
!       .. Local Arrays ..
        Character (6)                      :: st(6)
!       .. Executable Statements ..
!       Display titles
        If (k>1) Then
          Write (nout,99999)
        Else If (k==1) Then
          Write (nout,99998)
        End If

        Do l = 1, maxlag
          Do j = 1, k
            sum = parlag(1,j,l)
            st(j) = '.'
            If (sum>1.96E0_nag_wp*se(1,j,l)) Then
              st(j) = '+'
            End If
            If (sum<-1.96E0_nag_wp*se(1,j,l)) Then
              st(j) = '-'
            End If
          End Do
          If (k==1) Then
            Write (nout,99997) l, (parlag(1,j,l),j=1,k), (st(i2),i2=1,k), &
              qq(1,1,l), x(l), pvalue(l)
            Write (nout,99996)(se(1,j,l),j=1,k)
          Else If (k==2) Then
            Write (nout,99995) l, (parlag(1,j,l),j=1,k), (st(i2),i2=1,k), &
              qq(1,1,l), x(l), pvalue(l)
            Write (nout,99994)(se(1,j,l),j=1,k)
          Else If (k==3) Then
            Write (nout,99993) l, (parlag(1,j,l),j=1,k), (st(i2),i2=1,k), &
              qq(1,1,l), x(l), pvalue(l)
            Write (nout,99992)(se(1,j,l),j=1,k)
          Else If (k==4) Then
            Write (nout,99991) l
            Write (nout,99986)(parlag(1,j,l),j=1,k), (st(i2),i2=1,k), &
```

```
              qq(1,1,l), x(l), pvalue(l)
            Write (nout,99990)(se(1,j,l),j=1,k)
          End If

          Do i = 2, k

            Do j = 1, k
              sum = parlag(i,j,l)
              st(j) = '.'
              If (sum>1.96E0_nag_wp*se(i,j,l)) Then
                st(j) = '+'
              End If
              If (sum<-1.96E0_nag_wp*se(i,j,l)) Then
                st(j) = '-'
              End If
            End Do
            If (k==2) Then
              Write (nout,99989)(parlag(i,j,l),j=1,k), (st(i2),i2=1,k), &
                qq(i,i,l)
              Write (nout,99994)(se(i,j,l),j=1,k)
            Else If (k==3) Then
              Write (nout,99988)(parlag(i,j,l),j=1,k), (st(i2),i2=1,k), &
                qq(i,i,l)
              Write (nout,99992)(se(i,j,l),j=1,k)
            Else If (k==4) Then
              Write (nout,99987)(parlag(i,j,l),j=1,k), (st(i2),i2=1,k), &
                qq(i,i,l)
              Write (nout,99990)(se(i,j,l),j=1,k)
            End If

          End Do
        End Do

        If (ifail==2) Then
          Write (nout,99985) 'Recursive equations broke down at ', maxlag + 1
        End If

        Return

99999   Format (' Partial Autoregression Matrices',4X,'Indicator',2X, &
          'Residual',3X,'Chi-Square',2X,'Pvalue'/37X,'Symbols',3X,'Variances', &
          3X,'Statistic'/' -------------------------------',4X,'---------',2X, &
          '---------',2X,'-----------',1X,'------')
99998   Format (' Partial Autoregression Function',4X,'Indicator',2X, &
          'Residual',3X,'Chi-Square',2X,'Pvalue'/37X,'Symbols',3X,'Variances', &
          3X,'Statistic'/' -------------------------------',4X,'---------',2X, &
          '---------',2X,'-----------',1X,'------')
99997   Format (/' Lag',I3,1X,':',F7.3,22X,A1,F14.3,3X,F10.3,F9.3)
99996   Format (9X,'(',F6.3,')')
99995   Format (/' Lag',I3,1X,':',2F8.3,14X,2A1,F13.3,3X,F10.3,F9.3)
99994   Format (10X,'(',F6.3,')(',F6.3,')')
99993   Format (/' Lag',I3,1X,':',3F8.3,6X,3A1,F12.3,3X,F10.3,F9.3)
99992   Format (10X,'(',F6.3,')(',F6.3,')(',F6.3,')')
99991   Format (/' Lag',I3)
99990   Format (2X,'(',F6.3,')(',F6.3,')(',F6.3,')(',F6.3,')')
99989   Format (9X,2F8.3,14X,2A1,F13.3)
99988   Format (9X,3F8.3,6X,3A1,F12.3)
99987   Format (1X,4F8.3,5X,4A1,F12.3)
99986   Format (1X,4F8.3,5X,4A1,F12.3,3X,F10.3,F9.3)
99985   Format (1X,A,I0)
      End Subroutine zprint
    End Module g13dpfe_mod
    Program g13dpfe

!     G13DPF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: g13dpf, nag_wp
      Use g13dpfe_mod, Only: nin, nout, zprint
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Local Scalars ..
      Integer                              :: i, ifail, k, kmax, l, lwork, m,  &
                                              maxlag, mk, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable      :: loglhd(:), parlag(:,:,:),        &
                                              pvalue(:), qq(:,:,:), se(:,:,:), &
                                              work(:), x(:), z(:,:)
      Integer, Allocatable                 :: iwork(:)
!     .. Executable Statements ..
      Write (nout,*) 'G13DPF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the problem size
      Read (nin,*) k, n, m

      kmax = k
      mk = m*k
      l = mk + 1
      lwork = (k+1)*k + l*(4+k)*2*l**2
      Allocate (z(kmax,n),parlag(kmax,kmax,m),se(kmax,kmax,m),x(m),pvalue(m), &
        loglhd(m),work(lwork),qq(kmax,kmax,m),iwork(mk))

!     Read in series
      Do i = 1, k
        Read (nin,*) z(i,1:n)
      End Do

!     Calculate sample partial autoregression matrices
      ifail = -1
      Call g13dpf(k,n,z,kmax,m,maxlag,parlag,se,qq,x,pvalue,loglhd,work,lwork, &
        iwork,ifail)
      If (ifail/=0) Then
        If (ifail/=2) Then
          Go To 100
        End If
      End If

!     Display results
      Call zprint(k,m,kmax,maxlag,parlag,se,qq,x,pvalue,nout,ifail)

100   Continue

    End Program g13dpfe
```

9.2 Program Data

```
G13DPF Example Program Data
2 48 10                                                         :: K,N,M
-1.490 -1.620  5.200  6.230  6.210  5.860  4.090  3.180
 2.620  1.490  1.170  0.850 -0.350  0.240  2.440  2.580
 2.040  0.400  2.260  3.340  5.090  5.000  4.780  4.110
 3.450  1.650  1.290  4.090  6.320  7.500  3.890  1.580
 5.210  5.250  4.930  7.380  5.870  5.810  9.680  9.070
 7.290  7.840  7.550  7.320  7.970  7.760  7.000  8.350
 7.340  6.350  6.960  8.540  6.620  4.970  4.550  4.810
 4.750  4.760 10.880 10.010 11.620 10.360  6.400  6.240
 7.930  4.040  3.730  5.600  5.350  6.810  8.270  7.680
 6.650  6.080 10.250  9.140 17.750 13.300  9.630  6.800
 4.080  5.060  4.940  6.650  7.940 10.760 11.890  5.850
 9.010  7.500 10.020 10.380  8.150  8.370 10.730 12.140 :: End of Z
```

9.3 Program Results

```
G13DPF Example Program Results

Partial Autoregression Matrices    Indicator  Residual   Chi-Square  Pvalue
                                    Symbols   Variances   Statistic
-------------------------------    ---------  ---------  ----------- ------

Lag  1 :   0.757   0.062              +.        2.731       49.884    0.000
         ( 0.092)( 0.092)
           0.061   0.570              .+        5.440
         ( 0.129)( 0.130)

Lag  2 :  -0.161  -0.135              ..        2.530        3.347    0.502
         ( 0.145)( 0.109)
          -0.093  -0.065              ..        5.486
         ( 0.213)( 0.160)

Lag  3 :   0.237   0.044              ..        1.755       13.962    0.007
         ( 0.128)( 0.095)
           0.047  -0.248              ..        5.291
         ( 0.222)( 0.165)

Lag  4 :  -0.098   0.152              ..        1.661        7.071    0.132
         ( 0.134)( 0.099)
           0.402  -0.194              ..        4.786
         ( 0.228)( 0.168)

Lag  5 :   0.257  -0.026              ..        1.504        5.184    0.269
         ( 0.141)( 0.106)
           0.400  -0.021              ..        4.447
         ( 0.242)( 0.183)

Lag  6 :  -0.075   0.112              ..        1.480        2.083    0.721
         ( 0.156)( 0.111)
           0.196  -0.106              ..        4.425
         ( 0.269)( 0.192)

Lag  7 :  -0.054   0.097              ..        1.478        5.074    0.280
         ( 0.166)( 0.121)
           0.574  -0.080              +.        3.838
         ( 0.267)( 0.195)

Lag  8 :   0.147   0.041              ..        1.415       10.991    0.027
         ( 0.188)( 0.128)
           0.916  -0.242              +.        2.415
         ( 0.246)( 0.167)

Lag  9 :  -0.039   0.099              ..        1.322        3.936    0.415
         ( 0.251)( 0.140)
          -0.500   0.173              ..        2.196
         ( 0.324)( 0.181)

Lag 10 :   0.189   0.131              ..        1.206        3.175    0.529
         ( 0.275)( 0.157)
          -0.183  -0.040              ..        2.201
         ( 0.371)( 0.212)
```