NAG Fortran Library Routine Document

G13DBF

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

G13DBF calculates the multivariate partial autocorrelation function of a multivariate time series.

2 Specification

```
 SUBROUTINE G13DBF (C0, C, NSM, NS, NL, NK, P, V0, V, D, DB, W, WB, NVP,
1                   WA, IWA, IFAIL)
 INTEGER            NSM, NS, NL, NK, NVP, IWA, IFAIL
 double precision   C0(NSM,NS), C(NSM,NSM,NL), P(NK), V0, V(NK),
1                   D(NSM,NSM,NK), DB(NSM,NS), W(NSM,NSM,NK),
2                   WB(NSM,NSM,NK), WA(IWA)
```

3 Description

The input is a set of lagged autocovariance matrices $C_0, C_1, C_2, \ldots, C_m$. These will generally be sample values such as are obtained from a multivariate time series using G13DMF.

The main calculation is the recursive determination of the coefficients in the finite lag (forward) prediction equation

$$x_t = \Phi_{l,1} x_{t-1} + \cdots + \Phi_{l,l} x_{t-l} + e_{l,t}$$

and the associated backward prediction equation

$$x_{t-l-1} = \Psi_{l,1} x_{t-l} + \cdots + \Psi_{l,l} x_{t-1} + f_{l,t}$$

together with the covariance matrices D_l of $e_{l,t}$ and G_l of $f_{l,t}$.

The recursive cycle, by which the order of the prediction equation is extended from l to $l+1$, is to calculate

$$M_{l+1} = C_{l+1}^{\mathrm{T}} - \Phi_{l,1} C_l^{\mathrm{T}} - \cdots - \Phi_{l,l} C_1^{\mathrm{T}} \qquad (1)$$

then $\Phi_{l+1,l+1} = M_{l+1} D_l^{-1}, \quad \Psi_{l+1,l+1} = M_{l+1}^{\mathrm{T}} G_l^{-1}$

from which

$$\Phi_{l+1,j} = \Phi_{l,j} - \Phi_{l+1,l+1} \Psi_{l,l+1-j}, \quad j = 1, 2, \ldots, l \qquad (2)$$

and

$$\Psi_{l+1,j} = \Psi_{l,j} - \Psi_{l+1,l+1} \Phi_{l,l+1-j}, \quad j = 1, 2, \ldots, l. \qquad (3)$$

Finally, $D_{l+1} = D_l - M_{l+1} \Phi_{l+1,l+1}^{\mathrm{T}}$ and $G_{l+1} = G_l - M_{l+1}^{\mathrm{T}} \Psi_{l+1,l+1}^{\mathrm{T}}$.

(Here T denotes the transpose of a matrix.)

The cycle is initialized by taking (for $l = 0$)

$$D_0 = G_0 = C_0.$$

In the step from $l = 0$ to 1, the above equations contain redundant terms and simplify. Thus (1) becomes $M_1 = C_1^{\mathrm{T}}$ and neither (2) or (3) are needed.

Quantities useful in assessing the effectiveness of the prediction equation are generalized variance ratios

$$v_l = \det D_l / \det C_0, \quad l = 1, 2, \ldots$$

and multiple squared partial autocorrelations

$$p_l^2 = 1 - v_l / v_{l-1}.$$

4 References

Akaike H (1971) Autoregressive model fitting for control *Ann. Inst. Statist. Math.* **23** 163–180

Whittle P (1963) On the fitting of multivariate autoregressions and the approximate canonical factorization of a spectral density matrix *Biometrika* **50** 129–134

5 Parameters

1: C0(NSM,NS) – ***double precision*** array *Input*

On entry: contains the zero lag cross-covariances between the NS series as returned by G13DMF. (C0 is assumed to be symmetric, upper triangle only is used.)

2: C(NSM,NSM,NL) – ***double precision*** array *Input*

On entry: contains the cross-covariances at lags 1 to NL. $\mathrm{C}(i,j,k)$ must contain the cross-covariance, c_{ijk}, of series i and series j at lag k. Series j leads series i.

3: NSM – INTEGER *Input*

On entry: the first dimension of the arrays C0, C, D, DB, W and WB and the second dimension of the arrays C, D, W and WB as declared in the (sub)program from which G13DBF is called.

Constraint: $\mathrm{NSM} \geq \max(\mathrm{NS}, 1)$.

4: NS – INTEGER *Input*

On entry: k, the number of time series whose cross-covariances are supplied in C and C0.

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

5: NL – INTEGER *Input*

On entry: m, the maximum lag for which cross-covariances are supplied in C.

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

6: NK – INTEGER *Input*

On entry: the number of lags to which partial auto-correlations are to be calculated.

Constraint: $1 \leq \mathrm{NK} \leq \mathrm{NL}$.

7: P(NK) – ***double precision*** array *Output*

On exit: the multiple squared partial autocorrelations from lags 1 to NVP; that is, $\mathrm{P}(l)$ contains p_l^2, for $l = 1, 2, \ldots, \mathrm{NVP}$. For lags $\mathrm{NVP} + 1$ to NK the elements of P are set to zero.

8: V0 – ***double precision*** *Output*

On exit: the lag zero prediction error variance (equal to the determinant of C0).

9: V(NK) – ***double precision*** array *Output*

On exit: the prediction error variance ratios from lags 1 to NVP; that is, $\mathrm{V}(l)$ contains v_l, for $l = 1, 2, \ldots, \mathrm{NVP}$. For lags $\mathrm{NVP} + 1$ to NK the elements of V are set to zero.

10: D(NSM,NSM,NK) – ***double precision*** array *Output*

On exit: the prediction error variance matrices at lags 1 to NVP.

Element (i,j,k) of D contains the prediction error covariance of series i and series j at lag k, for $k = 1, 2, \ldots, \mathrm{NVP}$. Series j leads series i; that is, the (i,j)th element of D_k. For lags $\mathrm{NVP} + 1$ to NK the elements of D are set to zero.

11: DB(NSM,NS) – ***double precision*** array *Output*

On exit: the backward prediction error variance matrix at lag NVP.

DB(i,j) contains the backward prediction error covariance of series i and series j; that is, the (i,j)th element of the G_k, where $k = \mathrm{NVP}$.

12: W(NSM,NSM,NK) – ***double precision*** array *Output*

On exit: the prediction coefficient matrices at lags 1 to NVP.

W(i,j,l) contains the jth prediction coefficient of series i at lag l; that is, the (i,j)th element of Φ_{kl}, where $k = \mathrm{NVP}$, for $l = 1, 2, \ldots, \mathrm{NVP}$. For lags $\mathrm{NVP} + 1$ to NK the elements of W are set to zero.

13: WB(NSM,NSM,NK) – ***double precision*** array *Output*

On exit: the backward prediction coefficient matrices at lags 1 to NVP.

WB(i,j,l) contains the jth backward prediction coefficient of series i at lag l; that is, the (i,j)th element of Ψ_{kl}, where $k = \mathrm{NVP}$, for $l = 1, 2, \ldots, \mathrm{NVP}$. For lags $\mathrm{NVP} + 1$ to NK the elements of WB are set to zero.

14: NVP – INTEGER *Output*

On exit: the maximum lag, L, for which calculation of P, V, D, DB, W and WB was successful. If the routine completes successfully NVP will equal NK.

15: WA(IWA) – ***double precision*** array *Workspace*
16: IWA – INTEGER *Input*

On entry: the dimension of the array WA as declared in the (sub)program from which G13DBF is called.

Constraint: $\mathrm{IWA} \geq (2 \times \mathrm{NS} + 1) \times \mathrm{NS}$.

17: 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 Chapter P01 for details.

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

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

On entry, $\mathrm{NSM} < 1$,
or $\mathrm{NS} < 1$,
or $\mathrm{NS} > \mathrm{NSM}$,
or $\mathrm{NL} < 1$,

or NK < 1,
or NK > NL,
or IWA < $(2 \times \text{NS} + 1) \times \text{NS}$.

IFAIL = 2

C0 is not positive-definite.

V0, V, P, D, DB, W, WB and NVP are set to zero.

IFAIL = 3

At lag $k = \text{NVP} + 1 \leq \text{NK}$, D_k was found not to be positive-definite. Up to lag NVP, V0, V, P, D, W and WB contain the values calculated so far and from lag NVP + 1 to lag NK the matrices contain zero. DB contains the backward prediction coefficients for lag NVP.

7 Accuracy

The conditioning of the problem depends on the prediction error variance ratios. Very small values of these may indicate loss of accuracy in the computations.

8 Further Comments

The time taken by G13DBF is roughly proportional to $\text{NK}^2 \times \text{NS}^3$.

If sample autocorrelation matrices are used as input, then the output will be relevant to the original series scaled by their standard deviations. If these autocorrelation matrices are produced by G13DMF, you must replace the diagonal elements of C_0 (otherwise used to hold the series variances) by 1.

9 Example

This example reads the autocovariance matrices for four series from lag 0 to 5. It calls G13DBF to calculate the multivariate partial autocorrelation function and other related matrices of statistics up to lag 3. It prints the results.

9.1 Program Text

```
*     G13DBF Example Program Text
*     Mark 14 Revised. NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NSMAX, NSM, NLMAX, NKMAX, IWA
      PARAMETER        (NSMAX=6,NSM=NSMAX,NLMAX=5,NKMAX=NLMAX,
     +                 IWA=(2*NSMAX+1)*NSMAX)
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
*     .. Local Scalars ..
      DOUBLE PRECISION V0
      INTEGER          I, I1, IFAIL, J, J1, K, NK, NL, NS, NVP
*     .. Local Arrays ..
      DOUBLE PRECISION C(NSM,NSM,NLMAX), C0(NSM,NSMAX),
     +                 D(NSM,NSM,NKMAX), DB(NSM,NSMAX), P(NKMAX),
     +                 V(NKMAX), W(NSM,NSM,NKMAX), WA(IWA),
     +                 WB(NSM,NSM,NKMAX)
*     .. External Subroutines ..
      EXTERNAL         G13DBF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'G13DBF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
*     Read series length, and numbers of lags
      READ (NIN,*) NS, NL, NK
      IF (NS.GT.0 .AND. NS.LE.NSMAX .AND. NL.GT.0 .AND. NL.LE.
     +    NLMAX .AND. NK.GT.0 .AND. NK.LE.NKMAX) THEN
*        Read autocovariances
         READ (NIN,*) ((C0(I,J),J=1,NS),I=1,NS)
```

```
         READ (NIN,*) (((C(I,J,K),J=1,NS),I=1,NS),K=1,NL)
*        Call routine to calculate multivariate partial autocorrelation
*        function
         IFAIL = 1
*
         CALL G13DBF(C0,C,NSM,NS,NL,NK,P,V0,V,D,DB,W,WB,NVP,WA,IWA,
     +               IFAIL)
*
         WRITE (NOUT,*)
         IF (IFAIL.NE.0) THEN
            WRITE (NOUT,99999) 'G13DBF fails. IFAIL =', IFAIL
            WRITE (NOUT,*)
         END IF
         IF (IFAIL.EQ.0 .OR. IFAIL.EQ.3) THEN
            WRITE (NOUT,99998) 'Number of valid parameters =', NVP
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Multivariate partial autocorrelations'
            WRITE (NOUT,99997) (P(I1),I1=1,NK)
            WRITE (NOUT,*)
            WRITE (NOUT,*)
     +        'Zero lag predictor error variance determinant'
            WRITE (NOUT,*) 'followed by error variance ratios'
            WRITE (NOUT,99997) V0, (V(I1),I1=1,NK)
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Prediction error variances'
            DO 40 K = 1, NK
               WRITE (NOUT,*)
               WRITE (NOUT,99996) 'Lag =', K
               DO 20 I = 1, NS
                  WRITE (NOUT,99997) (D(I,J1,K),J1=1,NS)
   20          CONTINUE
   40       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Last backward prediction error variances'
            WRITE (NOUT,*)
            WRITE (NOUT,99996) 'Lag =', NVP
            DO 60 I = 1, NS
               WRITE (NOUT,99997) (DB(I,J1),J1=1,NS)
   60       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Prediction coefficients'
            DO 100 K = 1, NK
               WRITE (NOUT,*)
               WRITE (NOUT,99996) 'Lag =', K
               DO 80 I = 1, NS
                  WRITE (NOUT,99997) (W(I,J1,K),J1=1,NS)
   80          CONTINUE
  100       CONTINUE
            WRITE (NOUT,*)
            WRITE (NOUT,*) 'Backward prediction coefficients'
            DO 140 K = 1, NK
               WRITE (NOUT,*)
               WRITE (NOUT,99996) 'Lag =', K
               DO 120 I = 1, NS
                  WRITE (NOUT,99997) (WB(I,J1,K),J1=1,NS)
  120          CONTINUE
  140       CONTINUE
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,I3)
99998 FORMAT (1X,A,I10)
99997 FORMAT (1X,5F12.5)
99996 FORMAT (1X,A,I5)
      END
```

9.2 Program Data

```
G13DBF Example Program Data
     4     5     3   500
  .10900E-01 -.77917E-02  .13004E-02  .12654E-02
 -.77917E-02  .57040E-01  .24180E-02  .14409E-01
  .13004E-02  .24180E-02  .43960E-01 -.21421E-01
  .12654E-02  .14409E-01 -.21421E-01  .72289E-01
  .45889E-02  .46510E-03 -.13275E-03  .77531E-02
 -.24419E-02 -.11667E-01 -.21956E-01 -.45803E-02
  .11080E-02 -.80479E-02  .13621E-01 -.85868E-02
 -.50614E-03  .14045E-01 -.10087E-02  .12269E-01
  .18652E-02 -.64389E-02  .88307E-02 -.24808E-02
 -.11865E-01  .72367E-02 -.19802E-01  .59069E-02
 -.80307E-02  .14306E-01  .14546E-01  .13510E-01
 -.21791E-02 -.29528E-01 -.15887E-01  .88308E-03
 -.80550E-04 -.37759E-02  .75463E-02 -.42276E-02
  .41447E-02 -.37987E-02  .19332E-02 -.17564E-01
 -.10582E-01  .67733E-02  .69832E-02  .61747E-02
  .41352E-02 -.16013E-01  .17043E-01 -.13412E-01
  .76079E-03 -.10134E-02  .11870E-01 -.41651E-02
  .36014E-02 -.36375E-02 -.25571E-01  .50218E-02
 -.13924E-01  .11718E-01 -.59088E-02  .59297E-02
  .10739E-01 -.14571E-01  .13816E-01 -.12588E-01
 -.64365E-03 -.44556E-02  .51334E-02  .71587E-03
  .63617E-02  .15217E-03  .27270E-02 -.22261E-02
 -.85855E-02  .14468E-02 -.28698E-02  .44384E-02
  .68339E-02 -.21790E-02  .13759E-01  .28217E-03
```

9.3 Program Results

```
 G13DBF Example Program Results

 Number of valid parameters =          3

 Multivariate partial autocorrelations
      0.64498     0.92669     0.84300

 Zero lag predictor error variance determinant
 followed by error variance ratios
      0.00000     0.35502     0.02603     0.00409

 Prediction error variances

 Lag =    1
      0.00811    -0.00511     0.00159    -0.00029
     -0.00511     0.04089     0.00757     0.01843
      0.00159     0.00757     0.03834    -0.01894
     -0.00029     0.01843    -0.01894     0.06760

 Lag =    2
      0.00354    -0.00087    -0.00075    -0.00105
     -0.00087     0.01946     0.00535     0.00566
     -0.00075     0.00535     0.01900    -0.01071
     -0.00105     0.00566    -0.01071     0.04058

 Lag =    3
      0.00301    -0.00087    -0.00054     0.00065
     -0.00087     0.01824     0.00872     0.00247
     -0.00054     0.00872     0.00935    -0.00216
      0.00065     0.00247    -0.00216     0.02254

 Last backward prediction error variances

 Lag =    3
      0.00331    -0.00392    -0.00106     0.00592
     -0.00392     0.01890     0.00348    -0.00330
     -0.00106     0.00348     0.01003    -0.01054
      0.00592    -0.00330    -0.01054     0.03336
```

```
Prediction coefficients

Lag =    1
     0.81861     0.23399    -0.17097     0.09256
     0.06738    -0.48720    -0.14064     0.04295
     0.15036     0.11924    -0.36725    -0.42092
    -0.70971     0.02998     0.59779     0.34610

Lag =    2
    -0.34049    -0.13370     0.40610    -0.02183
    -1.27574    -0.13591    -0.65779    -0.11267
    -0.45439     0.19379     0.63420     0.33920
    -0.43237    -0.54848    -0.62897     0.16670

Lag =    3
     0.16437     0.13858     0.01290     0.03463
     0.39291     0.07407    -0.08802    -0.15361
    -1.29240    -0.24489     0.30235     0.39442
     0.89768    -0.39040     0.25151    -0.28304

Backward prediction coefficients

Lag =    1
     0.41541     0.06149     0.15319     0.05079
     0.12370    -0.26471    -0.22721     0.48503
    -0.86933    -0.47373     0.37924     0.13814
     1.30779    -0.09178    -1.45398    -0.21967

Lag =    2
    -0.06740    -0.12255    -0.13673    -0.09730
    -1.24801     0.03090     0.51706    -0.28925
     0.98045    -0.20194     0.16307    -0.10869
    -1.68389    -0.74589     0.52900     0.41580

Lag =    3
     0.03794     0.10491    -0.21635     0.08015
     0.75392     0.22603    -0.25661    -0.47450
    -0.00338     0.05636    -0.08818     0.12723
     0.55022    -0.41232     0.71649    -0.14565
```