NAG Library Routine Document

S19ARF

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

S19ARF returns an array of values for the Kelvin function $\mathrm{kei}\, x$.

2 Specification

```
SUBROUTINE S19ARF (N, X, F, IVALID, IFAIL)
INTEGER            N, IVALID(N), IFAIL
REAL (KIND=nag_wp) X(N), F(N)
```

3 Description

S19ARF evaluates an approximation to the Kelvin function $\mathrm{kei}\, x_i$ for an array of arguments x_i, for $i = 1, 2, \ldots, n$.

Note: for $x < 0$ the function is undefined, so we need only consider $x \geq 0$.

The routine is based on several Chebyshev expansions:

For $0 \leq x \leq 1$,

$$\mathrm{kei}\, x = -\frac{\pi}{4} f(t) + \frac{x^2}{4}[-g(t)\log(x) + v(t)]$$

where $f(t)$, $g(t)$ and $v(t)$ are expansions in the variable $t = 2x^4 - 1$;

For $1 < x \leq 3$,

$$\mathrm{kei}\, x = \exp\left(-\frac{9}{8}x\right) u(t)$$

where $u(t)$ is an expansion in the variable $t = x - 2$;

For $x > 3$,

$$\mathrm{kei}\, x = \sqrt{\frac{\pi}{2x}} e^{-x/\sqrt{2}} \left[\left(1 + \frac{1}{x}\right) c(t) \sin\beta + \frac{1}{x} d(t) \cos\beta\right]$$

where $\beta = \dfrac{x}{\sqrt{2}} + \dfrac{\pi}{8}$, and $c(t)$ and $d(t)$ are expansions in the variable $t = \dfrac{6}{x} - 1$.

For $x < 0$, the function is undefined, and hence the routine fails and returns zero.

When x is sufficiently close to zero, the result is computed as

$$\mathrm{kei}\, x = -\frac{\pi}{4} + \left(1 - \gamma - \log\left(\frac{x}{2}\right)\right)\frac{x^2}{4}$$

and when x is even closer to zero simply as

$$\mathrm{kei}\, x = -\frac{\pi}{4}.$$

For large x, $\mathrm{kei}\, x$ is asymptotically given by $\sqrt{\dfrac{\pi}{2x}} e^{-x/\sqrt{2}}$ and this becomes so small that it cannot be computed without underflow and the routine fails.

4 References

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

5 Arguments

1: N – INTEGER *Input*

On entry: n, the number of points.

Constraint: $\mathrm{N} \geq 0$.

2: X(N) – REAL (KIND=nag_wp) array *Input*

On entry: the argument x_i of the function, for $i = 1, 2, \ldots, \mathrm{N}$.

Constraint: $\mathrm{X}(i) \geq 0.0$, for $i = 1, 2, \ldots, \mathrm{N}$.

3: F(N) – REAL (KIND=nag_wp) array *Output*

On exit: $\mathrm{kei}\, x_i$, the function values.

4: IVALID(N) – INTEGER array *Output*

On exit: IVALID(i) contains the error code for x_i, for $i = 1, 2, \ldots, \mathrm{N}$.

IVALID(i) $= 0$
No error.

IVALID(i) $= 1$
x_i is too large, the result underflows. F(i) contains zero. The threshold value is the same as for IFAIL $= 1$ in S19ADF, as defined in the Users' Note for your implementation.

IVALID(i) $= 2$
$x_i < 0.0$, the function is undefined. F(i) contains 0.0.

5: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

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, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

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, at least one value of X was invalid.
Check IVALID for more information.

IFAIL $= 2$

On entry, N $= \langle value \rangle$.
Constraint: N ≥ 0.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Let E be the absolute error in the result, and δ be the relative error in the argument. If δ is somewhat larger than the machine representation error, then we have:

$$E \simeq \left| \frac{x}{\sqrt{2}}(-\mathrm{ker}_1\, x + \mathrm{kei}_1\, x) \right| \delta.$$

For small x, errors are attenuated by the function and hence are limited by the ***machine precision***.

For medium and large x, the error behaviour, like the function itself, is oscillatory and hence only absolute accuracy of the function can be maintained. For this range of x, the amplitude of the absolute error decays like $\sqrt{\frac{\pi x}{2}} e^{-x/\sqrt{2}}$, which implies a strong attenuation of error. Eventually, $\mathrm{kei}\, x$, which is asymptotically given by $\sqrt{\frac{\pi}{2x}} e^{-x/\sqrt{2}}$, becomes so small that it cannot be calculated without causing underflow and therefore the routine returns zero. Note that for large x, the errors are dominated by those of the standard function exp.

8 Parallelism and Performance

S19ARF is not threaded in any implementation.

9 Further Comments

Underflow may occur for a few values of x close to the zeros of $\mathrm{kei}\, x$, below the limit which causes a failure with IFAIL $= 1$.

10 Example

This example reads values of X from a file, evaluates the function at each value of x_i and prints the results.

10.1 Program Text

```
    Program s19arfe

!     S19ARF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, s19arf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: f(:), x(:)
      Integer, Allocatable             :: ivalid(:)
!     .. Executable Statements ..
      Write (nout,*) 'S19ARF Example Program Results'

!     Skip heading in data file
      Read (nin,*)

      Write (nout,*)
      Write (nout,*) '     X           F          IVALID'
      Write (nout,*)

      Read (nin,*) n

      Allocate (x(n),f(n),ivalid(n))

      Read (nin,*) x(1:n)

      ifail = 0
      Call s19arf(n,x,f,ivalid,ifail)

      Do i = 1, n
        Write (nout,99999) x(i), f(i), ivalid(i)
      End Do

99999 Format (1X,1P,2E12.3,I5)
    End Program s19arfe
```

10.2 Program Data

```
S19ARF Example Program Data

7

0.0 0.1 1.0 2.5 5.0 10.0 15.0
```

10.3 Program Results

```
 S19ARF Example Program Results

      X           F          IVALID

    0.000E+00  -7.854E-01    0
    1.000E-01  -7.769E-01    0
    1.000E+00  -4.950E-01    0
    2.500E+00  -1.107E-01    0
    5.000E+00   1.119E-02    0
    1.000E+01  -3.075E-04    0
    1.500E+01   7.963E-06    0
```