nag_prob_density_vavilov (g01muc) returns the value of the Vavilov density function \( \phi_V(\lambda; \kappa, \beta^2) \). It is intended to be used after a call to nag_init_vavilov (g01zuc).

3 Description

nag_prob_density_vavilov (g01muc) evaluates an approximation to the Vavilov density function \( \phi_V(\lambda; \kappa, \beta^2) \) given by

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

where \( \kappa > 0 \) and \( 0 \leq \beta^2 \leq 1 \), \( e \) is an arbitrary real constant and

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

\( E_1(x) = \int_0^x t^{-1} (1 - e^{-t}) dt \) 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


5 Parameters

1: \( x \) – double

Input

On entry: the argument \( \lambda \) of the function.

2: \( \text{comm}_\text{arr}[322] \) – const double

Input

On entry: this must be the same parameter \( \text{comm}_\text{arr} \) as returned by a previous call to nag_init_vavilov (g01zuc).

6 Error Indicators and Warnings

None.

7 Accuracy

At least 5 significant digits are usually correct.
8 Further Comments

nag_prob_density_vavilov (g01muc) can be called repeatedly with different values of \( \lambda \) provided that the values of \( \kappa \) and \( \beta^2 \) remain unchanged between calls. Otherwise, nag_init_vavilov (g01zuc) must be called again. This is illustrated in Section 9.

9 Example

The example program evaluates \( \phi_V(\lambda; \kappa, \beta^2) \) at \( \lambda = 2.5, \kappa = 0.4 \) and \( \beta^2 = 0.1 \), and prints the results.

9.1 Program Text

/* nag_prob_density_vavilov (g01muc) Example Program. *
 * Copyright 2002 Numerical Algorithms Group.
 * Mark 7, 2002.
 */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg01.h>
#include <nagx02.h>

int main(void)
{
    /* Scalars */
    double c1, c2, x, rkappa, beta2, xl, xu, y;
    Integer exit_status, mode;
    NagError fail;
    #define WKMAX 322
    double comm_arr[WKMAX];
    mode = 0;
    INIT_FAIL(fail);
    exit_status = 0;
    c1 = -X02ALC;
    c2 = -X02ALC;
    Vprintf(" g01muc Example Program Results\n\n");
    /* Skip heading in data file */
    Vscanf("%*[^
\n] ");
    while (scanf("%lf%lf%lf%*[\n] ", &x, &rkappa, &beta2) != EOF)
    {
        if ((rkappa != c1) || (beta2 != c2 ))
        {
            g01zuc(rkappa, beta2, mode, &xl, &xu, comm_arr, &fail);
            if (fail.code != NE_NOERROR)
            {
                Vprintf("Error from g01zuc.%s\n", fail.message);
                exit_status = 1;
                goto END;
            }
        }
        y = g01muc(x, comm_arr);
        Vprintf(" X Rkappa Beta2 Y\n\n");
        Vprintf("%3.1f %3.1f %3.1f %12.4e\n", x, rkappa, beta2, y);
        c1 = rkappa;
    }

c2 = beta2;
    
END:
    return exit_status;
}

9.2 Program Data

\textit{g01muc} Example Program Data

2.5 0.4 0.1 : Values of $X$, $\text{RKAPPA}$ and $\text{BETA2}$

9.3 Program Results

\textit{g01muc} Example Program Results

\begin{tabular}{llll}
\hline
$X$ & RKappa & Beta2 & Y \\
\hline
2.5 & 0.4 & 0.1 & 8.3675e-02 \\
\hline
\end{tabular}