NAG Library Function Document

nag_complex_bessel_j_seq (s18gkc)

1 Purpose

nag_complex_bessel_j_seq (s18gkc) returns a sequence of values for the Bessel functions $J_{\alpha+n-1}(z)$ or $J_{\alpha-n+1}(z)$ for complex $z$, non-negative $\alpha < 1$ and $n = 1, 2, \ldots, |N| + 1$.

2 Specification

```c
#include <nag.h>
#include <nags.h>
void nag_complex_bessel_j_seq (Complex z, double a, Integer nl, Complex b[], NagError *fail)
```

3 Description

nag_complex_bessel_j_seq (s18gkc) evaluates a sequence of values for the Bessel function of the first kind $J_{\alpha}(z)$, where $z$ is complex and nonzero and $\alpha$ is the order with $0 \leq \alpha < 1$. The $(|N| + 1)$-member sequence is generated for orders $\alpha, \alpha + 1, \ldots, \alpha + |N|$ when $N \geq 0$. Note that $+$ is replaced by $-$ when $N < 0$. For positive orders the function may also be called with $z = 0$, since $J_{q}(0) = 0$ when $q > 0$. For negative orders the formula

$$J_{-q}(z) = \cos(\pi q)J_{q}(z) - \sin(\pi q)Y_{q}(z)$$

is used to generate the required sequence. The appropriate values of $J_{q}(z)$ and $Y_{q}(z)$ are obtained by calls to nag_complex_bessel_y (s17dcc) and nag_complex_bessel_j (s17dec).

4 References


5 Arguments

1: $z$ – Complex

   Input

   On entry: the argument $z$ of the function.

   Constraint: $z \neq (0.0, 0.0)$ when $nl < 0$.

2: $a$ – double

   Input

   On entry: the order $\alpha$ of the first member in the required sequence of function values.

   Constraint: $0.0 \leq a < 1.0$.

3: $nl$ – Integer

   Input

   On entry: the value of $N$.

   Constraint: $\text{abs}(nl) \leq 101$.

4: $b[\text{abs}(nl) + 1]$ – Complex

   Output

   On exit: with fail.code = NE_NOERROR or NW_SOME_PRECISION_LOSS, the required sequence of function values: $b[n-1]$ contains $J_{\alpha+n-1}(z)$ if $nl \geq 0$ and $J_{\alpha-n+1}(z)$ otherwise, for $n = 1, 2, \ldots, \text{abs}(nl) + 1$.
6  Error Indicators and Warnings

**NE_ALLOC_FAIL**
Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

**NE_BAD_PARAM**
On entry, argument `<value>` had an illegal value.

**NE_INT**
On entry, `|nl| = <value>`.
Constraint: `|nl| ≤ 101`.
On entry, `nl = <value>`.
Constraint: when `nl < 0, z ≠ (0.0, 0.0)`.

**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 contact NAG for assistance.
An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

**NE_NO_LICENCE**
Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

**NE_OVERFLOW_LIKELY**
Computation abandoned due to the likelihood of overflow.

**NE_REAL**
On entry, `a = <value>`.
Constraint: `a < 1.0`.
On entry, `a = <value>`.
Constraint: `a ≥ 0.0`.

**NE_TERMINATION_FAILURE**
Computation abandoned due to failure to satisfy the termination condition.

**NE_TOTAL_PRECISION_LOSS**
Computation abandoned due to total loss of precision.

**NW_SOME_PRECISION_LOSS**
Computation completed but some precision has been lost.

7  Accuracy

All constants in `nag_complex_bessel_y (s17dcc)` and `nag_complex_bessel_j (s17dec)` are specified to approximately 18 digits of precision. If `t` denotes the number of digits of precision in the floating-point arithmetic being used, then clearly the maximum number of correct digits in the results obtained is
limited by \( p = \min(t, 18) \). Because of errors in argument reduction when computing elementary functions inside \( \text{nag\_complex\_bessel\_y} \) (s17dcc) and \( \text{nag\_complex\_bessel\_j} \) (s17dec), the actual number of correct digits is limited, in general, by \( p - s \), where \( s \approx \max(1, |\log_{10}|z|, |\log_{10}|\alpha|) \) represents the number of digits lost due to the argument reduction. Thus the larger the values of \( |z| \) and \( |\alpha| \), the less the precision in the result.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

This example evaluates \( J_0(z) \), \( J_1(z) \), \( J_2(z) \) and \( J_3(z) \) at \( z = 0.6 - 0.8i \), and prints the results.

10.1 Program Text

/* \( \text{nag\_complex\_bessel\_j\_seq} \) Example Program. */
* * Copyright 2014 Numerical Algorithms Group. *
* * Mark 7, 2002. */
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nags.h>

int main(void)
{
    Integer exit_status = 0;
    Complex z, b[20];
    double a, alpha;
    Integer i, nl;
    NagError fail;

    INIT_FAIL(fail);

    /* \( \text{nag\_complex\_bessel\_j\_seq} \). */
    nag_complex_bessel_j_seq(z, a, nl, b, &fail);
    if (fail.code == NE_NOERROR)
    {
        printf(" z a nl
");
        printf(" (%7.3f,%7.3f) %lf %"%NAG_IFMT"
", z.re, z.im, a, nl); /* Nag\_stdio\_fmt */
    }

    /* \( \text{nag\_complex\_bessel\_j\_seq} \). */
    /* Bessel function of the 1st kind \( J_\alpha(n) \). */
    nag_complex_bessel_j_seq(z, a, nl, b, &fail);
    if (fail.code == NE_NOERROR)
    {
        printf(" z a nl
");
        printf(" (%7.3f,%7.3f) %lf %"%NAG_IFMT"
", z.re, z.im, a, nl);
    }

Mark 25
printf("Requested values of \text{J}_\text{alpha}(z)\n\n");
alpha = a;
printf(" alpha J_alpha(z)\n");
for (i = 0; i < \text{ABS}(nl) + 1; i++)
  {
    printf("%13.4e \text{ (13.4e,}\%13.4e)\n", alpha, b[i].re,
            b[i].im);
    if (nl > 0)
      alpha += 1.0;
    else
      alpha -= 1.0;
  }
else
  {
    printf("Error from \text{nag\_complex\_bessel\_j\_seq} (s18gkc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
  }
END:
  return exit_status;
}

10.2 Program Data

\text{nag\_complex\_bessel\_j\_seq} (s18gkc) Example Program Data
\( (0.6,-0.8) \quad 0.0 \quad 3 \) - Values of \( z, a \) and \( nl \)

10.3 Program Results

\text{nag\_complex\_bessel\_j\_seq} (s18gkc) Example Program Results
\( z \quad a \quad nl \)
\( (0.600, -0.800) \quad 0.000000 \quad 3 \)

Requested values of \( \text{J}_\text{alpha}(z) \)

\begin{array}{ll}
\text{alpha} & \text{J}_\text{alpha}(z) \\
0.0000e+00 & (1.0565e+00, \quad 2.4811e-01) \\
1.0000e+00 & (3.5825e-01, \quad -3.7539e-01) \\
2.0000e+00 & (-2.5974e-02, \quad -1.2538e-01) \\
3.0000e+00 & (-1.9369e-02, \quad -8.6380e-03) \\
\end{array}