NAG C Library Function Document

nag_generate_agarchII (g05hlc)

1 Purpose
nag_generate_agarchII (g05hlc) generates a given number of terms of a type II AGARCH\((p, q)\) process (see Engle and Ng (1993)).

2 Specification
#include <nag.h>
#include <nag05.h>

void nag_generate_agarchII (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 type II AGARCH\((p, q)\) process can be represented by:

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

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

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\((p, q)\) processes are generated when \(\gamma\) is zero, otherwise asymmetric GARCH\((p, q)\) sequences are generated with \(\gamma\) specifying the amount by which positive (or negative) shocks are to be enhanced.

4 Parameters

1: num – Integer
Input

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

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

2: p – Integer
Input

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

Constraint: \(p \geq 0\).

3: q – Integer
Input

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

Constraint: \(q \geq 1\).

4: theta[\text{q}+\text{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 \(\beta_j, j = 1, \ldots, p\).
gamma – double
*Input*
On entry: the asymmetry parameter \( \gamma \) for the GARCH\((p, q)\) sequence.

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

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

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.

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_agarchII, with
fcall = Nag_Garch_Fcall_False.

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.

NE_INT_ARG_LT
On entry, \( p \) must not be less than 0: \( p = \text{<value>} \).
On entry, \( q \) must not be less than 1: \( q = \text{<value>} \).
On entry, \( \text{num} \) must not be less than 1: \( \text{num} = \text{<value>} \).
On entry, \( \text{num} = \text{<value>} \) while \( p+q+1 = \text{<value>} \)
These parameters must satisfy \( \text{num} \geq p+q+1 \).

6 Further Comments

6.1 Accuracy
Not applicable.

6.2 References
Kingdom Inflation Econometrica 50 987–1008
307–327
1749–1777
7 See Also
None.

8 Example
See the example for nag_estimate_agarchII (g13fcc).