NAG C Library Function Document

nag_generate_garchGJR (g05hmc)

1 Purpose

nag_generate_garchGJR (g05hmc) generates a given number of a GJR GARCH\((p, q)\) process (see Glosten, et al. (1993)).

2 Specification

```c
#include <nag.h>
#include <nag05.h>

void nag_generate_garchGJR (Integer num, Integer p, Integer q,
                            const double theta[], double gamma, double ht[], double et[],
                            Nag_Garch_Fcall_Type fcall, double rvec[], NagError *fail)
```

3 Description

A GJR GARCH\((p, q)\) process is 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 \(\epsilon_t \geq 0\).

Here \(T\) is the number of observations in the sequence, \(\epsilon_t\) is the observed value of the GARCH\((p, q)\) process at time \(t\), \(h_t\) is the conditional variance at time \(t\), and \(\psi_t\) the information set of all information up to time \(t\). Symmetric GARCH\((p, q)\) sequences are generated when \(\gamma\) is zero, otherwise asymmetric GARCH\((p, q)\) sequences are generated with \(\gamma\) specifying the amount by which negative shocks are to be enhanced.

4 Parameters

1: num – Integer

   On entry: the number of terms in the sequence, \(T\).

   Constraints:
   \[
   \text{num} \geq 1,
   \text{num} > p+q+1.
   \]

2: p – Integer

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

   Constraint: \(p \geq 0\).

3: q – Integer

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

   Constraint: \(q \geq 1\).
4. **theta[q+p+1]** – const double
   
   *Input*
   
   *On entry:* the first element contains the coefficient $\alpha_0$, the next $q$ elements contain the coefficients $\alpha_i$, $i = 1, \ldots, q$. The remaining $p$ elements are the coefficients $\beta_j$, $j = 1, \ldots, p$.

5. **gamma** – double
   
   *Input*
   
   *On entry:* the asymmetry parameter $\gamma$ for the GARCH($p, q$) sequence.

6. **ht[num]** – double
   
   *Output*
   
   *On exit:* the conditional variances $h_t$, $t = 1, \ldots, T$ for the GARCH($p, q$) sequence.

7. **et[num]** – double
   
   *Output*
   
   *On exit:* the observations $\epsilon_t$, $t = 1, \ldots, T$ for the GARCH($p, q$) sequence.

8. **fcall** – Nag_Garch_Fcall_Type
   
   *Input*
   
   *On entry:* if **fcall** = Nag_Garch_Fcall_True then a new sequence is to be generated, else if **fcall** = Nag_Garch_Fcall.False a given sequence is to be continued using the information in **rvec**.

9. **rvec[2*(p+q+1)]** – double
   
   *Input/Output*
   
   *On entry:* the array contains information required to continue a sequence if **fcall** = Nag_Garch_Fcall.False.
   
   *On exit:* contains information that can be used in a subsequent call of nag_generate_garchGJR, with **fcall** = Nag_Garch_Fcall.False.

10. **fail** – NagError *
    
    *Input/Output*
    
    The NAG error parameter (see the Essential Introduction).

5  **Error Indicators and Warnings**

**NE.BAD.PARAM**

On entry, parameter **fcall** had an illegal value.

On entry, parameter **gamma** had an illegal value.

**NE.INT_ARG_LT**

On entry, $p$ must not be less than 0: $p = <value>$.

On entry, $q$ must not be less than 1: $q = <value>$.

On entry, **num** must not be less than 1: **num** = <value>.

On entry, **num** = <value> while $p+q+1 = <value>$

These parameters must satisfy $\text{num} \geq p+q+1$.

6  **Further Comments**

6.1  **Accuracy**

Not applicable.
6.2 References


7 See Also

None.

8 Example

See the example for nag_estimate_garchGJR (g13fec).