Mathematics Notes

Note 17

February 1970

R. D. Halbgewachs, 9422
Sandia Laboratories, Albuquerque

Abstract

Techniques have been selected, developed, and implemented for computing Bessel functions with complex arguments and real orders, both varying simultaneously over extremely wide ranges of values. This paper describes these techniques and how they overcome such difficulties as exponential overflow or underflow, storage of very large arrays, and loss of accuracy during multiple sequences of recursion relations in Bessel function calculations. A new approach to validation testing is presented and the computer routines which use these techniques are discussed.

Key word: mathematics

MaN 17

SC-M-70-3

DEVELOPMENTS IN TECHNIQUES FOR COMPUTATION OF BESSEL FUNCTIONS BY DIGITAL COMPUTERS

R. D. Halbgewachs, 9422
Sandia Laboratories, Albuquerque

February 1970

ABSTRACT

Techniques have been selected, developed, and implemented for computing Bessel functions with complex arguments and real orders, both varying simultaneously over extremely wide ranges of values. This paper describes these techniques and how they overcome such difficulties as exponential overflow or underflow, storage of very large arrays, and loss of accuracy during multiple sequences of recursion relations in Bessel function calculations. A new approach to validation testing is presented and the computer routines which use these techniques are discussed.

Key word: mathematics

CONTENTS

DEVELOPMENTS IN TECHNIQUES FOR COMPUTATION OF BESSEL FUNCTIONS BY DIGITAL COMPUTERS

1. Introduction

Two new computer routines have been written to compute Bessel functions over extensive ranges of both arguments and orders. It was recognized that, in many instances, users have needed certain mathematical tools for use on computers (i.e., mathematical computer routines) but many of these needed tools were either nonexistent or possibly lacking in some aspect. As a part of the Sandia Mathematical Library Project, these two routines have been written to provide the user with a broader range of quality mathematical routines in the area of special functions.

In the routine BESSEL a cylindrical Bessel function can have an argument ranging over all quadrants of the complex plane with magnitudes in the tens of thousands. The range of orders is extremely wide and can extend anywhere from zero to values beyond the magnitude of the arguments supplied. This feature permits the order to be simultaneously of the same magnitude as the argument. The second routine, BESSPH, provides this same feature for the spherical Bessel functions. The capability of these two routines allows for applications in which alternative, less-desirable methods would have been used because of inherent restrictions in existing routines. This paper describes some of the techniques employed to circumvent the intrinsic difficulties in Bessel function calculations.

2. Basic Theory

Solutions of the differential equation

$$z^2 \frac{d^2w}{dz^2} + z\frac{dw}{dz} + (z^2 - \upsilon^2)\, w = 0 \qquad (1)$$

are known as Bessel functions of argument z and order υ. Some particular functions are those of the first kind $J_\upsilon(z)$, of the second kind $Y_\upsilon(z)$, and of the third kind $H_\upsilon^{(1)}(z)$, $H_\upsilon^{(2)}(z)$. The functions $Y_\upsilon(z)$ are also known as Weber's functions and the $H_\upsilon^{(1)}(z)$, $H_\upsilon^{(2)}(z)$ functions are known as Hankel's functions.

Alternatively, the function $J_\upsilon(z)$ can be defined by the equation

$$J_\upsilon(z) = (\tfrac{1}{2}z)^\upsilon \sum_{k=0}^{\infty} \frac{(-\frac{1}{4}z^2)^k}{k!\,\Gamma(\upsilon+k+1)}\,, \qquad (2)$$

and if υ is an integer n, then

$$J_{-n}(z) = (-1)^n J_n(z).$$

Otherwise, $J_{-\upsilon}(z)$ can be defined by substituting $-\upsilon$ into equation (2), replacing υ.

$Y_\upsilon(z)$ is related to $J_\upsilon(z)$ by

$$Y_\upsilon(z) = \frac{J_\upsilon(z)\cos(\upsilon\pi) - J_{-\upsilon}(z)}{\sin(\upsilon\pi)} \qquad (3)$$

where the right-hand side of this equation is replaced by its limiting value if υ is an integer. The Hankel functions are related to $J_\upsilon(z)$ and $Y_\upsilon(z)$ by the equations

$$\begin{aligned} H_\upsilon^{(1)}(z) &= J_\upsilon(z) + i\, Y_\upsilon(z)\,, \\ H_\upsilon^{(2)}(z) &= J_\upsilon(z) - i\, Y_\upsilon(z). \end{aligned} \qquad (4)$$

In the case of real arguments, $H_\upsilon^{(2)}(z)$ is the complex conjugate of $H_\upsilon^{(1)}(z)$, but it should be noted that for complex arguments we have complex functions and this conjugation relation is no longer true.

Solutions of the differential equation

$$z^2 \frac{d^2w}{dz^2} + 2z \frac{dw}{dz} + [z^2 - n(n+1)]\, w = 0, \quad (n=0, \pm 1, \pm 2, \ldots) \tag{5}$$

are known as spherical Bessel functions of argument z and order n. Again we have particular solutions -- those of the first kind $j_n(z)$, of the second kind $y_n(z)$, and of the third kind $h_n^{(1)}(z)$, $h_n^{(2)}(z)$. The relationship between the spherical and cylindrical functions are

$$j_n(z) = \sqrt{\tfrac{1}{2}\pi/z}\, J_{n+\frac{1}{2}}(z) \ , \tag{6}$$

$$y_n(z) = \sqrt{\tfrac{1}{2}\pi/z}\, Y_{n+\frac{1}{2}}(z) = (-1)^{n+1} j_{-(n+1)}(z) \ , \tag{7}$$

$$h_n^{(1)}(z) = j_n(z) + i\, y_n(z) = \sqrt{\tfrac{1}{2}\pi/z}\ \ H^{(1)}_{n+\frac{1}{2}}(z) \ , \tag{8}$$

$$h_n^{(2)}(z) = j_n(z) - i\, y_n(z) = \sqrt{\tfrac{1}{2}\pi/z}\ \ H^{(2)}_{n+\frac{1}{2}}(z) \ . \tag{9}$$

Of principal concern in this paper is the description and discussion of the mathematical methods that are specifically used in the two routines for the evaluation of these Bessel functions. Before we begin, however, it should be noted that there are various other approaches which can be used in evaluating these functions. The treatise by Watson[1] covers many aspects of Bessel functions, including various series definitions, asymptotic forms, properties about the zeroes of the functions, etc. The reason for using the particular equations and techniques discussed here is a combination of three factors: (1) the methods provide results with as much or more accuracy as results using other approaches, (2) the methods provide results as fast or faster than other approaches, and (3) the methods permit the routines to provide results over a much larger range of arguments and orders than ever before (except for cases of extreme arguments or orders where special asymptotic expansions can be applied).

Depending on the ranges of the function parameters, several different mathematical methods are used for evaluating these functions. Foremost are the three-term recurrence relations

$$F_{\upsilon-1}(z) + F_{\upsilon+1}(z) = \frac{2\upsilon}{z} F_{\upsilon}(z) \ , \tag{10}$$

$$f_{n-1}(z) + f_{n+1}(z) = \frac{2n+1}{z} f_n(z) \ . \tag{11}$$

In (10), F denotes any of the above cylindrical functions or any linear combination of these functions and, similarly, in (11) f represents the spherical functions. Reasons for the use of these relations are the ease of application and the speed with which a set of function values can be obtained for a wide range of given orders. The equations (10) and (11) are the most commonly used relations although many others exist and can be used for special purposes such as evaluating derivatives, cross-products, etc. For example, the Handbook of Mathematical Functions[2] contains a long list of recurrence relations.

It has been shown by Gautschi[3] that for the particular functions $J_{\upsilon}(z)$ and $j_n(z)$ the recurrence relations of (10) and (11) are unstable in the "forward" direction, that is, in the direction of increasing order. This instability results from the effect of the dominant solutions of the differential equations (1) and (5) over the minimal solutions, i.e., the dominance of the functions $Y_{\upsilon}(z)$ and $y_n(z)$. However, this effect of dominance is minimized by applying the recurrence procedure in a "backward" direction (decreasing order). Miller[4] first developed an algorithm for recurring backward with strictly integer orders and Goldstein and Thaler[5] have done additional work with the algorithm.

In the following discussion we shall present the technique of backward recurrence as used in the routines BESSEL and BESSPH. Given an argument z, the functions $J_{\upsilon}(z)$ and $j_n(z)$ behave so that as order increases the function values decrease in magnitude. This behavior is used to our advantage in developing the backward recurrence. Let us denote these functions simply by $f_{\upsilon}(z)$ and when we deal with a specific function we will then return to the notation for that particular function.

Suppose the largest order of the function desired is N, where we denote the fractional part as

$$\eta = |N| - \text{Int } |N| \ . \tag{12}$$

Then our approach is to choose some integer M such that $M > N$ and assign values

$$f^*_{M+1+\eta}(z) = 0 \quad \text{and} \quad f^*_{M+\eta}(z) = \xi \ . \tag{13}$$

In some journal articles ξ is taken as unity for purposes of discussion but, for added control over the growth of the functions during application of the backward recurrence, it is to our advantage to make ξ some very small value expressible in the computer, e.g., $\xi = 10^{-200}$ for the CDC 6600 computer.

Olver[6] and Gautschi[3] indicate that the criterion for M is that it be sufficiently large to insure the desired degree of accuracy in the end results. In our work, the value

$$M = \text{Int} \left\{ \max. \left[\left(\frac{e}{2}|z| \ , N\right) \right] \right\} + 200 \tag{14}$$

has proved to supply this degree of accuracy (discussed in more detail in Section 5). Then from (13) we can calculate the string of values

$$f^*_{M-1+\eta}(z), \ f^*_{M-2+\eta}(z), \ \ldots, \ f^*_N(z), \ \ldots, \ f^*_\eta(z)$$

by using the recurrence relations. The very important point to recognize is that although $f^*_N(z), \ \ldots, \ f^*_\eta(z)$ are not correct function values, they are all properly related to each other in the same scale, provided M is large enough. Since we can multiply the appropriate recurrence relation by a constant factor without altering the relation, we need only find the proper normalizing factor α such that

$$f_\eta(z) = \alpha f^*_\eta(z), \ f_{\eta+1}(z) = \alpha f^*_{\eta+1}(z), \ \ldots, \ f_N(z) = \alpha f^*_N(z) \ .$$

There are several ways in which we can determine this normalizing factor α. First, the spherical functions have the very nice property of being expressed explicitly. Watson[1] shows the equation for $J_{n+\frac{1}{2}}(z)$ to be

$$J_{n+\frac{1}{2}}(z) = \sqrt{\frac{2}{\pi z}} \left[\sin\left(z - \tfrac{1}{2} n\pi\right) \sum_{r=0}^{\leq \frac{1}{2}n} \frac{(-1)^r (n+2r)!}{(2r)!(n-2r)!(2z)^{2r}} \right.$$

$$\left. + \cos\left(z - \tfrac{1}{2} n\pi\right) \sum_{r=0}^{\leq \frac{1}{2}(n-1)} \frac{(-1)^r (n+2r+1)!}{(2r+1)!(n-2r-1)!(2z)^{2r+1}} \right] \qquad (15)$$

which, for n = 0 and using equation (6), reduces the expression to

$$j_0(z) = \frac{\sin z}{z} . \qquad (16)$$

Hence, we have the normalizing factor α by simply making the division

$$\alpha = \frac{j_0(z)}{j_0^*(z)} . \qquad (17)$$

The problem of determining α for the case of cylindrical functions is a bit more involved. When the orders are integers, we know from Neumann's addition theorem[1] that

$$1 = J_0^2(z) + 2 \sum_{k=1}^{\infty} J_k^2(z) .$$

If we have

$$\beta = \left[J_0^*(z)\right]^2 + 2 \sum_{k=1}^{\infty} \left[J_k^*(z)\right]^2 , \qquad (18)$$

then let

$$J_k^{**}(z) = J_k^*(z) \; / \sqrt{\beta}$$

so that

$$\left[J_k^{**}(z)\right]^2 = \left[J_k^{*}(z)\right]^2/\beta$$

and

$$\left[J_0^{**}(z)\right]^2 + 2\sum_{k=1}^{\infty}\left[J_k^{**}(z)\right]^2 = \left\{\left[J_0^{*}(z)\right]^2 + 2\sum_{k=1}^{\infty}\left[J_k^{*}(z)\right]\right\}/\beta = 1.$$

Since each $J_k^*(z)$ was multiplied by a constant, the recurrence relation is still maintained and the functions $J_k^{**}(z)$ satisfy the addition theorem. Hence, we may assume $J_k(z) = J_k^{**}(z)$, $k = 0, 1, 2, \ldots$, and $\alpha = \beta^{-\frac{1}{2}}$. This summation theorem is only one of many that could be applied (see Handbook[2]).

However, in using the summation method, we first notice that an error is introduced since we can allow the summation to extend only over a finite range of orders and convergence becomes a part of the criteria for M. Furthermore, unlike the nice functional behavior for real arguments, the appearance of complex arguments can cause extreme functional behavior. As an example,

$$\text{for } z_1 = 100 \text{ and } z_2 = 100 + i\,100$$

we have

$$J_5(z_1) \approx -0.007 \text{ and } J_5(z_2) \approx 2.7 \times 10^{40} + i\,8.6 \times 10^{40}.$$

Several questions can be raised in light of this information. Clearly, the large magnitudes would indicate the summation may need to extend over a much larger range of orders than desired before (18) attains some degree of validity when a finite number of terms is used. But this leads us into the next question which is the problem of deciding whether, in practice, we can ever handle magnitudes such as these and expect to control the accuracy.

A more reasonable approach in determining the normalizing factor is to calculate a very good approximate value for $J_\eta(z)$, say $\hat{J}_\eta(z)$, and then let

$$\alpha = \hat{J}_\eta(z)/J_\eta^*(z). \qquad (19)$$

If we have $|z| \leq 10.0$, the series (2) provides us with the needed value $J_\eta(z)$ and for $|z| > 10.0$ we can determine $J_\eta(z)$, $\eta < 1$, from Hankel's asymptotic expansion. This expansion and some of the computational aspects of using the expansions are discussed in Appendix A. Should the circumstance arise where $\hat{J}_\eta(z)$ is zero, then the function $\hat{J}_{\eta+1}(z)$ is used in determining the normalization factor α. Otherwise, we would have $\alpha = 0$ and this would clearly be an incorrect normalizing factor for the sequence.

Let us now consider the evaluation of the functions of the second and third kinds. As a result of their dominant characteristics, the functions $Y_\upsilon(z)$ and $y_n(z)$ permit usage of the recurrence relations (10) and (11) in the forward direction. Hence the problem in this case reduces to determining good starting values to use in the recurrence.

For the spherical functions, the explicit expression makes everything straightforward. We have

$$y_0(z) = -\frac{\cos z}{z} \; ; \; y_1(z) = -\frac{1}{z}\left(\frac{\cos z}{z} + \sin z\right) . \qquad (20)$$

A few more calculations are necessary to get the cylindrical functions started. In the case where integer orders are to be used we need $Y_0(z)$ and $Y_1(z)$ to start. For noninteger orders we need to determine $Y_\eta(z)$ and $Y_{\eta+1}(z)$ where $\eta = |\upsilon| - \text{int}\,|\upsilon|$. If $|z| > 10.0$, the Hankel expansion (equation (A1), Appendix A) will provide the values $Y_0(z)$ and $Y_1(z)$ or $Y_\eta(z)$ and $Y_{\eta+1}(z)$. In the case $|z| \leq 10.0$, separate methods are needed for integer orders as opposed to noninteger orders. With integer orders, we can determine $Y_0(z)$ from the equation

$$Y_0(z) = \frac{2}{\pi}\{\ln(\tfrac{1}{2}z) + \gamma\} J_0(z) + \frac{2}{\pi}\left\{\frac{\tfrac{1}{4}z^2}{(1!)^2} - (1+\tfrac{1}{2})\frac{(\tfrac{1}{4}z^2)^2}{(2!)^2}\right.$$

$$\left. + (1+\tfrac{1}{2}+1/3)\frac{(\tfrac{1}{4}z^2)^3}{(3!)^2} - \ldots.\right\}$$

and then from the Wronskian

$$J_1(z)\, Y_0(z) - J_0(z)\, Y_1(z) = \frac{2}{\pi z}$$

we can obtain $Y_1(z)$.

In the case of noninteger orders the series (2) can be used to evaluate $J_\eta(z)$ and $J_{-\eta}(z)$ and then using (3) we have $Y_\eta(z)$. Similarly, we can determine $Y_{\eta+1}(z)$.

Because of the domination of the functions of the second kind over those of the first kind we are safe in using the recurrence relations in the forward direction to evaluate the remaining class of functions. From the definitions of the third kind of functions we can determine the starting values of these functions and then proceed to use the recurrence relation. For example, to determine the starting values for $h_n^{(1)}(z)$ we use (8) together with (16) and (20) to give

$$h_0^{(1)}(z) = \frac{1}{z} (\sin z - i \cos z)$$

and

$$h_1^{(1)}(z) = \frac{1}{z^2} (\sin z - i \cos z) - \frac{1}{z} (\cos z + i \sin z) .$$

In a similar manner, we would determine the lowest two orders for $J_\upsilon(z)$ and $Y_\upsilon(z)$ to provide us with the necessary H function starters.

One might ask why a recurrence on the Hankel functions is necessary when the functions of the first and second kind could be used and simply added to or subtracted from each other. As a result of the oscillatory nature of these functions, the functional values will many times be very nearly equal. If, in the finite precision of the computer representations, we have the numbers alike, then we say that for our purposes the functions are equal. We must be concerned when the values are nearly equal. For during the course of studying techniques and examining results, it was shown that although the functions of the first and second kind were calculated to, say, an accuracy of 10 significant places, upon combination little significance remained in the real or imaginary parts (in some cases, both) because of cancellation. But, by starting the Hankel functions with good values and allowing the functions to run through the recurrence, the result was that both real and imaginary parts of the functions were accurate.

There is a special situation where the recurrence relation should not be used to evaluate the third kind of functions. For strictly real arguments and positive orders, the first and second kinds of functions

provide strictly real results. Hence, both parts of the Hankel functions can be accurately obtained by combining the other two functions as shown in (4), (8), or (9). But why not use recurrence? Depending on the argument and order, of the two numbers that would be carried during recurrence (real and imaginary parts) one number could become much larger than the other. As recurrence progressed, the smaller would then either lose significance or become zero. Granted, if one were only concerned with the magnitude of the function, this condition would have little effect since the larger part would completely dominate. But in other cases, for example multiplication of functions, it is critical that both parts be accurate.

Aside from the equations and methods just discussed there are also some very important approaches to be considered in evaluating the functions of the first kind for a relatively small but still significant range of arguments and orders. This much-used region is that for which arguments are of magnitude $|z| < 10.0$. Basically, the approach is to use the series (2) for the evaluations in this range. For the spherical functions, $j_n(z)$, equations (2) and (6) are used. The series is modified first by breaking up the gamma function so that

$$J_\upsilon(z) = (z/2)^\upsilon \frac{1}{\Gamma(\upsilon+1)} \left\{ 1 + \sum_{k=1}^{\infty} \frac{(-z^2/4)^k}{k!(\upsilon+k) \ldots (\upsilon+1)} \right\}. \qquad (21)$$

Evaluation is simplified by considering the series in (21) as

$$\sum_{k=1}^{\infty} a_k$$

where

$$a_{k+1} = \left(\frac{-z^2/4}{(k+1)(\upsilon+k+1)} \right) a_k$$

and

$$a_0 = 1.$$

Hence, we have a two-term recurrence to evaluate the series of (21). For most applications equation (21) is calculated in a straightforward manner and, because of the limitation that $|z| < 10.0$, convergence can be achieved rapidly. For that matter, large orders or small arguments improve the convergence rate.

Since one of the major features of the routines BESSEL and BESSPH is to provide results for arguments and orders over large ranges, we must also face the fact that we should be capable of providing results for very small arguments and orders. For very small arguments, (21) reduces to the asymptotic form

$$J_\upsilon(z) \approx (z/2)^\upsilon \frac{1}{\Gamma(\upsilon+1)} . \tag{22}$$

Let us first consider the situation where the asymptotic form is to be used and the order is quite large. Our first reaction is to call the result zero, but with very little effort we can provide a nonzero result. That is, the logarithm can be calculated so that

$$\ln J_\upsilon(z) \sim \upsilon \ln(z/2) - \ln \Gamma(\upsilon+1) \tag{23}$$

and by using Stirling's approximation[2] we have

$$\ln \Gamma(\upsilon+1) = \ln \upsilon + \ln \Gamma(\upsilon) \sim (\upsilon+\tfrac{1}{2}) \ln \upsilon - \upsilon + \tfrac{1}{2} \ln 2\pi + \sum_{k=1}^{\infty} \frac{B_{2k}}{2k(2k-1)\upsilon^{2k-1}} , \ (\upsilon\to\infty). \tag{24}$$

The terms B_{2k} are known as Bernoulli numbers and

$$B_0 = 1, \qquad B_2 = 1/6, \qquad B_4 = -1/30,$$
$$B_6 = 1/42, \qquad B_8 = -1/30, \qquad B_{10} = 5/66 .$$

Equation (24) can be rewritten as

$$\ln \Gamma(\upsilon+1) \sim (\upsilon+\tfrac{1}{2}) \ln \upsilon - \upsilon + \tfrac{1}{2} \ln 2\pi + \frac{1}{12\upsilon} - \frac{1}{360\upsilon^3} + \frac{1}{1260\upsilon^5} - \ldots., \ (\upsilon\to\infty).$$

Note that we are restricting ourselves to large orders so the approximation is appropriate and convergence is again rapid. For smaller orders, the approximation becomes questionable and we can achieve better accuracy by calculating the gamma function outright first and then using logarithms of the function to finish.

During the calculation of the Bessel functions, certain argument- and order-dependent relations between the functions must be observed. These relations are discussed in detail in Appendix B. At this point we have the methods and equations necessary for evaluating the Bessel functions. However, there still remains part of the iceberg to be uncovered. That is, although the equations are certainly valid as they appear, several factors must now be considered in the process of computer applications.

3. Scaling

We have already noted that multiplying the terms in recurrence relations (10) and (11) by a constant will not alter the relationships, and hence the concept of scaling becomes very valuable. The reasons for scaling actually result from the behavior of the Bessel functions themselves.

By simply examining the tables in the Handbook[2] for real arguments it can be seen that, for functions of the second and third kinds, exponential growth is typical as the order increases. The growth rate is even more extreme when the arguments are complex. If we are to allow a wide range of orders to be used then scaling of some nature must be employed.

On the other hand, functions of the first kind actually decrease in magnitude as the order increases. This feature is utilized in the backward recurrence by starting the recurrence with very small values. But as the recurrence proceeds and order decreases, the growth pattern is such that the computed functions increase in magnitude until a relatively level plateau is reached (see Figure 1).

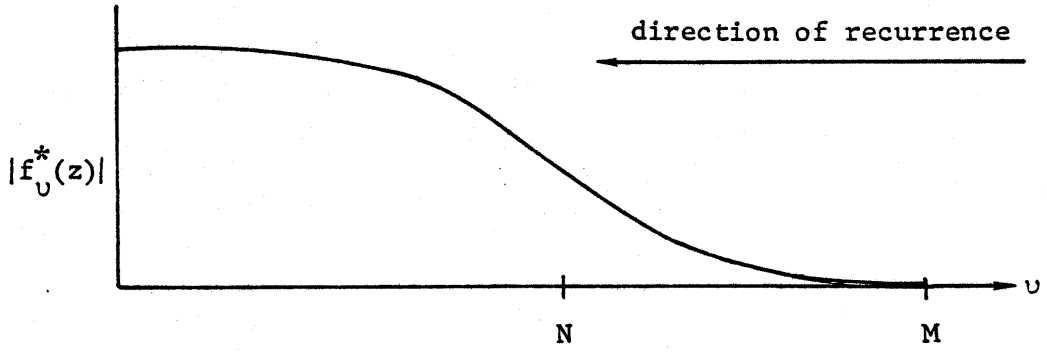

Figure 1.

To start, the order-over-argument term in (10) and (11) is the dominant factor and during recurrence the magnitudes of the computed functions increase. Also during the process of recurrence, however, this factor reaches a magnitude less than unity and is no longer

dominant. The role has been taken over by the computed functions and growth of the functions levels out. In using a computer, the problem to be aware of is that this plateau may not be reached before the exponential capacity of the computer is surpassed. In fact, this is the usual limiting restriction on most Bessel function computer routines, hence another reason for the use of scaling.

The clinching point which requires us to use scaling was demonstrated earlier in a simple example. The fact that we are dealing with complex arguments forces an awareness of the occurence of large functional values. We may encounter large values not only after long use of the recurrence (either forward or backward), but even in calculating starting values or correction factors.

The general scaling method used during recurrence for all of the functions is to extract exponential factors, e^a, from the function values whenever a certain level of magnitude, L, is reached. The recurrence is then continued with the new function values. As a result, the build-up shown in Figure 1 for backward recurrence is transformed to that of Figure 2.

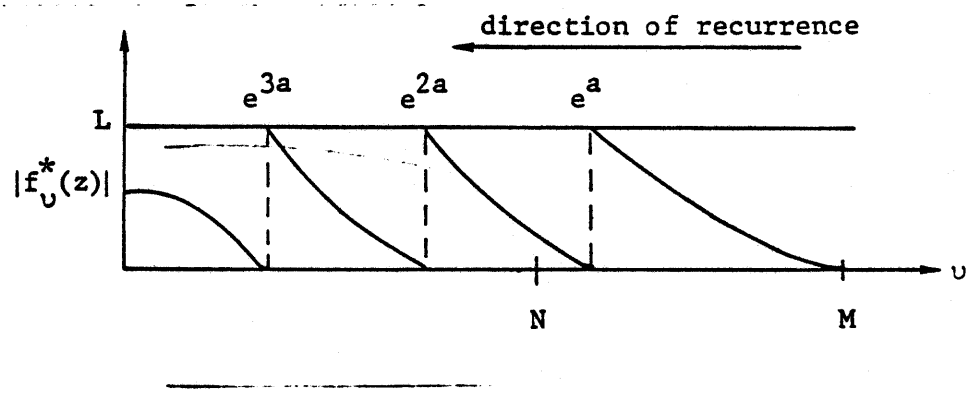

Figure 2.

We can picture the same sort of behavior for the forward recurrence as that of Figure 2 by simply considering the origin to be at the right.

The actual process of scaling can best be explained by following through an example. Suppose that we are in the recurrence process, either forward or backward, and the function values are increasing in magnitude as we proceed. From two previously calculated values we are determining a third. After computing this third value we check and find that the magnitude exceeds the limit L. The order of this function value--in a sense the location of this function value in the recurrence--is saved. The function value is then decreased by the factor e^a and also the function value just previously calculated on the last recurrence. Recurrence then proceeds from these two new function values. At the same time, the two new values are saved in an array for further use later in the routines. A counter is also incremented so that after the entire recurrence has been completed we have available the number of times, m, that scaling was required, and also the exponent of the final factor, e^{ma}. Now, at this point, the forward recurrence is completed and the function values are properly scaled.

However, the entire problem of factor scaling becomes much more muddled before we can say that we have finished with the backward recurrence and are able to return the true function values. From the earlier description of backward recurrence we recognize that we have only scaled the recurred values and a normalizing factor must still be determined. Consider the "true" function values of equations (17) and (19) and for purposes of discussion let us denote either function $j_0(z)$ or $J_\eta(z)$ by $f_t(z)$. If we have

$$-a \leq \ln |f_t(z)| \leq a$$

then no correction to the scaling factors is necessary and α can be calculated. In the case that $\ln |f_t(z)|$ lies outside these bounds, then all of the scaling factors must be adjusted to represent the true exponent of the function. This is easily accomplished by making the adjustment in multiples of a (i.e., $(m \pm k)a$, $k = 1, 2, \ldots$). Of course, before computing α, the true function value $f_t(z)$ must have the exponent appropriately adjusted.

The end representation of the scaling process is designed to provide the user of the routines with as much flexibility as possible. Suppose the function $f_v(z)$ were to be evaluated and returned. From

the particular routine selected would come a two-part result. First there would be a function value $f'_\upsilon(z)$ and then the corresponding exponential scaling factor for this returned value, κ, so that

$$f_\upsilon(z) = e^\kappa \, f'_\upsilon(z) \ . \tag{25}$$

Notice that for $\kappa \neq 0$, the true function value $f_\upsilon(z)$ will more than likely fall outside the exponent bounds of the computer. But since the user already knows the value of κ in the exponent factor e^κ he probably can use exponent arithmetic in his calculations and make use of the function $f'_\upsilon(z)$ without concern for the computer limitations. For example, suppose the equation

$$w(z) = \frac{J_\upsilon(z)\, Y_{\upsilon+1}(z) - J_{\upsilon+1}(z)}{H^{(2)}_{\upsilon-1}(z)\, Y_\upsilon(z)}$$

were to be calculated. The primed function values would be used in the multiplications, divisions, and subtractions and, through exponent bookkeeping, a true final result could be determined.

4. Calculation Efficiency and Storage

The storage of many function values is a major problem that must be confronted when developing routines such as BESSEL and BESSPH. In these two routines it is possible to compute four different Bessel functions simultaneously. We must immediately recognize that, when a very large range of orders is permitted, it is not feasible to attempt to carry all values in arrays during execution of the routines. For example, suppose we wished to evaluate all four functions, spherical or cylindrical, and each function were to be used with various orders between zero and 20,000. Since the functions are complex, this problem would require a minimum of 160,000 locations for array storage alone. Even if it were possible to fit the arrays into available storage, the core requirement would probably force most computers into dedicated use for this one problem while executing. For these two Bessel function routines, we are even permitting orders higher than that of the example. Hence, we are forced to investigate ways of reducing core requirements. But the execution time increase must also be considered when we begin to develop these schemes.

For the routines BESSEL and BESSPH the recurrence relations (10) and (11) can be used to work in our favor. In the following discussion we will use the term primary recurrence to refer to the recurrence actually applied through the entire range of orders requested. The term secondary recurrence refers to any recurrence that takes place over only a portion of the entire order range.

We first partition the entire order range requested into intervals of some size I. In the backward recurrence the entire range also includes those orders that lie between N, the maximum order requested, and M, the starting order for recurrence. For the forward recurrence, the primary recurrence extends only up to the order N. Then during the primary recurrence we retain only two function values in each interval. To clarify, suppose we go back to the example described above and show an array as it would be constructed under the partitioning. We assume there is some given argument z and N = 20000. Let us select I = 1000. A storage array of values retained would become

Array Location	Order
1	0
.	.
.	.
.	.
1000	999
1001	1000
1002	1999
1003	2000
1004	2999
1005	3000
.	.
.	.
.	.
.	.
.	.
1038	19999
1039	20000

Hence, for the computation of all four complex functions, our storage requirement is reduced to 8312 locations. We could have selected I = 500 and further reduced the amount of storage necessary. The selection of interval size is left to the user for his own particular problem.

Notice that the lower locations of the array are filled in sequence by the corresponding orders. These I locations become the working portion of the array. Should any or all orders between zero and 1000 be requested, the routine can immediately go to the array and pick up the answer. Suppose that a higher order or orders is requested. The routine will pick out the proper interval and the two values which represent that interval. A secondary recurrence is then performed with the values in the lower locations replaced by the complete interval requested. Any subsequent calls for orders within that interval are immediately available without additional secondary recurrences. With this scheme it is possible for the order requests to bounce around various intervals and the routine can still provide results for any order needed. Naturally, the most expedient method of requesting orders, if possible, is to call all needed orders in a single interval before moving outside the interval and into another.

The knowledge of the pattern of needed orders can aid in deciding the value of I that should be used. For example, if many orders of each interval are needed then a large value of I could be used (provided core is available) so that a single secondary recurrence would provide many results. On the other hand, if only a few orders are needed from each interval, then a smaller partition size could be selected for faster secondary recurrence around specific orders.

The principal reason for doing a second recurrence over an entire interval is to avoid the need for multiple secondary recurrences should more than one order be requested from the same interval. Repeated calls within an interval would be necessary, for example, in the problem of evaluating a series whose terms are Bessel functions. This would also be true in determining derivatives of Bessel functions (see Handbook[2]).

One might ask how all this affects the user who wants the value of a Bessel function for only one order. For a small order, we see from the example that the lower locations will automatically contain the results for the smaller orders. Suppose for some value of the argument we wished to know the value of a function of order 400. By specifying the maximum order as N = 400 and selecting I to be slightly larger, we would be guaranteed immediate return of the result without a secondary recurrence. Should a single, very large order be requested, the amount of time for a secondary recurrence over the particular partition for that order would be negligible compared to the primary recurrence which is necessary anyway.

Finally, we present a few fine points about the specific operation of this storage procedure. Consider again the above example where I = 1000. If we had specified that N = 20500, the array would contain two more values, namely, those for orders 20499 and 20500. That is, the array would have been completed by a fraction of an interval to permit all orders to be covered. Another important facet to the secondary recurrences is that it is quite possible, and in fact probable, that an exponent factor will have been removed somewhere in the interior of a partition. This may occur in several partitions or even several times in the same partition. In Section 3 we stated that when a factor was removed, two function values and an order were saved.

During a secondary recurrence, the order of the functions being regenerated is checked. Whenever an order corresponds to a factor point, the two saved related function values are brought in to complete the secondary recurrence from that point. The fact that this happens is reflected by changes in the factor value returned with each function value.

5. Program Validation

Several studies have been made on the effectiveness of the repeated usage of the recurrence procedures. Olver's[6] study was primarily concerned with the backward recurrence technique and the error propagation that may occur during the application of this algorithm. The studies made by Gautschi[3] and Oliver[7] deal with both forward and backward recurrence procedures. The conclusions from each of these studies points to some of the restrictions that must be followed in order to maintain control over errors. Gautschi emphasizes that the effectiveness of the algorithms is clearly enhanced if good estimates of the initial values of the functions are available. That is, in the forward recurrence we are required to make sure that the starting values are as nearly exact as possible. In the backward recurrence, we must make sure that the value from which a normalization factor is calculated, see equations (17) and (19), is as nearly exact as possible. Furthermore, each of studies emphasizes the fact that in backward recurrence we must insure that the starting order of recurrence, M, is sufficiently greater than the largest order N for which we expect results.

Many functions related by recurrence relations are solutions to differential equations. In many of these cases, there exist minimal or nonminimal (dominant) solutions to the differential equations. It must be clear into which category the functions fall. We cannot expect to apply one of the recurrence algorithms to the wrong category and emerge with good results. From their behavior it is clear that the Bessel functions of the first kind are minimal and must be determined in the backward direction. On the other hand, the functions of the second kind are dominant and forward recurrence is used. Functions of the third kind lie in a middle range and this is the motivation for using several different methods in the calculation of these functions. It is shown in the studies just mentioned that, provided a sufficiently large value for M is selected and nearly exact initial values are calculated, the error accumulation is held to very small values relative to the function values calculated. In addition, the error that is present for some function value is primarily dependent on the initial errors and the degree of accuracy that was first introduced.

Actual physical testing of the results from these routines presents an interesting problem. Although many arguments and orders fall into regions where other computational methods exist, these two routines are designed to provide results in regions heretofore not treated. The first part of this discussion will be concerned with some of the standard validation approaches which were used. The remainder of the discussion presents a new approach to validation. Specific degrees of accuracy and test results for the routines are not given here, but are discussed in two other reports.[8,9]

Because of priorities in the development of these routines, BESSPH was written first. Not only was there a greater urgency for the spherical functions at the time, but this also proved to be beneficial in the development and testing of BESSEL. From equations (6) through (9) we see that the spherical functions can be considered as special cases of the cylindrical functions for half-integer orders. The routine BESSPH was completely tested and checked before the validation tests were begun on BESSEL. Hence, we had a tested routine that provided comparison values for the special real orders of half-integers in BESSEL.

The first step in the validation of any special function routine is to determine if there already exist tables for some ranges which can be used for initial comparisons. The tables provided by the Handbook of Mathematical Functions,[2] the Harvard Computation Laboratory,[10] the National Bureau of Standards,[11, 12, 13, 14] and Jahnke and Emde[15] were examined. This process insured that no constant multiplying factors were present in the results. At least for the arguments and orders that could be used in these tables, the results were assured of being "in the ballpark."

Let us now consider those equations and methods that were available for use in testing. In Figure 3, we see the argument-order range split into three regions. Region I represents that area in which most methods are available, the region of relatively small orders where arguments vary in magnitudes from small to large.

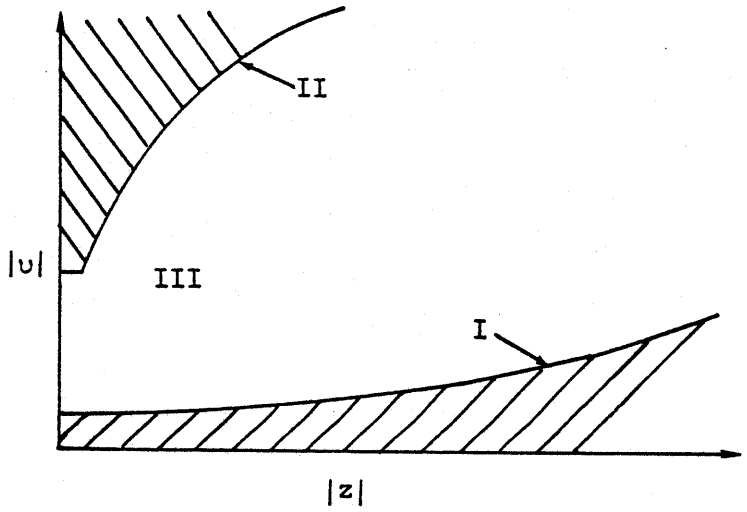

Figure 3.

The spherical functions present a definite advantage in testing in that they can be explicitly expressed as shown in (15). For a portion of Region I, these equations provide a very solid basis for comparison. Results from the cylindrical functions for real arguments and integer orders were compared to results obtained through a routine called BES[16] which is contained in the Sandia Mathematical Program Library. This routine has already been thoroughly tested and established as having very good accuracy, so that comparisons provided valid test results. In all of these tests, the argument size was limited by the accuracy and restrictions of the test routines. However, the asymptotic expansion described in Appendix A holds for larger arguments and small orders. This expansion applies to spherical as well as cylindrical functions. Under these conditions, the arguments could be tested for magnitudes in the tens of thousands; precisely one of those areas for which BESSEL and BESSPH are designed to operate. It is evident from the discussion that, so far, only a small portion of the argument-order range has been covered. These test methods are restricted to evaluating functions whose orders are relatively small (less in magnitude than a few hundred), and in some cases very small.

The other extreme of the argument-order range shown in Figure 3 is illustrated by Region II. In this region, the orders are considered to be very large and equation (21) becomes the workhorse of testing. Since the orders are large, Stirling's approximation is used

in calculating the gamma function. Because of the magnitude size of the function values, it is necessary to use logarithms in both Regions I and II for the validation testing.

Region III represents that area for which a new test approach was developed. The capability to evaluate Bessel functions in this region rounds out the features of the routines BESSEL and BESSPH. We begin a test sequence by specifying an argument and some order that will force the routines into Region II. Let this order represent the maximum order to be requested. We know that recurrence proceeds along the orders with the argument fixed. Hence, in the backward recurrence, we start in Region II and as recurrence proceeds we travel a vertical path downward into Region I. In the forward process, we travel from I to II. The important point to remember is that the recurrence relations are satisfied by the function values during the recurrence process. From our previous discussion we have shown that the function values which lie in Regions I and II can be verified. Therefore, a reasonable test hypothesis for Region III is the following:

1. Verification of the function values in the starting Region (I or II), and

2. verification of the function values in the stopping Region (I or II),

3. implies that the function values of Region III are valid.

A secondary test was established in checking functions of the first kind to lend further support to the validity of values in Region III. Let us assume the start point of recurrence to be in Region II and the stop point in Region I. We represent the travel between start point P_1 and the stop point as line A in Figure 4. Also consider a second recurrence where the start point does not fall in Region II, but starts at P_2.

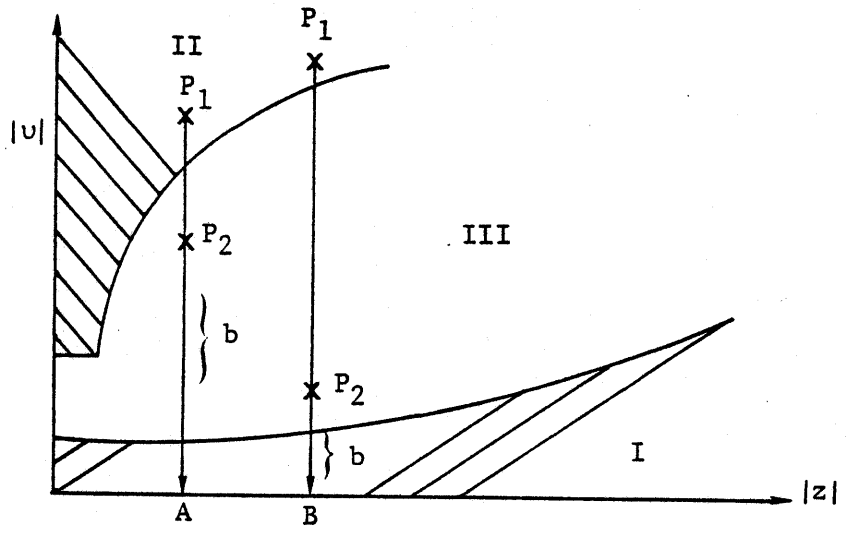

Figure 4

The question that must be answered is whether the function values that result for those orders that lie within the bracket b are exactly equal for the two recurrences. This is important for the functions of the first kind since we know that the first recurrence starts well beyond those values to be tested and we must make sure that the second recurrence also provides the necessary starting distance. The test can be emphasized even more in a situation similar to that represented by line B.

Through the extensive testing techniques just described it is determined whether the routine is functioning as designed for all portions of the range to be covered. Furthermore, the accuracy of the routine can also be demonstrated so the user will have substantial confidence that the values returned are valid.

REFERENCES

1. G. N. Watson, A Treatise on the Theory of Bessel Functions, 2nd Edition, Cambridge University Press, New York, 1952.

2. M. Abramowitz and I. Stegun, Handbook of Mathematical Functions, National Bureau of Standards, AMS 55, 1965.

3. W. Gautschi, "Computational Aspects of Three-Term Recurrence Relations," SIAM Review 9, 1967, pp. 24-82.

4. J. C. P. Miller, "Bessel Functions, Part II," Mathematical Tables X, British Association for Advancement of Science, Cambridge University Press, New York, 1952.

5. M. Goldstein and R. M. Thaler, "Recurrence Techniques for the Calculation of Bessel Functions," MTAC 13, 1959, pp. 102-108.

6. F. W. J. Olver, "Error Analysis of Miller's Recurrence Algorithm," Math. of Computation 18, 1964, pp. 65-74.

7. J. Oliver, "Relative Error Propagation in the Recursive Solution of Linear Recurrence Relations," Numerische Mathematik 9, 1967, pp. 323-340.

8. R. D. Halbgewachs, BESSEL: Cylindrical Bessel Functions of Complex Argument and Real Order, Sandia Laboratories, Albuquerque, New Mexico (to be published).

9. R. D. Halbgewachs, BESSPH: Spherical Bessel Functions of Complex Argument, Sandia Laboratories, Albuquerque, New Mexico (to be published).

10. Harvard University Computation Laboratory, Tables of the Bessel Functions of the First Kind, volumes 1-15, Harvard University Press, Cambridge, Mass., 1947-1950.

11. National Bureau of Standards, Tables of Bessel Functions $J_0(z)$ $J1(z)$ for Complex Arguments, Columbia University Press, 1947

12. National Bureau of Standards, Tables of Bessel Functions $Y_0(z)$ and $Y_1(z)$ for Complex Arguments, Columbia University Press, 1950.

13. National Bureau of Standards, Tables of the Bessel Functions $Y_0(x)$, $Y_1(x)$, $K_0(x)$, $K_1(x)$ for $0 \leq x \leq 1$, AMS 25, U. S. Government Printing Office, 1952.

14. National Bureau of Standards, Tables of Spherical Bessel Functions, Volumes I and II, Columbia University Press, 1947.

15. E. Jahnke and F. Emde, Tables of Functions, 4th Edition, Dover Publications, New York, 1945.

16. R. D. Halbgewachs, BES: A Routine for the Evaluation of Cylindrical Bessel Functions, SC-M-69-336, Sandia Laboratories, Albuquerque, New Mexico, October 1969.

APPENDIX A

Hankel's Asymptotic Expansions

APPENDIX A

Hankel's Asymptotic Expansions

This expansion is defined by considering υ fixed and $|z| \to \infty$ with

$$J_\upsilon(z) = \sqrt{\frac{2}{\pi z}} \{P(\upsilon,z) \cos \chi - Q(\upsilon,z) \sin \chi\} \tag{A1}$$

and

$$Y_\upsilon(z) = \sqrt{\frac{2}{\pi z}} \left\{ P(\upsilon,z) \sin \chi + Q(\upsilon,z) \cos \chi \right\} \tag{A2}$$

where

$$|\arg z| < \pi \text{ and } \chi = z - \tfrac{1}{2}\upsilon\pi - \tfrac{1}{4}\pi.$$

The terms $P(\upsilon,z)$ and $Q(\upsilon,z)$ are defined by

$$P(\upsilon,z) \sim \sum_{k=0}^{\infty} (-1)^k \frac{(\upsilon,2k)}{(2z)^{2k}} \tag{A3}$$

and

$$Q(\upsilon,z) \sim \sum_{k=0}^{\infty} (-1)^k \frac{(\upsilon,2k+1)}{(2z)^{2k+1}} \tag{A4}$$

where (υ,m) is Hankel's symbol and is defined to be

$$(\upsilon,m) = \frac{\Gamma(\frac{1}{2}+\upsilon+m)}{m!\,\Gamma(\frac{1}{2}+\upsilon-m)} . \tag{A5}$$

If we let $u = 4\upsilon^2$ and expand (A5) we have the forms for $P(\upsilon,z)$ and $Q(\upsilon,z)$ that appear most in the literature,

$$P(\upsilon,z) \sim 1 - \frac{(u-1)(u-9)}{2!\,(8z)^2} + \frac{(u-1)(u-9)(u-25)(u-49)}{4!\,(8z)^4} - \ldots ,$$

and

$$Q(\upsilon,z) \sim \frac{u-1}{8z} - \frac{(u-1)(u-9)(u-25)}{3!\,(8z)^3} + \ldots .$$

One of the most important characteristics of these expansions is the fact that the series (A3) and (A4) are not convergent and must be examined carefully during their use. Without loss of generality, in (A3) consider

$$P(\upsilon,z) \sim \sum_{k=0}^{\infty} a_k, \qquad a_0 = 1$$

where we let $a_{k+1} = r\, a_k$. By using the definition of (A5) we find

$$r = \frac{(2k+\upsilon+3/2)\,(2k+\upsilon+1/2)\,(2k-\upsilon+1/2)\,(2k-\upsilon+3/2)}{(2z)^2\,(2k+2)\,(2k+1)}$$

$$= \frac{(4k^2+6k-\upsilon^2+9/4)\,(4k^2+2k-\upsilon^2+1/4)}{(2z)^2\,(4k^2+6k+2)} .$$

For $\upsilon = 0$, we find that the terms a_k are monotonically decreasing until $k \approx \text{int}\,|z|$. The ratio r then assumes values greater than unity and the terms increase in magnitude. For values of $\upsilon \neq 0$, by examining (A5) we see that k must be slightly larger than z before a turnaround occurs. This behavior is demonstrated during this evaluation of the series by computer. In summary, we find the magnitude of the terms a_k strictly monotonically decreasing to some term a_m such that

$$|a_0| > |a_1| > \cdots > |a_m| .$$

Then the situation reverses and the series begins to diverge with the remaining terms monotonically increasing

$$|a_m| \leq |a_{m+1}| < |a_{m+2}| < \cdots .$$

It has been found from extensive tests that provided the orders of the functions are not too large and the series (A3) and (A4) are stopped at these minimum points, the resulting function values are good. Indeed, in most cases, convergence within the limitations of the computer occurs before this turning point. By convergence we

mean that a certain point is reached when the remaining terms in the descending part of the series no longer significantly contribute to the summation, i.e.,

$$\left|a_{N+1}\right| < \epsilon \left|\sum_{k=0}^{N} a_k\right| .$$

With certain restrictions placed on the argument and order, Watson[1] makes a stronger statement in the form of a theorem:

> If in the expansions for $J_\upsilon(z)$ and $Y_\upsilon(z)$
>
> z is real and positive,
>
> υ is real,
>
> $2m > |\upsilon| - \frac{1}{2}$ for $P(\upsilon,z)$, and
>
> $2m \geq |\upsilon| - 3/2$ for $Q(\upsilon,z)$,
>
> then in the oscillatory parts of the series for $P(\upsilon,z)$ and $Q(\upsilon,z)$ the remainders after m terms in the expansions are of the same sign as, and numerically less than, the first terms neglected.

From (A6) we can see that by increasing the size of the order, υ, we also increase the number of terms a_k available for evaluation of the series before cutoff becomes necessary. However, the rate of convergence slows down so more terms are necessary for evaluation. The slower convergence rate, plus the oscillatory behavior of the terms, gives rise to accuracy deterioration for increasing orders. Consider the relative error between a true function value, f_t, and a function value from the corresponding expansion, f_e, so that

$$\epsilon_r = \left|\frac{f_t - f_e}{f_t}\right| < 10^{-8} .$$

With this acceptance level on f_e, a rough estimate of the maximum order N that can be used in (A1) or (A2) is

$$N = \frac{3\sqrt{a}}{4}$$

where μ is determined from the positive solution of

$$\frac{(\mu-1)(\mu-9)}{2(8z)^2} = 1 .$$

For more strict acceptance levels, N decreases in size.

Notice that for orders $0 \leq \upsilon < 1$ and $|z| > 10.0$ the rate of convergence is very fast and there is no problem with the turning point. Hence, the expansions are particularly useful in calculating values for the normalization of $J_\upsilon(z)$ and initial values of $Y_\upsilon(z)$.

APPENDIX B

Argument-Order-Dependent Relations Between Bessel Functions

APPENDIX B

Argument-Order-Dependent Relations Between Bessel Functions

Since Bessel's differential equation is unaltered if z is replaced by -z, we must expect the functions $J_{\pm\upsilon}(-z)$ to be solutions of the equation satisfied by $J_{\pm\upsilon}(z)$. For that matter, we must consider negative arguments and orders for each of the Bessel functions, cylindrical or spherical. We first define z^{υ} to be

$$z^{\upsilon} = \exp(\upsilon \log z)$$

where the phase (argument) of z is taken to be such that

$$-\pi < \arg z \leq \pi \ .$$

Now let us consider Bessel functions of argument $ze^{m\pi i}$, where m is any integer, arg z has its principal value, and we define

$$\arg(ze^{m\pi i}) = \arg z + m\pi \ .$$

Watson[1] defines $J_{\upsilon}(z)$ so that

$$J_{\upsilon}(ze^{m\pi i}) = e^{m\upsilon\pi i} J_{\upsilon}(z) \ . \tag{B1}$$

From the relation

$$Y_{\upsilon}(z) = \frac{J_{\upsilon}(z) \cos \upsilon\pi - J_{-\upsilon}(z)}{\sin \upsilon\pi} \tag{B2}$$

we can define the function of the second kind for all values of the argument. Using (B2) we have

$$Y_{\upsilon}(ze^{m\pi i}) = \frac{e^{m\upsilon\pi i} J_{\upsilon}(z) \cos \upsilon\pi - e^{-m\upsilon\pi i} J_{-\upsilon}(z)}{\sin \upsilon\pi} \ . \tag{B3}$$

Now

$$e^{i\theta} J_\upsilon(z) = J_\upsilon(z) (\cos\theta - i \sin\theta + 2i \sin\theta)$$

$$= e^{-i\theta} J_\upsilon(z) + 2i \sin\theta J_\upsilon(z). \qquad \text{(B4)}$$

If we let $\theta = m\upsilon\pi$ and substitute (B4) into (B3) we get

$$Y_\upsilon(ze^{m\pi i}) = \frac{e^{-im\upsilon\pi} J_\upsilon(z) \cos\upsilon\pi - e^{-im\upsilon\pi} J_{-\upsilon}(z)}{\sin\upsilon\pi}$$

$$+ 2i J_\upsilon(z) \cot\upsilon\pi \sin m\upsilon\pi$$

$$= e^{-im\upsilon\pi} Y_\upsilon(z) + 2i J_\upsilon(z) \cot\upsilon\pi \sin m\upsilon\pi. \qquad \text{(B5)}$$

Since

$$H_\upsilon^{(1)}(z) = J_\upsilon(z) + iY_\upsilon(z)$$

and

$$H_\upsilon^{(2)}(z) = J_\upsilon(z) - iY_\upsilon(z) ,$$

we obtain additional relations for the functions of the third kind from the relations shown in (B1) and (B5). That is,

$$\sin\upsilon\pi H_\upsilon^{(1)}(ze^{m\pi i}) = -\sin\{(m-1)\upsilon\pi\} H_\upsilon^{(1)}(z)$$

$$-e^{-\upsilon\pi i} \sin m\upsilon\pi H_\upsilon^{(2)}(z) \qquad \text{(B6)}$$

and

$$\sin\upsilon\pi H_\upsilon^{(2)}(ze^{m\pi i}) = \sin\{(m+1)\upsilon\pi\} H_\upsilon^{(2)}(z)$$

$$+ e^{\upsilon\pi i} \sin m\upsilon\pi H_\upsilon^{(1)}(z). \qquad \text{(B7)}$$

When an argument is supplied to BESSEL or BESSPH, checks on the quadrant of the argument are made. Thereafter, a code is held and all calculations are performed on a transformed argument in the first quadrant. Any adjustments that are necessary as indicated by the code are then made before the function values are returned. We use the

equations shown above to transform the function values according to the argument as reflected by the code value. In making these transformations we select m = ±1 such that any rotations from one quadrant to another will not cross the negative real axis, a branch cut for the complex logarithm. Hence, we choose m = 1 for arguments which lie in the third quadrant and the rotation is counterclockwise into the first quadrant. For arguments in the second quadrant (including the negative real axis), m = -1 and rotation is clockwise into the fourth quadrant. In going from the fourth quadrant to the first, we have the conjugate relations

$$J_\upsilon(\bar{z}) = \overline{J_\upsilon(z)} \quad , \qquad Y_\upsilon(\bar{z}) = \overline{Y_\upsilon(z)} \quad ,$$

$$H_\upsilon^{(1)}(\bar{z}) = \overline{H_\upsilon^{(2)}(z)} \quad , \qquad H_\upsilon^{(2)}(\bar{z}) = \overline{H_\upsilon^{(1)}(z)} \quad .$$

The spherical Bessel functions are more easily handled since

$$j_n(ze^{m\pi i}) = e^{mn\pi i}\, j_n(z) \quad ,$$

$$y_n(ze^{m\pi i}) = (-1)^m\, e^{mn\pi i} y_n(z) \quad ,$$

$$h_n^{(1)}(ze^{(2m+1)\pi i}) = (-1)^n\, h_n^{(2)}(z) \quad ,$$

$$h_n^{(2)}(ze^{(2m+1)\pi i}) = (-1)^n\, h_n^{(1)}(z) \quad .$$

In addition to checking the argument, orders are also checked and all calculations are performed for positive orders. For the spherical functions, simple relationships exist between the various functions which permit us to evaluate functions of negative orders. These relations can all be deduced from the single equation

$$y_n(z) = (-1)^{n+1}\, j_{-(n+1)}(z).$$

Furthermore, we can determine all the relationships necessary to evaluate cylindrical functions of negative order from those of positive order by using equation (B2).

DISTRIBUTION:

D. R. Cotter, 100
W. C. Myre, 1210
O. M. Stuetzer, 1220
D. M. Olson, 1510
V. E. Blake, Jr., 1520
T. B. Lane, 1540
R. T. Othmer, 1541
R. L. Peurifoy, Jr., 1550
I. M. Moore, 1610
B. E. Arthur, Jr., 1640
R. R. Prairie, 1643
W. L. Stevens, 1650
C. Winter, 1710
D. R. Morrison, 1720
H. H. Wicke, 1721
R. J. Thompson, 1722
R. G. Clem, 1730
H. D. Sivinski, 1740
A. A. Lieber, 1910
C. B. McCampbell, 2310
K. L. Gillespie, 2320
J. P. Shoup, 2330
M. T. Abegg, 2340
W. C. Kraft, 2440
T. S. Church, 2450
G. H. Roth, 2490
C. M. Tapp, 2610
G. W. Rodgers, 2620
E. G. Franzak, 2630
J. A. Hood, 2650
G. W. Gobeli, 5110
O. E. Jones, 5130
J. E. Schirber, 5150
W. Herrmann, 5160
L. D. Bertholf, 5162
C. H. Karnes, 5165
A. W. Snyder, 5220
C. R. Mehl, 5230
J. H. Renken, 5231
E. H. Beckner, 5240
H. M. Stoller, 5310
J. E. McDonald, Actg., 5320
R. L. Schwoebel, 5330
R. G. Kepler, 5510
W. M. O'Neill, 5520
M. J. Davis, 5530
R. L. Brin, 7250
T. L. Pace, 7260
J. E. Stiegler, 7280
W. T. Moffat, 7290
E. K. Montoya, 7292
M. McWhirter, 7320
M. Cowan, 7340
F. W. Neilson, 7360
S. A. Moore, 7370
J. W. Jones, 7510
W. E. Caldes, 7520
R. J. Dill, 7530
J. W. Pearce, 8120
J. D. Benton, 8121
D. E. Gregson, 8130
L. E. Davies, 8150
R. A. Baroody, 8160
G. E. Brandvold, 8170
C. S. Selvage, 8180
R. H. Meinken, 8310
J. W. Weihe, 8320
J. N. Rogers, 8321
A. G. Schuknecht, 8322
G. W. Anderson, Jr., 8330
J. L. Wirth, 8340
A. N. Blackwell, 8350
C. D. Broyles, 9110
G. E. Hansche, 9120
H. E. Viney, 9130
J. R. Banister, 9150
M. L. Kramm, 9210
J. C. Eckhart, 9220
H. H. Patterson, 9230
R. E. Hepplewhite, 9240
R. C. Maydew, 9320
W. H. Curry, 9322
E. C. Rightley, 9323
H. R. Vaughn, 9325
D. F. McVey, 9328
K. J. Touryan, 9340
W. C. Scrivner, 9400
L. E. Mahuron, 9410
J. E. Wesbrook, 9411
C. E. Katzenberger, 9412
G. D. Horne, Jr., 9413
E. C. Domme, 9414
J. L. Tischhauser, 9420
A. R. Iacoletti, 9421
D. A. Young, 9422
R. D. Halbgewachs, 9422 (50)
D. K. Robins, 9424
B. T. Fox, 9427
A. J. Arenholz, 9428
A. J. Clark, Jr., 9510
G. J. Hildebrandt, 9520
L. C. Baldwin, 3412
G. C. McDonald, 3416
Attn: M. S. Goldstein (2)
B. R. Allen, 3422
C. H. Sproul, 3428-1 (65)
B. F. Hefley, 8232 (5)