NAG Toolbox

nag_tsa_uni_arima_estim (g13ae)

1 Purpose

nag_tsa_uni_arima_estim (g13ae) fits a seasonal autoregressive integrated moving average (ARIMA) model to an observed time series, using a nonlinear least squares procedure incorporating backforecasting. Parameter estimates are obtained, together with appropriate standard errors. The residual series is returned, and information for use in forecasting the time series is produced for use by the functions nag_tsa_uni_arima_update (g13ag) and nag_tsa_uni_arima_forecast_state (g13ah).

The estimation procedure is iterative, starting with initial parameter values such as may be obtained using nag_tsa_uni_arima_prelim (g13ad). It continues until a specified convergence criterion is satisfied, or until a specified number of iterations has been carried out. The progress of the procedure can be monitored by means of a user-supplied function.

2 Syntax

```
[par, c, icount, ex, exr, al, s, g, sd, h, st, nst, itc, zsp, isf, ifail] =
nag_tsa_uni_arima_estim(mr, par, c, x, iex, igh, ist, piv, kpiv, zsp, kzsp,
'npar', npar, 'kfc', kfc, 'nx', nx, 'nit', nit)
```

```
[par, c, icount, ex, exr, al, s, g, sd, h, st, nst, itc, zsp, isf, ifail] = g13ae
(mr, par, c, x, iex, igh, ist, piv, kpiv, zsp, kzsp, 'npar', npar, 'kfc', kfc,
'nx', nx, 'nit', nit)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 23: **nit** and **kfc** were made optional.

3 Description

The time series $x_1, x_2, \ldots, x_n$ supplied to nag_tsa_uni_arima_estim (g13ae) is assumed to follow a seasonal autoregressive integrated moving average (ARIMA) model defined as follows:

$$\nabla^d \nabla_s^D x_t - c = w_t,$$

where $\nabla^d \nabla_s^D x_t$ is the result of applying non-seasonal differencing of order d and seasonal differencing of seasonality s and order D to the series x_t, as outlined in the description of nag_tsa_uni_diff (g13aa). The differenced series is then of length $N = n - d'$, where $d' = d + (D \times s)$ is the generalized order of differencing. The scalar c is the expected value of the differenced series, and the series $w_1, w_2, \ldots, w_N$ follows a zero-mean stationary autoregressive moving average (ARMA) model defined by a pair of recurrence equations. These express w_t in terms of an uncorrelated series a_t, via an intermediate series e_t. The first equation describes the seasonal structure:

$$w_t = \Phi_1 w_{t-s} + \Phi_2 w_{t-2\times s} + \cdots + \Phi_P w_{t-P\times s} + e_t - \Theta_1 e_{t-s} - \Theta_2 e_{t-2\times s} - \cdots - \Theta_Q e_{t-Q\times s}.$$

The second equation describes the non-seasonal structure. If the model is purely non-seasonal the first equation is redundant and e_t above is equated with w_t:

$$e_t = \phi_1 e_{t-1} + \phi_2 e_{t-2} + \cdots + \phi_p e_{t-p} + a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}.$$

Estimates of the model parameters defined by

$$\begin{array}{c}\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q, \\ \Phi_1, \Phi_2, \ldots, \Phi_P, \Theta_1, \Theta_2, \ldots, \Theta_Q\end{array}$$

and (optionally) c are obtained by minimizing a quadratic form in the vector $w = (w_1, w_2, \ldots, w_N)'$.

This is $QF = w'V^{-1}w$, where V is the covariance matrix of w, and is a function of the model parameters. This matrix is not explicitly evaluated, since QF may be expressed as a 'sum of squares' function. When moving average parameters θ_i or Θ_i are present, so that the generalized moving average order $q' = q + s \times Q$ is positive, backforecasts $w_{1-q'}, w_{2-q'}, \ldots, w_0$ are introduced as nuisance parameters. The 'sum of squares' function may then be written as

$$S(pm) = \sum_{t=1-q'}^{N} a_t^2 - \sum_{t=1-q'-p'}^{-q'} b_t^2,$$

where pm is a combined vector of parameters, consisting of the backforecasts followed by the ARMA model parameters.

The terms a_t correspond to the ARMA model residual series a_t, and $p' = p + s \times P$ is the generalized autoregressive order. The terms b_t are only present if autoregressive parameters are in the model, and serve to correct for transient errors introduced at the start of the autoregression.

The equations defining a_t and b_t are precisely:

$$e_t = w_t - \Phi_1 w_{t-s} - \Phi_2 w_{t-2\times s} - \cdots - \Phi_P w_{t-P\times s} + \Theta_1 e_{t-s} + \Theta_2 e_{t-2\times s} + \cdots + \Theta_Q e_{t-Q\times s},$$
for $t = 1-q', 2-q', \ldots, n$.

$$a_t = e_t - \phi_1 e_{t-1} - \phi_2 e_{t-2} - \cdots - \phi_p e_{t-p} + \theta_1 a_{t-1} + \theta_2 a_{t-2} + \cdots + \theta_q a_{t-q},$$
for $t = 1-q', 2-q', \ldots, n$.

$$f_t = w_t - \Phi_1 w_{t+s} - \Phi_2 w_{t+2\times s} - \cdots - \Phi_P w_{t+P\times s} + \Theta_1 f_{t-s} + \Theta_2 f_{t-2\times s} + \cdots + \Theta_Q f_{t-Q\times s},$$
for $t = (1-q'-s\times P), (2-q'-s\times P), \ldots, (-q'+P)$

$$b_t = f_t - \phi_1 f_{t+1} - \phi_2 f_{t+2} - \cdots - \phi_p f_{t+p} + \theta_1 b_{t-1} + \theta_2 b_{t-2} + \cdots + \theta_q b_{t-q},$$
for $t = (1-q'-p'), (2-q'-p'), \ldots, (-q')$.

For all four of these equations, the following conditions hold:

$w_i = 0$ if $i < 1 - q'$

$e_i = 0$ if $i < 1 - q'$

$a_i = 0$ if $i < 1 - q'$

$f_i = 0$ if $i < 1 - q' - s \times P$

$b_i = 0$ if $i < 1 - q' - p'$

Minimization of S with respect to pm uses an extension of the algorithm of Marquardt (1963).

The first derivatives of S with respect to the parameters are calculated as

$$2 \times \sum a_t \times a_{t,i} - 2 \sum b_t \times b_{t,i} = 2 \times G_i,$$

where $a_{t,i}$ and $b_{t,i}$ are derivatives of a_t and b_t with respect to the ith parameter.

The second derivative of S is approximated by

$$2 \times \sum a_{t,i} \times a_{t,j} - 2 \times \sum b_{t,i} \times b_{t,j} = 2 \times H_{ij}.$$

Successive parameter iterates are obtained by calculating a vector of corrections dpm by solving the equations

$$(H + \alpha \times D) \times dpm = -G,$$

where G is a vector with elements G_i, H is a matrix with elements H_{ij}, α is a scalar used to control the search and D is the diagonal matrix of H.

The new parameter values are then $pm + dpm$.

The scalar α controls the step size, to which it is inversely related.

If a step results in new parameter values which give a reduced value of S, then α is reduced by a factor β. If a step results in new parameter values which give an increased value of S, or in ARMA model parameters which in any way contravene the stationarity and invertibility conditions, then the new parameters are rejected, α is increased by the factor β, and the revised equations are solved for a new parameter correction.

This action is repeated until either a reduced value of S is obtained, or α reaches the limit of 10^9, which is used to indicate a failure of the search procedure.

This failure may be due to a badly conditioned sum of squares function or to too strict a convergence criterion. Convergence is deemed to have occurred if the fractional reduction in the residual sum of squares in successive iterations is less than a value γ, while $\alpha < 1.0$.

The stationarity and invertibility conditions are tested to within a specified tolerance multiple δ of machine accuracy. Upon convergence, or completion of the specified maximum number of iterations without convergence, statistical properties of the estimates are derived. In the latter case the sequence of iterates should be checked to ensure that convergence is adequate for practical purposes, otherwise these properties are not reliable.

The estimated residual variance is

$$erv = S_{\mathrm{min}} / df,$$

where S_{min} is the final value of S, and the residual number of degrees of freedom is given by

$$df = N - p - q - P - Q \quad (-1 \text{ if } c \text{ is estimated}).$$

The covariance matrix of the vector of estimates pm is given by

$$erv \times H^{-1},$$

where H is evaluated at the final parameter values.

From this expression are derived the vector of standard deviations, and the correlation matrix for the whole parameter set. These are asymptotic approximations.

The differenced series w_t (now uncorrected for the constant), intermediate series e_t and residual series a_t are all available upon completion of the iterations over the range (extended by backforecasts)

$$t = 1 - q', 2 - q', \ldots, N.$$

The values a_t can only properly be interpreted as residuals for $t \geq 1 + p' - q'$, as the earlier values are corrupted by transients if $p' > 0$.

In consequence of the manner in which differencing is implemented, the residual a_t is the one step ahead forecast error for $x_{t+d'}$.

For convenient application in forecasting, the following quantities constitute the 'state set', which contains the minimum amount of time series information needed to construct forecasts:

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

(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 $(N - \max(p, Q \times s)) < t \leq N$,

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

This state set is available upon completion of the iterations. The function may be used purely for the construction of this state set, given a previously estimated model and time series x_t, by requesting zero iterations. Backforecasts are estimated, but the model parameter values are unchanged. If later observations become available and it is desired to update the state set, nag_tsa_uni_arima_update (g13ag) can be used.

4 References

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

Marquardt D W (1963) An algorithm for least squares estimation of nonlinear parameters *J. Soc. Indust. Appl. Math.* **11** 431

5 Parameters

5.1 Compulsory Input Parameters

1: **mr**(**7**) – INTEGER array

The orders vector (p, d, q, P, D, Q, s) of the ARIMA model whose parameters are to be estimated. p, q, P and Q refer respectively to the number of autoregressive (ϕ), moving average (θ), seasonal autoregressive (Φ) and seasonal moving average (Θ) parameters. d, D and s refer respectively to the order of non-seasonal differencing, the order of seasonal differencing and the seasonal period.

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 $s > 1,\ P + D + Q > 0$;
$d + s \times (P + D) \leq n$;
$p + d - q + s \times (P + D - Q) \leq n$.

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

The initial 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 that order.

3: **c** – REAL (KIND=nag_wp)

If **kfc** $= 0$, **c** must contain the expected value, c, of the differenced series.

If **kfc** $= 1$, **c** must contain an initial estimate of c.

4: **x**(**nx**) – REAL (KIND=nag_wp) array

The n values of the original undifferenced time series.

5: **iex** – INTEGER

The dimension of the arrays **ex**, **exr** and **al**.

Constraint: **iex** $\geq q + (Q \times s) + n$, which is equivalent to the exit value of **icount**(4).

6: **igh** – INTEGER

The dimension of the arrays **g** and **sd** and the second dimension of the arrays **h** and **hc**.

Constraint: **igh** $\geq q + (Q \times s) +$ **npar** $+$ **kfc** which is equivalent to the exit value of **icount**(6).

7: **ist** – INTEGER

The dimension of the array **st**.

Constraint: **ist** $\geq (P \times s) + d + (D \times s) + q + \max(p, Q \times s)$.

8: **piv** – SUBROUTINE, supplied by the NAG Library or the user.

piv is used to monitor the progress of the optimization.

```
piv(mr, par, npar, c, kfc, icount, s, g, h, ldh, igh, itc, zsp)
```

Input Parameters

1: **mr**(**7**) – INTEGER array
2: **par**(**npar**) – REAL (KIND=nag_wp) array
3: **npar** – INTEGER
4: **c** – REAL (KIND=nag_wp)
5: **kfc** – INTEGER
6: **icount**(**6**) – INTEGER array
7: **s** – REAL (KIND=nag_wp)
8: **g**(**igh**) – REAL (KIND=nag_wp) array
9: **h**(*ldh*, **igh**) – REAL (KIND=nag_wp) array
10: **ldh** – INTEGER
11: **igh** – INTEGER
12: **itc** – INTEGER
13: **zsp**(**4**) – REAL (KIND=nag_wp) array

All the arguments are defined as for nag_tsa_uni_arima_estim (g13ae) itself.

If **kpiv** $= 0$ a dummy **piv** must be supplied.

9: **kpiv** – INTEGER

Must be nonzero if the progress of the optimization is to be monitored using **piv**. Otherwise **kpiv** must contain 0.

10: **zsp**(**4**) – REAL (KIND=nag_wp) array

When **kzsp** $= 1$, the first four elements of **zsp** must contain the four values used to guide the search procedure. These are as follows.

zsp(1) contains α, the value used to constrain the magnitude of the search procedure steps.

zsp(2) contains β, the multiplier which regulates the value α.

zsp(3) contains δ, the value of the stationarity and invertibility test tolerance factor.

zsp(4) contains γ, the value of the convergence criterion.

If **kzsp** $\neq 1$ on entry, default values for **zsp** are supplied by the function.

These are 0.001, 10.0, 1000.0 and $\max(100 \times \textit{\textbf{machine precision}}, 0.0000001)$ respectively.

Constraint: if **kzsp** $= 1$, **zsp**$(1) > 0.0$, **zsp**$(2) > 1.0$, **zsp**$(3) \geq 1.0$, $0 \leq$ **zsp**$(4) < 1.0$.

11: **kzsp** – INTEGER

The value 1 if the function is to use the input values of **zsp**, and any other value if the default values of **zsp** are to be used.

5.2 Optional Input Parameters

1: **npar** – INTEGER

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

The total number of ϕ, θ, Φ and Θ parameters to be estimated.

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

2: **kfc** – INTEGER

Default: 1

Must be set to 1 if the constant, c, is to be estimated and 0 if it is to be held fixed at its initial value.

Constraint: **kfc** $= 0$ or 1.

3: **nx** – INTEGER

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

n, the length of the original undifferenced time series.

4: **nit** – INTEGER

Default: 100

The maximum number of iterations to be performed.

Constraint: **nit** ≥ 0.

5.3 Output Parameters

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

The latest values of the estimates of these parameters.

2: **c** – REAL (KIND=nag_wp)

If **kfc** $= 0$, **c** is unchanged.

If **kfc** $= 1$, **c** contains the latest estimate of c.

Therefore, if **c** and **kfc** are both zero on entry, there is no constant correction.

3: **icount**(**6**) – INTEGER array

Size of various output arrays.

icount(1)
Contains $q + (Q \times s)$, the number of backforecasts.

icount(2)
Contains $n - d - (D \times s)$, the number of differenced values.

icount(3)
Contains $d + (D \times s)$, the number of values of reconstitution information.

icount(4)
Contains $n + q + (Q \times s)$, the number of values held in each of the series **ex**, **exr** and **al**.

icount(5)
Contains $n - d - (D \times s) - p - q - P - Q -$ **kfc**, the number of degrees of freedom associated with S.

icount(6)
Contains **icount**(1) + **npar** + **kfc**, the number of parameters being estimated.

These values are always computed regardless of the exit value of **ifail**.

4: **ex**(**iex**) – REAL (KIND=nag_wp) array

The extended differenced series which is made up of:

icount(1) backforecast values of the differenced series.

icount(2) actual values of the differenced series.

icount(3) values of reconstitution information.

The total number of these values held in **ex** is **icount**(4).

If the function exits because of a faulty input argument, the contents of **ex** will be indeterminate.

5: **exr**(**iex**) – REAL (KIND=nag_wp) array

The values of the model residuals which is made up of:

icount(1) residuals corresponding to the backforecasts in the differenced series.

icount(2) residuals corresponding to the actual values in the differenced series.

The remaining **icount**(3) values contain zeros.

If the function exits with **ifail** holding a value other than 0 or 9, the contents of **exr** will be indeterminate.

6: **al**(**iex**) – REAL (KIND=nag_wp) array

The intermediate series which is made up of:

icount(1) intermediate series values corresponding to the backforecasts in the differenced series.

icount(2) intermediate series values corresponding to the actual values in the differenced series.

The remaining **icount**(3) values contain zeros.

If the function exits with **ifail** $\neq 0$, the contents of **al** will be indeterminate.

7: **s** – REAL (KIND=nag_wp)

The residual sum of squares after the latest series of parameter estimates has been incorporated into the model. If the function exits with a faulty input argument, **s** contains zero.

8: **g**(**igh**) – REAL (KIND=nag_wp) array

The latest value of the derivatives of S with respect to each of the parameters being estimated (backforecasts, **par** parameters, and where relevant the constant – in that order). The contents of **g** will be indeterminate if the function exits with a faulty input argument.

9: **sd**(**igh**) – REAL (KIND=nag_wp) array

The standard deviations corresponding to each of the parameters being estimated (backforecasts, **par** parameters, and where relevant the constant, in that order).

If the function exits with **ifail** containing a value other than 0 or 9, or if the required number of iterations is zero, the contents of **sd** will be indeterminate.

10: **h**(ldh, **igh**) – REAL (KIND=nag_wp) array

The second derivative of S and correlation coefficients.

(a) the latest values of an approximation to the second derivative of S with respect to each of the $(q + Q \times s + \mathbf{npar} + \mathbf{kfc})$ parameters being estimated (backforecasts, **par** parameters, and where relevant the constant – in that order), and

(b) the correlation coefficients relating to each pair of these parameters.

These are held in a matrix defined by the first $(q + Q \times s + \mathbf{npar} + \mathbf{kfc})$ rows and the first $(q + Q \times s + \mathbf{npar} + \mathbf{kfc})$ columns of **h**. (Note that **icount**(6) contains the value of this expression.) The values of (a) are contained in the upper triangle, and the values of (b) in the strictly lower triangle.

These correlation coefficients are zero during intermediate printout using **piv**, and indeterminate if **ifail** contains on exit a value other than 0 or 9.

All the contents of **h** are indeterminate if the required number of iterations are zero. The $(q + (Q \times s) + \mathbf{npar} + \mathbf{kfc} + 1)$th row of **h** is used internally as workspace.

11: **st**(**ist**) – REAL (KIND=nag_wp) array

The **nst** values of the state set array. If the function exits with **ifail** containing a value other than 0 or 9, the contents of **st** will be indeterminate.

12: **nst** – INTEGER

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

13: **itc** – INTEGER

The number of iterations performed.

14: **zsp**(**4**) – REAL (KIND=nag_wp) array

zsp contains the values, default or otherwise, used by the function.

15: **isf**(**4**) – INTEGER array

Contains success/failure indicators, one for each of the four types of parameter in the model (autoregressive, moving average, seasonal autoregressive, seasonal moving average), in that order.

Each indicator has the interpretation:

-2 On entry parameters of this type have initial estimates which do not satisfy the stationarity or invertibility test conditions.

-1 The search procedure has failed to converge because the latest set of parameter estimates of this type is invalid.

0 No parameter of this type is in the model.

1 Valid final estimates for parameters of this type have been obtained.

16: **ifail** – INTEGER

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

6 Error Indicators and Warnings

Note: nag_tsa_uni_arima_estim (g13ae) may return useful information for one or more of the following detected errors or 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 it against the constraints in Section 5),
or **kfc** $\neq 0$ or 1.

ifail $= 2$

On entry, **nx** $- d - D \times s \leq$ **npar** + **kfc**, i.e., the number of terms in the differenced series is not greater than the number of parameters in the model. The model is over-parameterised.

ifail $= 3$

On entry, one or more of the user-supplied criteria for controlling the iterative process are invalid,
or **nit** < 0,
or if **kzsp** $= 1$, **zsp**$(1) \leq 0.0$;
or if **kzsp** $= 1$, **zsp**$(2) \leq 1.0$;
or if **kzsp** $= 1$, **zsp**$(3) < 1.0$;

or if **kzsp** $= 1$, **zsp**$(4) < 0.0$;
or if **kzsp** $= 1$, **zsp**$(4) \geq 1.0$.

ifail $= 4$

On entry, the state set array **st** is too small. The output value of **nst** contains the required value (see the description of **ist** in Section 5 for the formula).

ifail $= 5$

On entry, the workspace array wa is too small. Check the value of iwa against the constraints in Section 5.

ifail $= 6$ (*warning*)

On entry, **iex** $< q + (Q \times s) +$ **nx**,
or **igh** $< q + (Q \times s) +$ **npar** $+$ **kfc**,
or $ldh \leq q + (Q \times s) +$ **npar** $+$ **kfc**.

ifail $= 7$ (*warning*)

This indicates a failure in the search procedure, with **zsp**$(1) \geq 1.0\text{e}09$.

Some output arguments may contain meaningful values; see Section 5 for details.

ifail $= 8$ (*warning*)

This indicates a failure to invert H.

Some output arguments may contain meaningful values; see Section 5 for details.

ifail $= 9$ (*warning*)

This indicates a failure in nag_linsys_real_posdef_solve_1rhs (f04as) which is used to solve the equations giving the latest estimates of the backforecasts.

Some output arguments may contain meaningful values; see Section 5 for details.

ifail $= 10$ (*warning*)

Satisfactory parameter estimates could not be obtained for all parameter types in the model. Inspect array **isf** for further information on the parameter type(s) in error.

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_estim (g13ae) is approximately proportional to **nx** $\times$ **itc** $\times$ $(q + Q \times s +$ **npar** $+$ **kfc**$)^2$.

9 Example

The following program reads 30 observations from a time series relating to the rate of the earth's rotation about its polar axis. Differencing of order 1 is applied, and the number of non-seasonal parameters is 3, one autoregressive (ϕ), and two moving average (θ). No seasonal effects are taken into account.

The constant is estimated. Up to 25 iterations are allowed.

The initial estimates of ϕ_1, θ_1, θ_2 and c are zero.

9.1 Program Text

```
     function g13ae_example

fprintf('g13ae example results\n\n');

% Data
x = [-217; -177; -166; -136; -110;  -95;  -64;  -37;  -14;  -25;
      -51;  -62;  -73;  -88; -113; -120;  -83;  -33;  -19;   21;
       17;   44;   44;   78;   88;  122;  126;  114;   85;   64];
nx = numel(x);

% Orders
mr = [nag_int(1);1;2;0;0;0;0];

% parameters and sizes
npar = mr(1) + mr(3) + mr(4) + mr(6);
par = zeros(npar,1);
iex = mr(3) + (mr(6)*mr(7)) + nx;
igh = mr(3) + (mr(6)*mr(7)) + npar + 1;
ist = (mr(4)+mr(5))*mr(7) + mr(2) + mr(3) + max(mr(1),(mr(6)*mr(7)));

% Control parameters
c = 0;
kpiv = nag_int(0);
zsp = [0.001; 10; 1000; 0.0001];
kzsp = nag_int(1);

% Fit ARIMA model
[par, c, icount, ex, exr, al, s, g, sd, h, st, nst, itc, zsp, isf, ifail] = ...
  g13ae( ...
         mr, par, c, x, iex, igh, ist, @piv, kpiv, zsp, kzsp);

% Display results
nex = icount(4);
ndf = icount(5);
ngh = icount(6);
fprintf('Convergence was achieved after %4d cycles\n\n',itc);
fprintf('Final values of par array and the constant c are as follows\n');
fprintf('%10.4f', par, c);
fprintf('\n\nResidual sum of squares is %10.3f with %4d %s\n\n', ...
        s, ndf, 'degrees of freedom');
fprintf('The final values of ZSP were\n');
fprintf('%15.4e', zsp);
fprintf('\n\nThe number of parameters estimated was %4d\n', ngh);
fprintf('( backward forecasts, par and c, in that order )\n\n');
fprintf('The corresponding G array holds\n');
fprintf('%9.4f', g);
if  itc>0
  fprintf('\n\nThe corresponding SD array holds\n');
  fprintf('%9.4f', sd);
  fprintf('\n\n')
  [ifail] = x04ca( ...
                   'General', ' ', h(1:ngh,:), 'Corresponding H matrix');

  fprintf('\nHolds second derivatives in the upper half ');
  fprintf('(including the main diagonal)\n');
  fprintf('and correlation coefficients in the lower triangle\n');
```

```
end
fprintf('\n%s%5d%s\n','EX, EXR, and AL each hold', nex, ...
        ' values made up of');
fprintf('%5d%s\n', icount(1), ' back forecast(s),');
fprintf('%5d%s\n', icount(2), ' differenced values, and');
fprintf('%5d%s\n\n', icount(3), ' element(s) of reconstituted information');
fprintf('  Ex\n');
for j = 1:5:nex
  fprintf('%11.5f', ex(j:min(j+4,nex)));
  fprintf('\n');
end
fprintf('\n  Exr\n');
for j = 1:5:nex
  fprintf('%11.5f', exr(j:min(j+4,nex)));
  fprintf('\n');
end
fprintf('\n  Al\n');
for j = 1:5:nex
  fprintf('%11.5f', al(j:min(j+4,nex)));
  fprintf('\n');
end
fprintf('\nThe state set consists of %4d values\n',nst);
fprintf('%11.5f', st);
fprintf('\n');

function [] = piv(mr, par, npar, c, kfc, icount, s, g, h, ih, igh, itc, zsp)

  fprintf('Iteration %d  residual sum f squares = %16.4', itc, s);
```

9.2 Program Results

```
     g13ae example results

Convergence was achieved after   16 cycles

Final values of par array and the constant c are as follows
   -0.0547   -0.5568   -0.6636    9.9807

Residual sum of squares is   9397.924 with   25 degrees of freedom

The final values of ZSP were
     1.0000e-15     1.0000e+01     1.0000e+03     1.0000e-04

The number of parameters estimated was    6
( backward forecasts, par and c, in that order )

The corresponding G array holds
  -0.1512  -0.2343  -6.4097  13.5617 -72.6232  -0.1642

The corresponding SD array holds
  14.8379  15.1887   0.3507   0.2709   0.1695   7.3893

 Corresponding H matrix
             1            2            3            4            5
 1    1.9416E+00  -6.1794E-01   2.4409E-01   1.7942E+00  -8.3579E-01
 2    3.4176E-01   1.9446E+00  -1.6544E-01  -2.5084E-01   1.7952E+00
 3   -1.0544E-02   5.5643E-03   9.0416E+03  -9.6825E+03   5.4626E+02
 4   -1.2113E-02   5.6011E-03   8.1322E-01   1.7031E+04  -5.6761E+03
 5   -2.3216E-03  -1.1495E-03   3.6741E-01   4.7942E-01   1.7028E+04
 6   -1.4580E-01  -2.6004E-01  -4.0877E-02  -4.8389E-02  -3.7442E-02

             6
 1    2.4106E-01
 2    8.5926E-01
 3    8.1847E-01
 4    6.9417E+00
 5    6.3308E+00
 6    7.4339E+00

Holds second derivatives in the upper half (including the main diagonal)
```

```
and correlation coefficients in the lower triangle

EX, EXR, and AL each hold   32 values made up of
    2 back forecast(s),
   29 differenced values, and
    1 element(s) of reconstituted information

  Ex
   19.52500    5.87533   40.00000   11.00000   30.00000
   26.00000   15.00000   31.00000   27.00000   23.00000
  -11.00000  -26.00000  -11.00000  -11.00000  -15.00000
  -25.00000   -7.00000   37.00000   50.00000   14.00000
   40.00000   -4.00000   27.00000    0.00000   34.00000
   10.00000   34.00000    4.00000  -12.00000  -29.00000
  -21.00000   64.00000

  Exr
   19.52500   -3.92787   19.57110   -5.62907   10.22209
   15.15821   -9.32757   16.42850   15.21154   -5.42106
  -27.34437  -18.30612    5.38901  -12.98124  -22.47672
  -15.21833    4.49436   33.68668   19.75860  -27.14696
   32.24262  -12.27651    1.69412   -1.84650   23.37721
  -10.45763   14.33018   -5.70614  -28.64010  -20.45020
   -2.72147    0.00000

  Al
   19.52500    5.87533   30.01926    1.01926   20.01926
   16.01926    5.01926   21.01926   17.01926   13.01926
  -20.98074  -35.98074  -20.98074  -20.98074  -24.98074
  -34.98074  -16.98074   27.01926   40.01926    4.01926
   30.01926  -13.98074   17.01926   -9.98074   24.01926
    0.01926   24.01926   -5.98074  -21.98074  -38.98074
  -30.98074    0.00000

The state set consists of    4 values
   64.00000  -30.98074  -20.45020   -2.72147
```