JOURNAL OF RESEARCH of the National Bureau of Standards — B. Mathematical Sciences
Vol. 77B, No. 3, July–September 1973

Certification of an Algorithm for Bessel Functions of Complex Argument

David J. Sookne [1]

Institute for Basic Standards, National Bureau of Standards, Washington, D.C. 20234

(June 4, 1973)

Computer tests of algorithm BESLCI are described, and the results of the test are given.

Key words: Backward recursion; Bessel function; bit comparison; error bounds; relative error.

1. Introduction

Bessel function values $I_n(z)$, $J_n(z)$ generated by BESLCI [1] were compared with check values calculated via the ascending series

$$J_n(z)=\left(\frac{z}{2}\right)^n\sum_{k=0}^{\infty}\frac{(-z^2/4)^k}{k!(n+k)!},\qquad I_n(z)=\left(\frac{z}{2}\right)^n\sum_{k=0}^{\infty}\frac{(z^2/4)^k}{k!(n+k)!},$$

using multiprecision arithmetic [2].[2] The 60-bit mantissa of each part of the test value was subtracted from the corresponding mantissa of the check value, and the result expressed as a multiple, m, say, of the last bit. For most errors mentioned in this certification, $|m| \leq 7$. Bit comparison was used in preference to calculation of relative error in order to simplify computations. An error of m bits in the mantissa corresponds to a relative error between $m\cdot 2^{-60}$ and $m\cdot 2^{-59}$.

This test is too strict, however, near a zero of the real or imaginary part of the function being tested. For complex values of z, it is more realistic to compare the bit error with the *greater* of the real and imaginary parts, since the relative error in a complex number $\xi+i\eta$ is $(\delta\xi+i\delta\eta)/|\xi+i\eta|$, $\delta\xi$ and $\delta\eta$ denoting the respective errors in the real and imaginary parts. For arguments in the neighborhoods of zeroes of Re $J_n(z)$, Im $J_n(z)$, Re $I_n(z)$, or Im $I_n(z)$, an assessment of absolute errors was made to 60 binary places by right-shifting the mantissa of both test and check values before subtraction. This was done whenever the order n was less than $|z|$ and the modulus of the check value less than $\frac{1}{2}$. These cases are distinguished by asterisks in the computer printout; statistics for this test are compiled separately.

A bit comparison test was also used to check the section of code involving the two-term ascending series for small $|z|$.

A second feature of BESLCI which was tested systematically is the error return. This portion of the code determines whether all orders $n=0, 1, \ldots,$ NB-1 can be calculated to the desired accuracy. A modification occurs if NB is so large compared with $|z|$ that there is underflow in the representation of $J_{\mathrm{NB}-1}(z)$ or $I_{\mathrm{NB}-1}(z)$. When this happens, the program determines the largest

AMS Subject Classification: 68A10.
[1] Present address: W.U.J.S. Institute, Arad, Israel.
[2] Figures in brackets indicate the literature references at the end of this paper.

value NCALC ($>|z|$) such that accurate values can be calculated for $n=0, 1, \ldots$, NCALC−1; these values are then calculated. To test this, BESLCI was called with NB = 5600. In each case, the program returned NCALC < NB, insuring that the error return loop had been executed. Then the values of NCALC and the Bessel function of order NCALC−1 were checked by calling BESLCI with NB set equal to NCALC; this involved another execution of the error return loop. In each case, the new NCALC was identical with the old. Finally, the Bessel function value of order $[|z|]$ was checked by calling BESLCI with NB $= [|z|]+1$; this involved no execution of the error return loop. In both cases, bit comparisons on 60 significant bits were performed.

Results of the tests are described in the next two sections.

2. The Bit Comparison Test

Arguments z were chosen as follows. Let R_j denote the set of rational numbers $2^j \cdot p$, where p is a 60-bit fraction, with $\frac{1}{2} \leq p < 1$. Two complex arguments were generated in each of the sets

$$A_j = \{x+iy : x \epsilon R_j,\ y=0\} \qquad j = -6(1)6$$

$$B_k = \{x+iy : x=0,\ y \epsilon R_k\} \qquad k = -6(1)6$$

$$C_{jk} = \{x+iy : x \epsilon R_j,\ y \epsilon R_k\} \qquad j, k = -6(1)6,$$

using a pseudorandom number generator [3]. One of the two arguments was used to generate I's and the other J's. For each argument, functions of order 0, 1, . . ., 10 were tested. It should be noted that the sets C_{jk} include complex arguments with real and imaginary parts of quite disparate magnitudes.

Summary statistics for the tests are:

Test on 60 Significant Bits

903	results correct to 60 bits*
1329	results in error by 1 multiple of last bit
967	results in error by 2 multiples of last bit
657	results in error by 3 multiples of last bit
404	results in error by 4 multiples of last bit
333	results in error by 5 multiples of last bit
247	results in error by 6 multiples of last bit
208	results in error by 7 multiples of last bit
1422	results in error by more than 7 multiples of last bit
6470	(Total)

Test on 60 Binary Places

320	results correct to 60 places**
479	results in error by 1 multiple of last place
279	results in error by 2 multiples of last place
160	results in error by 3 multiples of last place
82	results in error by 4 multiples of last place
59	results in error by 5 multiples of last place
45	results in error by 6 multiples of last place

*Excluding 202 results for pure real or imaginary argument, for which test and check values are both zero.

**Excluding 370 results for pure real or imaginary argument.

26 results in error by 7 multiples of last place
88 results in error by more than 7 multiples of last place

1538 (Total)

In the first test, the maximum error was $(6651)_8$. This occurred in the real part of $J_n(z)$ at $n=10$ and $z=(.1216\ldots)-i(.01928\ldots)$, for which $J_n(z)=-2^{-71}\cdot p-i\cdot 2^{-62}\cdot q$, with $\frac{1}{2}\leq p, q<1$. It is appropriate to assess this error relative to $|J_n(z)|$. This can be done fairly simply by dividing the bit error by 2^9, since 9 is the difference between the binary exponents of $\operatorname{Re} J_n(z)$ and $\operatorname{Im} J_n(z)$. This "relative error" is 7. On applying this technique to all the bit errors, the maximum relative error was found to be $(34)_8$. Allowing for the fact that one mantissa can be no more than twice the other, we see that the relative error is bounded by $(70)_8$, showing that BESLCI yielded values correct to 54 bits out of 60.

In the test on 60 binary places, the maximum error was found to be $(77)_8$, showing that at least 54 binary places are correct. (Indeed, the first 57 places were correct in 94% of the cases.)

The test on the two-term ascending series used only the argument set C_{jk} (see above), where j and k each take the values -16 and -100. The relative error was bounded by $(60)_8$, in comparison with $(70)_8$ for the test described above.

3. Testing of Error Return

The following arguments $z=x+iy$ were used:

$$\begin{array}{ll} 2^j+i\cdot 2^k & j,k=-9(3)12 \\ 2^j & j\ =-9(3)12 \\ i\cdot 2^k & k\ =-9(3)12; \end{array}$$

the values of NB are described in section 1. After the J's were calculated using $x+iy$ as the argument, the I's were calculated using $y+ix$. This showed that BESLCI gave results in accord (to 60 bits) with the equations

$$J_n(\bar{z})=\overline{J_n(z)}, \qquad I_n(z)=(-i)^n J_n(iz).$$

In the test on Bessel function values of order NCALC–1, the largest error was 5 multiples of the last bit; most were zero.

In the test on order $[|z|]$, the errors were larger, owing to the greater number of back-recursion steps. To calculate $J_{4096}(4096+i\cdot 2^{-9})$, for example, the subroutine call with NB$=5600$ involved approximately 900 more steps than the call with NB$=4097$. Bearing in mind that each step includes 2 multiplications and 3 additions for both the real and imaginary parts of the function value and also that the computer uses truncation in floating-point arithmetic rather than roundoff, we see that the error of $(2234)_8=1180$ in the last bit in the real part of $J_{4096}(4096+i\cdot 2^{-9})$ is to be expected. The imaginary part of $J_{4096}(4096+i\cdot 2^{-9})$ was about 2^{-15} times the size of the real part, and the bit error, $(72461407)_8$, was about 2^{14} times as big, which, again, is to be expected.

The errors mentioned in the preceding paragraphs were the worst cases. For all other arguments, the errors were within the expected tolerance, based on the number of recursion steps, and the relative sizes of x and y.

4. Summary and Conclusions

For $n=0(1)10$ and the values of z tested, BESLCI yielded $J_n(z)$ and $I_n(z)$ on the UNIVAC 1108 with either a relative error or an absolute error bounded by 10^{-16}. The latter assessment applied when $n < |z|$ and the magnitude of the real or imaginary part of the Bessel function value being tested is less than $\frac{1}{2}$; the former applies in all other cases.

The error return feature was tested and found to be executed accurately in all cases. This test also confirmed the error bounds described above for values of z in the square $|\mathrm{Re}\, z| < 64$ and $|\mathrm{Im}\, z| < 64$.

For other values of n and z, similar accuracy may be expected since the tests we have used include large and small values of $|z|$ and $|z|/n$.

Calculations were performed on a UNIVAC 1108 under EXEC 2. All were carried out with double-precision mantissa of 60 bits, corresponding to just over 18 figures. Subroutine executions took approximately 0.5N milliseconds, where

$$N = \max(|z|, \mathrm{NB}) + 10.$$

5. References and Note

[1] Algorithm BESLCI, see previous paper.

[2] Maximon, Leonard C., FORTRAN Program for Arbitrary, Precision Arithmetic, unpublished data.

[3] A linear congruential generator $X_{n+1} = aXn \pmod{2^{60}}$ with $Xo = a = 5^{13}$. The 60 bits generated were used as the mantissa, and the complement of the leftmost bit taken for the sign bit.

(Paper 77B3&4–388)