NAG C Library Function Document

nag_generate_agarchI (g05hkc)

1 Purpose

nag_generate_agarchI (g05hkc) generates a given number of terms of a type I AGARCH(p, q) process (see Engle and Ng (1993)).

2 Specification

```
#include <nag.h>
#include <nagg05.h>
void nag_generate_agarchI (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 I AGARCH(p, q) process can be represented by:

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

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

Here T is the number of observations in the sequence, ϵ_t is the *observed* value of the GARCH(p, q) process at time t, h_t is the conditional variance at time t, and ψ_t the information set of all information up to time t. Symmetric GARCH(p, q) sequences are generated when γ is zero, otherwise asymmetric GARCH(p, q) sequences are generated with γ 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:

num ≥ 1,

num $>$ **p**+**q**+1.

2: **p** – Integer *Input*

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

Constraint: **p** ≥ 0.

3: **q** – Integer *Input*

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

Constraint: **q** ≥ 1.

4: **theta[q+p+1]** – const double *Input*

On entry: the first element contains the coefficient α_o, the next **q** elements contain the coefficients α_i, $i = 1, \ldots, q$. The remaining **p** elements are the coefficients β_j, $j = 1, \ldots, p$.

5: **gamma** – double *Input*

On entry: the asymmetry parameter γ 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 ϵ_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_agarchI, 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.

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 **num** $\geq$ **p**+**q**+1.

6 Further Comments

6.1 Accuracy

Not applicable.

6.2 References

Engle R (1982) Autoregressive Conditional Heteroskedasticity with Estimates of the Variance of United Kingdom Inflation *Econometrica* **50** 987–1008

Bollerslev T (1986) Generalised Autoregressive Conditional Heteroskedasticity *Journal of Econometrics* **31** 307–327

Engle R and Ng V (1993) Measuring and Testing the Impact of News on Volatility *Journal of Finance* **48** 1749–1777

Hamilton J (1994) *Time Series Analysis* Princeton University Press

7 See Also

None.

8 Example

See the example for nag_estimate_agarchI (g13fac).