NAG Toolbox for MATLAB

g13be

1 Purpose

g13be fits a multi-input model relating one output series to the input series with a choice of three different estimation criteria: nonlinear least-squares, exact likelihood and marginal likelihood. When no input series are present, g13be fits a univariate ARIMA model.

2 Syntax

```
[para, xxy, zsp, itc, sd, cm, s, d, ndf, res, sttf, nsttf, ifail] =
g13be(mr, mt, para, kfc, nxxy, xxy, kef, nit, kzsp, zsp, kzef, isttf,
iwa, imwa, kpriv, 'nser', nser, 'npara', npara)
```

3 Description

3.1 The Multi-input Model

The output series y_t, for $t = 1, 2, \ldots, n$, is assumed to be the sum of (unobserved) components $z_{i,t}$ which are due respectively to the inputs $x_{i,t}$, for $i = 1, 2, \ldots, m$.

Thus $y_t = z_{1,t} + \cdots + z_{m,t} + n_t$ where n_t is the error, or output noise component.

A typical component z_t may be either

(a) a simple regression component, $z_t = \omega x_t$ (here x_t is called a simple input), or

(b) a transfer function model component which allows for the effect of lagged values of the variable, related to x_t by

$$z_t = \delta_1 z_{t-1} + \delta_2 z_{t-2} + \cdots + \delta_p z_{t-p} + \omega_0 x_{t-b} - \omega_1 x_{t-b-1} - \cdots - \omega_q x_{t-b-q}.$$

The noise n_t is assumed to follow a (possibly seasonal) ARIMA model, i.e., may be represented in terms of an uncorrelated series, a_t, by the hierarchy of equations

(i) $\nabla^d \nabla_s^D n_t = c + w_t$

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

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

as outlined in Section 3 of the document for g13ae.

Note: the orders p, q appearing in each of the transfer function models and the ARIMA model are not necessarily the same; $\nabla^d \nabla_s^D n_t$ is the result of applying non-seasonal differencing of order d and seasonal differencing of seasonality s and order D to the series n_t: the differenced series is then of length $N = n - d - s \times D$; the constant term parameter c may optionally be held fixed at its initial value (usually, but not necessarily zero) rather than being estimated.

For the purpose of defining an estimation criterion it is assumed that the series a_t is a sequence of independent Normal variates having mean 0 and variance σ_a^2. An allowance has to be made for the effects of unobserved data prior to the observation period. For the noise component an allowance is always made using a form of backforecasting.

For each transfer function input, you have to decide what values are to be assumed for the pre-period terms $z_0, z_{-1}, \ldots, z_{1-p}$ and $x_0, x_{-1}, \ldots, x_{1-b-q}$ which are in theory necessary to re-create the component series $z_1, z_2, \ldots, z_n$, during the estimation procedure.

The first choice is to assume that all these values are zero. In this case, in order to avoid undesirable transient distortion of the early values $z_1, z_2, \ldots,$ you are advised first to correct the input series x_t by subtracting from all the terms a suitable constant to make the early values $x_1, x_2, \ldots,$ close to zero. The series mean $\bar{x}$ is one possibility, but for a series with strong trend the constant might be simply x_1.

The second choice is to treat the unknown pre-period terms as nuisance parameters and estimate them along with the other parameters. This choice should be used with caution. For example, if $p = 1$ and $b = q = 0$, it is equivalent to fitting to the data a decaying geometric curve of the form $A\delta^t$, for $t = 1, 2, 3, \ldots$, along with the other inputs, this being the form of the transient. If the output y_t contains a strong trend of this form, which is not otherwise represented in the model, it will have a tendency to influence the estimate of δ away from the value appropriate to the transfer function model.

In most applications the first choice should be adequate, with the option possibly being used as a refinement at the end of the modelling process. The number of nuisance parameters is then $\max(p, b + q)$, with a corresponding loss of degrees of freedom in the residuals. If you align the input x_t with the output by using in its place the shifted series x_{t-b}, then setting $b = 0$ in the transfer function model, there is some improvement in efficiency. On some occasions when the model contains two or more inputs, each with estimation of pre-period nuisance parameters, these parameters may be co-linear and lead to failure of the function. The option must then be 'switched off' for one or more inputs.

3.2 The Estimation Criterion

This is a measure of how well a proposed set of parameters in the transfer function and noise ARIMA models matches the data. The estimation function searches for parameter values which minimize this criterion. For a proposed set of parameter values it is derived by calculating

(i) the components $z_{1,t}, z_{2,t}, \ldots, z_{m,t}$ as the responses to the input series $x_{1,t}, x_{2,t} \ldots, x_{m,t}$ using the equations or above,

(ii) the discrepancy between the output and the sum of these components, as the noise

$$n_t = y_t - \left(z_{1,t} + z_{2,t} + \cdots + z_{m,t}\right),$$

(iii) the residual series a_t from n_t by reversing the recursive equations (c), (d) and (e) above.

This last step again requires treatment of the effect of unknown pre-period values of n_t and other terms in the equations regenerating a_t. This is identical to the treatment given in Section 3 of the document for g13ae, and leads to a criterion which is a sum of squares function S, of the residuals a_t. It may be shown that the finite algorithm presented there is equivalent to taking the infinite set of past values $n_0, n_{-1}, n_{-2}, \ldots$, as (linear) nuisance parameters. There is no loss of degrees of freedom however, because the sum of squares function S may be expressed as including the corresponding set of past residuals; see page 273 of Box and Jenkins 1976, who prove that

$$S = \sum_{-\infty}^{n} a_t^2.$$

The function $D = S$ is the first of the three possible criteria, and is quite adequate for moderate to long series with no seasonal parameters. The second is the exact likelihood criterion which considers the past set n_0, n_{-1}, n_{-2} not as simple nuisance parameters, but as unobserved random variables with known distribution. Calculation of the likelihood of the observed set $n_1, n_2, \ldots, n_n$ requires theoretical integration over the range of the past set. Fortunately this yields a criterion of the form $D = M \times S$ (whose minimization is equivalent to maximizing the exact likelihood of the data), where S is exactly as before, and the multiplier M is a function calculated from the ARIMA model parameters. The value of M is always ≥ 1, and M tends to 1 for any fixed parameter set as the sample size n tends to ∞. There is a moderate computational overhead in using this option, but its use avoids appreciable bias in the ARIMA model parameters and yields a better conditioned estimation problem.

The third criterion of marginal likelihood treats the coefficients of the simple inputs in a manner analogous to that given to the past set $n_0, n_{-1}, n_{-2}, \ldots$. These coefficients, together with the constant term c used to represent the mean of w_t, are in effect treated as random variables with highly dispersed distributions. This leads to the criterion $D = M \times S$ again, but with a different value of M which now depends on the simple input series values x_t. In the presence of a moderate to large number of simple inputs, the marginal likelihood criterion can counteract bias in the ARIMA model parameters which is caused by estimation of the simple inputs. This is particularly important in relatively short series.

g13be can be used with no input series present, to estimate a univariate ARIMA model for the output alone. The marginal likelihood criterion is then distinct from exact likelihood only if a constant term is being estimated in the model, because this is treated as an implicit simple input.

3.3 The Estimation Procedure

This is the minimization of the estimation criterion or objective function D (for deviance). The function uses an extension of the algorithm of Marquardt 1963. The step size in the minimization is inversely related to a parameter α, which is increased or decreased by a factor β at successive iterations, depending on the progress of the minimization. Convergence is deemed to have occurred if the fractional reduction of D in successive iterations is less than a value γ, while $\alpha < 1$.

Certain model parameters (in fact all excluding the ωs) are subject to stability constraints which are checked throughout to within a specified tolerance multiple δ of machine accuracy. Using the least-squares criterion, the minimization may halt prematurely when some parameters 'stick' at a constraint boundary. This can happen particularly with short seasonal series (with a small number of whole seasons). It will not happen using the exact likelihood criterion, although convergence to a point on the boundary may sometimes be rather slow, because the criterion function may be very flat in such a region. There is also a smaller risk of a premature halt at a constraint boundary when marginal likelihood is used.

A positive, or zero number of iterations can be specified. In either case, the value D of the objective function at iteration zero is presented at the initial parameter values, except for estimation of any pre-period terms for the input series, backforecasts for the noise series, and the coefficients of any simple inputs, and the constant term (unless this is held fixed).

At any later iteration, the value of D is supplied after re-estimation of the backforecasts to their optimal values, corresponding to the model parameters presented at that iteration. This is not true for any pre-period terms for the input series which, although they are updated from the previous iteration, may not be precisely optimal for the parameter values presented, unless convergence of those parameters has occurred. However, in the case of marginal likelihood being specified, the coefficients of the simple inputs and the constant term are also re-estimated together with the backforecasts at each iteration, to values which are optimal for the other parameter values presented.

3.4 Further Results

The residual variance is taken as $erv = \frac{S}{df}$, where $df = N -$ (total number of parameters estimated), is the residual degrees of freedom. The pre-period nuisance parameters for the input series are included in the reduction of df, as is the constant if it is estimated.

The covariance matrix of the vector of model parameter estimates is given by

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

where H is the linearized least-squares matrix taken from the final iteration of the algorithm of Marquardt. From this expression are derived the vector of standard deviations, and the correlation matrix of parameter estimates. These are approximations which are only valid asymptotically, and must be treated with great caution when the parameter estimates are close to their constraint boundaries.

The residual series a_t is available upon completion of the iterations over the range $t = 1 + d + s \times D, \ldots, n$ corresponding to the differenced noise series w_t.

Because of the algorithm used for backforecasting, these are only true residuals for $t \geq 1 + q + s \times Q - p - s \times P - d - s \times D$, provided this is positive. Estimation of pre-period terms for the inputs will also tend to reduce the magnitude of the early residuals, sometimes severely.

The model component series $z_{1,t}, \ldots, z_{m,t}$ and n_t may optionally be returned in place of the supplied series values, in order to assess the effects of the various inputs on the output.

3.5 Forecasting Information

For the purpose of constructing forecasts of the output series at future time points $t = n + 1, n + 2, \ldots$ using g13bh, it is not necessary to use the whole set of observations y_t and $x_{1,t}, x_{2,t}, \ldots, x_{m,t}$, for $t = 1, 2, \ldots, m$. It is sufficient to retain a limited set of quantities constituting the 'state set' as follows: for each series which appears with lagged subscripts in equations , , , and above, include the values at times $n + 1 - k$ for $k = 1$ up to the maximum lag associated with that series in the equations. Note that implicitly includes past values of n_t and intermediate differences of n_t such as $\nabla^{d-1}\nabla_s^D$.

If later observations of the series become available, it is possible to update the state set (without re-estimating the model) using g13bg. If time series data is supplied with a previously estimated model, it is possible to construct the state set (and forecasts) using g13bj.

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) – int32 array**

The orders vector (p, d, q, P, D, Q, s) of the ARIMA model for the output noise component.

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: **mt(4,nser) – int32 array**

The transfer function model orders b, p and q of each of the input series. The order parameters for input series i are held in column i. Row 1 holds the value b_i, row 2 holds the value q_i and row 3 holds the value p_i. For a simple input, $b_i = q_i = p_i = 0$.

Row 4 holds the value r_i, where $r_i = 1$ for a simple input, $r_i = 2$ for a transfer function input for which no allowance is to be made for pre-observation period effects, and $r_i = 3$ for a transfer function input for which pre-observation period effects will be treated by estimation of appropriate nuisance parameters.

When $r_i = 1$, any nonzero contents of rows 1, 2, and 3 of column i are ignored.

3: **para(npara) – double array**

Initial values of the multi-input model parameters. These are in order, firstly the ARIMA model parameters: p values of ϕ parameters, q values of θ parameters, P values of Φ parameters and Q values of Θ parameters. These are followed by initial values of the transfer function model parameters $\omega_0, \omega_1, \ldots, \omega_{q_1}$, $\delta_1, \delta_2, \ldots, \delta_{p_1}$ for the first of any input series and similarly for each subsequent input series. The final component of **para** is the initial value of the constant c, whether it is fixed or is to be estimated.

4: **kfc – int32 scalar**

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

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

5: **nxxy – int32 scalar**

the (common) length of the original, undifferenced input and output time series.

6: **xxy(ldxxy,nser) – double array**

ldxxy, the first dimension of the array, must be at least **nxxy**.

The columns of **xxy** must contain the **nxxy** original, undifferenced values of each of the input series and the output series in that order.

7: **kef – int32 scalar**

Indicates the likelihood option.

kef $= 1$

Gives least-squares.

kef $= 2$

Gives exact likelihood.

kef $= 3$

Gives marginal likelihood.

Constraint: **kef** $= 1$, 2 or 3.

8: **nit – int32 scalar**

The maximum required number of iterations.

nit $= 0$

No change is made to any of the model parameters in array **para** except that the constant c (if **kfc** $= 1$) and any ω relating to simple input series are estimated. (Apart from these, estimates are always derived for the nuisance parameters relating to any backforecasts and any pre-observation period effects for transfer function inputs.)

Constraint: **nit** ≥ 0.

9: **kzsp – int32 scalar**

Must be set to 1 if the function is to use the input values of **zsp** in the minimization procedure, and to any other value if the default values of **zsp** are to be used.

10: **zsp(4) – double array**

If **kzsp** $= 1$, then **zsp** must contain the four values used to control the strategy of the search procedure.

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 of 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** $\neq 1$ before entry, default values of **zsp** are supplied by the function. These are 0.01, 10.0, 1000.0 and $\max(100 \times$ ***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: **kzef – int32 scalar**

Must not be set to 0, if the values of the input component series z_t and the values of the output noise component n_t are to overwrite the contents of **xxy** on exit, and must be set to 0 if **xxy** is to remain unchanged.

12: **isttf – int32 scalar**

Constraint: **isttf** $\geq (P \times s) + d + (D \times s) + q + \max(p, Q \times s) + ncg$, where $ncg = \sum(b_i + q_i + p_i)$ over all input series for which $r_i > 1$.

13: **iwa – int32 scalar**

It is not practical to outline a method for deriving the exact minimum permissible value of **iwa**, but the following gives a reasonably good conservative approximation. (It should be noted that if **iwa** is too small (but not grossly so) then the exact minimum is returned in **mwa**(i) and is also printed if **kpriv** $\neq 0$.)

Let $q' = q + (Q \times s)$ and $d' = d + (D \times s)$ where the orders of the output noise model are p, d, q, P, D, Q, s.

Let there be l input series, where $l =$ **nser** $- 1$.

Let

$$\begin{array}{ll} mx_i = \max(b_i + q_i, p_i), & \text{if } r_i = 3, \text{ for } i = 1, 2 \ldots, l \\ mx_i = 0, & \text{if } r_i \neq 3, \text{ for } i = 1, 2 \ldots, l \end{array}$$

where the transfer function model orders for input i are given by b_i, q_i, p_i, r_i.

Let $qx = \max(q', mx_1, mx_2, \ldots, mx_l)$.

Let $ncd = \textbf{npara} + \textbf{kfc} + qx + \sum_{i=1}^{l} mx_i$ and $nce = \textbf{nxxy} + d' + 6 \times qx$.

Finally, let $ncf =$ **nser**, and then increment ncf by 1 every time any of the following conditions is satisfied. (The last six conditions should be applied separately to each input series, so that, for example, if we have two input series and if $p_1 > 0$ and $p_2 > 0$ then ncf is incremented by 2.)

The conditions are:

$$p > 0$$
$$q > 0$$
$$P > 0$$
$$Q > 0$$
$$qx > 0$$
$$\mathbf{kfc} > 0$$
$$\left.\begin{array}{r} p > 0 \\ q > 0 \\ P > 0 \\ Q > 0 \end{array}\right\} \text{ and } q > 0 \text{ and } \mathbf{kef} > 1.$$
$$\left.\begin{array}{r} p > 0 \\ q > 0 \\ P > 0 \\ Q > 0 \end{array}\right\} \text{ and } \mathbf{kfc} > 0 \text{ and } \mathbf{kef} = 3.$$
$$\left.\begin{array}{r} mx_i > 0 \\ p_i > 0 \\ p > 0 \\ q > 0 \\ P > 0 \\ Q > 0 \end{array}\right\} \text{ and } r_i = 1 \text{ and } \mathbf{kef} > 3 \text{ separately, for } i = 1, 2, \ldots, l.$$

Then **iwa** $\geq 2 \times (ncd)^2 + (nce) \times (ncf + 4)$.

14: **imwa – int32 scalar**

Constraint: **imwa** $\geq$ (16 × **nser**) + (7 × ncd) + (3 × **npara**) + (3 × **kfc**) + 27, where the derivation of ncd is shown under **iwa**.

If **imwa** is too small then the exact minimum needed is returned in **imwa** and if **kpriv** $\neq 0$ it is also printed

15: **kpriv – int32 scalar**

Must not be set to 0, if it is required to monitor the course of the optimization or to print out the requisite minimum values of **iwa** or **imwa** in the event of an error of the type **ifail** = 6 or 7. The course of the optimization is monitored by printing out at each iteration the iteration count (**itc**), the residual sum of squares (**s**), the objective function (**d**) and a description and value for each of the parameters in the **para** array. The descriptions are PHI for ϕ, THETA for θ, SPHI for Φ, STHETA for Θ, OMEGA/SI for ω in a simple input, OMEGA for ω in a transfer function input, DELTA for δ and CONSTANT for c. In addition SERIES 1, SERIES 2, etc. indicate the input series relevant to the OMEGA and DELTA parameters.

kpriv must be set to 0 if the print-out of the above information is not required.

5.2 Optional Input Parameters

1: **nser – int32 scalar**

Default: The dimension of the arrays **mt**, **xxy**. (An error is raised if these dimensions are not equal.)

the total number of input and output series. There may be any number of input series (including none), but always one output series.

Constraints:

nser > 1 if there are no parameters in the model (that is, $p = q = P = Q = 0$ and **kfc** $= 0$);
nser ≥ 1.

2: **npara – int32 scalar**

Default: The dimension of the arrays **para**, **sd**, **cm**. (An error is raised if these dimensions are not equal.)

the exact number of $\phi, \theta, \Phi, \Theta$, ω, δ and c parameters.

Constraint: $\textbf{npara} = p + q + P + Q + \textbf{nser} + \sum(p_i + q_i)$, the summation being over all the input series. c must be included, whether fixed or estimated.

5.3 Input Parameters Omitted from the MATLAB Interface

ldxxy, ldcm, wa, mwa

5.4 Output Parameters

1: **para(npara) – double array**

The latest values of the estimates of these parameters.

2: **xxy(ldxxy,nser) – double array**

If $\textbf{kzef} = 0$, **xxy** remains unchanged on exit.

If $\textbf{kzef} \neq 0$, the columns of **xxy** hold the corresponding values of the input component series z_t in place of x_t and the output noise component n_t in place of y_t, in that order.

3: **zsp(4) – double array**

Contains the values, default or otherwise, used by the function.

4: **itc – int32 scalar**

The number of iterations carried out.

A value of $\textbf{itc} = -1$ on exit indicates that the only estimates obtained up to this point have been for the nuisance parameters relating to backforecasts, unless the marginal likelihood option is used, in which case estimates have also been obtained for simple input coefficients ω and for the constant c (if $\textbf{kfc} = 1$). This value of **itc** usually indicates a failure in a consequent step of estimating transfer function input pre-observation period nuisance parameters.

A value of $\textbf{itc} = 0$ on exit indicates that estimates have been obtained up to this point for the constant c (if $\textbf{kfc} = 1$), for simple input coefficients ω and for the nuisance parameters relating to the backforecasts and to transfer function input pre-observation period effects.

5: **sd(npara) – double array**

The **npara** values of the standard deviations corresponding to each of the parameters in **para**. When the constant is fixed its standard deviation is returned as zero. When the values of **para** are valid, the values of **sd** are usually also valid. However, if an exit value of $\textbf{ifail} = 3$, 8 or 10 is accompanied by a failure to invert the second derivative matrix (which would normally give $\textbf{ifail} = 9$), then the contents of **sd** will be indeterminate.

6: **cm(ldcm,npara) – double array**

The first **npara** rows and columns of **cm** contain the correlation coefficients relating to each pair of parameters in **para**. All coefficients relating to the constant will be zero if the constant is fixed. The contents of **cm** will be indeterminate under the same conditions as **sd**.

7: **s – double scalar**

The residual sum of squares, S, at the latest set of valid parameter estimates.

8: **d – double scalar**

The objective function, D, at the latest set of valid parameter estimates.

9: **ndf – int32 scalar**

The number of degrees of freedom associated with S.

10: **res(nxxy) – double array**

The values of the residuals relating to the differenced values of the output series. The remainder of the first **nxxy** terms in the array will be zero.

11: **sttf(isttf) – double array**

The **nsttf** values of the state set array.

12: **nsttf – int32 scalar**

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

13: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Note: g13be may return useful information for one or more of the following detected errors or warnings.

ifail $= 1$

On entry, **kfc** < 0,
or **kfc** > 1,
or **ldxxy** $<$ **nxxy**,
or **ldcm** $<$ **npara**,
or **kef** < 1,
or **kef** > 3,
or **nit** < 0,
or **nser** < 1,
or **nser** $= 1$ and there are no parameters in the model ($p = q = P = Q = 0$ and **kfc** $= 0$).

ifail $= 2$

On entry, there is inconsistency between **npara** and **kfc** on the one hand and the orders in arrays **mr** and **mt** on the other,
or one of the r_i, stored in **mt**$(4, i) \neq 1$, 2 or 3.

ifail $= 3$

On entry or during execution, one or more sets of δ parameters do not satisfy the stationarity or invertibility test conditions.

ifail $= 4$

On entry, when **kzsp** $= 1$, **zsp**$(1) \leq 0.0$,
or **zsp**$(2) \leq 1.0$,
or **zsp**$(3) < 1.0$,
or **zsp**$(4) < 0.0$,
or **zsp**$(4) \geq 1.0$.

ifail = 5

On entry, **iwa** is too small by a considerable margin. No information is supplied about the requisite minimum size.

ifail = 6

On entry, **iwa** is too small, but the requisite minimum size is returned in **mwa**(1), which is printed if **kpriv** $\neq 0$.

ifail = 7

On entry, **imwa** is too small, but the requisite minimum size is returned in **mwa**(1), which is printed if **kpriv** $\neq 0$.

ifail = 8

This indicates a failure in f04as which is used to solve the equations giving the latest estimates of the parameters.

ifail = 9

This indicates a failure in the inversion of the second derivative matrix. This is needed in the calculation of the correlation matrix and the standard deviations of the parameter estimates.

ifail = 10

On entry or during execution, one or more sets of the ARIMA (ϕ, θ, Φ or Θ) parameters do not satisfy the stationarity or invertibility test conditions.

ifail = 11

On entry, **isttf** is too small. The state set information will not be produced and if **kzef** $\neq 0$ array **xxy** will remain unchanged. All other parameters will be produced correctly.

ifail = 12

The function has failed to converge after **nit** iterations. If steady decreases in the objective function, D, were monitored up to the point where this exit occurred, then the exit probably occurred because **nit** was set too small, so the calculations should be restarted from the final point held in **para**.

ifail = 13

On entry, **isttf** is too small (see **ifail** = 11) and **nit** iterations were carried out without the convergence conditions being satisfied (see **ifail** = 12).

7 Accuracy

The computation used is believed to be stable.

8 Further Comments

The time taken by g13be is approximately proportional to **nxxy** $\times$ **itc** $\times$ **npara**^{2}.

9 Example

```
mr = [int32(1);
      int32(0);
      int32(0);
      int32(0);
      int32(0);
```

```
      int32(1);
      int32(4)];
mt = [int32(1), int32(0);
      int32(0), int32(0);
      int32(1), int32(0);
      int32(3), int32(0)];
para = [0;
      0;
      2;
      0.5;
      0];
kfc = int32(1);
nxxy = int32(40);
xxy = [8.074999999999999, 105;
      7.819, 119;
      7.366, 119;
      8.113, 109;
      7.38, 117;
      7.134, 135;
      7.222, 126;
      7.768, 112;
      7.386, 116;
      6.965, 122;
      6.478, 115;
      8.105, 115;
      8.06, 122;
      7.684, 138;
      7.58, 135;
      7.093, 125;
      6.129, 115;
      6.026, 108;
      6.679, 100;
      7.414, 96;
      7.112, 107;
      7.762, 115;
      7.645, 123;
      8.638999999999999, 122;
      7.667, 128;
      8.08, 136;
      6.678, 140;
      6.739, 122;
      5.569, 102;
      5.049, 103;
      5.642, 89;
      6.808, 77;
      6.636, 89;
      8.241, 94;
      7.968, 104;
      8.044, 108;
      7.791, 119;
      7.024, 126;
      6.102, 119;
      6.053, 103];
kef = int32(3);
nit = int32(20);
kzsp = int32(0);
zsp = [0;
      0;
      0;
      0];
kzef = int32(1);
isttf = int32(20);
iwa = int32(1500);
imwa = int32(200);
kpriv = int32(0);
[paraOut, xxyOut, zspOut, itc, sd, cm, s, d, ndf, res, sttf, nsttf,
ifail] = ...
        g13be(mr, mt, para, kfc, nxxy, xxy, kef, nit, kzsp, zsp, kzef,
isttf, iwa, imwa, kpriv)
```