NAG Fortran Library Routine Document

S18DCF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

S18DCF returns a sequence of values for the modified Bessel functions $K_{\nu+n}(z)$ for complex z, non-negative ν and $n = 0, 1, \ldots, \mathrm{N} - 1$, with an option for exponential scaling.

2 Specification

```
SUBROUTINE S18DCF(FNU, Z, N, SCALE, CY, NZ, IFAIL)
INTEGER            N, NZ, IFAIL
real               FNU
complex            Z, CY(N)
CHARACTER*1        SCALE
```

3 Description

This subroutine evaluates a sequence of values for the modified Bessel function $K_\nu(z)$, where z is complex, $-\pi < \arg z \leq \pi$, and ν is the real, non-negative order. The N-member sequence is generated for orders ν, $\nu + 1, \ldots, \nu + N - 1$. Optionally, the sequence is scaled by the factor e^z.

The routine is derived from the routine CBESK in Amos (1986).

Note: although the routine may not be called with ν less than zero, for negative orders the formula $K_{-\nu}(z) = K_\nu(z)$ may be used.

When N is greater than 1, extra values of $K_\nu(z)$ 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 $|z|$ is very small, near the machine underflow threshold, or $(\nu + N - 1)$ is too large, there is a risk of overflow and so no computation is performed. In all the above cases, a warning is given by the routine.

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 nonnegative order *ACM Trans. Math. Software* **12** 265–273

5 Parameters

1: FNU – ***real*** *Input*

On entry: the order, ν, of the first member of the sequence of functions.

Constraint: FNU $\geq$ 0.0.

2: Z – ***complex*** *Input*

On entry: the argument z of the functions.

Constraint: Z $\neq$ (0.0, 0.0).

3: N – INTEGER *Input*

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

Constraint: N $\geq$ 1.

4: SCALE – CHARACTER*1 *Input*

On entry: the scaling option.

If SCALE = 'U', the results are returned unscaled.

If SCALE = 'S', the results are returned scaled by the factor e^{z}.

Constraint: SCALE = 'U' or 'S'.

5: CY(N) – ***complex*** array *Output*

On exit: the N required function values: CY(i) contains $K_{\nu+i-1}(z)$, for $i = 1, 2, \ldots, N$.

6: NZ – INTEGER *Output*

On exit: the number of components of CY that are set to zero due to underflow. If NZ > 0 and Re $z \geq 0.0$, elements CY(1), CY(2), ..., CY(NZ) are set to zero. If Re $z < 0.0$, NZ simply states the number of underflows, and not which elements they are.

7: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, FNU > 0.0,
or Z = (0.0, 0.0),
or N < 1,
or SCALE $\neq$ 'U' or 'S'.

IFAIL = 2

No computation has been performed due to the likelihood of overflow, because ABS(Z) is less than a machine-dependent threshold value (given in the Users' Note for your implementation).

IFAIL = 3

No computation has been performed due to the likelihood of overflow, because FNU + N − 1 is too large – how large depends on Z and the overflow threshold of the machine.

IFAIL = 4

The computation has been performed, but the errors due to argument reduction in elementary functions make it likely that the results returned by S18DCF are accurate to less than half of

machine