2005 International Symposium on Nonlinear
Theory and its Applications (NOLTA2005)
Bruges, Belgium, October 18-21, 2005

Validated Computation of Bessel functions

Nobito Yamamoto† and Nozomu Matsuda†

†Department of Computer Science, The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585 Japan
Email: yamamoto@im.uec.ac.jp, matsuda@sazae.im.uec.ac.jp

Abstract—We propose a method to compute Bessel functions with guaranteed accuracy, which works on MATLAB. Using multiple-precision arithmetic, the method gives as precise results as one wants together with information of how precise the results are. When it is sufficient to get the results in double precision, one can use a fast version.

1. Introduction

Both validated computation and multiple-precision have close relationship with quality of computation. Validated computation insures the quality, and multiple-precision improves it. There are two situations in which both of them are used together.

1. High accuracy is required, and the result should be obtained in multiple-precision. Moreover, you want to know which digit contains error.

2. Getting the result in a standard precision, say double precision, the magnitude of the error is asked to be less than several times of the machine epsilon for the double precision. But you may have a larger error if the calculation of the whole process is carried out in double precision.

In the first case, validated computation tells us a bound of the error and up to which digit can be regarded as precise. In the second case, validated computation using enough long digits is recommended.

We propose here a method which gives validated values of Bessel functions together with the following features.

(1) The outputs are as accurate as the user wants.

(2) Fast calculation can be done for the outputs in double precision.

(3) Even if the variables are interval ones, the outputs have the features (1) and (2).

For the calculation of Bessel functions, it is a standard way to use the recursion formula. However we donot use it because of certain restrictions on error estimation of the recursion. Instead the power series expansion of Bessel function is adopted. Since the calculation of the power series causes a large amount of rounding errors, it should be carried out with multiple-precision even for the results of the double precision. Combining the feature (1), it can be said that we are in both the situations 1 and 2.

The programs are written using INTLAB, an library for interval arithmetics with validated computation which works on Matlab.

INTLAB has built-in functions which retern results in interval form with guaranteed accuracy, but they cover only elementary functions. Special functions, e.g. Bessel function, Gamma function and so on, are not built in INTLAB. Nevertheless special functions appear in many cases of numerical computation. For example, if you want to get a validated solution to one of Poisson equations in a circle domain, using a numerical verification method based on the Bessel-Fourier expansion is a considerable way. Then you need a method which returns the values of Bessel functions with guaranteed accuracy.

Our method supplies the following properties.

- ◇ The first and the second kind of Bessel functions with integer parameters n can be evaluated with guaranteed accuracy.

- ◇ If the user specifies the range of the parameter n and the variable x, then he can use a fast version of our method to get a result which has a bound of error up to several times of the machine epsilon of the double precision.

- ◇ If the user specifies the degree of accuracy, the method returns a validated result of that accuracy.

- ◇ The derivatives can also be calculated with guaranteed accuracy.

2. Method for Validation

Let $J_n(x)$ and $Y_n(x)$ be the first and the second kind of Bessel functions, respectively. When the parameter n is an integer, $J_n(x)$ and its derivatives have the power expansions as follows.

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!(n+k)!}\left(\frac{x}{2}\right)^{n+2k} \quad (1)$$

$$J_n^{(m)}(x) = \sum_{k=s}^{\infty} \frac{(-1)^k (n+2k)!}{2^m \cdot k!(n+k)!(n+2k-m)!}\left(\frac{x}{2}\right)^{n+2k-m} \quad (2)$$

Here,

$$s = \begin{cases} 0 & (m \le n) \\ \lceil \frac{m-n}{2} \rceil & (m > n) \end{cases}, \tag{3}$$

and $\lceil X \rceil$ means the smallest integer which is larger than or equal to X.

When the parameter n is negative, we have

$$J_{-n}^{(m)}(x) = (-1)^n J_n^{(m)}(x). \tag{4}$$

Validated calculation of $J_n^{(m)}(x)$ (including the case $m = 0$) can be carried out if an upper bound of the error of truncating the power expansion is given. We estimate the bound of the truncation error as follows.

Let

$$d_k = \frac{(-1)^k (n+2k)!}{2^m \cdot k!(n+k)!(n+2k-m)!} (\frac{x}{2})^{n+2k-m}, \tag{5}$$

then

$$d_k = -\frac{(n+2k)(n+2k-1)x^2}{4k(n+k)(n+2k-m)(n+2k-m-1)} \cdot d_{k-1} \tag{6}$$

holds. Put

$$c_k = \frac{(n+2k)(n+2k-1)x^2}{4k(n+k)(n+2k-m)(n+2k-m-1)}, \tag{7}$$

which appears in the right-hand side of (6). Note that

$$c_k = x^2/\{4k(n+k)(1 - \frac{m}{n+2k})(1 - \frac{m}{n+2k-1})\}, \tag{8}$$

then we find that the sequence $\{c_k\}$ is monotonously decreasing. Therefore there exists a positive integer K such that

$$|d_k| \le |d_{k-1}|, \quad \forall k \ge K \tag{9}$$

holds. For integers k larger than K, the sequence $\{d_k\}$ is an alternating one with monotonously decreasing absolute values. Then we can estimate a bound of the truncation error by

$$|J_n^{(m)}(x) - \sum_{k=s}^{K} d_k| < |d_{K+1}| \quad . \tag{10}$$

The second kind of Bessel function $Y_n(x)$ can be evaluated in a similar manner. Hereafter we describe our method only for $J_n(x)$ since the method for $Y_n(x)$ is almost same.

3. Using multiple-precision arithmetic

We may have a too wide interval as a result of validated computation of $J_n(x)$ by using the power expansion, since summing up an alternating series causes a large amount of rounding errors. To make the influence of the rounding errors as small as enough, multiple-precision arithmetic is adopted for the summation.

INTLAB has a type for multiple-precision. In our method, the user specifies the length of digits of the variable x as a variable of the multiple-precision type. One can download the latest version of INTLAB from

http://www.ti3.tu-harburg.de/ rump/intlab/

How long digits we need is estimated as follows, for example.

Suppose that it is asked that the relative error of $J_n(x)$ should be less than or around the machine epsilon of the double precision. The enough length of the digits of d_k depends on n and x. Take K as the largest integer k which satisfies

$$k < \frac{1}{2}\sqrt{n^2 + x^2} - \frac{1}{2}n, \tag{11}$$

then, from (7), the absolute value of d_k takes the maximum at $k = K$ and so the radius of the interval value of d_k is. Thus we decide the length of the digits in order that the radius of d_K is small enough compared with the absolute value of $J_n(x)$. For the decimal numbers, the length of the malti-precision variable x should be an integer larger than

$$\log_{10} |\frac{d_K}{\varepsilon \cdot J_n(x)}| \quad , \tag{12}$$

where ε is the machine epsilon of the double precision. Here we need the value of $J_n(x)$, but it does not have to be a validated value. The built-in function in Matlab 'besselj(n, x)' can be used.

Our numerical experiments show that until $|x| < 100$ the above strategy works well. But for larger $|x|$, the results might give less precision. Therefore some trials are necessary to decide the length of the digits for x in general.

4. Mean value form

Using the multiple-precision arithmetic makes the radius of the results small enough for point values of x that are not intervals. But when we take x as an interval, even if the radius is very small, the relative error of $J_n(x)$ becomes catastrophically large. This comes from interval inflation which is one of the properties of the interval arithmetic. To make the interval inflation as small as possible, we adopt an inclusion so called the mean value form.

Let $f(x)$ be a differentiable function of x, and x is given as an interval. Consider an interval enclosure $[f(x)]$ such that

$$f(x) \subset [f(x)] \tag{13}$$

holds for the set $f(x)$. We want $[J_n(x)]$ whose radius is small enough compared with the results obtained by the interval extension of our method.

Suppose we have obtained an interval enclosure $[f'(x)]$ for the derivative $f'(x)$ by a certain method. Let us take $F(x)$ as

$$F(x) = f(mid(x)) + [f'(x)](x - mid(x)), \tag{14}$$

then we call $F(x)$ the mean value form of $f(x)$. Here $mid(x)$ means the center of the interval x. The mean value theorem shows that $F(x)$ is an interval enclosure of $f(x)$.

When the radius of x is sufficiently small, the mean value form often gives a better result than the interval extension of $f(x)$.

Applying the mean value form to the computation of $J_n(x)$, we have

$$J_n(mid(x)) + [J'_n(x)](x - mid(x)) \quad . \tag{15}$$

But the numerical experiments shows that it may be insufficient for some pairs of n and x. The relative error of $J_n(x)$ is getting larger when $|n|$ becomes smaller and $|x|$ becomes larger. We have still large errors even if we apply the mean value form once for certain small n's and large x's. To improve the results, we apply the mean value form recursively as follows.

$$J_n(mid(x)) + (J'_n(mid(x)) + (J''_n(mid(x)) + (\cdots \\ )(x - mid(x)))(x - mid(x)))(x - mid(x)) \quad .$$

The user specifies the number of the recurrence. The values of the derivatives can be computed with guaranteed accuracy using multiple-precision arithmetic as is mentioned above. Note that

$$|J_n^{(m)}(x)| \leq 1 \tag{16}$$

holds for arbitrary integers n, m and an arbitrary real number x. This allows us to bound the absolute value of the derivative of the highest degree by 1.

5. Fast version of the computation

In order to reduce CPU time, we make a table of values of Bessel function and use Chebyshev interpolation. The results are restricted within an accuracy corresponding to the machine epsilon of the double precision.

Previously the user has to specify the range of the parameter n and the variable x. We make a table using validated values of Bessel functions on each n and selected points of x. For an arbitrary point (n,x) within the range, $J_n(x)$ is calculated by validated Chebyshev interpolation.

Chebyshev interpolation

Take m points in an interval region $[a, b]$ as follows.

$$\xi_i = \frac{a+b}{2} + \frac{b-a}{2} \cos \frac{\pi}{m}(i + \frac{1}{2}), \quad (i = 0, 1, \cdots, m-1) \quad . \tag{17}$$

These points are so called Chebyshev points. Define Chebyshev polynomial by

$$T_k(t) = \cos(k \cos^{-1}(t)). \tag{18}$$

Then we have Chebyshev interpolation $p(x)$ of a function $f(x)$ on $[a, b]$ by

$$p(x) = \frac{1}{2}c_0 + \sum_{k=1}^{m-1} c_k T_k(t), \quad t = (x - \frac{a+b}{2})/\frac{b-a}{2}, \tag{19}$$

where the coefficient c_k is as follows.

$$c_k = \frac{2}{m} \sum_{i=0}^{m-1} f(\xi_i) \cos(\frac{2i+1}{2m} k\pi) \quad . \tag{20}$$

We need error esitmation of Chebyshev interpolation for validated computation.

$$\|f(x) - p(x)\|_{C[a,b]} \leq \frac{2(b-a)^m}{4^m \cdot m!} \|f^{(m)}(x)\|_{C[a,b]} =: E \quad . \tag{21}$$

Here

$$\|f(x)\|_{C[a,b]} = \max_{a \leq x \leq b} |f(x)| \quad . \tag{22}$$

The table contains the values of c_k and E for corresponding Chebyshev points. Once we have made the table, a fast computation of the validation can be carried out as long as (n,x) is within the range.

When the variable x has an interval value, there occurs an increasing of the error because of interval inflation. Thus we have to apply the mean value form to Chebyshev interpolation, that is,

$$J_n(mid(x)) + J'_n(x)(x - mid(x)) \tag{23}$$

with

$$|J'_n(x)| \leq 1. \tag{24}$$

6. Numerical experiments

We test the efficiency of multiple-precision and the mean value form for $n = 0$ where the error of $J_n(x)$ takes the maximum.

Efficiency of multiple-precision

Table 1 shows the centers and the radii of intervals including $J_0(100)$ with respect to the number of bits of multiple-precision binary numbers.

Table 1. Values of $J_0(100)$ according to the digits of multiple-precision numbers

	$J_0(100)$	
	center	radius
92	$-6.075215159834 \times 10^{18}$	$7.673845534663 \times 10^{21}$
115	$-1.407272878080 \times 10^{11}$	$9.147936743096 \times 10^{14}$
138	$1.027710000000 \times 10^{5}$	$1.258291200000 \times 10^{8}$
161	$3.491795063019 \times 10^{-2}$	$1.300000000000 \times 10^{1}$
184	$1.998585024351 \times 10^{-2}$	$1.668930053711 \times 10^{-6}$
207	$1.998585030422 \times 10^{-2}$	$1.847411112976 \times 10^{-13}$
230	$1.998585030422 \times 10^{-2}$	$2.032879073410 \times 10^{-20}$
253	$1.998585030422 \times 10^{-2}$	$2.827277484312 \times 10^{-27}$
276	$1.998585030422 \times 10^{-2}$	$2.648153673532 \times 10^{-34}$
299	$1.998585030422 \times 10^{-2}$	$4.017802956912 \times 10^{-41}$

Efficiency of the mean value form

Table 2 shows the centers and the radii of intervals including $J_0(x)$ with respect to the number of recurrence of the mean value form. Here x is an interval with the center 100 and the radius 0.1.

Table 2. Values of $J_0(x)$ according to recurrence of mean value form

	$J_0(x)$	
	center	radius
0	$-5.312662293228 \times 10^{21}$	$3.466232109999 \times 10^{28}$
1	$1.998585030421 \times 10^{-2}$	$1.000000238419 \times 10^{-1}$
2	$1.998585030421 \times 10^{-2}$	$1.771461963654 \times 10^{-2}$
3	$1.998585030421 \times 10^{-2}$	$8.922219276428 \times 10^{-3}$
4	$1.998585030421 \times 10^{-2}$	$8.099079132080 \times 10^{-3}$
5	$1.998585030421 \times 10^{-2}$	$8.011221885681 \times 10^{-3}$
6	$1.998585030421 \times 10^{-2}$	$8.002996444702 \times 10^{-3}$
7	$1.998585030421 \times 10^{-2}$	$8.002161979675 \times 10^{-3}$
8	$1.998585030421 \times 10^{-2}$	$8.002042770386 \times 10^{-3}$
9	$1.998585030421 \times 10^{-2}$	$8.002042770386 \times 10^{-3}$
10	$1.998585030421 \times 10^{-2}$	$8.002042770386 \times 10^{-3}$

We can see that 3 or 4 times of recurrence are sufficient.

Verification of the recursion formula

Bessel functions satisfy the following recursion formula.

$$J_{n+1}(x) = \frac{2n}{x} J_n(x) - J_{n-1}(x). \tag{25}$$

Let us verify this. Table 3 shows the interval values of $J_0(10)$, $J_1(10)$, $J_2(10)$ and the result for $J_2(10)$ from the recursion formula (denoted by *rec* in the tables). Table 4 gives a view of overlap between the results for $J_2(10)$.

Table 3. $J_2(10)$ and the result from the recursion formula

	center	radius
J_0	$-2.459357644513 \times 10^{-1}$	$1.524659305058 \times 10^{-20}$
J_1	$4.347274616886 \times 10^{-2}$	$1.863472483959 \times 10^{-20}$
J_2	$2.546303136851 \times 10^{-1}$	$2.202285662861 \times 10^{-20}$
rec	$2.546303136851 \times 10^{-1}$	$1.863472483959 \times 10^{-20}$

Table 4. Overlap of the results for $J_2(10)$

J_2		*rec*
Upper	0.254630313685120622554051 0	
	0.254630313685120622551001 6	Upper
	0.254630313685120622513732 2	Lower
Lower	0.254630313685120622510005 3	

We can see that the intervals have common part, which is necessary for validated computation.

Verification of Bessel equation

Bessel functions satisfy Bessel equations as follows.

$$x^2 J_n''(x) + x J_n'(x) + (x^2 - n^2) J_n(x) = 0. \tag{26}$$

Table 5 shows the result of validated computation of the left-hand side (denoted by $left$ in the table), which must include 0. Here $n = 5$ and $x = 10$.

Table 5. Verification of the Bessel equation at $n = 5, x = 10$

	center	radius
J_5	$-2.340615281868 \times 10^{-1}$	$2.202285662861 \times 10^{-20}$
J_5'	$-1.025719220086 \times 10^{-1}$	$8.300922883092 \times 10^{-20}$
J_5''	$1.858033383410 \times 10^{-1}$	$3.083199928006 \times 10^{-19}$
$left$	$-2.930733997500 \times 10^{-19}$	$3.309527131512 \times 10^{-17}$

It is verified that the left-hand side includes 0.

Exact values up to 500 digits

The following number is the value of the derivative of the 20th degree at $n = 10$, $x = 30$. It is calculated such that the radius of the interval result is less than 1×10^{-500}. Therefore it has exact digits up to the 500th digit.

$J_{10}^{(20)}(30) =$
$-0.044087165659750665184051294870744890205631247733960153340369465129889891893975259825371960367118446167285134441986594501286557223523554862232244594952557949850675760421372674315082336977386946252927907180976255523570076435821608918849927589495322805631693577640601475497839027348336428192046791725548237653597163771360515106658502591403097840359961815640517750884912897206270929372335777611846675981170858993746421623901505020895513580426540355635418875558897531256962921787350853367553077000456389 01