

NAG Library Routine Document

G13AEF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

G13AEF 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 routines G13AGF and G13AHF.

The estimation procedure is iterative, starting with initial parameter values such as may be obtained using G13ADF. 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 routine.

2 Specification

```

SUBROUTINE G13AEF (MR, PAR, NPAR, C, KFC, X, NX, ICOUNT, EX, EXR, AL,      &
                  IEX, S, G, IGH, SD, H, LDH, ST, IST, NST, PIV, KPIV,    &
                  NIT, ITC, ZSP, KZSP, ISF, WA, IWA, HC, IFAIL)
INTEGER          MR(7), NPAR, KFC, NX, ICOUNT(6), IEX, IGH, LDH, IST,  &
                NST, KPIV, NIT, ITC, KZSP, ISF(4), IWA, IFAIL
REAL (KIND=nag_wp) PAR(NPAR), C, X(NX), EX(IEX), EXR(IEX), AL(IEX), S,  &
                G(IGH), SD(IGH), H(LDH,IGH), ST(IST), ZSP(4),          &
                WA(IWA), HC(LDH,IGH)
EXTERNAL        PIV

```

3 Description

The time series x_1, x_2, \dots, x_n supplied to G13AEF 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 G13AAF. 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, \dots, 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:

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

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 :

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

Estimates of the model parameters defined by

$$\phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q, \\ \Phi_1, \Phi_2, \dots, \Phi_P, \Theta_1, \Theta_2, \dots, \Theta_Q$$

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

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 $w_{1-q'}, w_{2-q'}, \dots, w_0$ 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:

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

for $t = 1 - q', 2 - q', \dots, n$.

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

for $t = 1 - q', 2 - q', \dots, n$.

$$f_t = w_t - \Phi_1 w_{t+s} - \Phi_2 w_{t+2 \times s} - \dots - \Phi_P w_{t+P \times s} + \Theta_1 f_{t-s} + \Theta_2 f_{t-2 \times s} + \dots + \Theta_Q f_{t-Q \times s},$$

for $t = (1 - q' - s \times P), (2 - q' - s \times P), \dots, (-q' + P)$

$$b_t = f_t - \phi_1 f_{t+1} - \phi_2 f_{t+2} - \dots - \phi_p f_{t+p} + \theta_1 b_{t-1} + \theta_2 b_{t-2} + \dots + \theta_q b_{t-q},$$

for $t = (1 - q' - p'), (2 - q' - p'), \dots, (-q')$.

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

$$w_i = 0 \text{ if } i < 1 - q'$$

$$e_i = 0 \text{ if } i < 1 - q'$$

$$a_i = 0 \text{ if } i < 1 - q'$$

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

$$b_i = 0 \text{ if } i < 1 - q' - p'$$

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 i th 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

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

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)

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

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 routine 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, G13AGF 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 Arguments

- 1: MR(7) – INTEGER array *Input*
- On entry:* 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:*
- $$\begin{aligned} p, d, q, P, D, Q, s &\geq 0; \\ p + q + P + Q &> 0; \\ s &\neq 1; \\ \text{if } s = 0, P + D + Q &= 0; \\ \text{if } s > 1, P + D + Q &> 0; \\ d + s \times (P + D) &\leq n; \\ p + d - q + s \times (P + D - Q) &\leq n. \end{aligned}$$
- 2: PAR(NPAR) – REAL (KIND=nag_wp) array *Input/Output*
- On entry:* 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.
- On exit:* the latest values of the estimates of these parameters.
- 3: NPAR – INTEGER *Input*
- On entry:* the total number of ϕ, θ, Φ and Θ parameters to be estimated.
- Constraint:* $\text{NPAR} = p + q + P + Q$.
- 4: C – REAL (KIND=nag_wp) *Input/Output*
- On entry:* if $\text{KFC} = 0$, C must contain the expected value, c , of the differenced series.
- If $\text{KFC} = 1$, C must contain an initial estimate of c .
- On exit:* if $\text{KFC} = 0$, C is unchanged.
- If $\text{KFC} = 1$, C contains the latest estimate of c .
- Therefore, if C and KFC are both zero on entry, there is no constant correction.
- 5: KFC – INTEGER *Input*
- On entry:* 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:* $\text{KFC} = 0$ or 1.
- 6: X(NX) – REAL (KIND=nag_wp) array *Input*
- On entry:* the n values of the original undifferenced time series.
- 7: NX – INTEGER *Input*
- On entry:* n , the length of the original undifferenced time series.
- 8: ICOUNT(6) – INTEGER array *Output*
- On exit:* 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 $n - d - (D \times s) - p - q - P - Q - \text{KFC}$, the number of degrees of freedom associated with S .

ICOUNT(6)

Contains $\text{ICOUNT}(1) + \text{NPAR} + \text{KFC}$, the number of parameters being estimated.

These values are always computed regardless of the exit value of IFAIL.

9: EX(IEX) – REAL (KIND=nag_wp) array *Output*

On exit: 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 routine exits because of a faulty input argument, the contents of EX will be indeterminate.

10: EXR(IEX) – REAL (KIND=nag_wp) array *Output*

On exit: 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 routine exits with IFAIL holding a value other than 0 or 9, the contents of EXR will be indeterminate.

11: AL(IEX) – REAL (KIND=nag_wp) array *Output*

On exit: 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 routine exits with IFAIL $\neq 0$, the contents of AL will be indeterminate.

12: IEX – INTEGER *Input*

On entry: the dimension of the arrays EX, EXR and AL as declared in the (sub)program from which G13AEF is called.

Constraint: $\text{IEX} \geq q + (Q \times s) + n$, which is equivalent to the exit value of ICOUNT(4).

- 13: S – REAL (KIND=nag_wp) Output
On exit: the residual sum of squares after the latest series of parameter estimates has been incorporated into the model. If the routine exits with a faulty input argument, S contains zero.
- 14: G(IGH) – REAL (KIND=nag_wp) array Output
On exit: 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 routine exits with a faulty input argument.
- 15: IGH – INTEGER Input
On entry: the dimension of the arrays G and SD and the second dimension of the arrays H and HC as declared in the (sub)program from which G13AEF is called.
Constraint: $IGH \geq q + (Q \times s) + NPAR + KFC$ which is equivalent to the exit value of ICOUNT(6).
- 16: SD(IGH) – REAL (KIND=nag_wp) array Output
On exit: the standard deviations corresponding to each of the parameters being estimated (backforecasts, PAR parameters, and where relevant the constant, in that order).
 If the routine 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.
- 17: H(LDH, IGH) – REAL (KIND=nag_wp) array Output
On exit: 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 $(q + Q \times s + NPAR + KFC)$ 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 $(q + Q \times s + NPAR + KFC)$ rows and the first $(q + Q \times s + NPAR + KFC)$ 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 $(q + (Q \times s) + NPAR + KFC + 1)$ th row of H is used internally as workspace.
- 18: LDH – INTEGER Input
On entry: the first dimension of the arrays H and HC as declared in the (sub)program from which G13AEF is called.
Constraint: $LDH \geq 1 + q + (Q \times s) + NPAR + KFC$, which is equivalent to the exit value of ICOUNT(6).
- 19: ST(IST) – REAL (KIND=nag_wp) array Output
On exit: the NST values of the state set array. If the routine exits with IFAIL containing a value other than 0 or 9, the contents of ST will be indeterminate.

- 20: IST – INTEGER *Input*
On entry: the dimension of the array ST as declared in the (sub)program from which G13AEF is called.
Constraint: $IST \geq (P \times s) + d + (D \times s) + q + \max(p, Q \times s)$.
- 21: NST – INTEGER *Output*
On exit: the number of values in the state set array ST.
- 22: PIV – SUBROUTINE, supplied by the NAG Library or the user. *External Procedure*
 PIV is used to monitor the progress of the optimization.

The specification of PIV is:

```
SUBROUTINE PIV (MR, PAR, NPAR, C, KFC, ICOUNT, S, G, H, LDH, IGH, &
                ITC, ZSP)
INTEGER          MR(7), NPAR, KFC, ICOUNT(6), LDH, IGH, ITC
REAL (KIND=nag_wp) PAR(NPAR), C, S, G(IGH), H(LDH,IGH), ZSP(4)
```

PIV is called on each iteration by G13AEF when the input value of KPIV is nonzero and is bypassed when it is 0.

The routine G13AFZ may be used as PIV. It prints the heading

```
G13AFZ MONITORING OUTPUT - ITERATION n
```

followed by the parameter values and the residual sum of squares. Output is directed to the advisory channel defined by X04ABF.

- | | | |
|-----|--|--------------|
| 1: | MR(7) – INTEGER array | <i>Input</i> |
| 2: | PAR(NPAR) – REAL (KIND=nag_wp) array | <i>Input</i> |
| 3: | NPAR – INTEGER | <i>Input</i> |
| 4: | C – REAL (KIND=nag_wp) | <i>Input</i> |
| 5: | KFC – INTEGER | <i>Input</i> |
| 6: | ICOUNT(6) – INTEGER array | <i>Input</i> |
| 7: | S – REAL (KIND=nag_wp) | <i>Input</i> |
| 8: | G(IGH) – REAL (KIND=nag_wp) array | <i>Input</i> |
| 9: | H(LDH, IGH) – REAL (KIND=nag_wp) array | <i>Input</i> |
| 10: | LDH – INTEGER | <i>Input</i> |
| 11: | IGH – INTEGER | <i>Input</i> |
| 12: | ITC – INTEGER | <i>Input</i> |
| 13: | ZSP(4) – REAL (KIND=nag_wp) array | <i>Input</i> |

On entry: all the arguments are defined as for G13AEF itself.

PIV must either be a module subprogram USED by, or declared as EXTERNAL in, the (sub) program from which G13AEF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

If KPIV = 0 a dummy PIV must be supplied.

- 23: KPIV – INTEGER *Input*
On entry: must be nonzero if the progress of the optimization is to be monitored using PIV. Otherwise KPIV must contain 0.
- 24: NIT – INTEGER *Input*
On entry: the maximum number of iterations to be performed.
Constraint: $NIT \geq 0$.

- 25: ITC – INTEGER *Output*
On exit: the number of iterations performed.
- 26: ZSP(4) – REAL (KIND=nag_wp) array *Input/Output*
On entry: 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 routine.
 These are 0.001, 10.0, 1000.0 and $\max(100 \times \textit{machine precision}, 0.0000001)$ respectively.
On exit: ZSP contains the values, default or otherwise, used by the routine.
Constraint: if KZSP = 1, ZSP(1) > 0.0, ZSP(2) > 1.0, ZSP(3) \geq 1.0, $0 \leq$ ZSP(4) < 1.0.
- 27: KZSP – INTEGER *Input*
On entry: the value 1 if the routine is to use the input values of ZSP, and any other value if the default values of ZSP are to be used.
- 28: ISF(4) – INTEGER array *Output*
On exit: 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.
- 29: WA(IWA) – REAL (KIND=nag_wp) array *Workspace*
 30: IWA – INTEGER *Input*
On entry: the dimension of the array WA as declared in the (sub)program from which G13AEF is called.
Constraint: $IWA \geq (F_1 \times F_2) + (9 \times NPAR)$,
 where $F_1 = NX + 1 + p + (P \times s) + q + (Q \times s)$;
 and $F_2 = 8$ if KFC = 1;
 $F_2 = 7$ if KFC = 0, $Q > 0$;
 $F_2 = 6$ if KFC = 0, $Q = 0$, $P > 0$;
 $F_2 = 5$ if KFC = 0, $Q = 0$, $P = 0$, $q > 0$;
 $F_2 = 4$ otherwise.

31: HC(LDH,IGH) – REAL (KIND=nag_wp) array Workspace

32: IFAIL – INTEGER Input/Output

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, because for this routine the values of the output arguments may be useful even if IFAIL \neq 0 on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Note: G13AEF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

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

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

This indicates a failure in the search procedure, with $ZSP(1) \geq 1.0E09$.

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

IFAIL = 8

This indicates a failure to invert H .

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

IFAIL = 9

Unable to calculate the latest estimates of the backforecasts.

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

IFAIL = 10

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.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -399

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The computations are believed to be stable.

8 Parallelism and Performance

G13AEF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G13AEF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The time taken by G13AEF is approximately proportional to $NX \times ITC \times (q + Q \times s + NPAR + KFC)^2$.

10 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.

10.1 Program Text

```
! G13AEF Example Program Text
! Mark 26 Release. NAG Copyright 2016.

Module g13aefe_mod

! G13AEF Example Program Module:
! Parameters and User-defined Routines

! .. Use Statements ..
Use nag_library, Only: nag_wp
! .. Implicit None Statement ..
Implicit None
! .. Accessibility Statements ..
Private
Public
! .. Parameters ..
Integer, Parameter, Public :: nin = 5, nout = 6
Contains
Subroutine piv(mr,par,npar,c,kfc,icount,s,g,h,ldh,igh,itc,zsp)

! .. Parameters ..
Integer, Parameter :: param_inds(4) = (/1,3,4,6/)
! .. Scalar Arguments ..
Real (Kind=nag_wp), Intent (In) :: c, s
Integer, Intent (In) :: igh, itc, kfc, ldh, npar
! .. Array Arguments ..
Real (Kind=nag_wp), Intent (In) :: g(igh), h(ldh,igh), par(npar), &
zsp(4)
Integer, Intent (In) :: icount(6), mr(7)
! .. Local Scalars ..
Integer :: i, param, start_ind
! .. Executable Statements ..
Write (nout,*)

If (itc==0) Then
  Write (nout,*) 'Begin monitoring output'
  Write (nout,99999) 'Degrees of freedom: ', icount(5)
End If

Write (nout,99999) 'Iteration: ', itc

param_loop: Do i = 1, 4
  param = mr(param_inds(i))
  If (param<=0) Then
    Cycle param_loop
  End If
  Select Case (param_inds(i))
  Case (1)
    Write (nout,*) 'Autoregressive parameters:'
    start_ind = 1
  Case (3)
    Write (nout,*) 'Moving-average parameters:'
  Case (4)
    Write (nout,*) 'Seasonal autoregressive parameters:'
  Case (6)
    Write (nout,*) 'Seasonal moving-average parameters:'
```

```

      End Select
      Write (nout,99998) par(start_ind:start_ind+param-1)
      start_ind = start_ind + param
    End Do param_loop

    If (kfc==1) Then
      Write (nout,*) 'Constant term:'
      Write (nout,99998) c
    End If

    Write (nout,*) 'Residual sum of squares:'
    Write (nout,99998) s

    Write (nout,*)
      'Parameters (alpha, beta, delta, gamma) controlling the search ', &
      'procedure:' &
    Write (nout,99998) zsp

    Return
99999  Format (1X,A,I3)
99998  Format (1X,E13.4)
    End Subroutine piv
  End Module g13aeefe_mod
  Program g13aeefe

!      G13AEF Example Main Program

!      .. Use Statements ..
  Use nag_library, Only: g13aeef, nag_wp, x04caf
  Use g13aeefe_mod, Only: nin, nout, piv
!      .. Implicit None Statement ..
  Implicit None
!      .. Local Scalars ..
  Real (Kind=nag_wp)          :: c, s
  Integer                    :: iex, ifail, igh, ist, itc, iwa, kfc, &
                             kpiv, kzsp, ldh, ndf, nex, ngh, nit, &
                             npar, nst, nx
!      .. Local Arrays ..
  Real (Kind=nag_wp), Allocatable :: al(:), ex(:), exr(:), g(:), h(:,,:), &
                             hc(:,,:), par(:), sd(:), st(:), &
                             wa(:), x(:)
  Real (Kind=nag_wp)          :: zsp(4)
  Integer                    :: icount(6), isf(4), mr(7)
!      .. Intrinsic Procedures ..
  Intrinsic                  :: max
!      .. Executable Statements ..
  Write (nout,*) 'G13AEF Example Program Results'
  Write (nout,*)

!      Skip heading in data file
  Read (nin,*)

!      Read in the problem size etc
  Read (nin,*) nx, kfc, c

!      Read in the orders
  Read (nin,*) mr(1:7)

!      Calculate NPAR and various array lengths
  npar = mr(1) + mr(3) + mr(4) + mr(6)
  iex = mr(3) + (mr(6)*mr(7)) + nx
  igh = mr(3) + (mr(6)*mr(7)) + npar + kfc
  ist = (mr(4)*mr(7)) + mr(2) + (mr(5)*mr(7)) + mr(3) + &
        max(mr(1), (mr(6)*mr(7)))
  iwa = ((nx+1+mr(1)+(mr(4)*mr(7))+mr(3)+(mr(6)*mr(7)))*8) + (9*npar)

  ldh = igh + 1
  Allocate (x(nx),par(npar),ex(iex),exr(iex),al(iex),g(igh),sd(igh), &
           h(ldh,igh),st(ist),wa(iwa),hc(ldh,igh))

!      Read in data

```

```

Read (nin,*) x(1:nx)

!   Read in initial values
Read (nin,*) par(1:npar)

!   Read in control parameters
Read (nin,*) kpiv, nit, kzsp
If (kzsp==1) Then
  Read (nin,*) zsp(1:4)
End If

!   Fit ARIMA model
ifail = -1
Call g13aef(mr,par,npar,c,kfc,x,nx,icount,ex,exr,al,iex,s,g,igh,sd,h,    &
  ldh,st,ist,nst,piv,kpiv,nit,itc,zsp,kzsp,isf,wa,iwa,hc,ifail)
If (ifail/=0) Then
  If (ifail<7) Then
    Go To 100
  End If
End If

!   Display results
nex = icount(4)
ndf = icount(5)
ngh = icount(6)
If (ifail==0) Then
  Write (nout,99998) 'Convergence was achieved after', itc, ' cycles'
Else
  Write (nout,99998) 'Iterative process ran for', itc, ' cycles'
End If
Write (nout,*)
Write (nout,*)
'Final values of the PAR parameters and the constant are as follows'    &
Write (nout,99997) par(1:npar), c
Write (nout,*)
Write (nout,99996) 'Residual sum of squares is', s, ' with', ndf,    &
  ' degrees of freedom'
Write (nout,*)
Write (nout,*) 'The final values of ZSP were'
Write (nout,99995) zsp(1:4)
Write (nout,*)
Write (nout,99999) 'The number of parameters estimated was', ngh
Write (nout,*) '( backward forecasts, PAR and C, in that order )'
Write (nout,*)
Write (nout,*) 'The corresponding G array holds'
Write (nout,99994) g(1:ngh)
If ((ifail==0 .Or. ifail==9) .And. itc>0) Then
  Write (nout,*)
  Write (nout,*) 'The corresponding SD array holds'
  Write (nout,99994) sd(1:ngh)
  Write (nout,*)
  Flush (nout)
  ifail = 0
  Call x04caf('General',' ',ngh,ngh,h,ldh,'Corresponding H matrix',    &
    ifail)
  Write (nout,*)
  'Holds second derivatives in the upper half (including the main ',    &
  'diagonal)'
  Write (nout,*) 'and correlation coefficients in the lower triangle'
End If
Write (nout,*)
Write (nout,99993) 'EX, EXR, and AL each hold', nex,    &
  ' values made up of', icount(1), ' back forecast(s),'
Write (nout,99992) icount(2), ' differenced values, and'
Write (nout,99992) icount(3), ' element(s) of reconstituted information'
Write (nout,*)
Write (nout,*) ' EX'
Write (nout,99991) ex(1:nex)
If (ifail==0 .Or. ifail==9) Then
  Write (nout,*)
  Write (nout,*) ' EXR'

```

```

      Write (nout,99991) exr(1:nex)
    End If
    If (ifail==0) Then
      Write (nout,*)
      Write (nout,*) ' AL'
      Write (nout,99991) al(1:nex)
    End If
    If (ifail==0 .Or. ifail==9) Then
      Write (nout,*)
      Write (nout,99998) 'The state set consists of', nst, ' values'
      Write (nout,99991) st(1:nst)
    End If

100  Continue

99999 Format (1X,A,I5)
99998 Format (1X,A,I3,A)
99997 Format (1X,4F10.4)
99996 Format (1X,A,F10.3,A,I4,A)
99995 Format (1X,4E15.4)
99994 Format (1X,10F9.4)
99993 Format (1X,A,I5,A,I5,A)
99992 Format (1X,I5,A)
99991 Format (1X,5F11.4)
      End Program g13aeffe

```

10.2 Program Data

G13AEF Example Program Data

```

30 1 0.0                :: NX,KFC,C
 1 1 2 0 0 0 0         :: MR
-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    :: X
0.0 0.0 0.0           :: PAR (initial values)
0 25 1                :: KPIV,NIT,KZSP
0.001 10.0 1000.0 0.0001 :: ZSP

```

10.3 Program Results

G13AEF Example Program Results

Convergence was achieved after 16 cycles

Final values of the PAR parameters and the constant 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
 0.1000E-14 0.1000E+02 0.1000E+04 0.1000E-03

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

19.5250	5.8753	40.0000	11.0000	30.0000
26.0000	15.0000	31.0000	27.0000	23.0000
-11.0000	-26.0000	-11.0000	-11.0000	-15.0000
-25.0000	-7.0000	37.0000	50.0000	14.0000
40.0000	-4.0000	27.0000	0.0000	34.0000
10.0000	34.0000	4.0000	-12.0000	-29.0000
-21.0000	64.0000			

EXR

19.5250	-3.9279	19.5711	-5.6291	10.2221
15.1582	-9.3276	16.4285	15.2115	-5.4211
-27.3444	-18.3061	5.3890	-12.9812	-22.4767
-15.2183	4.4944	33.6867	19.7586	-27.1470
32.2426	-12.2765	1.6941	-1.8465	23.3772
-10.4576	14.3302	-5.7061	-28.6401	-20.4502
-2.7215	0.0000			

AL

19.5250	5.8753	30.0193	1.0193	20.0193
16.0193	5.0193	21.0193	17.0193	13.0193
-20.9807	-35.9807	-20.9807	-20.9807	-24.9807
-34.9807	-16.9807	27.0193	40.0193	4.0193
30.0193	-13.9807	17.0193	-9.9807	24.0193
0.0193	24.0193	-5.9807	-21.9807	-38.9807
-30.9807	0.0000			

The state set consists of 4 values

64.0000	-30.9807	-20.4502	-2.7215
---------	----------	----------	---------
