

NAG Library Routine Document

G01EUF

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

G01EUF returns the value of the Vavilov distribution function $\Phi_V(\lambda; \kappa, \beta^2)$, via the routine name.

It is intended to be used after a call to G01ZUF.

2 Specification

```
FUNCTION G01EUF (X, RCOMM, IFAIL)
REAL (KIND=nag_wp) G01EUF
INTEGER IFAIL
REAL (KIND=nag_wp) X, RCOMM(322)
```

3 Description

G01EUF evaluates an approximation to the Vavilov distribution function $\Phi_V(\lambda; \kappa, \beta^2)$ given by

$$\Phi_V(\lambda; \kappa, \beta^2) = \int_{-\infty}^{\lambda} \phi_V(\lambda; \kappa, \beta^2) d\lambda,$$

where $\phi(\lambda)$ is described in G01MUF. The method used is based on Fourier expansions. Further details can be found in Schorr (1974).

4 References

Schorr B (1974) Programs for the Landau and the Vavilov distributions and the corresponding random numbers *Comp. Phys. Comm.* 7 215–224

5 Parameters

1: X – REAL (KIND=nag_wp) *Input*

On entry: the argument λ of the function.

2: RCOMM(322) – REAL (KIND=nag_wp) array *Communication Array*

On entry: this **must** be the same parameter RCOMM as returned by a previous call to G01ZUF.

3: 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:

$IFAIL = 1$

Either the initialization routine has not been called prior to the first call of this routine or a communication array has become corrupted.

$IFAIL = -99$

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

At least five significant digits are usually correct.

8 Parallelism and Performance

Not applicable.

9 Further Comments

G01EUF can be called repeatedly with different values of λ provided that the values of κ and β^2 remain unchanged between calls. Otherwise, G01ZUF must be called again. This is illustrated in Section 10.

10 Example

This example evaluates $\Phi_V(\lambda; \kappa, \beta^2)$ at $\lambda = 0.1$, $\kappa = 2.5$ and $\beta^2 = 0.7$, and prints the results.

10.1 Program Text

```

Program g01eufe

!      G01EUF Example Program Text
!
!      Mark 25 Release. NAG Copyright 2014.
!
!      .. Use Statements ..
!      Use nag_library, Only: g01euf, g01zuf, nag_wp, x02alf
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: mode = 1, nin = 5, nout = 6
!      .. Local Scalars ..
!      Real (Kind=nag_wp)         :: beta2, c1, c2, rkappa, x, x1, xu, y
!      Integer                    :: ifail
!      .. Local Arrays ..

```

```

      Real (Kind=nag_wp)          :: rcomm(322)
!      .. Executable Statements ..
      Write (nout,*) 'G01EUF Example Program Results'
      Write (nout,*)

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

!      Display titles
      Write (nout,99998) 'X', 'RKAPPA', 'BETA2', 'Y', 'IFAIL'
      Write (nout,*)

      c1 = -x02alf()
      c2 = -x02alf()
d_lp: Do
      Read (nin,*,Iostat=ifail) x, rkappa, beta2
      If (ifail/=0) Then
         Exit d_lp
      End If

      If ((rkappa/=c1) .Or. (beta2/=c2)) Then
!         If RKAPPA or BETA2 have changed, initialise array RCOMM
         ifail = 0
         Call g01zuf(rkappa,beta2,mode,xl,xu,rcomm,ifail)
      End If

!      Compute the value of the Vavilov distribution function
      ifail = 0
      y = g01euf(x,rcomm,ifail)

!      Display results
      Write (nout,99999) x, rkappa, beta2, y, ifail
      c1 = rkappa
      c2 = beta2
      End Do d_lp

99999 Format (1X,F4.1,5X,F4.1,5X,F4.1,3X,1P,E12.4,I6)
99998 Format (1X,A2,5X,A6,4X,A5,8X,A1,8X,A5)
      End Program g01eufe

```

10.2 Program Data

G01EUF Example Program Data
 0.1 2.5 0.7 : Values of X, RKAPPA and BETA2

10.3 Program Results

G01EUF Example Program Results

X	RKAPPA	BETA2	Y	IFAIL
0.1	2.5	0.7	9.9982E-01	0
