

NUMERICAL CALCULATION OF BESSEL FUNCTIONS

CHARLES SCHWARTZ

*Department of Physics, University of California
Berkeley, California 94720, USA
schwartz@physics.berkeley.edu*

Received 11 September 2012
Accepted 24 September 2012
Published 12 November 2012

A new computational procedure is offered to provide simple, accurate and flexible methods for using modern computers to give numerical evaluations of the various Bessel functions. The trapezoidal rule, applied to suitable integral representations, may become the method of choice for evaluation of the many special functions of mathematical physics.

Keywords: Bessel functions; trapezoidal rule; calculation of special functions.

PACS Nos.: 02.30.Gp, 02.60.Jh.

1. $K_\nu(z)$

Let me start with this nice integral representation for one type of Bessel function,

$$K_\nu(z) = \int_0^\infty dt \cosh(\nu t) e^{-z \cosh t}. \quad (1.1)$$

See Appendix A for reference.

What we offer is a particular technique of numerical integration that is simple and rapidly convergent for infinite integrals of smooth functions: the trapezoidal rule, wisely applied. The general theorem¹ says the following.

$$\sum_{n=-\infty}^{\infty} h f(nh) = \int_{-\infty}^{\infty} dx f(x) + \mathcal{E}, \quad (1.2)$$

$$\mathcal{E} = \sum_{m=1}^{\infty} \left[F\left(\frac{2\pi m}{h}\right) + F\left(-\frac{2\pi m}{h}\right) \right], \quad (1.3)$$

where F is the Fourier transform of the function f . If $f(x)$ is quite smooth, then its Fourier transform will decrease rapidly as the parameter h decreases.

Another way of describing this method is that we see all of the correction terms in the Euler–Maclaurin series vanish. This means that as the interval h decreases, the error in the trapezoidal rule decreases faster than any power of h . Precisely what that

formula for the convergence rate is depends on the detailed analytic behavior of the function $f(x)$. Typically, one sees some kind of exponential decrease: $\mathcal{E} \sim e^{-\alpha h^{-\beta}}$. I do not offer any general theory about this formula, although particular models have been studied in the past.^{1,2} For example, if $\beta = 1$, one sees the number of correct decimal places double as one halves the interval h . An essential virtue of this overall method is that the practitioner can see what the convergence looks like and can decide when to extend the calculation for better accuracy or to quit. The examples shown in the tables below offer such displays of rapidly converging output data.

For this method to be practical, one also needs the integrand to decrease rapidly as the integration variable goes toward infinity; and the sum over mesh points is just cut off when the contributions drop below the desired accuracy. This is sometimes helped by accelerating the rate of decay through a change of integration variable; and we shall show an example of this technique in Sec. 3.

For readers who are new to this method, I recommend the simple exercise of computing the infinite Gaussian integral, $\int dx e^{-x^2}$, using the trapezoidal rule (1.2). Here, the error formula turns out to be $\mathcal{E} \sim e^{-\pi^2/h^2}$. A recent mathematical review of this general technique may be found here.²

Applying this numerical technique directly to Eq. (1.1), we obtain the results shown in Tables 1(a) and 1(b). This looks quite good: rapid convergence to high accuracy at modest cost. The simple code for these calculations is shown in Appendix B; arbitrary (real, positive) values of z and ν may be input.

The number of mesh points used for the last line of data in Table 1(a) was 109, 73, 39, respectively.

The number of mesh points used for the last line of data in Table 1(b) was 150, 76, 31, respectively.

Table 1(a). Computations of Eq. (1.1) using method (1.2).

$1/h$	$K_0(z = 0.1)$	$K_0(z = 1.0) \times 10$	$K_0(z = 10) \times 10^5$
1	2.42704 11398 56250	4.20936 51061 48591	2.28987 96730 52002
2	2.42706 90280 99576	4.21024 43651 11141	1.77845 16875 44865
4	2.42706 90247 02016	4.21024 43824 07083	1.77800 62316 16017
8	2.42706 90247 02016	4.21024 43824 07082	1.77800 62316 16764
16	2.42706 90247 02016	4.21024 43824 07086	1.77800 62316 16764

Table 1(b). Computations of Eq. (1.1) using method (1.2).

$1/h$	$K_{2.718}(z = 0.01) \times 10^{-6}$	$K_{2.718}(z = 1.0)$	$K_{2.718}(z = 100) \times 10^{45}$
1	1.39714 10533 21390	4.54996 20838 02887	
2	1.40690 20983 29460	4.49904 64843 96175	9.30030 05343 36706
4	1.40690 07287 78440	4.49903 44319 18784	5.14859 62786 90992
8	1.40690 07287 78468	4.49903 44319 18744	4.83095 95172 64883
16	1.40690 07287 78469	4.49903 44319 18749	4.83095 57412 19501
32			4.83095 57412 19519

2. $J_\nu(x)$ and $N_\nu(x)$

To get at the other Bessel functions we need to move our variables into the complex plane. Here is one standard relation³:

$$H_\nu^{(1)}(x) = J_\nu(x) + iN_\nu(x) = \frac{2}{i\pi} e^{-i\nu\pi/2} K_\nu(-ix) \tag{2.1}$$

and for now I'll take x as a real variable. This function $N_\nu(x)$ is called $Y_\nu(x)$ by some authors.

Using the integral representation (1.1) now would not work well because what was previously a decaying exponential function of t is now an oscillatory function. However, we can move the contour of integration for that variable t so that it goes toward the line $\text{Im}(t) = \pi/2$; and this will restore the real exponential decay of the integrand at large distances. Here is one prescription:

$$t = \sinh(u) + \frac{i\pi}{2} \tanh(u) \tag{2.2}$$

and u is a new real integration variable which we will treat according to the general method of Eq. (1.2). For an alternative approach, see Appendix C.

The simple program written for calculating $K_\nu(x)$ is now expanded to accommodate complex variables; this runs to about 50 lines of code in C and some numerical results are shown in Tables 2 and 3. The convergence looks quite good.

Table 2. Calculation of $J_\nu(x)$ from Eq. (2.1) using (2.2) in (1.1).

$1/h$	$J_1(x = 0.1) \times 10^2$	$J_1(x = 1) \times 10$	$J_1(x = 10) \times 10^2$
1	31.34519 12483 38983	4.84678 01345 03115	-0.49254 74998 14282
2	10.04145 83361 50352	4.40211 90106 01766	1.00684 96120 06995
4	4.66764 17956 85866	4.40051 65097 30195	4.31962 34829 07726
8	4.99456 06293 39347	4.40050 58776 70964	4.34727 46212 95072
16	4.99375 25888 30283	4.40050 58574 49333	4.34727 46168 86134
32	4.99375 26036 24231	4.40050 58574 49336	4.34727 46168 86136
64	4.99375 26036 24215	4.40050 58574 49336	4.34727 46168 86136

Table 3. Calculation of $N_\nu(x)$ from Eq. (2.1) using (2.2) in (1.1).

$1/h$	$N_1(x = 0.1)$	$N_1(x = 1) \times 10$	$N_1(x = 10) \times 10$
1	-8.81445 14807 36515	-8.76505 96245 40165	5.92701 15605 77872
2	-6.94259 74076 35009	-7.79957 53906 29861	3.00580 17631 63178
4	-6.44231 94399 89834	-7.81226 14661 84539	2.48848 46077 69873
8	-6.45895 10404 44470	-7.81212 82132 14771	2.49015 42425 77341
16	-6.45895 10946 34644	-7.81212 82130 02891	2.49015 42420 69539
32	-6.45895 10947 02030	-7.81212 82130 02888	2.49015 42420 69539
64	-6.45895 10947 02026	-7.81212 82130 02890	2.49015 42420 69538

The number of mesh points used for the last line of data in these tables was 170, 143, 100, respectively.

3. $I_\nu(x)$

One can say that this Bessel function can almost always be effectively evaluated by using the power series. But, to complete the program offered in this paper, we should give an approach through an integral representation. Here is one, over an infinite range, which is found in the standard references.⁴

$$I_\nu(x) = \frac{1}{2\pi i} \int_{\infty-i\pi}^{\infty+i\pi} dt e^{\nu t} e^{x \cosh(t)}, \tag{3.1}$$

which we can represent by the contour

$$t = \cosh(u) + c + i\pi \tanh(u), \tag{3.2}$$

with the variable u treated by Eq. (1.2). The constant c is arbitrary but if we choose $c = -1 + \sinh^{-1}(\nu/x)$, then the point of stationary phase occurs at $u = 0$.

This works numerically but not quite as nicely as our previous examples: there is greater loss of accuracy due to the oscillations of the integrand, especially at small values of x .

So, let's look at an alternative integral representation for this Bessel function:

$$I_\nu(x) = \frac{(x/2)^\nu}{\sqrt{\pi}\Gamma(\nu + 1/2)} \int_0^\pi d\theta \sin^{2\nu}\theta \cosh(x \cos \theta). \tag{3.3}$$

This is a purely real integral, with no oscillations, and the factor out in front takes care of the behavior at small x . But, Is it amenable to the trapezoidal rule, Eq. (1.2), for effective numerical integration? If the index ν is an integer, the answer is yes, because the integrand is a periodic function over the interval of integration.

For more general application, however, we change variables to map the integration onto the entire real line:

$$\cos \theta = t = \tanh(u) \tag{3.4}$$

and then treat the integration over u by the trapezoidal rule. Some results of this calculation are shown in Table 4.

Table 4. Calculation of Eq. (3.3) using (3.4) and (1.2).

$1/h$	$I_2(z = 0.01) \times 10^5$	$I_2(z = 1.0) \times 10$	$I_2(z = 100) \times 10^{-42}$
1	1.30643 43443 38658	1.38166 26095 67052	1.20677 58487 72873
2	1.25004 98297 92758	1.35743 37442 58898	1.05923 08945 34915
4	1.25001 04167 00829	1.35747 66976 67279	1.05238 50353 94902
8	1.25001 04166 99218	1.35747 66976 70383	1.05238 43193 24316
16	1.25001 04166 99218	1.35747 66976 70383	1.05238 43193 24312

The number of mesh points used for the last line of data in Table 4 was 153, 154, 184, respectively.

In all cases one may try to accelerate the rate of decay of the integral by such further transformations as $u = \sinh(v)$ or $u = v^3$. For the same accuracy as shown in Table 4, this technique reduces the number of mesh points needed by a factor of 2 or 3. For the data shown in previous tables, this did not produce improvement.

All numerical results given in this paper were obtained by calculations in ordinary double precision (16 decimals).

For the general use of Eq. (3.3) one needs accurate evaluation of the Gamma function for arbitrary argument; and that particular topic is mentioned in the following section.

4. Discussion

There are standard libraries available for the numerical computation of Bessel functions, which rely on a variety of old techniques.^a They use power series for small argument, asymptotic series for large arguments, various schemes involving recurrence relations and interpolation for intermediate arguments. As far as I can tell, the technique presented here, which seems to cover all those bases in one grand sweep, is new and has not been recognized before, although the general principles behind this technique have been known for some time.

This technique appears to be uniquely valuable on several accounts: it gives high accuracy in rapid time; it is flexibly applicable to all sorts of Bessel functions; it is simple to program, relying on the standard computer routines for efficient evaluation of ordinary trigonometric functions (with real or complex arguments). It also invites the programmer (mathematician, physicist, engineer, or student) to be completely in charge of the analytical/numerical process, rather than relying on a big black box provided by some remote experts.

There is room for further exploration of various contours of the integration, since the simple ones used above may not be the most efficient. I expect that this general approach can be used for arbitrary complex values of the argument z and order ν , although I have not investigated those ideas. There are also some difficult problems in remote corners of Bessel function parameters⁵; and whether or not the present method might be better in all cases is an open question.

Finally, one can reasonably expect that other types of “special functions,” beyond Bessel, can be efficiently evaluated using these ideas if one has nice integral representations to start with.

For one example, the Gamma function can be nicely computed by using a contour integral:

$$\frac{1}{\Gamma(z)} = \frac{1}{2\pi i} \int dt e^{t} t^{-z}, \quad t = 2 - \cosh(u) + i \sinh(u) \quad (4.1)$$

^aI have been able to look into the open source libraries GSL and Ceres; how these things are done by Mathematica is unknown to me.

Table 5. Calculation of $\Gamma(z)$ using (4.1) and (1.2).

$1/h$	$\Gamma(z = 0.1)$	$\Gamma(z = 1 + i10)$
2	9.55495 13353 97527	
4	9.51350 56276 59931	3.88422 34738 42538e-07 + i1.13738 21528 01491e-06
8	9.51350 76986 68703	3.91892 92710 32289e-07 + i1.12844 79696 26640e-06
16	9.51350 76986 68744	3.91892 92708 81460e-07 + i1.12844 79695 84611e-06
32	9.51350 76986 68734	3.91892 92708 81405e-07 + i1.12844 79695 84617e-06
64		3.91892 92708 81394e-07 + i1.12844 79695 84628e-06

and the infinite integral over the real variable u is treated according to (1.2). See Table 5 for some numerical results.

Another famous function that is nicely handled this way is the zeta function.

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \frac{1}{2\Gamma(s)} \int_{-\infty}^{\infty} du e^{su} \frac{e^{-t/2}}{\sinh(t/2)}, \quad t = e^u. \quad (4.2)$$

With a further acceleration by the change of variables $u = \sinh(v)$, this gives efficient calculations with the trapezoidal rule.

Appendix A

Equation (1.1) may not be familiar. In fact, I do not find it in my standard reference book.³ So, let me show that it is correct. Apply the differential operator:

$$\begin{aligned} z^2 \left[\frac{d^2}{dz^2} + \frac{1}{z} \frac{d}{dz} - \frac{\nu^2}{z^2} - 1 \right] K_\nu(z) &= \int_0^\infty dt \cosh(\nu t) [z^2 \cosh^2 t - z \cosh t - \nu^2 - z^2] e^{-z \cosh t} \\ &= \int_0^\infty dt \cosh(\nu t) \left[\frac{d^2}{dt^2} - \nu^2 \right] e^{-z \cosh t} \end{aligned} \quad (A.1)$$

and that equals zero. So this is some Bessel function. Now look at how this integral behaves as $z \rightarrow \infty$: It will be dominated by small values of t and we readily find,

$$K_\nu(z) \rightarrow \sqrt{\frac{\pi}{2z}} e^{-z}, \quad (A.2)$$

which correctly identifies this particular Bessel function.

Appendix B

Here is the code used for calculation of the data in Tables 1(a) and 1(b).

```
#include <stdio.h>
#include <math.h>
int main (void)
{double z,t,term,sum,h,nu;
```

```

int n,k;
do {printf("Enter nu and z for Bessel function K_nu(z): ");
scanf("%lf %lf",&nu,&z);
h=2.0;
for(k=1;k<=8;k++) /* sequence of decreasing values of h */
{t=0.; /* the first point of the integral */
sum=0.5*exp(-z);
n=1;
do {t+=h; /* successive points of the integral */
term=cosh(nu*t)*exp(-z*cosh(t));
sum+=term;
n++;}
while(term/sum> 1.e-20); /* deciding when to quit */
printf("h= %lf, n=%d, ans = %24.16e\n",h, n, h*sum);
h=h/2.;}
} while(1);
return 0;}

```

Appendix C

Let me explore an alternative contour to the one given in Eq. (2.2). Move the contour integral of t — which originally went out along the real axis from zero to infinity — as follows. First leg is $t = (0, 0)$ to $t = (0, i\pi/2)$. Second leg is $t = (0, i\pi/2)$ to $t = (\infty, i\pi/2)$.

It will simplify this discussion if I just take the special case of $\nu = 0$. The result of doing this leads to the following integral formulas.

$$J_0(x) = \frac{2}{\pi} \int_0^{\pi/2} d\theta \cos(x \cos \theta), \tag{C.1}$$

$$N_0(x) = \frac{2}{\pi} \int_0^{\pi/2} d\theta \sin(x \cos \theta) - \frac{2}{\pi} \int_0^\infty ds e^{-x \sinh(s)}. \tag{C.2}$$

These formulas are found in the standard literature.³ But, are these integrals suitable for the particularly powerful method of numerical integration discussed in this paper? For J_0 the answer is yes; but this is only for special cases of the index ν ; and it works because the integrand is a periodic function. For N_0 the answer is no. Neither of the two integrals there satisfies the conditions for our method (although one could probably change the integration variables and make them conform to the type desired). An interesting question was whether the end-point correction terms from each of the two integrals at the pivot point $t = (0, i\pi/2)$ might cancel; and the answer appears to be negative. This is interpreted as emphasizing the role of continuous (analytic) functions and variables in getting the power of the present integration method. The contour discussed in this Appendix has a kink in it; and that is a spoiler.

C. Schwartz

References

1. C. Schwartz, *J. Comput. Phys.* **4**, 19 (1969).
2. J. Waldvogel, *Approximation and Computation* (Springer, New York, 2011).
3. W. Magnus and F. Oberhettinger, *Formulas and Theorems for the Functions of Mathematical Physics* (Chelsea Publishing, New York, 1949).
4. National Institute of Standards and Technology, Digital Library of Mathematical Functions, <http://dlmf.nist.gov/10.32.12>.
5. U. D. Jentschura and E. Lotstedt, *Comput. Phys. Commun.* **183**, 506 (2012).