NAG C Library Function Document

nag_tsa_transf_prelim_fit (g13bdc)

1 Purpose

nag_tsa_transf_prelim_fit (g13bdc) calculates preliminary estimates of the parameters of a transfer function model.

2 Specification

```
void nag_tsa_transf_prelim_fit (double r0, const double r[], Integer nl,
    Nag_TransfOrder *transfv, double s, double wds[], Integer isf[], NagError *fail)
```

3 Description

The routine calculates estimates of parameters $\delta_1, \delta_2, \ldots, \delta_p$, $\omega_0, \omega_1, \ldots, \omega_q$ in the transfer function model

$$y_t = \delta_1 y_{t-1} + \delta_2 y_{t-2} + \ldots + \delta_p y_{t-p} + \omega_0 x_{t-b} - \omega_1 x_{t-b-1} - \ldots - \omega_q x_{t-b-q}$$

given cross-correlations between the series x_t and lagged values of y_t:

$$r_{xy}(l), \quad l = 0, 1, \ldots, L$$

and the ratio of standard deviations s_y/s_x, as supplied by nag_tsa_cross_corr (g13bcc).

It is assumed that the series x_t used to calculate the cross-correlations is a sample from a time series with true autocorrelations of zero. Otherwise the cross-correlations between the series b_t and a_t, as defined in the description of nag_tsa_arma_filter (g13bac), should be used in place of those between y_t and x_t.

The estimates are obtained by solving for $\delta_1, \delta_2, \ldots, \delta_p$ the equations

$$r_{xy}(b+q+j) = \delta_1 r_{xy}(b+q+j-1) + \cdots + \delta_p r_{xy}(b+q+j-p), \quad j = 1, 2, \ldots, p$$

then calculating

$$\omega_i = \pm(s_y/s_x) r_{xy}(b+i) - \delta_1 r_{xy}(b+i-1) - \cdots - \delta_p r_{xy}(b+i-p), \quad i = 0, 1, \ldots, q$$

where the '$+$' is used for ω_0 and '$-$' for ω_i, $i > 0$.

Any value of $r_{xy}(l)$ arising in these equations for $l < b$ is taken as zero. The parameters $\delta_1, \delta_2, \ldots, \delta_p$ are checked as to whether they satisfy the stability criterion.

4 References

Box G E P and Jenkins G M (1976) *Time Series Analysis: Forecasting and Control* (Revised Edition) Holden–Day

5 Parameters

1: **r0** – double *Input*

On entry: the cross-correlation between the two series at lag 0, $r_{xy}(0)$.

Constraint: $-1.0 \leq$ **r0** ≤ 1.0.

2: **r**[**nl**] – const double *Input*

On entry: the cross-correlations between the two series at lags 1 to L, $r_{xy}(l)$, for $l = 1, 2, \ldots, L$.

Constraint: $-1.0 \leq$ **r**$[i] \leq 1.0$ for $i = 0, 1, \ldots,$ **nl** $- 1$.

3: **nl** – Integer *Input*

On entry: the number of lagged cross-correlations, L, in the array **r**.

Constraint: **nl** $\geq \max($**transfv.nag_b** + **transfv.nag_q** + **transfv.nag_p**$, 1)$.

4: **transfv** – Nag_TransfOrder *Input*

Note: transfv is a NAG defined structure. See Section 2.2.1.1 of the Essential Introduction.

On entry: The orders of the transfer function model where the triplet (**transfv.nag_b**, **transfv.nag_q**, **transfv.nag_p**) corresponds to the triplet (b, q, p) as described in Section 2.3.1 of the g13 Chapter Introduction.

Constraints:

transfv.nag_b ≥ 0;
transfv.nag_q ≥ 0;
transfv.nag_p ≥ 0.

5: **s** – double *Input*

On entry: the ratio of the standard deviation of the y series to that of the x series, s_y/s_x.

Constraint: **s** > 0.0.

6: **wds**$[dim]$ – double *Output*

Note: the dimension, dim, of the array **wds** must be at least **transfv.nag_q** + **transfv.nag_p** + 1.

On exit: the preliminary estimates of the parameters of the transfer function model in the order of $q + 1$ MA-like parameters followed by the p AR-like parameters. If the estimation of either type of parameter fails then these parameters are set to 0.0.

7: **isf**$[2]$ – Integer *Output*

On exit: indicators of the success of the estimation of MA-like and AR-like parameters respectively. A value 0 indicates that there are no parameters of that type to be estimated. A value of 1 or -1 indicates that there are parameters of that type in the model and the estimation of that type has been successful or unsuccessful respectively. Note that there is always at least one MA-like parameter in the model.

8: **fail** – NagError * *Input/Output*

The NAG error parameter (see the Essential Introduction).

6 Error Indicators and Warnings

NE_INT_4

On entry, **nl** $= \langle value \rangle$, **transfv.nag_b** $= \langle value \rangle$, **transfv.nag_q** $= \langle value \rangle$, **transfv.nag_p** $= \langle value \rangle$.
Constraint: **nl** $\geq \max($**transfv.nag_b** + **transfv.nag_q** + **transfv.nag_p**$, 1)$.

NE_CONSTRAINT

General constraint: **transfv.nag_b** ≥ 0 and

General constraint: **transfv.nag_q** ≥ 0 and

General constraint: **transfv.nag_p** ≥ 0.

NE_REAL

On entry, **s** $= \langle value \rangle$.
Constraint: **s** > 0.

On entry, **r0** lies outside $[-1.0, 1.0]$: **r0** $= \langle value \rangle$.

NE_REAL_ARRAY_ELEM_CONS

On entry, $\mathbf{r}[I-1]$ lies outside $[-1.0, 1.0]$: $I = \langle value \rangle$, $\mathbf{r}[I-1] = \langle value \rangle$.

NE_ALLOC_FAIL

Memory allocation failed.

NE_BAD_PARAM

On entry, parameter $\langle value \rangle$ had an illegal value.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

7 Accuracy

Equations used in the computations may become unstable, in which case results are reset to zero with array **isf** values set accordingly.

8 Further Comments

The time taken by the routine is roughly proportional to $(\mathbf{transfv.nag\_q} + \mathbf{transfv.nag\_p} + 1)^3$.

9 Example

The example program reads the cross-correlations between 2 series at lags 0 to 6. It then reads a (3,2,1) transfer function model and calculates and prints the preliminary estimates of the parameters of the model.

9.1 Program Text

```
/* nag_tsa_transf_prelim_fit (g13bdc) Example Program.
 *
 * Copyright 2001 Numerical Algorithms Group.
 *
 * Mark 7, 2001.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg13.h>

int main(void)
{
  /* Scalars */
  double r0, s;
  Integer exit_status, i, iwa, nl, nwds;

  /* Arrays */
  double *r=0, *wa=0, *wds=0;
  Integer isf[3];
  Nag_TransfOrder transfv;
  NagError fail;

  INIT_FAIL(fail);
  exit_status = 0;

  Vprintf("g13bdc Example Program Results\n");

  /* Skip heading in data file */
```

```
  Vscanf("%*[^\n] ");

  Vscanf("%ld%*[^\n] ", &nl);
  Vscanf("%lf%*[^\n] ", &r0);

  if (nl > 0)
    {
      /* Allocate array r */
      if ( !(r = NAG_ALLOC(nl, double)))
        {
          Vprintf("Allocation failure\n");
          exit_status = -1;
          goto END;
        }

      for (i = 1; i <= nl; ++i)
        Vscanf("%lf", &r[i-1]);
      Vscanf("%*[^\n] ");

      Vscanf("%ld %ld %ld", &transfv.nag_b, &transfv.nag_q, &transfv.nag_p);
      Vscanf("%lf%*[^\n] ", &s);

      nwds = transfv.nag_q + transfv.nag_p + 1;
      iwa = transfv.nag_p * (transfv.nag_p + 1);

      /* Allocate arrays wa and wds */
      if ( !(wa = NAG_ALLOC(iwa, double)) ||
           !(wds = NAG_ALLOC(nwds, double)))
        {
          Vprintf("Allocation failure\n");
          exit_status = -1;
          goto END;
        }

      g13bdc(r0, r, nl, &transfv, s, wds, isf, &fail);
      if (fail.code != NE_NOERROR)
        {
          Vprintf("Error from g13bdc.\n%s\n",   fail.message);
          exit_status = 1;
          goto END;
        }

      Vprintf("\n");
      Vprintf("Success/failure indicator%4ld%4ld\n", isf[0], isf[1]);
      Vprintf("\n");
      Vprintf("Transfer function model B, Q, P =");

      Vprintf("%4ld %4ld %4ld\n",
              transfv.nag_b, transfv.nag_q, transfv.nag_p);

      Vprintf("\n");
      Vprintf("Parameter initial estimates\n");

      for (i = 1; i <= nwds; ++i)
        Vprintf("%10.4f", wds[i-1]);
      Vprintf("\n");
    }

 END:
  if (r) NAG_FREE(r);
  if (wa) NAG_FREE(wa);
  if (wds) NAG_FREE(wds);

  return exit_status;
}
```

9.2 Program Data

```
g13bdc Example Program Data
     6
 -0.0155
  0.0339 -0.0374 -0.2895 -0.3430 -0.4518 -0.2787
     3     2     1
  1.9256
```

9.3 Program Results

```
g13bdc Example Program Results

Success/failure indicator   1   1

Transfer function model B, Q, P =   3    2    1

Parameter initial estimates
   -0.5575    0.3166    0.4626    0.6169
```