

nag_bessel_y0 (s17acc)

1. Purpose

nag_bessel_y0 (s17acc) returns the value of the Bessel function $Y_0(x)$.

2. Specification

```
#include <nag.h>
#include <nags.h>

double nag_bessel_y0(double x, NagError *fail)
```

3. Description

The function evaluates the Bessel function of the second kind, Y_0 , $x > 0$.

The approximation is based on Chebyshev expansions.

For x near zero, $Y_0(x) \simeq (2/\pi)(\ln(x/2) + \gamma)$, where γ denotes Euler's constant. This approximation is used when x is sufficiently small for the result to be correct to **machine precision**.

For very large x , it becomes impossible to provide results with any reasonable accuracy (see Section 6.1), hence the function fails. Such arguments contain insufficient information to determine the phase of oscillation of $Y_0(x)$; only the amplitude, $\sqrt{2/x}$, can be determined and this is returned. The range for which this occurs is roughly related to the **machine precision**: the function will fail if $x \gtrsim 1/\text{machine precision}$.

4. Parameters

x

Input: the argument x of the function.

Constraint: $x > 0.0$.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_REAL_ARG_GT

On entry, **x** must not be greater than $\langle \text{value} \rangle$: $x = \langle \text{value} \rangle$.

x is too large, the function returns the amplitude of the Y_0 oscillation, $\sqrt{2/\pi x}$.

NE_REAL_ARG_LE

On entry, **x** must not be less than or equal to 0.0: $x = \langle \text{value} \rangle$.

Y_0 is undefined, the function returns zero.

6. Further Comments

6.1. Accuracy

Let δ be the relative error in the argument and E be the absolute error in the result. (Since $Y_0(x)$ oscillates about zero, absolute error and not relative error is significant, except for very small x .)

If δ is somewhat larger than the machine representation error (e.g. if δ is due to data errors etc.), then E and δ are approximately related by $E \simeq |xY_1(x)| > \delta$ (provided E is also within machine bounds).

However, if δ is of the same order as the machine representation errors, then rounding errors could make E slightly larger than the above relation predicts.

For very small x , the errors are essentially independent of δ and the function should provide relative accuracy bounded by the **machine precision**.

For very large x , the above relation ceases to apply. In this region, $Y_0(x) \simeq \sqrt{2/\pi x} \sin(x - \pi/4)$. The amplitude $\sqrt{2/\pi x}$ can be calculated with reasonable accuracy for all x , but $\sin(x - \pi/4)$ cannot. If $x - \pi/4$ is written as $2N\pi + \theta$ where N is an integer and $0 \leq \theta < 2\pi$, then $\sin(x - \pi/4)$ is determined by θ only. If $x \gtrsim \delta^{-1}$, θ cannot be determined with any accuracy at all. Thus if x is greater than, or of the order of the inverse of **machine precision**, it is impossible to calculate the phase of $Y_0(x)$ and the function must fail.

6.2. References

Abramowitz M and Stegun I A (1968) *Handbook of Mathematical Functions* Dover Publications, New York ch 9 p 358.
 Clenshaw C W (1962) *Mathematical Tables, Chebyshev series for mathematical functions* National Physical Laboratory H.M.S.O. **5**.

7. See Also

nag_bessel_y1 (s17adc)

8. Example

The following program reads values of the argument x from a file, evaluates the function at each value of x and prints the results.

8.1. Program Text

```
/* nag_bessel_y0(s17acc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 2 revised, 1992.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nags.h>

main()
{
    double x, y;

    /* Skip heading in data file */
    Vscanf("%*[^\\n]");
    Vprintf("s17acc Example Program Results\\n");
    Vprintf("      x      y\\n");
    while (scanf("%lf", &x) != EOF)
    {
        y = s17acc(x, NAGERR_DEFAULT);
        Vprintf("%12.3e%12.3e\\n", x, y);
    }
    exit(EXIT_SUCCESS);
}
```

8.2. Program Data

```
s17acc Example Program Data
      0.5
      1.0
      3.0
      6.0
      8.0
     10.0
    1000.0
```

8.3. Program Results

s17acc Example Program Results

x	y
5.000e-01	-4.445e-01
1.000e+00	8.826e-02
3.000e+00	3.769e-01
6.000e+00	-2.882e-01
8.000e+00	2.235e-01
1.000e+01	5.567e-02
1.000e+03	4.716e-03
