NAG 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 G13DDF. 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, KMAX, TR, ID, DELTA, IP, IQ, MEAN, PAR,         &
                   LPAR, QQ, V, LMAX, PREDZ, SEFZ, REF, LREF, WORK,         &
                   LWORK, IWORK, LIWORK, IFAIL)
INTEGER            K, N, KMAX, ID(K), IP, IQ, LPAR, LMAX, LREF, LWORK,      &
                   IWORK(LIWORK), LIWORK, IFAIL
REAL (KIND=nag_wp) Z(KMAX,N), DELTA(KMAX,*), PAR(LPAR), QQ(KMAX,K),         &
                   V(KMAX,*), PREDZ(KMAX,LMAX), SEFZ(KMAX,LMAX),            &
                   REF(LREF), WORK(LWORK)
CHARACTER(1)       TR(K), MEAN
```

3 Description

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

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

where $\delta_i(B)$ is the differencing operator applied to the ith 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 ith series is d_i, then the differencing operator for the ith series is defined by $\delta_i(B) = 1 - \delta_{i1}B - \delta_{i2}B^2 - \cdots - \delta_{id_i}B^{d_i}$ where B is the backward shift operator; that is, $BZ_t = Z_{t-1}$. The differencing parameters δ_{ij}, for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, d_i$, must be supplied by you. If the ith series does not require differencing, then $d_i = 0$.

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

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

where $\epsilon_t = (\epsilon_{1t}, \epsilon_{2t}, \ldots, \epsilon_{kt})^{\mathrm{T}}$, for $t = 1, 2, \ldots, n$, 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 and θ_j are k by k matrices of parameters. The matrices ϕ_i, for $i = 1, 2, \ldots, p$, are the autoregressive (AR) parameter matrices, and the matrices θ_i, for $i = 1, 2, \ldots, q$, 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. 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

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

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

This may be rewritten as

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

or

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

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 minimizes the elements of $E\left\{e_n(l)e_n'(l)\right\}$ 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 minimizes the mean square error may be expressed as

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

The forecast error at t for lead l is then

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

Let $d = \max(d_i)$, for $i = 1, 2, \dots, k$. Unless $q = 0$ the routine requires estimates of ϵ_t, for $t = d+1, \dots, n$, which are obtainable from G13DDF. The terms ϵ_t are assumed to be zero, for $t = n+1, \dots, n+l_{\mathrm{max}}$. You 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 $Z_{n+l_{\mathrm{max}}+1}, Z_{n+l_{\mathrm{max}}+2}, \dots$, 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 quantities K, N, KMAX, IP, IQ, PAR, NPAR, QQ and V from G13DDF are suitable for input to G13DJF.

1: K – INTEGER *Input*

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

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

2: N – INTEGER *Input*

On entry: n, the number of observations in the series, Z_t, prior to differencing.

Constraint: $\mathrm{N} \geq 3$.

The total number of observations must exceed the total number of parameters in the model; that is if MEAN = 'Z', $\mathrm{N} \times \mathrm{K} > (\mathrm{IP} + \mathrm{IQ}) \times \mathrm{K} \times \mathrm{K} + \mathrm{K} \times (\mathrm{K} + 1)/2$;

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

(see the parameters IP, IQ and MEAN).

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

On entry: $\mathrm{Z}(i,t)$ must contain, z_{it}, the ith component of Z_t, for $i = 1, 2, \ldots, k$ and $t = 1, 2, \ldots, n$.

Constraints:

if $\mathrm{TR}(i)$ = 'L', $\mathrm{Z}(i,t) > 0.0$;
if $\mathrm{TR}(i)$ = 'S', $\mathrm{Z}(i,t) \geq 0.0$, 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, DELTA, QQ, V, PREDZ and SEFZ as declared in the (sub)program from which G13DJF is called.

Constraint: KMAX $\geq$ K.

5: TR(K) – CHARACTER(1) array *Input*

On entry: $\mathrm{TR}(i)$ indicates whether the ith time series is to be transformed, for $i = 1, 2, \ldots, k$.

$\mathrm{TR}(i)$ = 'N'
No transformation is used.

$\mathrm{TR}(i)$ = 'L'
A log transformation is used.

$\mathrm{TR}(i)$ = 'S'
A square root transformation is used.

Constraint: $\mathrm{TR}(i)$ = 'N', 'L' or 'S', for $i = 1, 2, \ldots, k$.

6: ID(K) – INTEGER array *Input*

On entry: $\mathrm{ID}(i)$ must specify, d_i, the order of differencing required for the ith series.

Constraint: $0 \leq \mathrm{ID}(i) < \mathrm{N} - \max(\mathrm{IP}, \mathrm{IQ})$, for $i = 1, 2, \ldots, k$.

7: DELTA(KMAX, $*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array DELTA must be at least $\max(1, d)$, where $d = \max(\mathrm{ID}(i))$.

On entry: if $\mathrm{ID}(i) > 0$, then $\mathrm{DELTA}(i,j)$ must be set equal to δ_{ij}, for $j = 1, 2, \ldots, d_i$ and $i = 1, 2, \ldots, k$.

If $d = 0$, DELTA is not referenced.

8: IP – INTEGER *Input*

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

Constraint: IP ≥ 0.

9: IQ – INTEGER *Input*

On entry: q, the number of MA parameter matrices.

Constraint: IQ ≥ 0.

10: MEAN – CHARACTER(1) *Input*

On entry: MEAN = 'M', if components of μ have been estimated and MEAN = 'Z', if all elements of μ are to be taken as zero.

Constraint: MEAN = 'M' or 'Z'.

11: PAR(LPAR) – REAL (KIND=nag_wp) array *Input*

On entry: must contain the parameter estimates read in row by row in the order $\phi_1, \phi_2, \ldots, \phi_p$, $\theta_1, \theta_2, \ldots, \theta_q$, μ.

Thus,

if IP > 0, 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 ϕ_l, for $l = 1, 2, \ldots, p$, $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, k$;

if IQ > 0, 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 θ_l, for $l = 1, 2, \ldots, q$, $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, k$;

if MEAN = 'M', PAR$((p+q) \times k \times k + i)$ must be set equal to an estimate of the ith component of μ, for $i = 1, 2, \ldots, k$.

Constraint: the first IP $\times$ K $\times$ K elements of PAR must satisfy the stationarity condition and the next IQ $\times$ K $\times$ 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', LPAR $\geq \max(1, (\mathrm{IP} + \mathrm{IQ}) \times \mathrm{K} \times \mathrm{K})$;
if MEAN = 'M', LPAR $\geq (\mathrm{IP} + \mathrm{IQ}) \times \mathrm{K} \times \mathrm{K} + \mathrm{K}$.

13: QQ(KMAX, K) – REAL (KIND=nag_wp) array *Input/Output*

On entry: QQ(i,j) must contain an estimate of the (i,j)th element of Σ. The lower triangle only is needed.

Constraint: QQ must be positive definite.

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

14: V(KMAX, $*$) – REAL (KIND=nag_wp) array *Input*

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

On entry: V(i,t) must contain an estimate of the ith component of ϵ_{t+d}, for $i = 1, 2, \ldots, k$ and $t = 1, 2, \ldots, n - d$.

If $q = 0$, V is not used.

15: LMAX – INTEGER *Input*

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

Constraint: LMAX ≥ 1.

16: PREDZ(KMAX, LMAX) – REAL (KIND=nag_wp) array *Output*

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

17: SEFZ(KMAX, LMAX) – REAL (KIND=nag_wp) array *Output*

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

18: REF(LREF) – REAL (KIND=nag_wp) array *Output*

On exit: the reference vector which may be used to update forecasts using G13DKF. The first $(\mathrm{LMAX}-1)\times\mathrm{K}\times\mathrm{K}$ elements contain the ψ weight matrices, $\psi_1, \psi_2, \ldots, \psi_{l_{\max}-1}$. The next $\mathrm{K}\times\mathrm{LMAX}$ elements contain the forecasts of the transformed series $\hat{Z}^*_{n+1}, \hat{Z}^*_{n+2}, \ldots, \hat{Z}^*_{n+l_{\max}}$ and the next $\mathrm{K}\times\mathrm{LMAX}$ 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: $\mathrm{LREF}\geq(\mathrm{LMAX}-1)\times\mathrm{K}\times\mathrm{K}+2\times\mathrm{K}\times\mathrm{LMAX}+\mathrm{K}$.

20: WORK(LWORK) – REAL (KIND=nag_wp) 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 $r=\max(\mathrm{IP},\mathrm{IQ})$ and $d=\max(\mathrm{ID}(i))$, for $i=1,2,\ldots,k$, $\mathrm{LWORK}\geq\max\left\{\mathrm{K}r(\mathrm{K}r+2),(\mathrm{IP}+d+2)\mathrm{K}^2+(\mathrm{N}+\mathrm{LMAX})\mathrm{K}\right\}$.

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

24: 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: $\mathrm{IFAIL}=0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry $\mathrm{IFAIL}=0$ 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, $\mathrm{K}<1$,
or $\mathrm{N}<3$,
or $\mathrm{KMAX}<\mathrm{K}$,
or $\mathrm{ID}(i)<0$ for some $i=1,2,\ldots,k$,
or $\mathrm{ID}(i)\geq\mathrm{N}-\max(\mathrm{IP},\mathrm{IQ})$ for some $i=1,2,\ldots,k$,
or $\mathrm{IP}<0$,
or $\mathrm{IQ}<0$,
or $\mathrm{MEAN}\neq$ 'M' or 'Z',

or $\text{LPAR} < (\text{IP} + \text{IQ}) \times \text{K} \times \text{K} + \text{K}$, and MEAN = 'M',
or $\text{LPAR} < (\text{IP} + \text{IQ}) \times \text{K} \times \text{K}$ and MEAN = 'Z',
or $\text{N} \times \text{K} \leq (\text{IP} + \text{IQ}) \times \text{K} \times \text{K} + \text{K} + \text{K}(\text{K} + 1)/2$, and MEAN = 'M',
or $\text{N} \times \text{K} \leq (\text{IP} + \text{IQ}) \times \text{K} \times \text{K} + \text{K}(\text{K} + 1)/2$ and MEAN = 'Z',
or $\text{LMAX} < 1$,
or $\text{LREF} < (\text{LMAX} - 1) \times \text{K} \times \text{K} + 2 \times \text{K} \times \text{LMAX} + \text{K}$,
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, you 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, you 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, you 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. You should check whether the transformations requested in the array TR are sensible. All output parameters are undefined.

IFAIL = −99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL = −399

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL = −999

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

The matrix computations are believed to be stable.

8 Parallelism and Performance

G13DJF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G13DJF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 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 ∇^2 to the first series and the first-order differencing operator ∇ to the second series:

$$w_{1t} = \nabla^2 z_{1t} = (1-B)^2 z_{1t} = (1-2B+B^2)Z_{1t}, \qquad \text{and}$$
$$w_{2t} = \nabla z_{2t} = (1-B)z_{2t}.$$

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

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

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.

G13DJF 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, you may use G13DXF, before calling this routine, to check that the VARMA model being used is invertible.

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

10 Example

This example computes 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. G13DDF is first called to fit an AR(1) model to the series. The mean vector μ is to be estimated and $\phi_1(2,1)$ constrained to be zero.

10.1 Program Text

```
!   G13DJF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.

    Module g13djfe_mod

!     G13DJF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                               :: fprint
!     .. Parameters ..
      Integer, Parameter, Public           :: iset = 1, nin = 5, nout = 6
    Contains
```

```
      Subroutine fprint(k,nm,lmax,predz,sefz,ldsefz,nout)

!       .. Scalar Arguments ..
        Integer, Intent (In)               :: k, ldsefz, lmax, nm, nout
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In)    :: predz(ldsefz,lmax),           &
                                              sefz(ldsefz,lmax)
!       .. Local Scalars ..
        Integer                            :: i, i2, j, l, l2, loop
!       .. Intrinsic Procedures ..
        Intrinsic                          :: min, mod
!       .. Executable Statements ..
        Write (nout,*) ' FORECAST SUMMARY TABLE'
        Write (nout,*) ' ----------------------'
        Write (nout,*)
        Write (nout,99999) ' Forecast origin is set at t = ', nm
        Write (nout,*)
        loop = lmax/5
        If (mod(lmax,5)/=0) Then
          loop = loop + 1
        End If

        Do j = 1, loop
          i2 = (j-1)*5
          l2 = min(i2+5,lmax)
          Write (nout,99998) 'Lead Time ', (i,i=i2+1,l2)
          Write (nout,*)
          i = 1
          Write (nout,99997) 'Series ', i, ' : Forecast    ', &
            (predz(1,l),l=i2+1,l2)
          Write (nout,99996) ' : Standard Error ', (sefz(1,l),l=i2+1,l2)
          Do i = 2, k
            Write (nout,99997) 'Series ', i, ' : Forecast    ', &
              (predz(i,l),l=i2+1,l2)
            Write (nout,99996) ' : Standard Error ', (sefz(i,l),l=i2+1,l2)
          End Do
          Write (nout,*)
        End Do
        Return

99999   Format (1X,A,I4)
99998   Format (1X,A,12X,5I10)
99997   Format (1X,A,I2,A,5F10.2)
99996   Format (10X,A,4(F7.2,3X),F7.2)
      End Subroutine fprint
    End Module g13djfe_mod
    Program g13djfe

!     G13DJF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: g13ddf, g13djf, g13dlf, nag_wp, x04abf
      Use g13djfe_mod, Only: fprint, iset, nin, nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                 :: cgetol, rlogl
      Integer                            :: d, i, ifail, ip, iprint, iq,     &
                                            ishow, k, kmax, ldcm, liwork,    &
                                            lmax, lpar, lref, lwork, maxcal, &
                                            n, nadv, nd, niter, r, tddelta
      Logical                            :: exact, meanl
      Character (1)                      :: mean
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable    :: cm(:,:), delta(:,:), g(:),       &
                                            par(:), predz(:,:), qq(:,:),     &
                                            ref(:), sefz(:,:), v(:,:),       &
                                            w(:,:), work(:), workl(:), z(:,:)
      Integer, Allocatable               :: id(:), iwork(:)
      Logical, Allocatable               :: parhld(:)
      Character (1), Allocatable         :: tr(:)
```

```
!     .. Intrinsic Procedures ..
      Intrinsic                            :: max, maxval
!     .. Executable Statements ..
      Write (nout,*) 'G13DJF Example Program Results'
      Write (nout,*)

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

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

      Allocate (tr(k),id(k))

!     Read in differencing
      Read (nin,*) id(1:k)

      d = maxval(id(1:k))
      tddelta = max(d,1)
      nd = n - d
      kmax = k
      Allocate (delta(kmax,tddelta),w(kmax,nd),workl(k*n),z(kmax,n))

!     Read in series and the transformation flag
      Read (nin,*)(z(i,1:n),i=1,k)
      Read (nin,*) tr(1:k)

!     If required, read in delta
      If (d>0) Then
        Read (nin,*)(delta(i,1:id(i)),i=1,k)
      End If

!     Difference and / or transform series
      ifail = 0
      Call g13dlf(k,n,z,kmax,tr,id,delta,w,nd,workl,ifail)

!     Read in information on the VARMA
      Read (nin,*) ip, iq, mean, lmax

!     Calculate number of parameters for the VARMA
      lpar = (ip+iq)*k*k
      meanl = .False.
      If (mean=='M' .Or. mean=='m') Then
        lpar = lpar + k
        meanl = .True.
      End If

!     Read in control parameters
      Read (nin,*) iprint, cgetol, maxcal, ishow

!     Read in exact likelihood flag
      Read (nin,*) exact

      ldcm = lpar
      kmax = k
      Allocate (par(lpar),qq(kmax,k),v(kmax,nd),g(lpar),cm(ldcm,lpar), &
        parhld(lpar))

!     Read in initial parameter estimates and free parameter flags
      Read (nin,*) par(1:lpar)
      Read (nin,*) parhld(1:lpar)

!     Read in initial values for covariance matrix Q
      Read (nin,*)(qq(i,1:i),i=1,k)

!     Set the advisory channel to NOUT for monitoring information
      If (iprint>=0 .Or. ishow/=0) Then
        nadv = nout
        Call x04abf(iset,nadv)
      End If
```

```
!     Fit a VARMA model
      ifail = -1
      Call g13ddf(k,nd,ip,iq,meanl,par,lpar,qq,kmax,w,parhld,exact,iprint, &
        cgetol,maxcal,ishow,niter,rlogl,v,g,cm,ldcm,ifail)
      If (ifail/=0) Then
        If (ifail<4) Then
          Go To 100
        End If
      End If

      lref = (lmax-1)*k*k + 2*k*lmax + k
      r = max(ip,iq)
      lwork = max(k*r*(k*r+2),(ip+d+2)*k**2+(n+lmax)*k)
      liwork = k*max(ip,iq)
      Allocate (predz(kmax,lmax),sefz(kmax,lmax),ref(lref),work(lwork), &
        iwork(liwork))

!     Perform forecast
      ifail = 0
      Call g13djf(k,n,z,kmax,tr,id,delta,ip,iq,mean,par,lpar,qq,v,lmax,predz, &
        sefz,ref,lref,work,lwork,iwork,liwork,ifail)

!     Display results
      Call fprint(k,n,lmax,predz,sefz,kmax,nout)

100   Continue

    End Program g13djfe
```

10.2 Program Data

```
G13DJF Example Program Data
2 48                                              :: K,N
0  0                                              :: ID
-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
 'N'  'N'                                         :: TR
1 0 'M' 5                                         :: IP,IQ,MEAN,LMAX
-1 0.0001 3000  0                                 :: IPRINT,CGETOL,MAXCAL,ISHOW
T                                                 :: EXACT
0.0 0.0 0.0 0.0 0.0 0.0                           :: PAR
 F   F   T   F   F   F                            :: PARHLD
0.0
0.0 0.0                                           :: QQ
```

10.3 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
```