

# NAG Fortran Library Routine Document

## G13DJF

**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

G13DJF computes forecasts of a multivariate time series. It is assumed that a vector ARMA model has already been fitted to the appropriately differenced/transformed time series using G13DCF. The standard deviations of the forecast errors are also returned. A reference vector is set up so that should future series values become available the forecasts and their standard errors may be updated by calling G13DKF.

### 2 Specification

```

SUBROUTINE G13DJF(K, N, Z, IK, TR, ID, DELTA, IP, IQ, MEAN, PAR, LPAR,
1                QQ, V, LMAX, PREDZ, SEFZ, REF, LREF, WORK, LWORK,
2                IWORK, LIWORK, IFAIL)
    INTEGER      K, N, IK, ID(K), IP, IQ, LPAR, LMAX, LREF, LWORK,
1                IWORK(LIWORK), LIWORK, IFAIL
    real         Z(IK,N), DELTA(IK,*), PAR(LPAR), QQ(IK,K), V(IK,*),
1                PREDZ(IK,LMAX), SEFZ(IK,LMAX), REF(LREF), WORK(LWORK)
    CHARACTER*1  TR(K), MEAN

```

### 3 Description

Let the vector  $Z_t = (z_{1t}, z_{2t}, \dots, z_{kt})^T$ , for  $t = 1, 2, \dots, n$ , denote a  $k$ -dimensional time series for which forecasts of  $Z_{n+1}, Z_{n+2}, \dots, Z_{n+l_{\max}}$  are required. Let  $W_t = (w_{1t}, w_{2t}, \dots, w_{kt})^T$  be defined as follows:

$$w_{it} = \delta_i(B)z_{it}^*, \quad i = 1, 2, \dots, k,$$

where  $\delta_i(B)$  is the differencing operator applied to the  $i$ th series and where  $z_{it}^*$  is equal to either  $z_{it}$ ,  $\sqrt{z_{it}}$  or  $\log_e z_{it}$  depending on whether or not a transformation was required to stabilize the variance before fitting the model.

If the order of differencing required for the  $i$ th series is  $d_i$ , then the differencing operator for the  $i$ th series is defined by  $\delta_i(B) = 1 - \delta_{i1}B - \delta_{i2}B^2 - \dots - \delta_{id_i}B^{d_i}$  where  $B$  is the backward shift operator; that is,  $BZ_t = Z_{t-1}$ . The differencing parameters  $\delta_{ij}$  for  $i = 1, 2, \dots, k$ ;  $j = 1, 2, \dots, d_i$  must be supplied by the user. If the  $i$ th series does not require differencing, then  $d_i = 0$ .

$W_t$  is assumed to follow a multivariate ARMA model of the form:

$$W_t - \mu = \phi_1(W_{t-1} - \mu) + \phi_2(W_{t-2} - \mu) + \dots + \phi_p(W_{t-p} - \mu) + \epsilon_t - \theta_1\epsilon_{t-1} - \dots - \theta_q\epsilon_{t-q}, \quad (1)$$

where  $\epsilon_t = (\epsilon_{1t}, \epsilon_{2t}, \dots, \epsilon_{kt})^T$ , for  $t = 1, 2, \dots, n$ , is a vector of  $k$  residual series assumed to be Normally distributed with zero mean and positive-definite covariance matrix  $\Sigma$ . The components of  $\epsilon_t$  are assumed to be uncorrelated at non-simultaneous lags. The  $\phi_i$  and  $\theta_j$  are  $k$  by  $k$  matrices of parameters. The matrices  $\phi_i$ , for  $i = 1, 2, \dots, p$ , are the autoregressive (AR) parameter matrices, and the matrices  $\theta_i$ , for  $i = 1, 2, \dots, q$ , the moving average (MA) parameter matrices. The parameters in the model are thus the  $p$  ( $k$  by  $k$ ) $\phi$ -matrices, the  $q$  ( $k$  by  $k$ ) $\theta$ -matrices, the mean vector  $\mu$  and the residual error covariance matrix  $\Sigma$ . The ARMA model (1) must be both stationary and invertible; see G13DXF for a method of checking these conditions.

The ARMA model (1) may be rewritten as

$$\phi(B)(\delta(B)Z_t^* - \mu) = \theta(B)\epsilon_t,$$

where  $\phi(B)$  and  $\theta(B)$  are the autoregressive and moving average polynomials and  $\delta(B)$  denotes the  $k$  by  $k$  diagonal matrix whose  $i$ th diagonal elements is  $\delta_i(B)$  and  $Z_t^* = (z_{1t}^*, z_{2t}^*, \dots, z_{kt}^*)^T$ .

This may be rewritten as

$$\phi(B)\delta(B)Z_t^* = \phi(B)\mu + \theta(B)\epsilon_t$$

or

$$Z_t^* = \tau + \psi(B)\epsilon_t = \tau + \epsilon_t + \psi_1\epsilon_{t-1} + \psi_2\epsilon_{t-2} + \dots$$

where  $\psi(B) = \delta^{-1}(B)\phi^{-1}(B)\theta(B)$  and  $\tau = \delta^{-1}(B)\mu$  is a vector of length  $k$ .

Forecasts are computed using a multivariate version of the procedure described in Box and Jenkins (1976). If  $\hat{Z}_n^*(l)$  denotes the forecast of  $Z_{n+l}^*$ , then  $\hat{Z}_n^*(l)$  is taken to be that linear function of  $Z_n^*, Z_{n-1}^*, \dots$  which minimises the elements of  $E\{e_n(l)e_n'(l)\}$  where  $e_n(l) = Z_{n+l}^* - \hat{Z}_n^*(l)$  is the forecast error.  $\hat{Z}_n^*(l)$  is referred to as the linear minimum mean square error forecast of  $Z_{n+l}^*$ .

The linear predictor which minimises the mean square error may be expressed as

$$\hat{Z}_n^*(l) = \tau + \psi_l\epsilon_n + \psi_{l+1}\epsilon_{n-1} + \psi_{l+2}\epsilon_{n-2} + \dots$$

The forecast error at  $t$  for lead  $l$  is then

$$e_n(l) = Z_{n+l}^* - \hat{Z}_n^*(l) = \epsilon_{n+l} + \psi_1\epsilon_{n+l-1} + \psi_2\epsilon_{n+l-2} + \dots + \psi_{l-1}\epsilon_{n+1}.$$

Let  $d = \max(d_i)$ , for  $i = 1, 2, \dots, k$ . Unless  $q = 0$  the routine requires estimates of  $\epsilon_t$ , for  $t = d+1, d+2, \dots, n$ , which are obtainable from G13DCF. The terms  $\epsilon_t$  are assumed to be zero, for  $t = n+1, n+2, \dots, n+l_{\max}$ . The user may use G13DKF to update these  $l_{\max}$  forecasts should further observations,  $Z_{n+1}, Z_{n+2}, \dots$ , become available. Note that when  $l_{\max}$  or more further observations are available then G13DJF must be used to produce new forecasts for  $Z_{n+l_{\max}+1}^*, Z_{n+l_{\max}+2}^*, \dots$ , should they be required.

When a transformation has been used the forecasts and their standard errors are suitably modified to give results in terms of the original series,  $Z_t$ ; see Granger and Newbold (1976).

## 4 References

Box G E P and Jenkins G M (1976) *Time Series Analysis: Forecasting and Control* (Revised Edition) Holden-Day

Granger C W J and Newbold P (1976) Forecasting transformed series *J. Roy. Statist. Soc. Ser. B* **38** 189–203

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

## 5 Parameters

The output quantities, K, N, IK, IP, IQ, PAR, NPAR, QQ and V from G13DCF are suitable for input to G13DJF.

1: K – INTEGER Input  
*On entry:* the dimension,  $k$ , of the multivariate time series.  
*Constraint:*  $K \geq 1$ .

2: N – INTEGER Input  
*On entry:* the number,  $n$ , of observations in the series,  $Z_t$ , prior to differencing.  
*Constraint:*  $N \geq 3$ .

The total number of observations must exceed the total number of parameters in the model; that is

if MEAN = 'Z', then  $N \times K > (IP + IQ) \times K \times K + K \times (K + 1)/2$ ;

if MEAN = 'M', then  $N \times K > (IP + IQ) \times K \times K + K \times (K + 1)/2$ ;

(see the parameters IP, IQ and MEAN).

- 3:  $Z(IK, N)$  – **real** array *Input*  
*On entry:*  $Z(i, t)$  must be set equal to the  $i$ th component of  $Z_t$ , for  $i = 1, 2, \dots, k$ ;  $t = 1, 2, \dots, n$ .  
*Constraints:*  
 if  $TR(i) = 'L'$ , then  $Z(i, t) > 0.0$  and  
 if  $TR(i) = 'S'$ , then  $Z(i, t) \geq 0.0$ , for  $i = 1, 2, \dots, k$ ;  $t = 1, 2, \dots, n$ .
- 4:  $IK$  – INTEGER *Input*  
*On entry:* the first dimension of the arrays  $Z$ ,  $QQ$ ,  $DELTA$ ,  $V$ ,  $PREDZ$  and  $SEFZ$  as declared in the (sub)program from which G13DJF is called.  
*Constraint:*  $IK \geq K$ .
- 5:  $TR(K)$  – CHARACTER\*1 array *Input*  
*On entry:*  $TR(i)$  indicates whether the  $i$ th time series is to be transformed, for  $i = 1, 2, \dots, k$ .  
 If  $TR(i) = 'N'$ , then no transformation is used;  
 If  $TR(i) = 'L'$ , then a log transformation is used;  
 If  $TR(i) = 'S'$ , then a square root transformation is used.  
*Constraint:*  $TR(i)$  must equal either 'N', 'L' or 'S', for  $i = 1, 2, \dots, k$ .
- 6:  $ID(K)$  – INTEGER array *Input*  
*On entry:*  $ID(i)$  must specify,  $d_i$ , the order of differencing required for the  $i$ th series.  
*Constraint:*  $0 \leq ID(i) < N - \max(IP, IQ)$ , for  $i = 1, 2, \dots, k$ .
- 7:  $DELTA(IK, *)$  – **real** array *Input*  
**Note:** the second dimension of the array  $DELTA$  must be at least  $\max(1, d)$ , where  $d = \max(ID(i))$ .  
*On entry:* if  $ID(i) > 0$ , then  $DELTA(i, j)$  must be set equal to  $\delta_{ij}$ , for  $j = 1, 2, \dots, d_i$ ;  $i = 1, 2, \dots, k$ . If  $d = 0$ , then  $DELTA$  is not referenced.
- 8:  $IP$  – INTEGER *Input*  
*On entry:* the number,  $p$ , of AR parameter matrices.  
*Constraint:*  $IP \geq 0$ .
- 9:  $IQ$  – INTEGER *Input*  
*On entry:* the number,  $q$ , of MA parameter matrices.  
*Constraint:*  $IQ \geq 0$ .
- 10:  $MEAN$  – CHARACTER\*1 *Input*  
*On entry:*  $MEAN$  must be set to 'M' if components of  $\mu$  have been estimated and 'Z' if all elements of  $\mu$  are to be taken as zero.  
*Constraint:*  $MEAN = 'M'$  or 'Z'.
- 11:  $PAR(LPAR)$  – **real** array *Input*  
*On entry:*  $PAR$  must contain the parameter estimates read in row by row in the order  $\phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q, \mu$ .  
 Thus, if  $IP > 0$ , then  $PAR((l-1) \times k \times k + (i-1) \times k + j)$  must be set equal to an estimate of the  $(i, j)$ th element of  $\phi_l$ , for  $l = 1, 2, \dots, p$ ;  $i = 1, 2, \dots, k$ ;  $j = 1, 2, \dots, k$ .  
 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 an estimate of the  $(i, j)$ th element of  $\theta_l$ , for  $l = 1, 2, \dots, q$ ;  $i = 1, 2, \dots, k$ ;  $j = 1, 2, \dots, k$ .

If MEAN has been set equal to 'M', then  $\text{PAR}((p+q) \times k \times k + i)$  must be set equal to an estimate of the  $i$ th component of  $\mu$ , for  $i = 1, 2, \dots, k$ .

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

12: LPAR – INTEGER *Input*

*On entry:* the dimension of the array PAR as declared in the (sub)program from which G13DJF is called.

*Constraints:*

if MEAN = 'Z', then  $\text{LPAR} \geq \max(1, (\text{IP} + \text{IQ}) \times \text{K} \times \text{K})$ ;  
if MEAN = 'M', then  $\text{LPAR} \geq (\text{IP} + \text{IQ}) \times \text{K} \times \text{K} + \text{K}$ .

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

*On entry:*  $\text{QQ}(i, j)$  must contain an estimate of the  $(i, j)$ th element of  $\Sigma$ . The lower triangle only is needed.

*Constraint:* QQ must be positive-definite.

*On exit:* if  $\text{IFAIL} \neq 1$ , then the upper triangle is set equal to the lower triangle.

14: V(IK,\*) – *real* array *Input*

**Note:** the second dimension of the array V must be at least  $\max(1, \text{N} - \max(\text{ID}(i)))$ .

*On entry:*  $\text{V}(i, t)$  must contain an estimate of the  $i$ th component of  $\epsilon_{t+d}$ , for  $i = 1, 2, \dots, k$ ;  $t = 1, 2, \dots, n - d$ , where  $d$  is the maximum value of the elements of the array ID. If  $q = 0$ , then V is not used.

15: LMAX – INTEGER *Input*

*On entry:* the number,  $l_{\max}$ , of forecasts required.

*Constraint:*  $\text{LMAX} \geq 1$ .

16: PREDZ(IK,LMAX) – *real* array *Output*

*On exit:*  $\text{PREDZ}(i, l)$  contains the forecast of  $z_{i,n+l}$ , for  $i = 1, 2, \dots, k$ ;  $l = 1, 2, \dots, l_{\max}$ .

17: SEFZ(IK,LMAX) – *real* array *Output*

*On exit:*  $\text{SEFZ}(i, l)$  contains an estimate of the standard error of the forecast of  $z_{i,n+l}$ , for  $i = 1, 2, \dots, k$ ;  $l = 1, 2, \dots, l_{\max}$ .

18: REF(LREF) – *real* array *Output*

*On exit:* the reference vector which may be used to update forecasts using G13DKF. The first  $(\text{LMAX} - 1) \times \text{K} \times \text{K}$  elements contain the  $\psi$  weight matrices,  $\psi_1, \psi_2, \dots, \psi_{l_{\max}-1}$ , stored column-wise. The next  $\text{K} \times \text{LMAX}$  elements contain the forecasts of the transformed series  $\hat{Z}_{n+1}^*, \hat{Z}_{n+2}^*, \dots, \hat{Z}_{n+l_{\max}}^*$  and the next  $\text{K} \times \text{LMAX}$  contain the variances of the forecasts of the transformed variables. The last K elements are used to store the transformations for the series.

19: LREF – INTEGER *Input*

*On entry:* the dimension of the array REF as declared in the (sub)program from which G13DJF is called.

*Constraint:*  $\text{LREF} \geq (\text{LMAX} - 1) \times \text{K} \times \text{K} + 2 \times \text{K} \times \text{LMAX} + \text{K}$ .

20: WORK(LWORK) – *real* array Workspace  
 21: LWORK – INTEGER Input

*On entry:* the dimension of the array WORK as declared in the (sub)program from which G13DJF is called.

*Constraint:* if  $r = \max(\text{IP}, \text{IQ})$  and  $d = \max(\text{ID}(i))$ , for  $i = 1, 2, \dots, k$ , then  $\text{LWORK} \geq \max\{Kr(Kr + 2), (\text{IP} + d + 2)K^2 + (\text{N} + \text{LMAX})K\}$ .

22: IWORK(LIWORK) – INTEGER array Workspace  
 23: LIWORK – INTEGER Input

*On entry:* the dimension of the array IWORK as declared in the (sub)program from which G13DJF is called.

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

24: 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,  $K < 1$ ,  
 or  $N < 3$ ,  
 or  $\text{IK} < K$ ,  
 or  $\text{ID}(i) < 0$  for some  $i = 1, 2, \dots, k$ ,  
 or  $\text{ID}(i) \geq N - \max(\text{IP}, \text{IQ})$  for some  $i = 1, 2, \dots, k$ ,  
 or  $\text{IP} < 0$ ,  
 or  $\text{IQ} < 0$ ,  
 or  $\text{MEAN} \neq \text{'M'}$  or  $\text{'Z'}$ ,  
 or  $\text{LPAR} < (\text{IP} + \text{IQ}) \times K \times K + K$ , and  $\text{MEAN} = \text{'M'}$ ,  
 or  $\text{LPAR} < (\text{IP} + \text{IQ}) \times K \times K$  and  $\text{MEAN} = \text{'Z'}$ ,  
 or  $N \times K \leq (\text{IP} + \text{IQ}) \times K \times K + K + K(K + 1)/2$ , and  $\text{MEAN} = \text{'M'}$ ,  
 or  $N \times K \leq (\text{IP} + \text{IQ}) \times K \times K + K(K + 1)/2$  and  $\text{MEAN} = \text{'Z'}$ ,  
 or  $\text{LMAX} < 1$ ,  
 or  $\text{LREF} < (\text{LMAX} - 1) \times K \times K + 2 \times K \times \text{LMAX} + K$ ,  
 or LWORK is too small,  
 or LIWORK is too small.

IFAIL = 2

On entry, at least one of the first  $k$  elements of TR is not equal to 'N', 'L' or 'S'.

IFAIL = 3

On entry, one or more of the transformations requested cannot be computed; that is, the user may be trying to log or square-root a series, some of whose values are negative.

IFAIL = 4

On entry, either QQ is not positive-definite or the autoregressive parameter matrices are extremely close to or outside the stationarity region, or the moving average parameter matrices are extremely close to or outside the invertibility region. To proceed, the user must supply different parameter estimates in the arrays PAR and QQ.

IFAIL = 5

This is an unlikely exit brought about by an excessive number of iterations being needed to evaluate the eigenvalues of the matrices required to check for stationarity and invertibility; see G13DXF. All output parameters are undefined.

IFAIL = 6

This is an unlikely exit which could occur if QQ is nearly non positive-definite. In this case the standard deviations of the forecast errors may be non-positive. To proceed, the user must supply different parameter estimates in the array QQ.

IFAIL = 7

This is an unlikely exit. For one of the series, overflow will occur if the forecasts are computed. The user should check whether the transformations requested in the array TR are sensible. All output parameters are undefined.

## 7 Accuracy

The matrix computations are believed to be stable.

## 8 Further Comments

The same differencing operator does not have to be applied to all the series. For example, suppose we have  $k = 2$ , and wish to apply the second order differencing operator  $\nabla^2$  to the first series and the first-order differencing operator  $\nabla$  to the second series:

$$\begin{aligned} w_{1t} &= \nabla^2 z_{1t} = (1 - B)^2 z_{1t} = (1 - 2B + B^2)Z_{1t}, \quad \text{and} \\ w_{2t} &= \nabla z_{2t} = (1 - B)z_{2t}. \end{aligned}$$

Then  $d_1 = 2, d_2 = 1, d = \max(d_1, d_2) = 2$ , and

$$\text{DELTA} = \begin{bmatrix} \delta_{11} & \delta_{12} \\ \delta_{21} & \delta_{22} \end{bmatrix} = \begin{bmatrix} 2 & -1 \\ 1 & 0 \end{bmatrix}.$$

**Note:** although differencing may already have been applied prior to the model fitting stage, the differencing parameters supplied in DELTA are part of the model definition and are still required by this routine to produce the forecasts.

This routine should not be used when the moving average parameters lie close to the boundary of the invertibility region. The routine does test for both invertibility and stationarity but if in doubt, the user may use G13DXF, before calling this routine, to check that the VARMA model being used is invertible.

On a successful exit, the output quantities K, LMAX, IK, REF and LREF will be suitable for input to G13DKF.

## 9 Example

A program to compute forecasts of the next five values in two series each of length 48. No transformation is to be used and no differencing is to be applied to either of the series. G13DCF is first called to fit an AR(1) model to the series. The mean vector  $\mu$  is to be estimated and  $\phi_1(2, 1)$  constrained to be zero.

## 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.

```
*      G13DJF Example Program Text
*      Mark 15 Release. NAG Copyright 1991.
*      .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          KMAX, IK, IPMAX, IQMAX, NMAX, NPARMX, LWORK,
+      LIWORK, ICM, MMAX, IDMAXL, LREF
      PARAMETER        (KMAX=3,IK=KMAX,IPMAX=3,IQMAX=3,NMAX=100,
+      NPARMX=(IPMAX+IQMAX)*KMAX*KMAX+KMAX,LWORK=2000,
+      LIWORK=100,ICM=NPARMX,MMAX=10,IDMAXL=2,
+      LREF=MMAX*KMAX*(KMAX+2))
*      .. Local Scalars ..
      real             CGETOL, RLOGL
      INTEGER          I, I2, IDMAX, IDMIN, IFAIL, IP, IPRINT, IQ,
+      ISHOW, J, K, L, L2, LMAX, LOOP, MAXCAL, N, ND,
+      NITER, NPAR
      LOGICAL          EXACT, MEANL
      CHARACTER        MEAN
*      .. Local Arrays ..
      real             CM(ICM,NPARMX), DELTA(IK,IDMAXL), G(NPARMX),
+      PAR(NPARMX), PREDZ(IK,MMAX), QQ(IK,KMAX),
+      REF(LREF), SEFZ(IK,MMAX), V(IK,NMAX), W(IK,NMAX),
+      WORK(LWORK), Z(IK,NMAX)
      INTEGER          ID(KMAX), IWORK(LIWORK)
      LOGICAL          PARHLD(NPARMX)
      CHARACTER        TR(KMAX)
*      .. External Subroutines ..
      EXTERNAL          G13DCF, G13DJF, G13DLF
*      .. Intrinsic Functions ..
      INTRINSIC          MAX, MIN, MOD
*      .. Executable Statements ..
      WRITE (NOUT,*) 'G13DJF Example Program Results'
*      Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) K, N, IP, IQ, MEAN, LMAX
      NPAR = (IP+IQ)*K*K
      MEANL = .FALSE.
      IF (MEAN.EQ.'M' .OR. MEAN.EQ.'m') THEN
         NPAR = NPAR + K
         MEANL = .TRUE.
      END IF
      IF (K.GT.0 .AND. K.LE.KMAX .AND. N.GE.1 .AND. N.LE.NMAX .AND.
+      NPAR.GE.1 .AND. NPAR.LE.NPARMX .AND. LMAX.GE.1 .AND. LMAX.LE.
+      MMAX) THEN
         READ (NIN,*) (ID(I),I=1,K)
         IDMIN = 0
         IDMAX = 0
         DO 20 I = 1, K
            IDMIN = MIN(ID(I),IDMIN)
            IDMAX = MAX(ID(I),IDMAX)
20      CONTINUE
*
         IF (IDMIN.GE.0 .AND. IDMAX.LE.IDMAXL) THEN
            DO 40 I = 1, K
               READ (NIN,*) (Z(I,J),J=1,N)
40      CONTINUE
            READ (NIN,*) (TR(I),I=1,K)
*
            IF (IDMAX.GT.0) THEN
               DO 60 I = 1, K
                  READ (NIN,*) (DELTA(I,J),J=1,ID(I))
60      CONTINUE
            END IF
            IFAIL = 0
*

```

```

      CALL G13DLF(K,N,Z,IK,TR,ID,DELTA,W,ND,WORK,IFAIL)
*
      DO 80 I = 1, NPAR
        PAR(I) = 0.0e0
        PARHLD(I) = .FALSE.
80      CONTINUE
      DO 120 I = 1, K
        DO 100 J = 1, I
          QQ(I,J) = 0.0e0
100      CONTINUE
120      CONTINUE
      PARHLD(3) = .TRUE.
      EXACT = .TRUE.
*      ** Set IPRINT .lt. 0 for no monitoring
      IPRINT = -1
      CGETOL = 0.0001e0
      MAXCAL = 40*NPAR*(NPAR+5)
*      ** Set ISHOW = 0 for no printing of results from G13DCF
      ISHOW = 0
      IFAIL = 1
*
      CALL G13DCF(K,ND,IP,IQ,MEANL,PAR,NPAR,QQ,IK,W,PARHLD,EXACT,
+          IPRINT,CGETOL,MAXCAL,ISHOW,NITER,RLOGL,V,G,CM,
+          ICM,WORK,LWORK,IWORK,LIWORK,IFAIL)
*
      IF (IFAIL.EQ.0 .OR. IFAIL.GE.4) THEN
        IFAIL = 0
*
      CALL G13DJF(K,N,Z,IK,TR,ID,DELTA,IP,IQ,MEAN,PAR,NPAR,QQ,
+          V,LMAX,PREDZ,SEFZ,REF,LREF,WORK,LWORK,IWORK,
+          LIWORK,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,*) ' FORECAST SUMMARY TABLE '
      WRITE (NOUT,*) ' -----'
      WRITE (NOUT,*)
      WRITE (NOUT,99998) ' Forecast origin is set at t = ', N
      WRITE (NOUT,*)
      LOOP = LMAX/5
      IF (MOD(LMAX,5).NE.0) LOOP = LOOP + 1
      DO 160 J = 1, LOOP
        I2 = (J-1)*5
        L2 = MIN(I2+5,LMAX)
        WRITE (NOUT,99997) 'Lead Time ', (I,I=I2+1,L2)
        WRITE (NOUT,*)
        I = 1
        WRITE (NOUT,99996) 'Series ', I, ' : Forecast ',
+          (PREDZ(1,L),L=I2+1,L2)
        WRITE (NOUT,99995) ' : Standard Error ',
+          (SEFZ(1,L),L=I2+1,L2)
        DO 140 I = 2, K
          WRITE (NOUT,99996) 'Series ', I, ' : Forecast ',
+          (PREDZ(I,L),L=I2+1,L2)
          WRITE (NOUT,99995) ' : Standard Error ',
+          (SEFZ(I,L),L=I2+1,L2)
140      CONTINUE
        WRITE (NOUT,*)
160      CONTINUE
      END IF
    END IF
  END IF
  STOP
*
99998 FORMAT (1X,A,I4)
99997 FORMAT (1X,A,12X,5I10)
99996 FORMAT (1X,A,I2,A,5F10.2)
99995 FORMAT (10X,A,4(F7.2,3X),F7.2)
END

```



## 9.2 Program Data

G13DJF Example Program Data

```

2 48 1 0 'M' 5 : K, N, IP, IQ, MEAN, M
0 0 : ID(I), I=1, K
-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 time series
'N' 'N' : TR(1), TR(2)

```

## 9.3 Program Results

G13DJF Example Program Results

### FORECAST SUMMARY TABLE

-----

Forecast origin is set at t = 48

Lead Time		1	2	3	4	5
Series 1 : Forecast		7.82	7.28	6.77	6.33	5.95
: Standard Error		1.72	2.23	2.51	2.68	2.79
Series 2 : Forecast		10.31	9.25	8.65	8.30	8.10
: Standard Error		2.32	2.68	2.78	2.82	2.83

---