NAG Library Routine Document

G05PFF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G05PFF generates a given number of terms of a GJR GARCH(p, q) process (see Glosten *et al.* (1993)).

2 Specification

```
SUBROUTINE G05PFF (DIST, NUM, IP, IQ, THETA, GAMMA, DF, HT, ET, FCALL,     &
                   R, LR, STATE, IFAIL)

INTEGER             NUM, IP, IQ, DF, LR, STATE(*), IFAIL
REAL (KIND=nag_wp) THETA(IQ+IP+1), GAMMA, HT(NUM), ET(NUM), R(LR)
LOGICAL             FCALL
CHARACTER(1)        DIST
```

3 Description

A GJR GARCH(p, q) process is represented by:

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

where $I_t = 1$ if $\epsilon_t < 0$, $I_t = 0$ if $\epsilon_t \geq 0$, 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. Symmetric GARCH sequences are generated when γ is zero, otherwise asymmetric GARCH sequences are generated with γ specifying the amount by which negative shocks are to be enhanced.

One of the initialization routines G05KFF (for a repeatable sequence if computed sequentially) or G05KGF (for a non-repeatable sequence) must be called prior to the first call to G05PFF.

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

1: DIST – CHARACTER(1) *Input*

On entry: 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 *Input*

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

Constraint: NUM > 0.

3: IP – INTEGER *Input*

On entry: the number of coefficients, β_i, for $i = 1, 2, \ldots, p$.

Constraint: IP ≥ 0.

4: IQ – INTEGER *Input*

On entry: the number of coefficients, α_i, for $i = 1, 2, \ldots, q$.

Constraint: IQ ≥ 1.

5: THETA(IQ + IP + 1) – REAL (KIND=nag_wp) array *Input*

On entry: the first element must contain the coefficient α_o, the next IQ elements must contain the coefficients α_i, for $i = 1, 2, \ldots, q$. The remaining IP elements must contain the coefficients β_j, for $j = 1, 2, \ldots, p$.

Constraints:

$$\sum_{i=2}^{\text{IQ+IP+1}} \text{THETA}(i) < 1.0;$$

$\text{THETA}(i) \geq 0.0$, for $i = 1$ and $i = \text{IQ} + 2, \ldots, \text{IQ} + \text{IP} + 1$.

6: GAMMA – REAL (KIND=nag_wp) *Input*

On entry: the asymmetry parameter γ for the GARCH(p, q) sequence.

Constraint: $\text{GAMMA} + \text{THETA}(i) \geq 0.0$, for $i = 2, 3, \ldots, \text{IQ} + 1$.

7: DF – INTEGER *Input*

On entry: 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.

8: HT(NUM) – REAL (KIND=nag_wp) array *Output*

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

9: ET(NUM) – REAL (KIND=nag_wp) array *Output*

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

10: FCALL – LOGICAL *Input*

On entry: if FCALL = .TRUE., a new sequence is to be generated, otherwise a given sequence is to be continued using the information in R.

11: R(LR) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the array contains information required to continue a sequence if FCALL = .FALSE..

On exit: contains information that can be used in a subsequent call of G05PFF, with FCALL = .FALSE..

12: LR – INTEGER *Input*

On entry: the dimension of the array R as declared in the (sub)program from which G05PFF is called.

Constraint: LR $\geq 2 \times (\mathrm{IP} + \mathrm{IQ} + 2)$.

13: STATE(∗) – INTEGER array *Communication Array*

Note: the actual argument supplied **must** be the array STATE supplied to the initialization routines G05KFF or G05KGF.

On entry: contains information on the selected base generator and its current state.

On exit: contains updated information on the state of the generator.

14: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, DIST is not valid: DIST $= \langle value \rangle$.

IFAIL $= 2$

On entry, NUM $= \langle value \rangle$.
Constraint: NUM ≥ 0.

IFAIL $= 3$

On entry, IP $= \langle value \rangle$.
Constraint: IP ≥ 0.

IFAIL $= 4$

On entry, IQ $= \langle value \rangle$.
Constraint: IQ ≥ 1.

IFAIL $= 5$

On entry, THETA($\langle value \rangle$) $= \langle value \rangle$ and $\gamma = \langle value \rangle$.
Constraint: $\alpha_i + \gamma \geq 0$.

IFAIL $= 7$

On entry, DF $= \langle value \rangle$.
Constraint: DF ≥ 3.

IFAIL $= 11$

IP or IQ is not the same as when R was set up in a previous call.
Previous value of IP $= \langle value \rangle$ and IP $= \langle value \rangle$.
Previous value of IQ $= \langle value \rangle$ and IQ $= \langle value \rangle$.

IFAIL $= 12$

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

IFAIL $= 13$

On entry, STATE vector has been corrupted or not initialized.

IFAIL $= 51$

On entry, THETA($\langle value \rangle$) $= \langle value \rangle$.
Constraint: THETA(i) ≥ 0.0.

IFAIL $= 52$

On entry, sum of THETA(i) $= \langle value \rangle$.
Constraint: sum of THETA(i), for $i = 1, 2, \ldots, \text{IP} + \text{IQ}$ is < 1.0.

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

G05PFF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

None.

10 Example

This example first calls G05KFF to initialize a base generator then calls G05PFF to generate two realizations, each consisting of ten observations, from a GJR GARCH$(1,1)$ model.

10.1 Program Text

```
    Program g05pffe

!     G05PFF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g05kff, g05pff, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: lseed = 1, nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: gamma
      Integer                          :: df, genid, i, ifail, ip, iq, lr,     &
                                          lstate, ltheta, nreal, num, rn, subid
      Logical                          :: fcall
      Character (1)                    :: dist
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: et(:), ht(:), r(:), theta(:)
      Integer                          :: seed(lseed)
      Integer, Allocatable             :: state(:)
!     .. Executable Statements ..
      Write (nout,*) 'G05PFF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the base generator information and seed
      Read (nin,*) genid, subid, seed(1)

!     Initial call to initialiser to get size of STATE array
      lstate = 0
      Allocate (state(lstate))
      ifail = 0
      Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!     Reallocate STATE
      Deallocate (state)
      Allocate (state(lstate))

!     Initialize the generator to a repeatable sequence
      ifail = 0
      Call g05kff(genid,subid,seed,lseed,state,lstate,ifail)

!     Read in sample size and number of realizations
      Read (nin,*) num, nreal

!     Read in number of coefficients
      Read (nin,*) ip, iq
```

```
      lr = 2*(ip+iq+2)
      ltheta = ip + iq + 1
      Allocate (theta(ltheta),ht(num),et(num),r(lr))

!     Read in error distribution
      Read (nin,*) dist

!     Read in degrees of freedom if required
      If (dist=='T' .Or. dist=='t') Then
        Read (nin,*) df
      End If

!     Read in rest of series parameters
      Read (nin,*) theta(1:ltheta)
      Read (nin,*) gamma

!     Set FCALL for first realization
      fcall = .True.

!     Generate NREAL realizations
      Do rn = 1, nreal

        ifail = 0
        Call g05pff(dist,num,ip,iq,theta,gamma,df,ht,et,fcall,r,lr,state, &
          ifail)

!       Display the results
        Write (nout,99998) 'Realization Number ', rn
        Write (nout,*) '   I            HT(I)            ET(I)'
        Write (nout,*) '  ------------------------------------'
        Write (nout,99999)(i,ht(i),et(i),i=1,num)
        Write (nout,*)

!       Set FCALL flag for any further realizations
        fcall = .False.
      End Do

99999 Format (1X,I5,1X,F16.4,1X,F16.4)
99998 Format (1X,A,I0)
    End Program g05pffe
```

10.2 Program Data

```
G05PFF Example Program Data
1  1  1762543      :: GENID,SUBID,SEED(1)
10 2               :: NUM,NREAL
1  1               :: IP,IQ
'N'
0.4 0.1 0.7        :: THETA
0.1                :: GAMMA
```

10.3 Program Results

```
 G05PFF Example Program Results

 Realization Number 1
    I            HT(I)            ET(I)
   ------------------------------------
     1           1.8000           0.4679
     2           1.6819          -1.6152
     3           2.0991           0.9592
     4           1.9614           1.1701
     5           1.9099          -1.7355
     6           2.3393          -0.0289
     7           2.0377          -0.4201
     8           1.8617           1.0865
     9           1.8212          -0.0061
    10           1.6749           0.5754

 Realization Number 2
```

I	HT(I)	ET(I)
1	1.6055	-2.0776
2	2.3872	-1.0034
3	2.2724	0.4756
4	2.0133	-2.2871
5	2.8554	0.4012
6	2.4149	-0.9125
7	2.2570	-1.0732
8	2.2102	3.7105
9	3.3239	2.3530
10	3.2804	0.1388