NAG Toolbox for MATLAB

g13ff

1 Purpose

g13ff forecasts the conditional variances, h_t, for $t = T+1, \ldots, T+\xi$ from a GJR GARCH(p, q) sequence, where ξ is the forecast horizon and T is the current time (see Glosten *et al.* 1993).

2 Syntax

```
[fht, ifail] = g13ff(nt, ip, iq, theta, gamma, ht, et, 'num', num)
```

3 Description

Assume the GARCH(p, q) process can be represented by:

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

where $\epsilon_t \mid \psi_{t-1} = N(0, h_t)$ or $\epsilon_t \mid \psi_{t-1} = S_t(df, h_t)$, and $S_t = 1$, if $\epsilon_t < 0$, or $S_t = 0$, if $\epsilon_t \geq 0$, has been modelled by g13fe, and the estimated conditional variances and residuals are contained in the arrays **ht** and **et** respectively.

g13ff will then use the last $\max(p, q)$ elements of the arrays **ht** and **et** to estimate the conditional variance forecasts, $h_t \mid \psi_T$, where $t = T+1, \ldots, T+\xi$ and ξ is the forecast horizon.

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: **nt – int32 scalar**

ξ, the forecast horizon.

Constraint: **nt** > 0.

2: **ip – int32 scalar**

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

Constraints:

$\max(\mathbf{ip}, \mathbf{iq}) \leq 20$;
$\mathbf{ip} \geq 0$.

3: **iq – int32 scalar**

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

Constraints:

$\max(\mathbf{ip}, \mathbf{iq}) \leq 20$;
$\mathbf{iq} \geq 1$.

4: **theta**$(\mathbf{iq} + \mathbf{ip} + \mathbf{1})$ **– double array**

The first element must contain the coefficient α_o and the next **iq** elements must contain the coefficients α_i, for $i = 1, \ldots, q$. The remaining **ip** elements must contain the coefficients β_j, for $j = 1, \ldots, p$.

5: **gamma – double scalar**

The asymmetry parameter γ for the GARCH(p, q) sequence.

6: **ht**$(\mathbf{num})$ **– double array**

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

7: **et**$(\mathbf{num})$ **– double array**

The sequence of past residuals for the GARCH(p, q) process, ϵ_t, for $t = 1, \ldots, T$.

5.2 Optional Input Parameters

1: **num – int32 scalar**

Default: The dimension of the arrays **ht**, **et**. (An error is raised if these dimensions are not equal.)
the number of terms in the arrays **ht** and **et** from the modelled sequence.

Constraint: $\max(\mathbf{ip}, \mathbf{iq}) \leq \mathbf{num}$.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **fht**$(\mathbf{nt})$ **– double array**

The forecast values of the conditional variance, h_t, for $t = T + 1, \ldots, T + \xi$.

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, $\mathbf{num} < \max(\mathbf{ip}, \mathbf{iq})$,
or $\mathbf{iq} < 1$,
or $\mathbf{ip} < 0$,
or $\max(\mathbf{ip}, \mathbf{iq}) > 20$,
or $\mathbf{nt} \leq 0$.

7 Accuracy

Not applicable

8 Further Comments

None.

9 Example

```
nt = int32(4);
iseed = zeros(4, 1, 'int32');
iseed(1) = int32(111);
igen = int32(0);
rwsav = zeros(9, 1);
rvec = zeros(40, 1);
dist = 'N';
num = int32(2000);
param = [0.4; 0.1; 0.7];
bx = [1.5;2.5;3];
theta = [0.2; 0.05; 0.35; 0.05; 2; 0.75; 1.25];
gamma = 0.1;
df = 5.1;
ip = int32(1);
iq = int32(1);
nreg = int32(2);
mn = int32(1);
hp = 0;
mean = 4;
copts = [true; true];
maxit = int32(100);
tol = 1e-05;
x = zeros(2000, 3);
for i = 1:num
  fac1 = double(i)*0.01;
  x(i,2) = 0.01 + 0.7*sin(fac1);
  x(i,1) = 0.5 + fac1*0.1;
  x(i,3) = 1.0;
end
[igen, iseed] = g05kb(igen, iseed);
fcall = true;
[ht, yt, rvec, iseed, rwsav, ifail] = ...
     g05hm(dist, int32(200), ip, iq, param, gamma, df, fcall, rvec, igen,
iseed, rwsav);
fcall = false;
[ht, yt, rvec, iseed, rwsav, ifail] = ...
    g05hm(dist, num, ip, iq, param, gamma, df, fcall, rvec, igen, iseed,
rwsav);

for i = 1:num
  xterm = 0;
  for k = 1:nreg
    xterm = xterm + x(i,k)*bx(k);
  end
  if (mn == 1)
    yt(i) = mean + xterm + yt(i);
  else
    yt(i) = xterm + yt(i);
  end
end

[theta, se, sc, covar, hp, et, ht, lgf, ifail] = ...
    g13fe(dist, yt, x, ip, iq, nreg, mn, theta, hp, copts, maxit, tol);

[fht, ifail] = g13ff(nt, ip, iq, theta, gamma, ht, et)
```

```
fht =
    1.4802
    1.5719
    1.6463
    1.7067
ifail =
           0
```