1 Purpose

nag_forecast_garchGJR (g13ffc) forecasts the conditional variances, $h_t$, $t = 1, \ldots, \tau$ from a GJR \text{GARCH}(p, q)$ sequence, where $\tau$ is the forecast horizon (see Glosten \textit{et al.} (1993)).

2 Specification

```c
#include <nag.h>
#include <nagg13.h>

void nag_forecast_garchGJR (Integer num, Integer nt, Integer p, Integer q,
       const double theta[], double gamma, double fht[], const double ht[],
       const double et[], NagError *fail)
```

3 Description

Assume the \text{GARCH}(p, q) process can be represented by:

$$
\epsilon_t | \psi_{t-1} \sim N(0, h_t)
$$

$$
h_t = \alpha_0 + \sum_{i=1}^{q} (\alpha_i + \gamma S_{t-i}) \epsilon_{t-i}^2 + \sum_{i=1}^{p} \beta_i h_{t-i}, \quad t = 1, \ldots, T.
$$

where $S_t = 1$, if $\epsilon_t < 0$, and $S_t = 0$, if $\epsilon_t \geq 0$ has been modelled by nag_estimate_garchGJR (g13fec) and the estimated conditional variances and residuals are contained in the arrays \text{ht} and \text{et} respectively. Then nag_forecast_garchGJR (g13ffc) will use the last $\max(p, q)$ elements of the arrays \text{ht} and \text{et} to estimate the conditional variance forecasts, $h_t | \psi_T$, where $t = T + 1, \ldots, T + \tau$ and $\tau$ is the forecast horizon.

4 References


5 Arguments

1: \textbf{num} – Integer

\textit{Input}

\textit{On entry:} the number of terms in the arrays \text{ht} and \text{et} from the modelled sequence.

\textit{Constraint:} $\max(p, q) \leq \text{num}$. 

Mark 25
\[ I n t e g e r \]

On entry: \( \tau \), the forecast horizon.

Constraint: \( nt > 0 \).

\[ p - I n t e g e r \]

On entry: the GARCH\((p, q)\) argument \( p \).

Constraint: \( 0 < \text{max}(p, q) \leq \text{num} \), \( p \geq 0 \).

\[ q - I n t e g e r \]

On entry: the GARCH\((p, q)\) argument \( q \).

Constraint: \( 0 < \text{max}(p, q) \leq \text{num} \), \( q \geq 1 \).

\[ \text{theta}[q+p+1] - \text{const double} \]

On entry: the first element must contain the coefficient \( \alpha_0 \), and the next \( q \) elements must contain the coefficients \( \alpha_i \), for \( i = 1, 2, \ldots, q \). The remaining \( p \) elements must contain the coefficients \( \beta_j \), for \( j = 1, 2, \ldots, p \).

\[ \text{gamma} - \text{double} \]

On entry: the asymmetry argument \( \gamma \) for the GARCH\((p, q)\) sequence.

\[ \text{fht}[nt] - \text{double} \]

On exit: the forecast values of the conditional variance, \( h_t \), for \( t = 1, 2, \ldots, \tau \).

\[ \text{ht}[\text{num}] - \text{const double} \]

On entry: the sequence of past conditional variances for the GARCH\((p, q)\) process, \( h_t \), for \( t = 1, 2, \ldots, T \).

\[ \text{et}[\text{num}] - \text{const double} \]

On entry: the sequence of past residuals for the GARCH\((p, q)\) process, \( \epsilon_t \), for \( t = 1, 2, \ldots, T \).

\[ \text{fail} - \text{NagError} * \]

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

**NE_2_INT_ARG_LT**

On entry, \( \text{num} = \langle \text{value} \rangle \) while \( \text{max}(p, q) = \langle \text{value} \rangle \). These arguments must satisfy \( \text{num} \geq \text{max}(p, q) \).

**NE_ALLOC_FAIL**

Dynamic memory allocation failed.

**NE_INT_ARG_LT**

On entry, \( nt = \langle \text{value} \rangle \).

Constraint: \( nt \geq 1 \).

On entry, \( \text{num} = \langle \text{value} \rangle \).

Constraint: \( \text{num} \geq 0 \).

On entry, \( p = \langle \text{value} \rangle \).

Constraint: \( p \geq 0 \).
On entry, $q = \langle \text{value} \rangle$.
Constraint: $q \geq 1$.

7 Accuracy
Not applicable.

8 Parallelism and Performance
Not applicable.

9 Further Comments
None.

10 Example
See the example for nag_estimate_agarchII (g13fcc).