NAG Toolbox for MATLAB

s17dl

1 Purpose

s17dl returns a sequence of values for the Hankel functions $H^{(1)}_{\nu+n}(z)$ or $H^{(2)}_{\nu+n}(z)$ for complex z, nonnegative ν and $n = 0, 1, \ldots, N - 1$, with an option for exponential scaling.

2 Syntax

```
[cy, nz, ifail] = s17dl(m, fnu, z, n, scal)
```

3 Description

s17dl evaluates a sequence of values for the Hankel function $H^{(1)}_{\nu}(z)$ or $H^{(2)}_{\nu}(z)$, where z is complex, $-\pi < \arg z \leq \pi$, and ν is the real, nonnegative order. The N-member sequence is generated for orders ν, $\nu + 1, \ldots, \nu + N - 1$. Optionally, the sequence is scaled by the factor e^{-iz} if the function is $H^{(1)}_{\nu}(z)$ or by the factor e^{iz} if the function is $H^{(2)}_{\nu}(z)$.

Note: although the function may not be called with ν less than zero, for negative orders the formulae $H^{(1)}_{-\nu}(z) = e^{\nu\pi i}H^{(1)}_{\nu}(z)$, and $H^{(2)}_{-\nu}(z) = e^{-\nu\pi i}H^{(2)}_{\nu}(z)$ may be used.

The function is derived from the function CBESH in Amos 1986. It is based on the relation

$$H^{(m)}_{\nu}(z) = \frac{1}{p}e^{-p\nu}K_{\nu}(ze^{-p}),$$

where $p = \frac{i\pi}{2}$ if $m = 1$ and $p = -\frac{i\pi}{2}$ if $m = 2$, and the Bessel function $K_{\nu}(z)$ is computed in the right half-plane only. Continuation of $K_{\nu}(z)$ to the left half-plane is computed in terms of the Bessel function $I_{\nu}(z)$. These functions are evaluated using a variety of different techniques, depending on the region under consideration.

When N is greater than 1, extra values of $H^{(m)}_{\nu}(z)$ 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 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 nonnegative order *ACM Trans. Math. Software* **12** 265–273

5 Parameters

5.1 Compulsory Input Parameters

1: **m – int32 scalar**

The kind of functions required.

If **m** $= 1$, the functions are $H^{(1)}_{\nu}(z)$.

If **m** $= 2$, the functions are $H_\nu^{(2)}(z)$.

Constraint: **m** $= 1$ or **m** $= 2$.

2: **fnu – double scalar**

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

Constraint: **fnu** ≥ 0.0.

3: **z – complex scalar**

The argument z of the functions.

Constraint: **z** $\neq (0.0, 0.0)$.

4: **n – int32 scalar**

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

Constraint: **n** ≥ 1.

5: **scal – string**

The scaling option.

scal = 'U'

The results are returned unscaled.

scal = 'S'

The results are returned scaled by the factor e^{-iz} when **m** $= 1$, or by the factor e^{iz} when **m** $= 2$.

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: **cy**(i) contains $H_{\nu+i-1}^{(\mathbf{m})}(z)$, 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 **nz** > 0, then if $\mathrm{Im}(z) > 0.0$ and **m** $= 1$, or $\mathrm{Im}(z) < 0.0$ and **m** $= 2$, elements **cy**(1), **cy**$(2), \ldots,$ **cy**$($**nz**$)$ are set to zero. In the complementary half-planes, **nz** simply states the number of underflows, and not which elements they are.

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, $\mathbf{m} \neq 1$ and $\mathbf{m} \neq 2$,
or $\mathbf{fnu} < 0.0$,
or $\mathbf{z} = (0.0, 0.0)$,
or $\mathbf{n} < 1$,
or **scal** $\neq$ '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.

ifail $= 3$

No computation has been performed due to the likelihood of overflow, because $\mathbf{fnu} + \mathbf{n} - 1$ 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 s17dl are accurate to less than half of ***machine precision***. This error exit may occur if either ABS(**z**) or $\mathbf{fnu} + \mathbf{n} - 1$ is greater than a machine-dependent threshold value.

ifail $= 5$

No computation has been performed because the errors due to argument reduction in elementary functions mean that all precision in results returned by s17dl would be lost. This error exit may occur when either of ABS(**z**) or $\mathbf{fnu} + \mathbf{n} - 1$ is greater than a machine-dependent threshold value.

ifail $= 6$

No results are returned because the algorithm termination condition has not been met. This may occur because the parameters supplied to s17dl would have caused overflow or underflow.

7 Accuracy

All constants in s17dl 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 s17dl, 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 s17dl 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 s17dl 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 s17dl is approximately proportional to the value of **n**, plus a constant. In general it is much cheaper to call s17dl with **n** greater than 1, rather than to make N separate calls to s17dl.

Paradoxically, for some values of z and ν, it is cheaper to call s17dl 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.

9 Example

```
m = int32(1);
fnu = 0;
z = complex(0.3, +0.4);
n = int32(2);
scal = 'U';
[cy, nz, ifail] = s17dl(m, fnu, z, n, scal)
```

```
cy =
   0.3466 - 0.5588i
  -0.7912 - 0.8178i
nz =
           0
ifail =
           0
```