# NAG Toolbox

# nag_tsa_multi_varma_update (g13dk)

## 1 Purpose

nag_tsa_multi_varma_update (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 nag_tsa_multi_varma_forecast (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 nag_tsa_multi_varma_update (g13dk) the reference vector is updated so that should future series values become available these forecasts may be updated by recalling nag_tsa_multi_varma_update (g13dk).

## 2 Syntax

```
[mlast, ref, v, predz, sefz, ifail] = nag_tsa_multi_varma_update(k, mlast, z, ref, predz, sefz, 'lmax', lmax, 'm', m, 'lref', lref)
[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 nag_tsa_multi_varma_forecast (g13dj). Given $m$ further observations $Z_{n+1}, Z_{n+2}, \ldots, Z_{n+m}$, where $m < l_{\max}$, nag_tsa_multi_varma_update (g13dk) updates the forecasts of $Z_{n+m+1}, Z_{n+m+2}, \ldots, Z_{n+l_{\max}}$ and their corresponding standard errors.

nag_tsa_multi_varma_update (g13dk) uses a multivariate version of the procedure described in Box and Jenkins (1976). The forecasts are updated using the $\psi$ weights, computed in nag_tsa_multi_varma_forecast (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 $\tau$ is a constant vector of length $k$ involving the differencing parameters and the mean vector $\mu$. 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 nag_tsa_multi_varma_update (g13dk) 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 nag_tsa_multi_varma_forecast (g13dj) are suitable for input to nag_tsa_multi_varma_update (g13dk).

## 5.1 Compulsory Input Parameters

1: **k** – INTEGER

$k$, the dimension of the multivariate time series.

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

2: **mlast** – INTEGER

On the first call to nag_tsa_multi_varma_update (g13dk), since calling nag_tsa_multi_varma_forecast (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 nag_tsa_multi_varma_update (g13dk).

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

3: $\mathbf{z}(kmax, \mathbf{m})$ – REAL (KIND=nag_wp) array

$kmax$, the first dimension of the array, must satisfy the constraint $kmax \geq \mathbf{k}$.

$\mathbf{z}(i,j)$ 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 nag_tsa_multi_varma_forecast (g13dj).

*Constraint*: if the transformation defined in **tr** in nag_tsa_multi_varma_forecast (g13dj) for the $i$th 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: $\mathbf{ref}(\mathbf{lref})$ – REAL (KIND=nag_wp) 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 nag_tsa_multi_varma_forecast (g13dj) (or a previous call to nag_tsa_multi_varma_update (g13dk)).

5: $\mathbf{predz}(kmax, \mathbf{lmax})$ – REAL (KIND=nag_wp) array

$kmax$, the first dimension of the array, must satisfy the constraint $kmax \geq \mathbf{k}$.

Nonupdated values are kept intact.

6: $\mathbf{sefz}(kmax, \mathbf{lmax})$ – REAL (KIND=nag_wp) array

$kmax$, the first dimension of the array, must satisfy the constraint $kmax \geq \mathbf{k}$.

Nonupdated values are kept intact.

## 5.2 Optional Input Parameters

1: **lmax** – INTEGER

*Default*: the second 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 nag_tsa_multi_varma_forecast (g13dj).

*Constraint*: **lmax** $\geq 2$.

2: **m** – INTEGER

*Default*: the second dimension of the array **z**.

$m$, the number of new observations available since the last call to either nag_tsa_multi_varma_forecast (g13dj) or nag_tsa_multi_varma_update (g13dk). The number of new observations since the last call to nag_tsa_multi_varma_forecast (g13dj) is then **m** + **mlast**.

*Constraint*: $0 <$ **m** $<$ **lmax** $-$ **mlast**.

3: **lref** – INTEGER

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

The dimension of the array **ref**.

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

## 5.3 Output Parameters

1: **mlast** – INTEGER

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

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

2: **ref**(**lref**) – REAL (KIND=nag_wp) array

The elements of **ref** are updated. The first (**lmax** $- 1$) $\times$ **k** $\times$ **k** elements store the $\psi$ 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 nag_tsa_multi_varma_forecast (g13dj). The last **k** elements are not updated.

3: **v**($kmax$, **m**) – REAL (KIND=nag_wp) array

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

4: **predz**($kmax$, **lmax**) – REAL (KIND=nag_wp) array

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

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

5: **sefz**($kmax$, **lmax**) – REAL (KIND=nag_wp) 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** $+ 1, \ldots, l_{\max}$.

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

6: **ifail** – INTEGER

**ifail** $= 0$ unless the function detects an error (see Section 5).

## 6 Error Indicators and Warnings

Errors or warnings detected by the function:

**ifail** $= 1$

On entry, **k** $< 1$,
or **lmax** $< 2$,
or **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 nag_tsa_multi_varma_forecast (g13dj) (or nag_tsa_multi_varma_update (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.

**ifail** $= -99$

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

**ifail** $= -399$

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

**ifail** $= -999$

Dynamic memory allocation failed.

## 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 nag_tsa_multi_varma_update (g13dk) with **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 **mlast** $+ 1,$ **mlast** $+ 2, \ldots,$ **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 **mlast** $+ 1,$ **mlast** $+ 2, \ldots,$ **mlast** $+ m^*$ of **sefz** are set to zero.

# 9 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. nag_tsa_multi_varma_estimate (g13dd) is first called to fit an AR(1) model to the series. $\mu$ is to be estimated and $\phi_1(2,1)$ constrained to be zero. A call to nag_tsa_multi_varma_forecast (g13dj) 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.

## 9.1 Program Text

```
     function g13dk_example

fprintf('g13dk example results\n\n');

% Series
z = [-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];
[k,n] = size(z);

% Difference /transform series
tr    = {'N'; 'N'};
id    = [nag_int(0);0];
delta = [0;  0];
[w, nd, ifail] = g13dl( ...
 z, tr, id, delta);

% VARMA info
ip    = nag_int(1);
iq    = nag_int(0);
mean_p = true;

% Initial parameter estimates and free parameter flags
par    = zeros(6, 1);
parhld = [false;  false;  true;  false;  false;  false];

% Exact likelihood
exact  = true;
% control parameters
iprint = nag_int(-1);
cgetol = 0.0001;
ishow  = nag_int(0);

qq = [0, 0; 0, 0];

% Fit VARMA
[par, qq, ~, ~, v, ~, ~, ifail] = ...
  g13dd( ...
  ip, iq, mean_p, par, qq, w, parhld, exact, iprint, cgetol, ...
  ishow, 'n', nd);

% Perform forecast
lmax = nag_int(5);
lref = nag_int(150);
mean_p = 'M';
[qq, predz, sefz, ref, ifail] = ...
  g13dj( ...
  z, tr, id, delta, ip, iq, mean_p, par, qq, v, lmax, lref);
```

```
% Display results
g13dk_print(k,n,lmax,predz,sefz);

% Update forecasts
m = nag_int([1; 1]);
z = [ 8.1  8.5;
     10.2 10.0];
mlast = nag_int(0);
for j = 1:numel(m)
  [mlast, ref, ~, predz, sefz, ifail] = ...
    g13dk( ...
    nag_int(k), mlast, z(:,j), ref, predz, sefz);
  % Display results
  g13dk_print(k,n+mlast,lmax,predz,sefz);
end

function g13dk_print(k,n,lmax,predz,sefz)
  fprintf('\n Forecast Summary Table\n');
  fprintf(' ----------------------\n\n');
  fprintf(' Forecast origin is set at t = %4d\n\n', n);
  loop = lmax/5;
  if mod(lmax,5)~=0
    loop = loop + 1;
  end
  for j = 1:loop
    i2 = (j-1)*5;
    l2 = min(i2+5,lmax);
    fprintf('Lead Time %14s',' ');
    fprintf('%7d', [i2+1:l2]);
    fprintf('\n\n');
    for i = 1:k
      fprintf('Series %d : Forecast       ', i);
      fprintf('%7.2f',predz(i,i2+1:l2));
      fprintf('\n%8s : Standard Error ', ' ');
      fprintf('%7.2f',sefz(i,i2+1:l2));
      fprintf('\n');
    end
  end
```

## 9.2 Program Results

```
     g13dk example 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
```