JOURNAL OF COMPUTATIONAL PHYSICS **21**, 343–350 (1976)

Note

On the Numerical Evaluation of the Ordinary Bessel Function of the Second Kind

1. Introduction

1.1. *Definitions and Relevant Properties*

The ordinary Bessel function of the first kind

$$J_\nu(z) = (z/2)^\nu \sum_{k=0}^{\infty} \frac{(-z^2/4)^k}{\Gamma(\nu + k + 1)\, k!} \tag{1.1}$$

and the ordinary Bessel function of the second kind

$$Y_\nu(z) = [\cos \nu\pi\, J_\nu(z) - J_{-\nu}(z)]/\sin \nu\pi \tag{1.2}$$

are two linearly independent solutions of the difference equation

$$f_{\nu+1} - (2\nu/z) f_\nu + f_{\nu-1} = 0. \tag{1.3}$$

This equation can be used to compute $Y_{\nu+n}$ for $n = 2, 3,\ldots$ when Y_ν and $Y_{\nu+1}$ are given. In the forward direction the recurrence formula (1.3) for Y_ν is numerically stable, whereas it is unstable for J_ν (see Gautschi [1]).

The ordinary Bessel functions of the third kind are the Hankel functions

$$H_\nu^{(1)}(z) = J_\nu(z) + iY_\nu(z),\ H_\nu^{(2)}(z) = J_\nu(z) - iY_\nu(z). \tag{1.4}$$

Important for the representation of the Hankel functions for large $|z|$ are the functions $P(\nu, z)$ and $Q(\nu, z)$ defined by

$$H_\nu^{(1,2)}(z) = [2/(\pi z)]^{1/2}\, e^{\pm i\chi}[P(\nu, z) \pm iQ(\nu, z)], \tag{1.5}$$

where the $+$ sign is used for $H_\nu^{(1)}$, the $-$ sign is used for $H_\nu^{(2)}$ and

$$\chi = z - \pi(2\nu + 1)/4. \tag{1.6}$$

For large $|z|$, P and Q are slowly varying and the oscillatory behavior of $H_\nu^{(1)}$ and

$H_\nu^{(2)}$ is contained in the exponential function in (1.5). From (1.4) and (1.5) we obtain

$$\begin{aligned} Y_\nu(z) &= [2/(\pi z)]^{1/2}\,[P(\nu, z)\sin\chi + Q(\nu, z)\cos\chi] \\ J_\nu(z) &= [2/(\pi z)]^{1/2}\,[P(\nu, z)\cos\chi - Q(\nu, z)\sin\chi]. \end{aligned} \tag{1.7}$$

Again, the oscillatory behavior of J_ν and Y_ν is fully described by the circular functions in (1.7).

The connection between the ordinary Bessel functions and the modified Bessel functions follows from

$$\begin{aligned} H_\nu^{(1)}(z) &= -2i\pi^{-1}e^{-\nu\pi i/2}K_\nu(ze^{-i\pi/2}) \qquad (-\tfrac{1}{2}\pi < \arg z \leqslant \pi), \\ H_\nu^{(2)}(z) &= 2i\pi^{-1}e^{\nu\pi i/2}K_\nu(ze^{i\pi/2}) \qquad (-\pi < \arg z \leqslant \tfrac{1}{2}\pi). \end{aligned} \tag{1.8}$$

From the Wronskian

$$J_{\nu+1}(z)\,Y_\nu(z) - J_\nu(z)\,Y_{\nu+1}(z) = 2/(\pi z)$$

and (1.7) it easily follows that

$$P(\nu, z)\,P(\nu + 1, z) + Q(\nu, z)\,Q(\nu + 1, z) = 1. \tag{1.9}$$

1.2. *Contents of the Paper*

We give algorithms for the computation of Y_ν and $Y_{\nu+1}$ and we use the methods of our previous paper on the computation of K_ν and $K_{\nu+1}$ (see Temme [6]). Our results in [6] can be used for complex values of z. Here we give the explicit results for Y_ν and $Y_{\nu+1}$ and these results follow immediately from [6] by using (1.8).

For the computation of J_ν the reader is referred to Gautschi [1], where an algorithm is given for the computation of $J_{\nu+n}(z)$, $n = 0, 1, 2,\ldots, N$. See also Gautschi [2]. In Luke [4] rational approximations for J_ν and Y_ν are given based on Padé-representations for large $|z|$. In Luke [5] a double series of Chebyshev polynomials and values of the coefficients are given for both Y_ν, J_ν for $z \geqslant 5$. In Goldstein and Thaler [3] the computation of Y_ν is based on series expansions in ordinary Bessel functions of the first kind, but the treatment of small $|\nu|$-values is not satisfactory.

2. The Computation for Small $|z|$

In order to obtain a more symmetric representation in (1.2) we write

$$\cos\nu\pi\,J_\nu(z) - J_{-\nu}(z) = J_\nu(z) - J_{-\nu}(z) - 2\sin^2(\nu\pi/2)\,J_\nu(z). \tag{2.1}$$

Furthermore we introduce the following notation

$$\begin{aligned}
c_k &= (-z^2/4)^k/k!,\\
p_k &= (\nu/\sin \nu\pi)\,(z/2)^{-\nu}/\Gamma(k+1-\nu),\\
q_k &= (\nu/\sin \nu\pi)\,(z/2)^{\nu}/\Gamma(k+1+\nu),\\
f_k &= (p_k - q_k)/\nu,\\
g_k &= f_k + 2\nu^{-1}\sin^2(\nu\pi/2)\, q_k\,,\\
h_k &= -kg_k + p_k\,,
\end{aligned}$$

where $k = 0, 1, \ldots.$ We have for $k = 1, 2, \ldots$ the recurrence relations

$$\begin{aligned}
p_k &= p_{k-1}/(k-\nu),\ q_k = q_{k-1}/(k+\nu),\\
f_k &= (kf_{k-1} + p_{k-1} + q_{k-1})/(k^2 - \nu^2).
\end{aligned}$$

Substitution of (1.1) in (1.2) and using (2.1) yields

$$Y_\nu(z) = -\sum_{k=0}^{\infty} c_k\, g_k\,. \tag{2.2}$$

Considering (2.1) with ν replaced by $\nu + 1$ and using (1.3) we have

$$\begin{aligned}
&\cos(\nu+1)\,\pi\, J_{\nu+1}(z) - J_{-\nu-1}(z)\\
&\quad = -[J_{\nu+1}(z) - J_{-\nu+1}(z)] + (2\nu/z)\, J_{-\nu}(z) + 2\sin^2(\nu\pi/2)\, J_{\nu+1}(z).
\end{aligned}$$

We obtain by substitution of (1.1)

$$Y_{\nu+1}(z) = -(2/z)\sum_{k=0}^{\infty} c_k h_k\,. \tag{2.3}$$

As in [6], f_0 can be represented in such a way that it can be computed with a satisfactorily small relative error.

For small values of $|z|$ the series in (2.2) and (2.3) converge rapidly. But cancellation may occur in summing the series numerically. A strict error analysis, as for the modified Bessel function, can not easily be given, but from numerical experiments it turns out that for $|z| < 3$ the computation is stable.

3. The Computation for $|z| \geqslant 3$

For $|z| \geqslant 3$ we compute $P(\nu, z)$, $P(\nu+1, z)$, $Q(\nu, z)$ and $Q(\nu+1, z)$, by using the functions $k_n(z)$ introduced in our previous paper [6]. For K_ν and $K_{\nu+1}$ we needed $k_0(z)$ and $k_1(z)$. From (1.8) it turns out that for the P- and Q-functions the functions $k_0(\pm iz)$ and $k_1(\pm iz)$ can be used. The application of the method in [6] is straightforward. However, the determination of the starting index N for the Miller algorithm caused some trouble, since our error analysis in [6] was based on the case of real variables. But trying out the results of [6] for the P- and Q-functions we noticed that the determination of the starting index N can indeed be based upon the estimations given in [6].

4. ALGOL 60 Procedures

The algorithms for the computation of $Y_\nu(z)$ and $Y_{\nu+1}(z)$ are given as an ALGOL 60 procedure for the case of real values of ν and z, $z > 0$. For convenience we write $\nu = a$ and $z = x$.

The procedure *bessya* computes for $x > 0$ and $a \in \mathbb{R}$ the functions $Y_a(x)$ and $Y_{a+1}(x)$; *bessya* calls for three nonlocal procedures *sinh*, *recip gamma*, and *besspqa*. For the text of *sinh*, and *recip gamma* the reader is referred to [6]. In *besspqa* the functions $P(a, x)$, $P(a + 1, x)$, $Q(a, x)$ and $Q(a + 1, x)$ are computed. We supply *besspqa* as a separate procedure since it can also be used for the computation of the Bessel functions $J_a(x)$ and $J_{a+1}(x)$ (see (1.7)). In *bessya* the procedure *besspqa* is called for $x \geqslant 3$ and $|a| < .5$, but the algorithm in *besspqa* converges for all x and a ($x > 0$). It is recommended, however, to take $x > \max(|a|, 3)$. For $|a| > x$ the recurrence relations

$$P(a + 1, x) = P(a - 1, x) - 2a/x\, Q(a, x)$$
$$Q(a + 1, x) = Q(a - 1, x) + 2a/x\, P(a, x)$$

can be used. These relations are valid for real a and x. They can be derived by substitution of (1.5) in (1.3). However, for $|a| + 1 > x$, computation of $J_a(x)$ and $J_{a+1}(x)$ by using (1.7) will cause a loss of correct significant digits.

The precision in the procedures *bessya* and *besspqa* can be controlled by using the variable *eps*. For *besspqa* its entry value corresponds to the desired relative accuracy in *pa*, *pa* 1, *qa* and *qa* 1. Also in *bessya* it corresponds to relative accuracy, except in the neighborhoods of zeros of $Y_a(x)$ or $Y_{a+1}(x)$. In that case *ya* or *ya* 1 are given with absolute accuracy *eps*.

The procedures *bessya* and *besspqa* were tested on the CD CYBER 73 of SARA, Amsterdam. For $a = 0, 0.2, 0.4$, $x = .5, 1, 2, 3, 5, 7, 10, 20, 50, 100$ and $eps = 10^{-15}$ we checked relation (1.9). The output of $|pa.pa\,1 + qa.qa\,1 - 1|$ is given in Table I. The procedure *bessya* was also tested in the neighborhood of $x = 3$. For $x^{\pm} = 3 \pm 2^{-46}$ we computed the numerical values of the expressions

$$d_0 = \{Y_a(x^-) - Y_a(x^+)\},$$
$$d_1 = \{Y_{a+1}(x^-) - Y_{a+1}(x^+)\}.$$

In Table II we give d_0, d_1, the maximum number of terms (n) used in (2.1), and the starting index N for the Miller algorithm.

TABLE I

x \ a	0.0	0.2	0.4
0.5	$1.4_{10} - 14$	$7.1_{10} - 15$	$0.0_{10} + 00$
1.0	$0.0_{10} + 00$	$7.1_{10} - 15$	$7.1_{10} - 15$
2.0	$7.1_{10} - 15$	$2.8_{10} - 14$	$7.1_{10} - 15$
3.0	$7.1_{10} - 15$	$0.0_{10} + 00$	$0.0_{10} + 00$
5.0	$7.1_{10} - 15$	$1.4_{10} - 14$	$0.0_{10} + 00$
7.0	$7.1_{10} - 15$	$7.1_{10} - 15$	$1.4_{10} - 14$
10.0	$7.1_{10} - 15$	$7.1_{10} - 15$	$7.1_{10} - 15$
20.0	$0.0_{10} + 00$	$7.1_{10} - 15$	$0.0_{10} + 00$
50.0	$2.1_{10} - 14$	$1.4_{10} - 14$	$0.0_{10} + 00$
100.0	$2.1_{10} - 14$	$7.1_{10} - 15$	$7.1_{10} - 15$

TABLE II

	eps	$5.0_{10} - 06$	$5.0_{10} - 09$	$5.0_{10} - 12$	$5.0_{10} - 14$
a					
0.0	$d0$	$5.2_{10} - 08$	$4.3_{10} - 11$	$3.4_{10} - 14$	$5.3_{10} - 15$
	$d1$	$6.4_{10} - 08$	$1.8_{10} - 11$	$3.6_{10} - 14$	$5.3_{10} - 15$
	(n, N)	(9, 17)	(11, 37)	(13, 64)	(14, 87)
0.2	$d0$	$4.8_{10} - 08$	$5.3_{10} - 11$	$5.0_{10} - 14$	$1.8_{10} - 15$
	$d1$	$9.4_{10} - 08$	$4.9_{10} - 11$	$2.2_{10} - 14$	$1.3_{10} - 14$
	(n, N)	(9, 17)	(11, 36)	(13, 63)	(14, 86)
0.4	$d0$	$6.8_{10} - 09$	$2.2_{10} - 11$	$2.1_{10} - 14$	$8.9_{10} - 15$
	$d1$	$2.3_{10} - 08$	$1.1_{10} - 10$	$2.5_{10} - 14$	$2.3_{10} - 14$
	(n, N)	(10, 15)	(11, 33)	(13, 59)	(14, 81)
0.6	$d0$	$2.0_{10} - 07$	$8.2_{10} - 12$	$3.4_{10} - 14$	$1.6_{10} - 14$
	$d1$	$9.9_{10} - 08$	$4.8_{10} - 11$	$1.6_{10} - 14$	$2.4_{10} - 14$
	(n, N)	(8, 15)	(11, 33)	(13, 59)	(14, 81)
0.8	$d0$	$3.5_{10} - 08$	$4.7_{10} - 12$	$4.1_{10} - 14$	$1.1_{10} - 14$
	$d1$	$5.7_{10} - 08$	$4.7_{10} - 11$	$0.0_{10} + 00$	$2.1_{10} - 14$
	(n, N)	(9, 17)	(11, 36)	(13, 63)	(14, 86)
1.0	$d0$	$6.4_{10} - 08$	$1.8_{10} - 11$	$3.2_{10} - 14$	$3.6_{10} - 15$
	$d1$	$9.5_{10} - 08$	$5.5_{10} - 11$	$7.1_{10} - 15$	$1.4_{10} - 14$
	(n, N)	(9, 17)	(11, 37)	(13, 64)	(14, 87)

```
procedure bessya(a,x,eps,ya,ya1); value a,x,eps; real a,x,eps,ya,ya1;
begin real b,c,d,e,f,g,h,p,pi,q,r,s; integer n,na; Boolean rec, rev;
    pi:= 4 × arctan(1); na:= entier(a+.5); rec:= a ≥ .5;
    rev:= a < −.5; if rev ∨ rec then a:= a−na;
    if a = −.5 then
    begin p:= sqrt(2/pi/x); f:= p × sin(x); g:= −p × cos(x) end else
    if x < 3 then
    begin b:= x/2; d:= −ln(b); e:= a × d;
        c:= if abs(a) < ₁₀−15 then 1/pi else a/sin(a × pi);
        s:= if abs(e) < ₁₀−15 then 1 else sinh(e)/e;
        e:= exp(e); g:= recip gamma(a, p, q) × e; e:= (e + 1/e)/2;
        f:= 2 × c × (p × e + q × s × d); e:= a × a;
        p:= g × c; q:= 1/g/pi; c:= a × pi/2;
        r:= if abs(c) < ₁₀−15 then 1 else sin(c)/c; r:= pi × c × r × r;
        c:= 1; d:= −b × b; ya:= f + r × q; ya1:= p;
        for n:= 1, n + 1 while
        abs(g/(1 + abs(ya))) + abs(h/(1 + abs(ya1))) > eps do
        begin f:= (f × n + p + q)/(n × n − e); c:= c × d/n;
            p:= p/(n − a); q:= q/(n + a);
            g:= c × (f + r × q); h:= c × p − n × g;
            ya:= ya + g; ya1:= ya1 + h
        end;
        f:= −ya; g:= −ya1/b
    end else
    begin b:= x − pi × (a + .5)/2; c:= cos(b); s:= sin(b);
        d:= sqrt(2/x/pi);
        besspqa(a,x,eps,p,q,b,h);
        f:= d × (p × s + q × c); g:= d × (h × s − b × c)
    end;
    if rev then
    begin x:= 2/x; na:= −na − 1;
        for n:= 0 step 1 until na do
        begin h:= x × (a − n) × f − g; g:= f; f:= h end
    end else if rec then
    begin x:= 2/x;
        for n:= 1 step 1 until na do
        begin h:= x × (a + n) × g − f; f:= g; g:= h end
    end;
    ya:= f; ya1:= g
end bessya;
```

```
procedure besspqa(a,x,eps,pa,qa,pa1,qa1); value a,x,eps;
        real a,x,eps,pa,qa,pa1,qa1;
begin real b,c,d,e,f,g,p,p0,q,q0,r,s; integer n,na; Boolean rec,rev;
    rev:= a < -.5; if rev then a:= -a-1;
    rec:= a ⩾ .5; if rec then
    begin na:= entier(a+.5); a:= a - na end;
    if a = -.5 then
    begin pa:= pa1:= 1; qa:= qa1:= 0 end else
    begin c:= .25 - a × a; b:= x + x; p:= 4 × arctan(1);
        e:= (x × cos(a × p)/p/eps)↑2; p:= 1; q:= -x; r:= s:= 1 + x × x;
        for n:= 2, n + 1 while r × n × n < e do
        begin d:= (n - 1 + c/n)/s; p:= (2 × n - p × d)/(n + 1);
            q:= (-b + q × d)/(n + 1); s:= p × p + q × q; r:= r × s
        end;
        f:= p:= p/s; g:= q:= -q/s;
        for n:= n, n - 1 while n > 0 do
        begin r:= (n+1) × (2-p) - 2; s:= b + (n+1) × q; d:= (n - 1 + c/n)/
            (r × r + s × s); p:= d × r; q:= d × s; e:= f;
            f:= p × (e + 1) - g × q; g:= q × (e + 1) + p × g
        end;
        f:= 1 + f; d:= f × f + g × g;
        pa:= f/d; qa:= -g/d; d:= a + .5 - p; q:= q + x;
        pa1:= (pa × q - qa × d)/x;
        qa1:= (qa × q + pa × d)/x
    end;
    if rec then
    begin x:= 2/x; b:= (a + 1) × x;
        for n:= 1 step 1 until na do
        begin p0:= pa - qa1 × b; q0:= qa + pa1 × b;
            pa:= pa1; pa1:= p0; qa:= qa1, qa1:= q0; b:= b + x
        end
    end;
    if rev then
    begin p0:= pa1; pa1:= pa; pa:= p0;
        q0:= qa1; qa1:= qa; qa:= q0
    end
end besspqa;
```

REFERENCES

1. W. GAUTSCHI, *SIAM Rev.* **9** (1967), 24–82.
2. W. GAUTSCHI, *Comm. ACM* 7 (1964), 479–480.
3. M. GOLDSTEIN AND R. M. THALER, *Math. Tables Aids Comput.* **13** (1959), 102–108.
4. Y. L. LUKE, "The special functions and their approximations," Vols. 1 and 2, Academic Press, New York and London, 1969.
5. Y. L. LUKE, *Math. Comp.* **26** (1972), 237–240.
6. N. M. TEMME, *J. Comput. Phys.* **19** (1975).

RECEIVED: November 8, 1975.

N. M. TEMME

Department of Applied Mathematics,
Mathematical Centre,
2e Boerhaavestraat 49,
Amsterdam 1005, The Netherlands