NAG Library Routine Document

G13DKF

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

G13DKF accepts a sequence of new observations in a multivariate time series and updates both the forecasts and the standard deviations of the forecast errors. A call to G13DJF must be made prior to calling this routine in order to calculate the elements of a reference vector together with a set of forecasts and their standard errors. On a successful exit from G13DKF the reference vector is updated so that should future series values become available these forecasts may be updated by recalling G13DKF.

2 Specification

```
SUBROUTINE G13DKF (K, LMAX, M, MLAST, Z, KMAX, REF, LREF, V, PREDZ,          &
                   SEFZ, WORK, IFAIL)

INTEGER            K, LMAX, M, MLAST, KMAX, LREF, IFAIL
REAL (KIND=nag_wp) Z(KMAX,M), REF(LREF), V(KMAX,M), PREDZ(KMAX,LMAX),        &
                   SEFZ(KMAX,LMAX), WORK(K*M)
```

3 Description

Let $Z_t = (z_{1t}, z_{2t}, \ldots, z_{kt})^{\mathrm{T}}$, for $t = 1, 2, \ldots, n$, denote a k-dimensional time series for which forecasts of $\hat{Z}_{n+1}, \hat{Z}_{n+2}, \ldots, \hat{Z}_{n+l_{\max}}$ have been computed using G13DJF. Given m further observations $Z_{n+1}, Z_{n+2}, \ldots, Z_{n+m}$, where $m < l_{\max}$, G13DKF updates the forecasts of $Z_{n+m+1}, Z_{n+m+2}, \ldots, Z_{n+l_{\max}}$ and their corresponding standard errors.

G13DKF uses a multivariate version of the procedure described in Box and Jenkins (1976). The forecasts are updated using the ψ weights, computed in G13DJF. If Z_t^* denotes the transformed value of Z_t and $\hat{Z}_t^*(l)$ denotes the forecast of Z_{t+l}^* from time t with a lead of l (that is the forecast of Z_{t+l}^* given observations $Z_t^*, Z_{t-1}^*, \ldots$), then

$$\hat{Z}_{t+1}^*(l) = \tau + \psi_l \epsilon_{t+1} + \psi_{l+1} \epsilon_t + \psi_{l+2} \epsilon_{t-1} + \cdots$$

and

$$\hat{Z}_t^*(l+1) = \tau + \psi_{l+1} \epsilon_t + \psi_{l+2} \epsilon_{t-1} + \cdots$$

where τ is a constant vector of length k involving the differencing parameters and the mean vector μ. By subtraction we obtain

$$\hat{Z}_{t+1}^*(l) = \hat{Z}_t^*(l+1) + \psi_l \epsilon_{t+1}.$$

Estimates of the residuals corresponding to the new observations are also computed as $\epsilon_{n+l} = Z_{n+l}^* - \hat{Z}_n^*(l)$, for $l = 1, 2, \ldots, m$. These may be of use in checking that the new observations conform to the previously fitted model.

On a successful exit, the reference array is updated so that G13DKF may be called again should future series values become available, see Section 9.

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, LMAX, KMAX, REF and LREF from G13DJF are suitable for input to G13DKF.

1: K – INTEGER *Input*

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

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

2: LMAX – INTEGER *Input*

On entry: the number, $l_{\max}$, of forecasts requested in the call to G13DJF.

Constraint: $\mathrm{LMAX} \geq 2$.

3: M – INTEGER *Input*

On entry: m, the number of new observations available since the last call to either G13DJF or G13DKF. The number of new observations since the last call to G13DJF is then $\mathrm{M} + \mathrm{MLAST}$.

Constraint: $0 < \mathrm{M} < \mathrm{LMAX} - \mathrm{MLAST}$.

4: MLAST – INTEGER *Input/Output*

On entry: on the first call to G13DKF, since calling G13DJF, MLAST must be set to 0 to indicate that no new observations have yet been used to update the forecasts; on subsequent calls MLAST must contain the value of MLAST as output on the previous call to G13DKF.

On exit: is incremented by m to indicate that $\mathrm{MLAST} + \mathrm{M}$ observations have now been used to update the forecasts since the last call to G13DJF.

MLAST must not be changed between calls to G13DKF, unless a call to G13DJF has been made between the calls in which case MLAST should be reset to 0.

Constraint: $0 \leq \mathrm{MLAST} < \mathrm{LMAX} - \mathrm{M}$.

5: Z(KMAX, M) – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{Z}(i,j)$ must contain the value of $z_{i,n+\mathrm{MLAST}+j}$, for $i = 1,2,\ldots,k$ and $j = 1,2,\ldots,m$, and where n is the number of observations in the time series in the last call made to G13DJF.

Constraint: if the transformation defined in TR in G13DJF for the ith series is the log transformation, then $\mathrm{Z}(i,j) > 0.0$, and if it is the square-root transformation, then $\mathrm{Z}(i,j) \geq 0.0$, for $j = 1,2,\ldots,m$ and $i = 1,2,\ldots,k$.

6: KMAX – INTEGER *Input*

On entry: the first dimension of the arrays Z, PREDZ, SEFZ and V as declared in the (sub)program from which G13DKF is called.

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

7: REF(LREF) – REAL (KIND=nag_wp) array *Input/Output*

On entry: must contain the first $(\mathrm{LMAX} - 1) \times \mathrm{K} \times \mathrm{K} + 2 \times \mathrm{K} \times \mathrm{LMAX} + \mathrm{K}$ elements of the reference vector as returned on a successful exit from G13DJF (or a previous call to G13DKF).

On exit: the elements of REF are updated. The first $(\mathrm{LMAX}-1) \times \mathrm{K} \times \mathrm{K}$ elements store the ψ weights $\psi_1, \psi_2, \ldots, \psi_{l_{\max}-1}$. The next $\mathrm{K} \times \mathrm{LMAX}$ elements contain the forecasts of the transformed series and the next $\mathrm{K} \times \mathrm{LMAX}$ elements contain the variances of the forecasts of the transformed variables; see G13DJF. The last K elements are not updated.

8: LREF – INTEGER *Input*

On entry: the dimension of the array REF as declared in the (sub)program from which G13DKF is called.

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

9: $\mathrm{V}(\mathrm{KMAX}, \mathrm{M})$ – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{V}(i,j)$ contains an estimate of the ith component of $\epsilon_{n+\mathrm{MLAST}+j}$, for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, m$.

10: $\mathrm{PREDZ}(\mathrm{KMAX}, \mathrm{LMAX})$ – REAL (KIND=nag_wp) array *Input/Output*

On entry: nonupdated values are kept intact.

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

The columns of PREDZ corresponding to the new observations since the last call to either G13DJF or G13DKF are set equal to the corresponding columns of Z.

11: $\mathrm{SEFZ}(\mathrm{KMAX}, \mathrm{LMAX})$ – REAL (KIND=nag_wp) array *Input/Output*

On entry: nonupdated values are kept intact.

On exit: $\mathrm{SEFZ}(i,j)$ contains an estimate of the standard error of the corresponding element of PREDZ, for $i = 1, 2, \ldots, k$ and $j = \mathrm{MLAST} + \mathrm{M} + 1, \ldots, l_{\max}$.

The columns of SEFZ corresponding to the new observations since the last call to either G13DJF or G13DKF are set equal to zero.

12: $\mathrm{WORK}(\mathrm{K} \times \mathrm{M})$ – REAL (KIND=nag_wp) array *Workspace*

13: 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 $\mathrm{LMAX} < 2$,
or $\mathrm{M} \leq 0$,

or MLAST + M $\geq$ LMAX,
or MLAST $<$ 0,
or KMAX $<$ K,
or LREF $<$ (LMAX $-$ 1) $\times$ K $\times$ K + 2 $\times$ K $\times$ LMAX + K.

IFAIL $=$ 2

On entry, some of the elements of the reference vector, REF, have been corrupted since the most recent call to G13DJF (or G13DKF).

IFAIL $=$ 3

On entry, one or more of the elements of Z is invalid, for the transformation being used; that is you may be trying to log or square root a series, some of whose values are negative.

IFAIL $=$ 4

This is an unlikely exit. For one of the series, overflow will occur if the forecasts are updated. You should check whether the elements of REF have been corrupted.

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

G13DKF is not threaded by NAG in any implementation.

G13DKF 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

If a further m^* observations, $Z_{n+\mathrm{MLAST}+1}, Z_{n+\mathrm{MLAST}+2}, \ldots, Z_{n+\mathrm{MLAST}+m^*}$, become available, then forecasts of $Z_{n+\mathrm{MLAST}+m^*+1}, Z_{n+\mathrm{MLAST}+m^*+2}, \ldots, Z_{n+l_{\mathrm{max}}}$ may be updated by recalling G13DKF with $\mathrm{M} = m^*$. Note that M and the contents of the array Z are the only quantities which need updating; MLAST is updated on exit from the previous call. On a successful exit, V contains estimates of $\epsilon_{n+\mathrm{MLAST}+1}, \epsilon_{n+\mathrm{MLAST}+2}, \ldots, \epsilon_{n+\mathrm{MLAST}+m^*}$; columns $\mathrm{MLAST}+1, \mathrm{MLAST}+2, \ldots, \mathrm{MLAST}+m^*$ of PREDZ contain the new observed values $Z_{n+\mathrm{MLAST}+1}, Z_{n+\mathrm{MLAST}+2}, \ldots, Z_{n+\mathrm{MLAST}+m^*}$ and columns $\mathrm{MLAST}+1, \mathrm{MLAST}+2, \ldots, \mathrm{MLAST}+m^*$ of SEFZ are set to zero.

10 Example

This example shows how to update the forecasts of two series each of length 48. No transformation has been used and no differencing applied to either of the series. G13DDF is first called to fit an AR(1) model to the series. μ is to be estimated and $\phi_1(2,1)$ constrained to be zero. A call to G13DJF is then made in order to compute forecasts of the next five series values. After one new observation becomes available the four forecasts are updated. A further observation becomes available and the three forecasts are updated.

10.1 Program Text

```
!   G13DKF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.

    Module g13dkfe_mod

!     G13DKF 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,*)
        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) loop = loop + 1
        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 g13dkfe_mod
    Program g13dkfe

!     G13DKF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: g13ddf, g13djf, g13dkf, g13dlf, nag_wp, x04abf
      Use g13dkfe_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, m,      &
                                              maxcal, mlast, n, nadv, nd,      &
                                              niter, r, tddelta, tdv
      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,*) 'G13DKF Example Program Results'
      Write (nout,*)

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

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

      Allocate (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 (z(kmax,n),tr(k),delta(kmax,tddelta),w(kmax,nd),workl(k*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
      If (mean=='M' .Or. mean=='m') Then
        lpar = lpar + k
        meanl = .True.
      Else
        meanl = .False.
      End If

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

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

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

!     Forecast from VARMA
      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)

!     Update forecasts
      mlast = 0
d_lp: Do
        Read (nin,*,Iostat=ifail) m
        If (ifail/=0) Then
          Exit d_lp
        End If
        Read (nin,*,Iostat=ifail)(z(1:k,i),i=1,m)
        If (ifail/=0) Then
          Exit d_lp
        End If

!       Reallocate V if required
        If (tdv<m) Then
```

```
          Deallocate (v)
          Allocate (v(kmax,m))
        End If

!       Reallocate WORK if required
        If (lwork<k*m) Then
          Deallocate (work)
          Allocate (work(lwork))
        End If

!       Update forecast
        ifail = 0
        Call g13dkf(k,lmax,m,mlast,z,kmax,ref,lref,v,predz,sefz,work,ifail)

        Call fprint(k,n+mlast,lmax,predz,sefz,kmax,nout)
      End Do d_lp

100   Continue

    End Program g13dkfe
```

10.2 Program Data

```
G13DKF 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                                   :: End of QQ
1                                         :: M (update 1)
8.1 10.2                                  :: Z (update 1)
1                                         :: M (update 2)
8.5 10.0                                  :: Z (update 2)
```

10.3 Program Results

```
 G13DKF 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

  FORECAST SUMMARY TABLE
  ----------------------

  Forecast origin is set at t =   49

 Lead Time                     1          2          3          4          5

 Series  1 : Forecast         8.10       7.49       6.94       6.46       6.06
           : Standard Error    0.00       1.72       2.23       2.51       2.68
 Series  2 : Forecast        10.20       9.19       8.61       8.28       8.08
           : Standard Error    0.00       2.32       2.68       2.78       2.82

  FORECAST SUMMARY TABLE
  ----------------------

  Forecast origin is set at t =   50

 Lead Time                     1          2          3          4          5

 Series  1 : Forecast         8.10       8.50       7.80       7.18       6.65
           : Standard Error    0.00       0.00       1.72       2.23       2.51
 Series  2 : Forecast        10.20      10.00       9.08       8.54       8.24
           : Standard Error    0.00       0.00       2.32       2.68       2.78
```