1 SUMMARY

To **compute the modified Bessel functions** *Ibessnu* **and** $K\nu$ for a range of orders ν=0,1,2,...,n, at the same argument x.

ATTRIBUTES — **Version:** 1.0.0. **Types:** FF07A; FF07AD. **Calls:** FD05A. **Original date:** April 1983. **Origin:** A. R. Curtis, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list and calling sequence

The single precision version

```
CALL FF07A(X,N,A,B,NAB)
```

The double precision version

```
CALL FF07AD(X,N,A,B,NAB)
```

X is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to the value of $\pm x$ (the subroutine uses the absolute value of X). A zero value is not allowed when $K_\nu(x)$ is requested, i.e. when argument N<0. This argument is not altered by the subroutine.

N is an INTEGER variable which must be set by the user to $\pm n$, where n is the highest order required. If N<0, both $I_\nu(x)$ and $K_\nu(x)$ are computed; if N>0, only $I_\nu(x)$ is computed. A zero value is not allowed. This argument is not altered by the subroutine.

A is a REAL (DOUBLE PRECISION in the D version) array of dimension NAB in which the subroutine will return the values of $I_\nu(x)$, ν=0,1,2,...,n, i.e. $I_0(x)$ in A(1), $I_1(x)$ in A(2) up to $I_n(x)$ in A(n+1).

B is a REAL (DOUBLE PRECISION in the D version) array of dimension NAB in which the subroutine will return the values of $K_\nu(x)$, ν=0,1,2,...,n, i.e. $K_0(x)$ in B(1), $K_1(x)$ in B(2) up to $K_n(x)$ in B(n+1). If N>0 the array B is not altered.

NAB is an INTEGER variable which must be set by the user to the dimension of the arrays A and B; it must be at least n+1. This argument is not altered by the subroutine.

3 GENERAL INFORMATION

Use of common: none.

Workspace: None.

Other routines called directly: calls DEXP, DSQRT and DLOG.

Input/output: Error warning and diagnostic messages on unit 6.

Restrictions:

$n > 0$;

$x > 0$ if $K_\nu(x)$ required,

else $x \geq 0$.

4 METHOD

4.1 Method

The method is based on the recurrence relation (9.6.26 in Abramowitz and Stegun[1])

$$u_{\nu+1} - u_{\nu-1} = \frac{2\nu}{x} u_\nu \tag{1}$$

whose general solution is

$$u_\nu = a I_\nu(x) + b(-1)^\nu K_\nu(x) \tag{2}$$

where a and b are constants.

To compute $I_\nu(x)$, the recurrence relation is solved for $u_{\nu-1}$ and used downwards from $\nu = m > n$, chosen so that the second term in (2) is negligible compared with the first for $\nu \le n$. At the same time, the sum

$$s = u_0 + 2\sum_{\nu=1}^{m} u_\nu \approx a e^x \tag{3}$$

is computed, and the normalizing factor $a^{-1} = s^{-1} e^x$ is finally applied to the stored values in the array A.

To compute $K_\nu(x)$ (if required), the recurrence relation is solved for $u_{\nu+1}$ and used upwards from accurate values of $K_0(x)$ and $K_1(x)$. The first is computed from new highly accurate Chebyshev series, and $K_1(x)$ is then computed as (9.6.15[1])

$$K_1(x) = \frac{1/x - I_1(x)K_0(x)}{I_0(x)}, \tag{4}$$

which does not suffer serious loss of accuracy through cancellation.

4.2 Accuracy and timing

The computation of $I_0(x)$ is good almost to full computer accuracy . The values of $I_\nu(x)$ in Table 9.11 in Abramowitz and Stegun[1] are reproduced exactly (apart from those which underflow). It may be necessary to take precautions against overflow. For small x and large n, underflows will occur and zero values will be returned; a call could be made to a library subroutine to mask off underflow interrupts.

The accuracy of $K_\nu(x)$ is also about 15 significant figures using the new Chebyshev series for $K_0(x)$. All $K_\nu(x)$ values in Table 9.11[1] for $x \le 10$ are reproduced (except for a few discrepancies by one in the last digit of the table values, and for values which would overflow); however, for $x = 50$ and $x = 100$, many errors in the last digit of the table values where found, and the accuracy of `FF07AD` was confirmed independently by using the asymptotic series 9.7.2[1]. For small x and large n, overflows would occur on the IBM computer; these are avoided, and the largest value represented on the machine is returned instead.

A 4-byte arithmetic version on some computers is not recommended because of possible accumulation of rounding errors; since the result arrays `A` and `B` are used as work-space, this consideration prevents internal 8-byte arithmetic computation being used to give 4-byte precision results. If `FF07AD` is used on other computers, the constants used in accuracy and overflow tests should be reconsidered.

Execution time depends more or less linearly on n, and also somewhat on x. For moderate x values, it varies on an IBM/3081 from about 75 μsec at $n = 10$ to about 650 μsec at $n = 100$ for $I_\nu(x)$ only; if $K_\nu(x)$ is also computed, typical times are 125 μsec and 800 μsec. Values of x in the range 10 to 100 are more expensive at small n, but less expensive at large n.

References

1. M.Abramowitz and I.A.Stegun, 'Handbook of Mathematical Functions', Dover (New York) 1965.
2. C.W.Clenshaw, 'N.P.L. Mathematical Tables', Vol. 5, HMSO (1962).