NAG Toolbox for MATLAB

g13dk

1 Purpose

g13dk 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 g13dj must be made prior to calling this function 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 g13dk the reference vector is updated so that should future series values become available these forecasts may be updated by recalling g13dk.

2 Syntax

```
[mlast, ref, v, predz, sefz, ifail] = g13dk(k, mlast, z, ref, predz,
sefz, 'lmax', lmax, 'm', m, 'lref', lref)
```

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 g13dj. Given m further observations $Z_{n+1}, Z_{n+2}, \ldots, Z_{n+m}$, where $m < l_{\max}$, g13dk updates the forecasts of $Z_{n+m+1}, Z_{n+m+2}, \ldots, Z_{n+l_{\max}}$ and their corresponding standard errors.

g13dk uses a multivariate version of the procedure described in Box and Jenkins 1976. The forecasts are updated using the ψ weights, computed in g13dj. 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 g13dk may be called again should future series values become available, see Section 8.

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 g13dj are suitable for input to g13dk.

5.1 Compulsory Input Parameters

1: **k – int32 scalar**

k, the dimension of the multivariate time series.

Constraint: $\mathbf{k} \geq 1$.

2: **mlast – int32 scalar**

On the first call to g13dk, since calling g13dj, **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 g13dk.

Constraint: $0 \leq \mathbf{mlast} < \mathbf{lmax} - \mathbf{m}$.

3: **z(kmax,m) – double array**

kmax, the first dimension of the array, must be at least **k**.

$\mathbf{z}(i,j)$ must contain the value of $z_{i,n+\mathbf{mlast}+j}$, 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 g13dj.

Constraint: if the transformation defined in **tr** in g13dj for the ith series is the log transformation, then $\mathbf{z}(i,j) > 0.0$, and if it is the square-root transformation, then $\mathbf{z}(i,j) \geq 0.0$, for $j = 1,2,\ldots,m$ and $i = 1,2,\ldots,k$

4: **ref(lref) – double array**

Must contain the first $(\mathbf{lmax} - 1) \times \mathbf{k} \times \mathbf{k} + 2 \times \mathbf{k} \times \mathbf{lmax} + \mathbf{k}$ elements of the reference vector as returned on a successful exit from g13dj (or a previous call to g13dk).

5: **predz(kmax,lmax) – double array**

kmax, the first dimension of the array, must be at least **k**.

Nonupdated values are kept intact.

6: **sefz(kmax,lmax) – double array**

kmax, the first dimension of the array, must be at least **k**.

Nonupdated values are kept intact.

5.2 Optional Input Parameters

1: **lmax – int32 scalar**

Default: The dimension of the arrays **predz**, **sefz**. (An error is raised if these dimensions are not equal.)

the number, $l_{\max}$, of forecasts requested in the call to g13dj.

Constraint: $\mathbf{lmax} \geq 2$.

2: **m – int32 scalar**

Default: The dimension of the arrays **z**, **v**. (An error is raised if these dimensions are not equal.)

m, the number of new observations available since the last call to either g13dj or g13dk. The number of new observations since the last call to g13dj is then $\mathbf{m} + \mathbf{mlast}$.

Constraint: $0 < \mathbf{m} < \mathbf{lmax} - \mathbf{mlast}$.

3: **lref – int32 scalar**

Default: The dimension of the array **ref**.

Constraint: **lref** $\geq$ (**lmax** $-$ 1) $\times$ **k** $\times$ **k** $+$ 2 $\times$ **k** $\times$ **lmax** $+$ **k**.

5.3 Input Parameters Omitted from the MATLAB Interface

kmax, work

5.4 Output Parameters

1: **mlast – int32 scalar**

Is incremented by m to indicate that **mlast** $+$ **m** observations have now been used to update the forecasts since the last call to g13dj.

mlast must not be changed between calls to g13dk, unless a call to g13dj has been made between the calls in which case **mlast** should be reset to 0.

2: **ref**(**lref**) **– double array**

The elements of **ref** are updated. The first (**lmax** $-$ 1) $\times$ **k** $\times$ **k** elements store the ψ weights $\psi_1, \psi_2, \ldots, \psi_{l_{\max}-1}$. The next **k** $\times$ **lmax** elements contain the forecasts of the transformed series and the next **k** $\times$ **lmax** elements contain the variances of the forecasts of the transformed variables; see g13dj. The last **k** elements are not updated.

3: **v**(**kmax,m**) **– double array**

v(i,j) contains an estimate of the ith component of $\epsilon_{n+\mathbf{mlast}+j}$, for $i = 1, 2, \ldots, k$ and $j = 1, 2, \ldots, m$.

4: **predz**(**kmax,lmax**) **– double array**

predz(i,j) contains the updated forecast of $z_{i,n+j}$, for $i = 1, 2, \ldots, k$ and $j =$ **mlast** $+$ **m** $+ 1,$ **mlast** $+$ **m** $+ 2, \ldots, l_{\max}$.

The columns of **predz** corresponding to the new observations since the last call to either g13dj or g13dk are set equal to the corresponding columns of **z**.

5: **sefz**(**kmax,lmax**) **– double array**

sefz(i,j) contains an estimate of the standard error of the corresponding element of **predz**, for $i = 1, 2, \ldots, k$ and $j =$ **mlast** $+$ **m** $+ 1,$ **mlast** $+$ **m** $+ 2, \ldots, l_{\max}$.

The columns of **sefz** corresponding to the new observations since the last call to either g13dj or g13dk are set equal to zero.

6: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **k** < 1,
or **lmax** < 2,
or **m** ≤ 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 g13dj (or g13dk).

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.

7 Accuracy

The matrix computations are believed to be stable.

8 Further Comments

If a further m^* observations, $Z_{n+\mathbf{mlast}+1}, Z_{n+\mathbf{mlast}+2}, \ldots, Z_{n+\mathbf{mlast}+m^*}$, become available, then forecasts of $Z_{n+\mathbf{mlast}+m^*+1}, Z_{n+\mathbf{mlast}+m^*+2}, \ldots, Z_{n+l_{\max}}$ may be updated by recalling g13dk with $\mathbf{m} = m^*$. 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 $\epsilon_{n+\mathbf{mlast}+1}, \epsilon_{n+\mathbf{mlast}+2}, \ldots, \epsilon_{n+\mathbf{mlast}+m^*}$; columns $\mathbf{mlast}+1, \mathbf{mlast}+2, \ldots, \mathbf{mlast}+m^*$ of **predz** contain the new observed values $Z_{n+\mathbf{mlast}+1}, Z_{n+\mathbf{mlast}+2}, \ldots, Z_{n+\mathbf{mlast}+m^*}$ and columns $\mathbf{mlast}+1, \mathbf{mlast}+2, \ldots, \mathbf{mlast}+m^*$ of **sefz** are set to zero.

9 Example

```
k = int32(2);
mlast = int32(0);
z = [8.1;
     10.2];
ref = [0.8016071892386086;
     0;
     0.0648134906597352;
     0.575015951133362;
     0.6425740858390225;
     0;
     0.08922375105047406;
     0.330643344057805;
     0.5150920068269873;
     0;
     0.09295254958469001;
     0.1901251969693142;
     0.4129014557918555;
     0;
     0.08683410968309672;
     0.109325020969728;
     7.82042808779155;
     10.30633951031062;
     7.277073498724811;
     9.251955479776221;
     6.773178244308308;
     8.645667843598654;
     6.329956567700314;
     8.297042781821611;
     5.952071278119719;
     8.096577810334958;
     2.964154253391392;
```

```
5.379903126133676;
4.957660260306922;
7.158732286465553;
6.297464315601706;
7.746890308545223;
7.191419098742513;
7.941360843800069;
7.783032455102286;
8.005661231897443;
100;
100;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
0;
```

```
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0];
predz   =   [7.82042808779155,   7.277073498724811,   6.773178244308308,
6.329956567700314, 5.952071278119719;
             10.30633951031062,   9.251955479776221,   8.645667843598654,
8.297042781821611, 8.096577810334958];
sefz   =   [1.721671935471852,   2.226580396102266,   2.509474908342721,
2.681682139766477, 2.789808677150153;
             2.319461818209922,   2.675580738169856,   2.783323608304507,
2.81804202307206, 2.829427721624541];
[mlastOut, refOut, v, predzOut, sefzOut, ifail] = g13dk(k, mlast, z, ref,
predz, sefz)
```

```
mlastOut =
          1
refOut =
     array elided
v =
    0.2796
   -0.1063
predzOut =
    8.1000    7.4943    6.9433    6.4641    6.0583
   10.2000    9.1908    8.6105    8.2768    8.0850
sefzOut =
         0    1.7217    2.2266    2.5095    2.6817
```

```
         0    2.3195    2.6756    2.7833    2.8180
ifail =
           0
```