NAG Toolbox for MATLAB

s18de

1 Purpose

s18de returns a sequence of values for the modified Bessel functions $I_{\nu+n}(z)$ for complex z, nonnegative ν and $n = 0, 1, \ldots, N-1$, with an option for exponential scaling.

2 Syntax

```
[cy, nz, ifail] = s18de(fnu, z, n, scal)
```

3 Description

s18de evaluates a sequence of values for the modified Bessel function $I_\nu(z)$, where z is complex, $-\pi < \arg z \leq \pi$, and ν is the real, nonnegative order. The N-member sequence is generated for orders $\nu, \nu+1, \ldots, \nu+N-1$. Optionally, the sequence is scaled by the factor $e^{-|\mathrm{Re}(z)|}$.

The function is derived from the function CBESI in Amos 1986.

Note: although the function may not be called with ν less than zero, for negative orders the formula $I_{-\nu}(z) = I_\nu(z) + \frac{2}{\pi}\sin(\pi\nu)K_\nu(z)$ may be used (for the Bessel function $K_\nu(z)$, see s18dc).

When N is greater than 1, extra values of $I_\nu(z)$ are computed using recurrence relations.

For very large $|z|$ or $(\nu+N-1)$, argument reduction will cause total loss of accuracy, and so no computation is performed. For slightly smaller $|z|$ or $(\nu+N-1)$, the computation is performed but results are accurate to less than half of ***machine precision***. If $\mathrm{Re}(z)$ is too large and the unscaled function is required, there is a risk of overflow and so no computation is performed. In all the above cases, a warning is given by the function.

4 References

Abramowitz M and Stegun I A 1972 *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

Amos D E 1986 Algorithm 644: A portable package for Bessel functions of a complex argument and non-negative order *ACM Trans. Math. Software* **12** 265–273

5 Parameters

5.1 Compulsory Input Parameters

1: **fnu – double scalar**

ν, the order of the first member of the sequence of functions.

Constraint: **fnu** ≥ 0.0.

2: **z – complex scalar**

The argument z of the functions.

3: **n – int32 scalar**

N, the number of members required in the sequence $I_\nu(z), I_{\nu+1}(z), \ldots, I_{\nu+N-1}(z)$.

Constraint: **n** ≥ 1.

4: **scal – string**

The scaling option.

scal = 'U'

The results are returned unscaled.

scal = 'S'

The results are returned scaled by the factor $e^{-|\mathrm{Re}(z)|}$.

Constraint: **scal** = 'U' or 'S'.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **cy(n) – complex array**

The N required function values: $\mathbf{cy}(i)$ contains $I_{\nu+i-1}(z)$, for $i = 1, 2, \ldots, N$.

2: **nz – int32 scalar**

The number of components of **cy** that are set to zero due to underflow.

If $\mathbf{nz} > 0$, then elements $\mathbf{cy}(\mathbf{n} - \mathbf{nz} + 1), \mathbf{cy}(\mathbf{n} - \mathbf{nz} + 2), \ldots, \mathbf{cy}(\mathbf{n})$ are set to zero.

3: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **fnu** < 0.0,
or **n** < 1,
or **scal** ≠ 'U' or 'S'.

ifail = 2

No computation has been performed due to the likelihood of overflow, because real(**z**) is greater than a machine-dependent threshold value. This error exit can only occur when **scal** = 'U'.

ifail = 3

The computation has been performed, but the errors due to argument reduction in elementary functions make it likely that the results returned by s18de are accurate to less than half of ***machine precision***. This error exit may occur when either ABS(**z**) or **fnu** + **n** − 1 is greater than a machine-dependent threshold value.

ifail = 4

No computation has been performed because the errors due to argument reduction in elementary functions mean that all precision in results returned by s18de would be lost. This error exit may occur when either ABS(**z**) or **fnu** + **n** − 1 is greater than a machine-dependent threshold value.

ifail = 5

No results are returned because the algorithm termination condition has not been met. This may occur because the parameters supplied to s18de would have caused overflow or underflow.

7 Accuracy

All constants in s18de are given to approximately 18 digits of precision. Calling the number of digits of precision in the floating-point arithmetic being used t, 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 s18de, the actual number of correct digits is limited, in general, by $p - s$, where $s \approx \max(1, |\log_{10}|z||, |\log_{10}\nu|)$ represents the number of digits lost due to the argument reduction. Thus the larger the values of $|z|$ and ν, the less the precision in the result. If s18de is called with $\mathbf{n} > 1$, then computation of function values via recurrence may lead to some further small loss of accuracy.

If function values which should nominally be identical are computed by calls to s18de with different base values of ν and different **n**, the computed values may not agree exactly. Empirical tests with modest values of ν and z have shown that the discrepancy is limited to the least significant 3 – 4 digits of precision.

8 Further Comments

The time taken for a call of s18de is approximately proportional to the value of **n**, plus a constant. In general it is much cheaper to call s18de with **n** greater than 1, rather than to make N separate calls to s18de.

Paradoxically, for some values of z and ν, it is cheaper to call s18de with a larger value of **n** than is required, and then discard the extra function values returned. However, it is not possible to state the precise circumstances in which this is likely to occur. It is due to the fact that the base value used to start recurrence may be calculated in different regions for different **n**, and the costs in each region may differ greatly.

Note that if the function required is $I_0(x)$ or $I_1(x)$, i.e., $\nu = 0.0$ or 1.0, where x is real and positive, and only a single function value is required, then it may be much cheaper to call s18ae, s18af, s18ce or s18cf, depending on whether a scaled result is required or not.

9 Example

```
fnu = 0;
z = complex(0.3, -0.4);
n = int32(2);
scal = 'U';
[cy, nz, ifail] = s18de(fnu, z, n, scal)
```

```
cy =
   0.9817 - 0.0595i
   0.1427 - 0.2027i
nz =
           0
ifail =
           0
```