NAG C Library Function Document

nag_bessel_zeros (s17alc)

1 Purpose

nag_bessel_zeros (s17alc) determines the leading $N$ zeros of one of the Bessel functions $J_\alpha(x)$, $Y_\alpha(x)$, $J'_\alpha(x)$ or $Y'_\alpha(x)$ for real $x$ and non-negative $\alpha$.

2 Specification

```c
void nag_bessel_zeros (double a, Integer n, Integer mode, double rel,
                      double x[], NagError *fail)
```

3 Description

This routine attempts to find the leading $N$ zeros of one of the Bessel functions $J_\alpha(x)$, $Y_\alpha(x)$, $J'_\alpha(x)$ or $Y'_\alpha(x)$, where $x$ is real. When $\alpha$ is real, these functions each have an infinite number of real zeros, all of which are simple with the possible exception of $x = 0$. If $\alpha \geq 0$, the $n$th positive zero is denoted by $j_{\alpha,n}$, $y_{\alpha,n}$, and $y'_{\alpha,n}$, respectively, for $n = 1, 2, \ldots, N$, except that $x = 0$ is counted as the first zero of $J'_\alpha(x)$ when $\alpha = 0$. Since $J'_1(x) = -J_1(x)$, it therefore follows that $j'_0 = 0$ and $j'_n = -j_{1,n-1}$, for $n = 2, 3, \ldots, N - 1$. Further details can be found in Abramowitz and Stegun (1972), 9.5.

nag_bessel_zeros is based on Algol 60 procedures given by Temme N M (1979). Initial approximations to the zeros are computed from asymptotic expansions. These are then improved by higher-order Newton iteration making use of the differential equation for the Bessel functions.

4 Parameters

1: a – double

   On entry: the order $\alpha$ of the function.

   Constraint: $0.0 \leq a \leq 10000.0$.

2: n – Integer

   On entry: the number $N$ of zeros required.

   Constraint: $n \geq 1$.

3: mode – Integer

   On entry: specifies the form of the function whose zeros are required as follows:

   - if mode = 1, then the zeros of $J_\alpha(x)$ are required;
   - if mode = 2, then the zeros of $Y_\alpha(x)$ are required;
   - if mode = 3, then the zeros of $J'_\alpha(x)$ are required;
   - if mode = 4, then the zeros of $Y'_\alpha(x)$ are required.

   Constraint: $1 \leq \text{mode} \leq 4$.

4: rel – double

   On entry: the relative accuracy to which the zeros are required.

   Suggested value: the square root of the machine precision.

   Constraint: rel > 0.0.
s17alc

5: \( x[n] \) – double

*Output*

On exit: the \( N \) required zeros of the function specified by \texttt{mode}.

6: \texttt{fail} – NagError *

*Input/Output*

The NAG error parameter (see the Essential Introduction).

5 Error Indicators and Warnings

\textbf{NE_INT}

On entry, \( n = \texttt{<value>} \).
Constraint: \( n \geq 1 \).

On entry, \texttt{mode} = \texttt{<value>}.  
Constraint: \( 1 \leq \texttt{mode} \leq 4 \).

\textbf{NE_REAL}

On entry, \( a = \texttt{<value>} \).
Constraint: \( 0.0 \leq a \leq 100000.0 \).

On entry, \texttt{rel} = \texttt{<value>}.  
Constraint: \( \texttt{rel} > 0.0 \).

\textbf{NE_INTERNAL_ERROR}

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

6 Further Comments

6.1 Accuracy

If the value of \texttt{rel} is set to \( 10^{-d} \), then the required zeros should have approximately \( d \) correct significant digits.

6.2 References


7 See Also

None.

8 Example

To determine the leading five positive zeros of the Bessel function \( J_0(x) \).
8.1 Program Text

/* nag_bessel_zeros (s17alc) Example Program. */
/* Copyright 2000 Numerical Algorithms Group. */
/* NAG C Library */
/* Mark 6, 2000. */
*/

#include <math.h>
#include <nag.h>
#include <nag_stdtlib.h>
#include <nags.h>
#include <nagx02.h>

int main(void)
{

#define NMAX 100

double a, rel;
double *x=0;
Integer i;
Integer exit_status=0;
Integer mode;
Integer n;
NagError fail;

INIT_FAIL(fail);
Vprintf("s17alc Example Program Results\n\n");

if (! (x = NAG_ALLOC(NMAX, double)))
{
    Vprintf("Allocation failure\n");
    exit_status=-1;
}

/* Skip heading in data file */
Vscanf("%*[\n"];
rel = sqrt(XO2AJC);
Vscanf("%lf %ld %ld", &a, &n, &mode);
s17alc (a, n, mode, rel, x, &fail);

if (fail.code == NE_NOERROR)
{
    Vprintf(" a n mode rel\n");
    Vprintf(" (machine-dependent)\n");
    Vprintf(" %4.1f%3ld%ld %9.1e\n", a, n, mode, rel);
    if (mode == 1)
    {
        Vprintf("Leading N positive zeros of J\n");
    }
    else if (mode == 2)
    {
        Vprintf("Leading N positive zeros of Y\n");
    }
    else if (mode == 3)
    {
    
}
if (a == 0.0)
{
  Vprintf("Leading N non-negative zeros of J\n\n");
} else
{
  Vprintf("Leading N positive zeros of J\n\n");
}
else if (mode == 4)
{
  Vprintf("Leading N positive zeros of Y\n\n");
  for (i = 0; i <= n-1; ++i)
  {
    Vprintf("x = %2.4e\n", x[i]);
  }
  Vprintf("\n");
} else
{
  Vprintf("Error from sl7alc.\n\n", fail.message);
  exit_status = 1;
  goto END;
}
END:
if (x) NAG_FREE (x);
return exit_status;

8.2 Program Data

sl7alc Example Program Data
0.0 5 1 : Values of a, n and mode

8.3 Program Results

sl7alc Example Program Results

<table>
<thead>
<tr>
<th>a</th>
<th>n</th>
<th>mode</th>
<th>rel</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>5</td>
<td>1</td>
<td>1.1e-08</td>
</tr>
</tbody>
</table>

Leading N positive zeros of J
x = 2.4048e+00
x = 5.5201e+00
x = 8.6537e+00
x = 1.1792e+01
x = 1.4931e+01