NAG Toolbox

nag_rand_times_garch_exp (g05pg)

1 Purpose

nag_rand_times_garch_exp (g05pg) generates a given number of terms of an exponential GARCH(p, q) process (see Engle and Ng (1993)).

2 Syntax

```
[ht, et, r, state, ifail] = nag_rand_times_garch_exp(dist, num, ip, iq, theta, df, fcall, r, state, 'lr', lr)
[ht, et, r, state, ifail] = g05pg(dist, num, ip, iq, theta, df, fcall, r, state, 'lr', lr)
```

3 Description

An exponential GARCH(p, q) process is represented by:

$$ln(h_t) = \alpha_0 + \sum_{i=1}^{q} \alpha_i z_{t-i} + \sum_{i=1}^{q} \phi_i (|z_{t-i}| - E[|z_{t-i}|]) + \sum_{j=1}^{p} \beta_j ln\left(h_{t-j}\right), \quad t = 1, 2, \ldots, T;$$

where $z_t = \dfrac{\epsilon_t}{\sqrt{h_t}}$, $E[|z_{t-i}|]$ denotes the expected value of $|z_{t-i}|$, and $\epsilon_t \mid \psi_{t-1} = N(0, h_t)$ or $\epsilon_t \mid \psi_{t-1} = S_t(df, h_t)$. Here S_t is a standardized Student's t-distribution with df degrees of freedom and variance h_t, 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 set of all information up to time t.

One of the initialization functions nag_rand_init_repeat (g05kf) (for a repeatable sequence if computed sequentially) or nag_rand_init_nonrepeat (g05kg) (for a non-repeatable sequence) must be called prior to the first call to nag_rand_times_garch_exp (g05pg).

4 References

Bollerslev T (1986) Generalised autoregressive conditional heteroskedasticity *Journal of Econometrics* **31** 307–327

Engle R (1982) Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation *Econometrica* **50** 987–1008

Engle R and Ng V (1993) Measuring and testing the impact of news on volatility *Journal of Finance* **48** 1749–1777

Glosten L, Jagannathan R and Runkle D (1993) Relationship between the expected value and the volatility of nominal excess return on stocks *Journal of Finance* **48** 1779–1801

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

5 Parameters

5.1 Compulsory Input Parameters

1: **dist** – CHARACTER(1)

The type of distribution to use for ϵ_t.

dist = 'N'
A Normal distribution is used.

dist = 'T'
A Student's t-distribution is used.

Constraint: **dist** = 'N' or 'T'.

2: **num** – INTEGER

T, the number of terms in the sequence.

Constraint: **num** ≥ 0.

3: **ip** – INTEGER

The number of coefficients, β_i, for $i = 1, 2, \ldots, p$.

Constraint: **ip** ≥ 0.

4: **iq** – INTEGER

The number of coefficients, α_i, for $i = 1, 2, \ldots, q$.

Constraint: **iq** ≥ 1.

5: **theta**$(\mathbf{2} \times \mathbf{iq} + \mathbf{ip} + \mathbf{1})$ – REAL (KIND=nag_wp) array

The initial parameter estimates for the vector θ. The first element must contain the coefficient α_o and the next **iq** elements must contain the autoregressive coefficients α_i, for $i = 1, 2, \ldots, q$. The next **iq** elements must contain the coefficients ϕ_i, for $i = 1, 2, \ldots, q$. The next **ip** elements must contain the moving average coefficients β_j, for $j = 1, 2, \ldots, p$.

Constraints:

$$\sum_{i=1}^{p} \beta_i \neq 1.0;$$
$$\frac{\alpha_0}{1 - \sum_{i=1}^{p} \beta_i} \leq -\log(\text{x02am}).$$

6: **df** – INTEGER

The number of degrees of freedom for the Student's t-distribution.

If **dist** = 'N', **df** is not referenced.

Constraint: if **dist** = 'T', **df** > 2.

7: **fcall** – LOGICAL

If **fcall** = *true*, a new sequence is to be generated, otherwise a given sequence is to be continued using the information in **r**.

8: **r**(**lr**) – REAL (KIND=nag_wp) array

The array contains information required to continue a sequence if **fcall** = *false*.

9: **state**(:) – INTEGER array

Note: the actual argument supplied **must** be the array **state** supplied to the initialization routines nag_rand_init_repeat (g05kf) or nag_rand_init_nonrepeat (g05kg).

Contains information on the selected base generator and its current state.

5.2 Optional Input Parameters

1: **lr** – INTEGER

Default: the dimension of the array **r**.

The dimension of the array **r**.

Constraint: **lr** $\geq 2 \times (\mathbf{ip} + 2 \times \mathbf{iq} + 2)$.

5.3 Output Parameters

1: **ht**(**num**) – REAL (KIND=nag_wp) array

The conditional variances h_t, for $t = 1, 2, \ldots, T$, for the GARCH(p, q) sequence.

2: **et**(**num**) – REAL (KIND=nag_wp) array

The observations ϵ_t, for $t = 1, 2, \ldots, T$, for the GARCH(p, q) sequence.

3: **r**(**lr**) – REAL (KIND=nag_wp) array

Contains information that can be used in a subsequent call of nag_rand_times_garch_exp (g05pg), with **fcall** = *false*.

4: **state**(:) – INTEGER array

Contains updated information on the state of the generator.

5: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **dist** is not valid.

ifail $= 2$

Constraint: **num** ≥ 0.

ifail $= 3$

Constraint: **ip** ≥ 0.

ifail $= 4$

Constraint: **iq** ≥ 1.

ifail $= 6$

Constraint: **df** ≥ 3.

ifail $= 10$

ip or **iq** is not the same as when **r** was set up in a previous call.

ifail $= 11$

On entry, **lr** is not large enough, **lr** $= \langle value \rangle$: minimum length required .

ifail $= 12$

On entry, **state** vector has been corrupted or not initialized.

ifail $= 20$

Invalid sequence generated, use different parameters.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example first calls nag_rand_init_repeat (g05kf) to initialize a base generator then calls nag_rand_times_garch_exp (g05pg) to generate two realizations, each consisting of ten observations, from an exponential GARCH$(1, 1)$ model.

9.1 Program Text

```
function g05pg_example

fprintf('g05pg example results\n\n');

% Initialize the generator to a repeatable sequence
seed = [nag_int(1762543)];
genid = nag_int(1);
subid = nag_int(1);
[state, ifail] = g05kf( ...
                        genid, subid, seed);

% Input parameters
dist  = 'N';
num   = nag_int(10);
ip    = nag_int(1);
iq    = nag_int(1);
theta = [0.1; -0.3; 0.1; 0.9];
df    = nag_int(0);
fcall = true;
r     = zeros(2*(2*ip+iq+2),1);
```

```
% Generate the first realisation
[ht, et, r, state, ifail] = g05pg( ...
                                   dist, num, ip, iq, ...
                                   theta, df, fcall, r, state);
% Display the results
fprintf('\n Realisation Number 1\n');
fprintf('   i            ht(i)            et(i)\n');
fprintf('  ------------------------------------\n');
for i=1:num
  fprintf('  %2d  %16.4f %16.4f\n', i, ht(i), et(i));
end

% Generate a second realisation
fcall = false;
[ht, et, r, state, ifail] = g05pg( ...
                                   dist, num, ip, iq, ...
                                   theta, df, fcall, r, state);
% Display the results
fprintf('\n Realisation Number 2\n');
fprintf('   i            ht(i)            et(i)\n');
fprintf('  ------------------------------------\n');
for i=1:num
  fprintf('  %2d  %16.4f %16.4f\n', i, ht(i), et(i));
end
```

9.2 Program Results

```
     g05pg example results

 Realisation Number 1
   i            ht(i)            et(i)
  ------------------------------------
   1            2.5098           0.5526
   2            2.1785          -1.8383
   3            3.3844           1.2180
   4            2.6780           1.3672
   5            2.0953          -1.8178
   6            3.2813          -0.0343
   7            2.9958          -0.5094
   8            3.0815           1.3978
   9            2.3961          -0.0070
  10            2.2445           0.6661

 Realisation Number 2
   i            ht(i)            et(i)
  ------------------------------------
   1            1.9327          -2.2795
   2            3.5577          -1.2249
   3            4.1461           0.6424
   4            3.4455          -2.9920
   5            5.9199           0.5777
   6            4.8221          -1.2894
   7            5.3174          -1.6473
   8            6.1095           6.1689
   9            3.1579           2.2935
  10            2.2189           0.1141
```