

# NAG Library Routine Document

## G01MUF

**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

G01MUF returns the value of the Vavilov density 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 G01MUF (X, RCOMM, IFAIL)
  REAL (KIND=nag_wp) G01MUF
  INTEGER IFAIL
  REAL (KIND=nag_wp) X, RCOMM(322)
```

### 3 Description

G01MUF evaluates an approximation to the Vavilov density function  $\phi_V(\lambda; \kappa, \beta^2)$  given by

$$\phi_V(\lambda; \kappa, \beta^2) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} e^{\lambda s} f(s; \kappa, \beta^2) ds,$$

where  $\kappa > 0$  and  $0 \leq \beta^2 \leq 1$ ,  $c$  is an arbitrary real constant and

$$f(s; \kappa, \beta^2) = C(\kappa, \beta^2) \exp \left\{ s \ln \kappa + (s + \kappa \beta^2) \left[ \ln \left( \frac{s}{\kappa} \right) + E_1 \left( \frac{s}{\kappa} \right) \right] - \kappa \exp \left( -\frac{s}{\kappa} \right) \right\}.$$

$E_1(x) = \int_0^x t^{-1} (1 - e^{-t}) dt$  is the exponential integral,  $C(\kappa, \beta^2) = \exp \{ \kappa (1 + \gamma \beta^2) \}$  and  $\gamma$  is Euler's constant.

The method used is based on Fourier expansions. Further details can be found in Schorr (1974).

For values of  $\kappa \leq 0.01$ , the Vavilov distribution can be replaced by the Landau distribution since  $\lambda_V = (\lambda_L - \ln \kappa)/\kappa$ . For values of  $\kappa \geq 10$ , the Vavilov distribution can be replaced by a Gaussian distribution with mean  $\mu = \gamma - 1 - \beta^2 - \ln \kappa$  and variance  $\sigma^2 = (2 - \beta^2)/2\kappa$ .

### 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 Arguments

- |    |  |                            |
|----|--|----------------------------|
| 1: | X – REAL (KIND=nag_wp)   | <i>Input</i>               |
|    | <i>On entry:</i> the argument $\lambda$ of the function.   |                            |
| 2: | RCOMM(322) – REAL (KIND=nag_wp) array  | <i>Communication Array</i> |
|    | <i>On entry:</i> this <b>must</b> be the same argument 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 argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -399

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

## 7 Accuracy

At least five significant digits are usually correct.

## 8 Parallelism and Performance

G01MUF is not threaded in any implementation.

## 9 Further Comments

G01MUF can be called repeatedly with different values of  $\lambda$  provided that the values of  $\kappa$  and  $\beta^2$  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  $\lambda = 2.5$ ,  $\kappa = 0.4$  and  $\beta^2 = 0.1$ , and prints the results.

## 10.1 Program Text

```

Program g01mufe

!      G01MUF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
      Use nag_library, Only: g01muf, g01zuf, nag_wp, x02alf
!      .. Implicit None Statement ..
      Implicit None
!      .. Parameters ..
      Integer, Parameter          :: mode = 0, nin = 5, nout = 6
!      .. Local Scalars ..
      Real (Kind=nag_wp)          :: beta2, c1, c2, rkappa, x, xl, xu, y
      Integer                     :: ifail
!      .. Local Arrays ..
      Real (Kind=nag_wp)          :: rcomm(322)
!      .. Executable Statements ..
      Write (nout,*) 'G01MUF Example Program Results'
      Write (nout,*)

!      Skip heading in data file
      Read (nin,*)
      c1 = -x02alf()
      c2 = -x02alf()

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

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
!         Initialize the RCOMM array if RKAPPA or BETA2 has changed
         ifail = 0
         Call g01zuf(rkappa,beta2,mode,xl,xu,rcomm,ifail)
      End If

!      Compute the value of the Vavilov density function
      ifail = 0
      y = g01muf(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)
End Program g01mufe

```

## 10.2 Program Data

G01MUF Example Program Data  
 2.5 0.4 0.1 : Values of X, RKAPPA and BETA2

### 10.3 Program Results

G01MUF Example Program Results

X	RKAPPA	BETA2	Y	IFAIL
2.5	0.4	0.1	8.3675E-02	0

---