NAG Toolbox

nag_tsa_uni_arima_update (g13ag)

1 Purpose

nag_tsa_uni_arima_update (g13ag) accepts a series of new observations of a time series, the model of which is already fully specified, and updates the 'state set' information for use in constructing further forecasts. The previous specifications of the time series model should have been obtained by using nag_tsa_uni_arima_estim (g13ae) or nag_tsa_uni_arima_estim_easy (g13af) to estimate the relevant parameters. The supplied state set will originally have been produced by nag_tsa_uni_arima_estim (g13ae) or nag_tsa_uni_arima_estim_easy (g13af), but may since have been updated by earlier calls to nag_tsa_uni_arima_update (g13ag).

A set of residuals corresponding to the new observations is returned. These may be of use in checking that the new observations conform to the previously fitted model.

2 Syntax

```
[st, anexr, ifail] = nag_tsa_uni_arima_update(st, mr, par, c, anx, 'nst', nst,
'npar', npar, 'nuv', nuv)
```

```
[st, anexr, ifail] = g13ag(st, mr, par, c, anx, 'nst', nst, 'npar', npar, 'nuv',
nuv)
```

3 Description

The time series model is specified as outlined in Section 3 in nag_tsa_uni_arima_estim (g13ae) or nag_tsa_uni_arima_estim_easy (g13af). This also describes how the state set, which contains the minimum amount of time series information needed to construct forecasts, is made up of

(i) the differenced series w_t (uncorrected for the constant c), for $(N - P \times s) < t \leq N$,

(ii) the d' values required to reconstitute the original series x_t from the differenced series w_t,

(iii) the intermediate series e_t, for $(N - \max(p, Q \times s)) < t \leq N$, and

(iv) the residual series a_t, for $(N - q) < t \leq N$.

If the number of original undifferenced observations was n, then $d' = d + (D \times s)$ and $N = n - d'$.

To update the state set, given a number of new undifferenced observations x_t, $t = n + 1, n + 2, \ldots, n + k$, the four series above are first reconstituted.

Differencing and residual calculation operations are then applied to the new observations and k new values of w_t, e_t and a_t are derived.

The first k values in these three series are then discarded and a new state set is obtained.

The residuals in the a_t series corresponding to the k new observations are preserved in an output array. The parameters of the time series model are not changed in this function.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **st**(**nst**) – REAL (KIND=nag_wp) array

The state set derived from nag_tsa_uni_arima_estim (g13ae) or nag_tsa_uni_arima_estim_easy (g13af), or as modified using earlier calls of nag_tsa_uni_arima_update (g13ag).

2: **mr**(**7**) – INTEGER array

The orders vector (p, d, q, P, D, Q, s) of the ARIMA model, in the usual notation.

Constraints:

$p, d, q, P, D, Q, s \geq 0$;
$p + q + P + Q > 0$;
$s \neq 1$;
if $s = 0$, $P + D + Q = 0$;
if $s > 1$, $P + D + Q > 0$.

3: **par**(**npar**) – REAL (KIND=nag_wp) array

The estimates of the p values of the ϕ parameters, the q values of the θ parameters, the P values of the Φ parameters and the Q values of the Θ parameters in the model – in that order, using the usual notation.

4: **c** – REAL (KIND=nag_wp)

The constant to be subtracted from the differenced data.

5: **anx**(**nuv**) – REAL (KIND=nag_wp) array

The new undifferenced observations which are to be used to update **st**.

5.2 Optional Input Parameters

1: **nst** – INTEGER

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

The number of values in the state set array **st**.

Constraint: **nst** $= P \times s + D \times s + d + q + \max(p, Q \times s)$. (As returned by nag_tsa_uni_arima_estim (g13ae) or nag_tsa_uni_arima_estim_easy (g13af)).

2: **npar** – INTEGER

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

The number of ϕ, θ, Φ and Θ parameters in the model.

Constraint: **npar** $= p + q + P + Q$.

3: **nuv** – INTEGER

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

k, the number of new observations in **anx**.

5.3 Output Parameters

1: **st**(**nst**) – REAL (KIND=nag_wp) array

The updated values of the state set.

2: **anexr**(**nuv**) – REAL (KIND=nag_wp) array

The residuals corresponding to the new observations in **anx**.

3: **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, **npar** $\neq p + q + P + Q$,
or the orders vector **mr** is invalid (check the constraints in Section 5).

ifail $= 2$

On entry, **nst** $\neq P \times s + D \times s + d + q + \max(Q \times s, p)$.

ifail $= 3$

On entry, **nuv** ≤ 0.

ifail $= 4$

On entry, $nwa < 4 \times$ **npar** $+ 3 \times$ **nst**.

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 computations are believed to be stable.

8 Further Comments

The time taken by nag_tsa_uni_arima_update (g13ag) is approximately proportional to **nuv** × **npar**.

9 Example

The following program is based on data derived from a study of monthly airline passenger totals (in thousands) to which a logarithmic transformation had been applied. The time series model was based on seasonal and non-seasonal differencing both of order 1, with seasonal period 12. The number of parameters estimated was two: a non-seasonal moving average parameter θ_1 with value 0.327 and a seasonal moving average parameter Θ_1 with value 0.6270. There was no constant correction. These, together with the state set array, were obtained using nag_tsa_uni_arima_estim (g13ae).

Twelve new observations are supplied. The function updates the state set and outputs a set of residuals corresponding to the new observations.

9.1 Program Text

```
     function g13ag_example

fprintf('g13ag example results\n\n');

% Previously calculated state set (g13ae, g13af or prior g13ag)
st = [0.0118;  -0.0669;   0.1296;  -0.0394;   0.0422;   0.1809;   0.1211;
      0.0281;  -0.2231;  -0.1181;  -0.1468;   0.0835;   5.8201;  -0.0157;
     -0.0361;  -0.0266;  -0.0199;   0.0298;   0.0290;   0.0147;   0.0373;
     -0.0931;   0.0223;  -0.0172;  -0.0353;  -0.0413];

% Orders
mr = [nag_int(0);1;1;0;1;1;12];
par = [0.327;    0.627];
c = 0;

% New update observations
anx = [5.8861;  5.8348;   6.0064;   5.9814;   6.0403;   6.1570;
       6.3063;  6.3261;   6.1377;   6.0088;   5.8916;   6.0039];
nux = numel(anx);

% Update state set
[st, anexr, ifail] = g13ag( ...
                            st, mr, par, c, anx);

%  Display results
nst = numel(st);
fprintf('The updated state set array now holds the values\n');
for j = 1:8:nst
  fprintf('%8.4f', st(j:min(j+7,nst)));
  fprintf('\n');
end
fprintf('\nThe residuals corresponding to the %4d values used to\n', nux);
fprintf('update the system are\n');
for j = 1:8:nux
  fprintf('%8.4f', anexr(j:min(j+7,nux)));
  fprintf('\n');
end
```

9.2 Program Results

```
     g13ag example results

The updated state set array now holds the values
  0.0660 -0.0513  0.1716 -0.0250  0.0589  0.1167  0.1493  0.0198
 -0.1884 -0.1289 -0.1172  0.1123  6.0039  0.0444 -0.0070  0.0253
  0.0019  0.0354 -0.0460  0.0374  0.0151 -0.0237  0.0032  0.0188
  0.0067  0.0126

The residuals corresponding to the   12 values used to
update the system are
  0.0309  0.0031  0.0263  0.0105  0.0388 -0.0333  0.0265  0.0238
 -0.0159 -0.0020  0.0182  0.0126
```