MaN 16

SC-M-69-336

BES: A ROUTINE FOR THE EVALUATION OF CYLINDRICAL BESSEL FUNCTIONS

R. D. Halbgewachs, 9422
Sandia Laboratories, Albuquerque

October 1969

ABSTRACT

BES evaluates the Bessel functions $J_n(x)$, $Y_n(x)$, $I_n(x)$, and $K_n(x)$ for real arguments and integer orders. The functions $J_n(x)$ and $I_n(x)$ may have positive or negative arguments, while the functions $Y_n(x)$ and $K_n(x)$ are restricted to positive arguments. The order for each of the functions may be positive or negative. In addition, the Hankel functions $H_n^{(1)}(x)$ and $H_n^{(2)}(x)$ may be determined through the use of this routine.

FOREWORD

The Sandia Laboratories Mathematical Program Library consists of a number of dependable, high-quality, general-purpose, mathematical computing routines. The standards established for the library require that these routines be mathematically sound, effectively implemented, extensively tested, and thoroughly documented. This report documents one such routine.

The library emphasizes the effective coverage of various distinct mathematical areas with a minimum number of routines. Nevertheless, it may contain other routines similar in nature but complementary to the one described here. Additional information on the mathematical program library, a description of the standard format for documenting these routines, and a guide to other routines in the library are contained in SC-M-69-337.

This report is also identified within Sandia Laboratories as Computing Publication ML0004/ALL. The routine was originally documented in July 1968. This report and its corresponding library routine are expected to be available from COSMIC shortly after publication.

CONTENTS

BES: A ROUTINE FOR THE EVALUATION OF BESSEL FUNCTIONS

1. Introduction

1.1 Background

Each of the routines previously used at Sandia for the computation of Bessel functions for real arguments and integer orders had certain drawbacks. Some were limited to orders of zero and one, some were vague about ranges of application, and some did not indicate the accuracy of the functions; none of them organized the most commonly used functions into one routine. BES was developed to provide an accurate, clearly defined, multifunction routine.

The routine by Gilbert and Baker,[1] although it was limited to evaluating $J_n(x)$ and $I_n(x)$ and had several drawbacks, did serve as the basis to begin writing the essentially new BES routine. Some of the ideas used by Gilbert and Baker were incorporated during the programming.

1.2 Applicable Programming Languages and Computer Systems

The programming language is a common subset of Control Data 3600 FORTRAN and Control Data 6600 FORTRAN.

Applicable computing systems are the Control Data 3600 Tape SCOPE and Control Data 6600 SCOPE.

In each of these systems the BES routine is maintained in a library file for the convenience of the user. The routine is accessible by means of a few machine-dependent control cards which are described in Appendices D and E.

1.3 Considerations Regarding Use

When discussing Bessel functions, there are two variables that must be considered. Using the notation $f_n(x)$ to represent a Bessel function, we refer to x as the argument of the function f and n as the order of the function.

The routine BES is designed to make function evaluations for one argument at a time. For each argument, however, the user can request that the routine return values for either a single, particular order or a range of orders. For example, suppose we wish to evaluate a function of argument x for all integer orders between 0 and n. By providing the routine with the largest order needed, in this case, n, then the values $f_0(x)$, $f_1(x)$, ..., $f_n(x)$ will be returned in an array of the calling sequence. Let the array be named T1. In this array we would then have $T1(1) = f_0(x)$, $T1(2) = f_1(x)$, ..., $T1(n+1) = f_n(x)$. At the same time, the array element T1(n+1) is equated to another member of the calling sequence, say RSLT1, and this value is returned also. This provides the user with the feature of requesting a function value for a single order and having that value returned directly into his variable name in the calling sequence. Hence, the user can avoid the problem of equating a particular variable name to an array element later in his own routine. To help clarify this point, suppose we request a value be returned for a function of argument x and a specific order k. Then the value k would be entered in the calling sequence as the maximum order needed and the returned value for $f_k(x)$ would be in RSLT1. (Note that this value could also be found in the array element T1(k+1) and furthermore, all integer orders between 0 and k would also be available in the array T1.)

This routine is restricted to real arguments and integer orders (positive, zero, or negative) and cannot be used for arguments or orders beyond these restrictions. For the functions $Y_n(x)$ and $K_n(x)$, negative arguments may produce complex results. Since the output of BES is assumed to be real, negative arguments for these two functions are not permitted. A complex Bessel function routine (BESSEL) has been developed for the Sandia Mathematical Program Library and will soon be released.

By using $J_n(x)$ and $Y_n(x)$, the Hankel functions $H_n^{(1)}(x)$ and $H_n^{(2)}(x)$ may also be calculated by the user (see Section 3).

2. Usage

2.1 Entry

The calling sequence is:

CALL BES(X,NO,KODE,RSLT1,RSLT2,T1,T2,IERR)

2.2 Description of Arguments

In the following discussion we will refer to an argument which must be preset to some value by the calling program prior to the call of the subroutine as an input argument. An argument which will return with a value determined in the called subroutine is referred to as an output argument. BES does not have any dual purpose arguments, i.e., input data becomes replaced by output data. Each of the input arguments can appear as either a variable name, a constant, or an expression, but an output argument can only appear as a variable name (includes array names). Examples of calling sequences are given in Section 2.4.

X - input, type real, the Bessel function argument. Specific restrictions on the argument ranges for each of the functions are machine dependent and are described in Appendices B and C.

NO - input, integer order of the function desired for a single value to be returned, or the maximum order desired if an array of values is to be returned. Restrictions on the magnitude of the order are dependent upon the magnitude of the argument, X, and are described in Appendices B and C.

KODE - input, an integer indicator for the particular function (functions) to be computed.

KODE = 10 -- function $J_n(x)$ only
= 11 -- function $Y_n(x)$ only
= 12 -- $J_n(x)$ and $Y_n(x)$
= 20 -- function $I_n(x)$ only
= 21 -- function $K_n(x)$ only
= 22 -- $I_n(x)$ and $K_n(x)$

RSLT1 - output, type real, contains the function value for $J_n(x)$ or $I_n(x)$ corresponding to the order NO and argument X supplied, depending on the KODE value. If KODE = 11 or 21, RSLT1 contains no computed results.

RSLT2 - output, type real, contains the function value of $Y_n(x)$ or $K_n(x)$ corresponding to the order NO and argument X supplied, depending on the KODE value. If KODE = 10 or 20, RSLT2 contains no computed results.

T1 - output, type real, is a work area within BES which will return with the array of function values for $J_n(x)$ or $I_n(x)$ of orders zero through NO. T1 must be dimensioned in the calling program and must contain at least M cells of storage where

$$M = \max\{|NO|, \text{int}(2|x|)\} + 51 .$$

Upon return from the subroutine, T1(1) = value for function of order zero, ..., T1(NO+1) = value for function of order NO.

T2 - output, type real, is similar to T1 for the functions $Y_n(x)$ and $K_n(x)$. An exception is that if T2 is not to be used (only $J_n(x)$ or $I_n(x)$ called), then T2 may have a dimension of 1 since the parameter must still appear in the calling sequence. Otherwise, T2 must be dimensioned at least M.

IERR - output, an integer indicator to the calling program that an error condition exists. IERR normally has the value 1, and a value greater than 1 indicates an error condition. Error codes and descriptions are listed in Section 2.8.

2.3 Restrictions Between Arguments

The bounds on the maximum absolute order, $|n|$, that can be requested for a function evaluation are interrelated to the magnitude of the argument for that function. These interrelated restrictions on order and argument ranges are provided in Appendices B and C. The restrictions shown in these appendices are checked by the routine and any violation will cause an error condition.

In this routine, it is necessary that each of the output arguments be distinct from any of the other arguments.

2.4 Principal Uses with Examples

(a) Problem: Given an argument x and order n, find $J_n(x)$.

Example: Consider n = 2, x = 1.5. We must define the dimension of T1 to be at least M = 54 and T2 will have a dimension of 1.

```
DIMENSION T1(54), T2(1)
X = 1.5
CALL BES(X,2,10,RSLT1, RSLT2,T1,T2, IERR)
```

Then RSLT1 = $J_2(1.5)$.

(b) Problem: Given an argument x and largest order N, find $I_0(x)$, $I_1(x),\ldots,I_N(x)$.

Example: Consider N = 20, x = -40.0. We must define the dimension of T1 to be at least M = 131 and T2 will have a dimension of 1.

```
DIMENSION T1(131), T2(1)
NO = 20
CALL BES(-40.0, NO,20,RSLT1,RSLT2,T1,T2,IERR)
```

Then T1(1) = $I_0(-40.0),\ldots,$T1(21) = $I_{20}(-40.0)$.

(c) Problem: Given an argument x and largest negative order -N, find $J_{-N}(x)$ and $Y_0(x)$, $Y_{-1}(x),\ldots,Y_{-N}(x)$.

Example: Consider N = 30 and x = 4.6. Suppose that the parameter KODE takes on different values in the calling program according to different sections of the program. In this particular instance we have KODE = 12. We must dimension both T1 and T2 with at least M = 81.

```
DIMENSION T1(81), T2(81)
X = 4.6
CALL BES(X,-30,KODE,RSLT1,RSLT2,T1,T2,IERR)
```

Then RSLT1 = $J_{-30}(4.6)$ and

T2(1) = $Y_0(4.6),\ldots,$T2(31) = $Y_{-30}(4.6)$.

2.5 Library Routines Explicitly Required

The standard Fortran subroutines used are IABS, ABS, IFIX, EXP, ALOG, and SQRT. In addition, the mathematical library routine ERRCHK[2] is used to process error codes and messages.

2.6 User-Supplied Routines Required

BES does not require any user supplied subroutines.

2.7 Cautions and Restrictions

The error conditions are tested in the routine prior to any major computation, and therefore the output arguments do not contain meaningful results upon an error return. In addition, failure to provide sufficient core storage in the dimensioning of the arrays T1 and T2 is not checked by the routine and should this problem arise, the routine will execute improperly.

2.8 Error Conditions, Messages, and Codes

Error conditions except improperly dimensioned arrays (T1 and T2) are reflected in the parameter IERR of the calling sequence. Error messages are processed by the error routine ERRCHK[2] and unless preset by the user by a call to the routine ERRSET, messages are printed, and all error conditions checked are considered fatal. If the user desires, error conditions may be preset as nonfatal and an error checked by examining IERR. For example, prior to any calls to the subroutine the user could have the statement

CALL ERRSET(100,0).

In this case, any errors encountered are set as nonfatal and up to 100 error messages could be printed. Should more than 100 errors occur (say, during a repeated loop of calls) additional messages would no longer be printed, but execution would continue.

Input parameters to BES are not altered and are returned to the calling program as originally supplied. If the nonfatal option is chosen and an error condition occurs, the user may wish to print out the input parameters of the calling sequence. Values for IERR and the corresponding messages are

- = 1, normal - no errors, no message printed
- = 2, ARGUMENT IS OUT OF RANGE (for J(x) and Y(x))
- = 3, ORDER REQUESTED WAS TOO LARGE FOR GIVEN ARGUMENT
- = 4, ARGUMENT TOO LARGE FOR I(X) AND K(X)
- = 5, NEGATIVE ARGUMENTS FOR Y(X) OR K(X) MAY PRODUCE COMPLEX RESULTS (negative argument for these two functions is not allowed)
- = 6, PARAMETER KODE INCORRECT FOR CALL (if KODE has any value other than 10, 11, 12, 20, 21, or 22).

3. Mathematical Methods

3.1 Statement of Problem

To evaluate the Bessel functions $J_n(x)$, $Y_n(x)$, $I_n(x)$, and $K_n(x)$ for real arguments, x, and integer orders, n.

3.2 Methods Used

The primary method used in the evaluation of these functions is that of repeated application of the recurrence relations described in Section 3.4. There is, however, quite a distinction between the ways in which the recurrence relations are used for each of the individual functions. For the functions of the first kind, $J_n(x)$ and $I_n(x)$, the recurrence relations are utilized in a backwards direction. These techniques were first introduced by Miller[3] and then carried further in the development by Goldstein and Thaler.[4] For the functions of the second kind, $Y_n(x)$ and $K_n(x)$, the recurrence relations are used in the "normal," forward direction.

Although discussed in more detail in Section 3.4, we note here that in the backwards technique, a correction to calculated values is made after the recurrence takes place. However, in the forward recurrence, the initial values of the functions must be calculated to within some given degree of accuracy to start the recurrence. This problem introduces the other mathematical methods used in the routines. In the evaluation of $Y_n(x)$ and $K_n(x)$, the initial values used for starting the recurrence are the zero and first order values. A simple equation to obtain these starting values is not readily available. Instead, more complex equations are necessary. To determine $Y_0(x)$, a series expression involving the $J_n(x)$ functions is used and $K_0(x)$ is determined from polynomial approximations. For both functions, the first order values are determined by utilizing Wronskian relations.

Special treatment is given to the functions of the first kind when $|x| < 1.0$. Using the largest order requested, N, we calculate $J_N(x)$, $J_{N-1}(x)$, $I_N(x)$, and $I_{N-1}(x)$. For these small arguments, x, ascending series are used for the evaluation since convergence is very fast. Using these true values, the recurrence relations are then used to determine the remaining orders. This method is used in preference to the backwards recurrence algorithm since the results produced are more accurate for this particular range of arguments.

It should be noted that there are various other approaches that can be used in evaluating these functions. The treatise by Watson[5] covers many aspects of Bessel functions, including various series definitions, asymptotic forms, properties about the zeros of the functions, etc. The reason for using the particular equations and techniques discussed here and in Section 3.4 is a combination of two factors: (1) the methods provide results with as much or more accuracy as results using other approaches, and (2) the methods provide results as fast or faster than other approaches.

3.3 Mathematical Range and Domain

The routine is restricted to real arguments and integer orders, and the function values returned are real. Specific computer range restrictions of the variables are described in Appendices B and C.

3.4 Equations and Discussion

In discussing the methods used in BES, we first describe the general recurrence relations and then elaborate on the backwards recurrence techniques. Furthermore, the series used to evaluate the functions of the first kind for $|x| < 1.0$ are shown. Keeping in mind that recurrence relations are used in the forward direction for functions of the second kind, we then describe the starting methods for these functions.

General recurrence relations:

Let $f_n(x)$ represent either $J_n(x)$ or $Y_n(x)$. Then the recurrence relations for these two functions is

$$f_{n-1}(x) + f_{n+1}(x) = \frac{2n}{x} f_n(x) . \qquad (1)$$

For the function $I_n(x)$ we have

$$I_{n-1}(x) - I_{n+1}(x) = \frac{2n}{x} I_n(x) \qquad (2)$$

and for $K_n(x)$ we have

$$K_{n+1}(x) - K_{n-1}(x) = \frac{2n}{x} K_n(x) \ . \qquad (3)$$

Backwards recurrence techniques:

By Miller's technique of backward recurrence,[3] we determine the values of $J_n(x)$ and $I_n(x)$. Given an argument, x, and some order N (largest order desired for an array of function values), let

$$M = \max \{|N|, \text{int}(2|x|)\} + 51 \ . \qquad (4)$$

Then let

$$J_M^*(x) = I_M^*(x) = 0.0$$

and

$$J_{M-1}^*(x) = e^{-x} \{I_{M-1}^*(x)\} = \xi$$

where ξ is a very small value, depends on the computer used, and is defined specifically in Appendices B and C.

Then by applying equation (1), recur to $J_0^*(x)$. Since recurrence relations remain valid when a constant is multiplied through the relations, we can consider e^{-x} as a constant and apply equation (2) in recurring down to $e^{-x}\{I_0^*(x)\}$.

These values determined are not the true function values desired, but they do have the important property of being properly related in the same scale. Because of this, we need only determine what the scale adjustment factors, α_1 and α_2, are such that

$$J_0(x) = \alpha_1 J_0^*(x), \ \ldots, \ J_N(x) = \alpha_1 J_N^*(x) \quad \text{and}$$

$$I_0(x) = \alpha_2 \, e^{-x} \{I_0^*(x)\}, \ \ldots, \ I_N(x) = \alpha_2 \, e^{-x} \{I_N^*(x)\} \ .$$

From Abramowitz and Stegun[6] we have the associated series

$$J_0(x) + 2\sum_{k=1}^{\infty} J_{2k}(x) = 1 \text{ , and}$$

$$I_0(x) + 2\sum_{k=1}^{\infty} I_k(x) = e^x \text{ .}$$

Let

$$\beta_1 = J_0^*(x) + 2\sum_{k=1}^{M/2} J_{2k}^*(x) \text{ ,} \tag{5}$$

and

$$\beta_2 = e^{-x}\left\{I_0^*(x) + 2\sum_{k=1}^{M} I_k^*(x)\right\} \text{ .} \tag{6}$$

Since the multiplying factors α_1 and α_2 occur for each function value in making the corrections, we can define the factors by using (5) and (6) so that

$$\alpha_1 = 1/\beta_1 \quad \text{and} \quad \alpha_2 = e^x/\beta_2 \text{ .} \tag{7}$$

The final step in the process is to make the necessary corrections to the calculated values $J_n^*(x)$ and $I_n^*(x)$ by the factors determined in (7).

Series for functions of the first kind, $|x| < 1.0$:

For arguments in the range $0.0 \le |x| < 1.0$, $J_n(x)$ and $I_n(x)$ for orders NO and NO-1 are evaluated by using the ascending series

$$J_n(x) = (x/2)^n \sum_{k=0}^{\infty} \frac{(-x^2/4)^k}{k!\,(n+k)!} \text{ , and} \tag{8}$$

$$I_n(x) = (x/2)^n \sum_{k=0}^{\infty} \frac{(x^2/4)^k}{k!(n+k)!} . \tag{9}$$

From these values, the recurrence relations (1) and (2) are used to determine the lower ordered functions.

Evaluation of functions of the second kind:

As previously mentioned, the recurrence relations (1) and (3), respectively, are used to evaluate $Y_n(x)$ and $K_n(x)$. Before this is done, however, we must first calculate the starting values for these functions. For $Y_0(x)$ we use the equation

$$Y_0(x) \sim \frac{2}{\pi} \{\ln(x/2) + \gamma\} J_0(x) + \frac{4}{\pi} \sum_{k=1}^{M/2} (-1)^k \frac{J_{2k}(x)}{k} , \tag{10}$$

where γ is Euler's constant (=0.577215665...).

By using this value for $Y_0(x)$ we can determine $Y_1(x)$ from the Wronskian relation

$$W\{J_n(x), Y_n(x)\} = J_{n+1}(x)Y_n(x) - J_n(x)Y_{n+1}(x) = \frac{2}{\pi x} , \tag{11}$$

where we let $n = 0$.

Since the $J_n(x)$ functions are necessary in evaluating (10) and (11), the routine will compute these functions even if they have not been requested via KODE. The user must not assume, however, that the $J_n(x)$ function values will be present in the array T1 unless so requested by KODE.

To determine the value of $K_0(x)$ we use the polynomial approximations given in Abramowitz and Stegun.[6]

For $0 < x \le 2$,

$$K_0(x) = -\ln(x/2)\, I_0(x) - a_0 + a_1(x/2)^2 + a_2(x/2)^4 + a_3(x/2)^6 + a_4(x/2)^8 + a_5(x/2)^{10} + a_6(x/2)^{12} + \varepsilon_1(x) , \qquad (12)$$

and for $2 < x < \infty$,

$$x^{1/2} e^x K_0(x) = b_0 - b_1(2/x) + b_2(2/x)^2 - b_3(2/x)^3 + b_4(2/x)^4 - b_5(2/x)^5 + b_6(2/x)^6 + \varepsilon_2(x) , \qquad (13)$$

where

$a_0 = 0.57721566$	,	$a_4 = 0.00262698$	
$a_1 = 0.42278420$	,	$a_5 = 0.00010750$	
$a_2 = 0.23069756$	,	$a_6 = 0.00000740$	
$a_3 = 0.03488590$	,	$\lvert\varepsilon_1(x)\rvert < 1 \times 10^{-8}$	
$b_0 = 1.25331414$	,	$b_4 = 0.00587872$	
$b_1 = 0.07832358$	,	$b_5 = 0.00251540$	
$b_2 = 0.02189568$	,	$b_6 = 0.00053208$	
$b_3 = 0.01062446$	,	$\lvert\varepsilon_2(x)\rvert < 1.9 \times 10^{-7}$.	

Then with the value of $K_0(x)$ from (12) or (13) we can determine $K_1(x)$ from the Wronskian relation

$$W\{K_n(x), I_n(x)\} = K_{n+1}(x)\, I_n(x) + K_n(x)\, I_{n+1}(x) = \frac{1}{x} \qquad (14)$$

where we let $n = 0$.

Additional equations used:

BES is designed for negative or positive orders on any of the functions, and negative or positive arguments for the functions $J_n(x)$

and $I_n(x)$. All calculations are first performed for positive arguments and orders and then if modifications are necessary to complete the function evaluations for negative orders or arguments, the following additional equations are used:

$$J_{-n}(x) = (-1)^n J_n(x) \quad , \quad Y_{-n}(x) = (-1)^n Y_n(x)$$
$$I_{-n}(x) = I_n(x) \quad , \quad K_{-n}(x) = K_n(x)$$
$$J_n(-x) = (-1)^n J_n(x) \quad , \quad I_n(-x) = (-1)^n I_n(x) .$$

It should be noted that the above relations are true for integer orders but are not necessarily true for functions of noninteger orders.

As mentioned earlier in Section 1.3, by using this routine the Hankel functions $H_n^{(1)}(x)$ and $H_n^{(2)}(x)$ may also be determined since

$$H_n^{(1)}(x) = J_n(x) + iY_n(x)$$

and

$$H_n^{(2)}(x) = J_n(x) - iY_n(x) .$$

3.5 Error Analysis, Bounds, and Estimates

Much of the error study conducted on BES is machine dependent and hence, is also discussed in Appendices B and C. Theoretical error analyses of Miller's recurrence algorithm have been made by Olver[7] and Gautschi.[8] In these papers, they show that with appropriate starting conditions and recurrence relations, and provided the starting order M is large enough, the functions can be computed to any desired accuracy. By examining the testing results and studies made, it can be seen that for the functions $J_n(x)$, $Y_n(x)$, and $I_n(x)$, accuracy is limited by the precision of the particular computer used.

For $K_n(x)$ we see from equations (12) and (13) that ε_1 and ε_2 are very close estimates to the "worst" deviations that can be encountered. If we consider $K_0^*(x)$ to be the value obtained from the polynomial approximation, then we can write

$$|K_0(x) - K_0^*(x)| \leq |\epsilon(x)| < \alpha$$

where ϵ represents either ϵ_1 or ϵ_2, depending on the range of x, and α is the related numerical bound in (12) or (13). Errors arising during recurrence of the $K_n(x)$ functions are negligible relative to ϵ_1 and ϵ_2.

4. Programming Methods

The steps taken in the routine follow straightforward from the methods and equations discussed in Section 3.4. The range bounds on arguments and orders vary from computer to computer. Exponential overflow or underflow in the computation of these functions are the principal criteria used in determining the restrictions on the ranges of arguments and orders. The fact that e^x is calculated directly in (7) must be considered if the routine is to be modified for a computer other than those specified in Section 1.2. The criteria for convergence in equations (5), (6), (8), (9), and (10) are also computer dependent. To arrive at equation (4) for the determination of M, successive trial evaluations of $J_n(x)$ and $I_n(x)$ were conducted until a value was obtained which provided sufficient accuracy for both $J_n(x)$ and $I_n(x)$, and also provided a sufficient number of terms for evaluating equation (10).

5. Space, Time, and Accuracy Considerations

Because of machine dependence, timing and space requirements are shown in Appendices B and C. It should be noted that if the user has a need to evaluate Bessel functions over a range of orders, but for the same argument, then the array option of the routine can be most effective. In this case, the average computation time per evaluation is considerably less than that of repeated function calls.

As mentioned in Section 3.5, with the exception of the $K_n(x)$ functions, accuracy of the routine is directly related to the precision of the particular computer being used. Results of the testing performed on the functions are detailed in Appendices B and C.

6. Testing Methods

6.1 General

Many relations that are normally considered as tests for the accuracy of computed Bessel functions are implicitly used in BES. Using these particular standard procedures simply does not provide adequate checks of the routine. It was found in studying other Bessel function routines that instances did occur where the testing had fallen into traps of this nature. Although this alone does not invalidate a routine, on the other hand, it certainly cannot validate a routine.

Several of the standard procedures that could not be applied to this routine are briefly discussed. This points out that although considered, these tests were rejected as providing inconclusive results.

Recurrence Relations -- These relations are nicely expressed and are easily handled in testing procedures. But, however good the test results appear, we have not proven the accuracy of the routine since recurrence is used implicitly in the routine. Using these relations would only show that the functions obey the rules of recurrence and this is already known to be true. Furthermore, all of the computed values could be off by a multiplicative constant and still satisfy the recurrence relations.

Wronskian Relations Involving the Same Function -- For example, we might consider

$$W\{J_\nu(x), J_{-\nu}(x)\} = J_{\nu+1}(x)J_{-\nu}(x) + J_\nu(x)J_{-(\nu+1)}(x) = \frac{2\sin(\nu\pi)}{x\pi} .$$

However, in this routine $\nu = n$, an integer, and the right-hand side is zero.

In Section 3.4 we show that the value of $J_{-n}(x)$ is determined from $J_n(x)$ and similarly, $J_{-(n+1)}(x)$ is determined from $J_{n+1}(x)$. Then the left-hand side of the equation is also zero, hence providing inconclusive results. Similar results occur for the other functions when Wronskian relations of this type are used.

Differential Equation Test -- Another approach in testing procedures is to determine if the computed function values satisfy the original defining differential equation. Suppose we consider the equation

$$x^2 \frac{d^2J_n(x)}{dx^2} + x \frac{dJ_n(x)}{dx} + (x^2 - n^2)J_n(x) = 0 .$$

The problem then reduces to that of expressing the derivatives in terms of known function values. It is exactly at this point where the test fails. Consider the relation

$$xJ_n'(x) + nJ_n(x) = xJ_{n-1}(x) .$$

By differentiating again and then making the proper substitutions into the differential equation so that we have the equation in terms of known function values, we find that the differential equation has been reduced to the recurrence relation shown in Section 3.4, equation (1). Again, similar results occur for the other functions.

6.2 Kinds of Tests Used

In testing the function values obtained from the routine, three principal methods were used. First, many comparisons were made to existing tables (Abramowitz and Stegun,[6] Harvard Computation Laboratory,[9] and National Bureau of Standards[10]) to insure that no constant factors (multipliers) were present in the results and also to examine, roughly, the kind of accuracy that was being obtained.

The second test was the use of the interlinked Wronskian relations:

$$W\{J_n(x), Y_n(x)\} = J_{n+1}(x)Y_n(x) - J_n(x)Y_{n+1}(x) = \frac{2}{\pi x} ,$$

$$W\{K_n(x), I_n(x)\} = I_n(x)K_{n+1}(x) + I_{n+1}(x)K_n(x) = \frac{1}{x} .$$

In this test, the true Wronskian value, W_t, was first determined from the right-hand side of the equations and then using the calculated function values from BES the test Wronskian value, W, was determined from the left-hand side of the equations. The test Wronskian is then compared to the true Wronskian by the relation

$$R = \left| \frac{W_t - W}{W_t} \right| .$$

The results from this test are described in Appendices B and C.

As a final and third test of the routine, the zeros of the functions $J_n(x)$ and $Y_n(x)$ for the lower orders were taken from Abramowitz and Stegun[6] and used as arguments for the routine. Section 6.4 details this particular test.

6.3 Normal Cases Tested

Arguments tested were 0.0(0.001)0.01, 0.0(0.01)2.0, 0.0(0.1)20.0, 0.0(1.0)100.0, and 0.0(10.0)1200.0. The restrictions on the magnitudes of orders were determined from the Wronskian tests when the results were not sufficiently accurate. In most cases, the exponents on the function values were the limiting factors.

6.4 Difficult Cases Tested

Although the zeros of the functions $J_n(x)$ and $Y_n(x)$ are not to be construed as "difficult or pathological" arguments, the most likely points of inaccuracy during recurrence may occur near the zeros. To examine the behavior of the routine near some of these points was deemed to be a part of good testing procedure. The values supplied in Abramowitz and Stegun[6] as being the zeros of $J_n(x)$ and $Y_n(x)$, $n = 0,1,\ldots,8$, were given as arguments to the routine. These values covered the first 20 zeros for each of the functions. In addition, arguments near these values were also checked. The results of the tests indicated that the routine did, in fact, behave properly in the neighborhood of zeros and that the function values tended to zero in these neighborhoods.

6.5 Range, Error, and Fault Checks Tested

All of the possible error conditions described in Section 2.8 were checked by calling the routine with faulty parameters.

7. Remarks

Should the user wish to evaluate Bessel functions for arguments or orders beyond the ranges prescribed for this routine, it is recommended that the routine BESSEL (cylindrical Bessel functions for complex arguments and real orders) be used. Although slower in computation time, BESSEL has a very much larger range of applicability.

8. Certification

This routine was subjected to a wide variety of tests. The performance of the routine throughout the tests was checked carefully. The nature of the tests, the reliability of the routine, the error analyses conducted, and the observed variation in accuracy are reported in this document. While it is believed that the facts recorded and the judgments expressed regarding accuracy and reliability are strong indications of the general quality and validity of the routine, the tests should not be considered to be exhaustive. The use of this routine outside of the stated range of application or in violation of stated restrictions may produce unspecified results. The statements made in this document are intended to apply only to those versions of the indicated routine which are released by the Sandia Laboratories Mathematical Program Library Project.

In the preparation for the writing of this routine nearly all of the current Bessel routines available to Sandia users were examined. An extensive investigation of the literature on Bessel functions was made and the author is of the conviction that the methods and testing procedures used in BES are the most suitable and accurate procedures available at this time.

References

1. G. Gilbert and G. Baker, "Bessel Function," CO-OP Library Routine C3 UCSD BES, 1961.

2. C. B. Bailey, Mathematical Program Library, SC-M-69-337, Sandia Laboratories, Albuquerque, New Mexico. In addition to the description of ERRCHK and ERRSET, SC-M-69-337 includes an index to other routines in the library and the control cards for each routine.

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

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

5. G. N. Watson, A Treatise on the Theory of Bessel Functions, 2nd edition, Cambridge Univ. Press, New York, 1952.

6. M. Abramowitz and I. Stegun, Handbook of Mathematical Functions, National Bureau of Standards, AMS 55, 1965, Chapter 9.

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

8. W. Gautschi, "Recursive Computation of the Repeated Integrals of the Error Function," Mathematics of Computation 15, 1961, pp. 227-232.

9. Harvard University Computation Laboratory, Tables of the Bessel Functions of the First Kind, volumes 3-15, Harvard University Press, Cambridge, Mass., 1947-1950.

10. National Bureau of Standards, Tables of the Bessel Functions $Y_0(x)$, $Y_1(x)$, $K_0(x)$, $K_1(x)$, AMS 25, 1952.

11. P. A. Lemke, Auxiliary Library Routines (PREP and COLLECT), Sandia Computing Publication UR0004/6600, June 1969, revised August, 1969.

APPENDIX A

The BES Listing

APPENDIX A

The BES Listing

NOTE: BES is more computer dependent than many of the routines of the mathematical library. For the CDC 3600 computer, certain statements are different than those shown in this listing.

The range test section, cards BES 1290 through BES 1480, and BES 3000 differ as indicated in Appendix B. In addition, the statements which contain embedded values for return in response to zero arguments are different. These cards are BES 1760, 1770, 1800, 1810. The statements BES 2240 and BES 3440 are also changed to test convergence at 10^{-9}.

To implement this routine on a computer other than the CDC 3600 or CDC 6600 requires certain statement changes similar to those just mentioned in the previous paragraph. Some changes must be made and some changes are not as critical to the operation of the routine. The bounds on arguments and orders that are described in Appendices B and C are designed to allow the user the maximum range possible without an exponent overflow or underflow. For other computers, if the routine is to be used for relatively small ranges of orders and arguments, then the range test sections do not necessarily need to be altered. In the case that a fairly wide range is desired, then new bounds would need to be evaluated.

Certain statements in BES must be checked and possibly changed for other computers. The DATA statements may need alteration and depending on the word size of the computer, the DIMENSION statement may also need to be changed. The following is a list of statements that must be checked before implementation:

BES 1760
1770
1800
1810
2240
2630
3440

```
      SUBROUTINE BES(X,NO,KODE,RSLT1,RSLT2,T1,T2,IERR)
C     ---------
C     ---------
C
C     SANDIA MATHEMATICAL PROGRAM LIBRARY
C     MATHEMATICAL COMPUTING SERVICES DIVISION 9422
C     SANDIA LABORATORIES
C     P. O. BOX 5800
C     ALBUQUERQUE, NEW MEXICO  87115
C
C     WRITTEN BY RONALD D. HALBGEWACHS, JULY, 1968.
C
C     CONTROL DATA 6600 VERSION
C
C     ABSTRACT
C
C             THIS ROUTINE CALCULATES THE BESSEL FUNCTIONS J(X),Y(X),
C             I(X), OR K(X) FOR REAL ARGUMENTS AND INTEGER ORDERS.
C
C     DESCRIPTION OF PARAMETERS
C
C             X      = INPUT,REAL ARGUMENT OF THE BESSEL FUNCTION.
C                      THE ARGUMENT MAY BE POSITIVE,ZERO, OR NEGATIVE
C                      (NEG. ARG. FOR Y(X) OR K(X) PRODUCES ERROR
C                      MESSAGE SINCE RESULTS MAY BE COMPLEX.)
C                      RESTRICTION ON RANGE IS
C                         FOR J(X), -1100.0 .LE. X .LE. 1100.0
C                         FOR Y(X), 0.0 .LE. X .LE. 1100.0
C                         FOR I(X), -600.0 .LE. X .LE. 600.0
C                         FOR K(X), 0.0 .LE. X .LE. 600.0
C
C             NO     = INPUT,INTEGER ORDER OF FUNCTION DESIRED FOR A
C                      SINGLE VALUE TO BE RETURNED, OR THE MAXIMUM ORDER
C                      DESIRED (+ OR -) IF AN ARRAY OF VALUES IS TO BE
C                      RETURNED.
C                      LET XX = ABS(X).  THEN BOUNDS ON ORDERS ARE
C
C                      1. FOR 0.0 .LE. XX .LE. 0.025,
C                            ABS(NO) .LE. INT(600.0*XX + 70.0)
C                      2. FOR 0.025 .LT. XX .LE. 0.20,
C                            ABS(NO) .LE. INT(140.0*XX + 83.0)
C                      3. FOR 0.20 .LT. XX .LE. 1.0,
C                          ABS(NO) .LE. INT(42.0*XX + 102.0)
C                      4. FOR 1.0 .LT. XX .LE. 20.0,
C                            ABS(NO) .LE. INT(0.02*XX**3 - 0.86*XX**2 +
C                                            17.15*XX + 124.0)
C                      5. FOR 20.0 .LT. XX .LE. 100.0,
C                            ABS(NO) .LE. INT(2.75*XX + 228.0)
C                      6. FOR 100.0 .LT. XX .LE. 400.0,
C                            ABS(NO) .LE. INT(1.67*XX + 336.0)
C                      7. FOR 400.0 .LT. XX .LE. 1100.0,
C                            ABS(NO) .LE. INT(1.33*XX + 470.0)
C
C             KODE   = INPUT,INTEGER INDICATOR FOR THE PARTICULAR
```

```
C                 FUNCTION TO BE COMPUTED.
C
C                   KODE = 10 -- FUNCTION J(X) ONLY
C                        = 11 --          Y(X) ONLY
C                        = 12 --          J(X) AND Y(X)
C
C                        = 20 --          I(X) ONLY
C                        = 21 --          K(X) ONLY
C                        = 22 --          I(X) AND K(X)
C
C         RSLT1 = OUTPUT,CONTAINS THE REAL FUNCTION VALUE FOR J(X)
C                 OR I(X) CORRESPONDING TO THE ORDER AND ARGUMENT
C                 SUPPLIED, DEPENDING ON THE KODE VALUE. THIS
C                 PARAMETER WOULD CONTAIN THE RESULT IF ONLY ONE
C                 FUNCTION VALUE IS TO BE RETURNED.
C
C         RSLT2 = OUTPUT,CONTAINS THE REAL FUNCTION VALUE FOR Y(X)
C                 OR K(X) IN A MANNER SIMILAR TO RSLT1.
C
C         T1    = OUTPUT,A WORK AREA WHICH WILL CONTAIN THE ARRAY OF
C                 REAL FUNCTION VALUES FOR J(X) OR I(X) OF ORDERS
C                 ZERO THROUGH NO, DEPENDING ON KODE.
C                 T1 MUST BE DIMENSIONED IN THE CALLING PROGRAM AND
C                 MUST CONTAIN AT LEAST M CELLS OF STORAGE, WHERE
C
C                     M = MAX(ABS(NO),INT(2*ABS(X))) + 51
C
C                 IN USING THE ARRAY, T1(1) = FUNCTION OF ORDER 0,
C                 --- T1(NO+1) = FUNCTION OF ORDER NO.
C
C         T2    = OUTPUT,SIMILAR TO T1 FOR THE FUNCTIONS Y(X) OR
C                 K(X). AN EXCEPTION IS THAT IF ONLY J(X) OR I(X)
C                 ARE CALLED, THEN T2 NEEDS NO DIMENSION IN THE
C                 CALLING PROGRAM, BUT THE PARAMETER MUST STILL
C                 APPEAR IN THE CALLING SEQUENCE.  OTHERWISE, T2
C                 MUST BE DIMENSIONED AT LEAST M.
C
C         IERR  = OUTPUT,ERROR FLAG FOR THE CONDITIONS
C
C                     =1, NORMAL - NO ERRORS
C                     =2, ARGUMENT OUT OF RANGE
C                     =3, ORDER TOO LARGE FOR ARGUMENT SUPPLIED
C                     =4, ARGUMENT TOO LARGE FOR I(X) AND K(X)
C                     =5, NEGATIVE ARGUMENTS FOR Y(X) OR K(X)
C                     =6, INCORRECT PARAMETER KODE
C
C     ---------
C     ---------
      DOUBLE PRECISION A0
      DIMENSION T1(1),T2(1),A(7),B(7),C(7)
      DIMENSION NERR1(4),NERR2(6),NERR3(5),NERR4(8),NERR5(5)
      DATA (PI=3.14159265358979),(EULER=0.577215664901533)
      DATA ((C(I),I=1,7)=0.36746691,0.60559366,-0.74350384,0.25300117,
     1-0.04261214,0.00427916,-0.00024846)
```

```
      DATA ((A(I),I=1,7)=0.57721566,0.42278420,0.23069756,0.03488590,
     10.00262698,0.00010750,0.00000740),((B(I),I=1,7)=1.25331414,
     20.07832358,0.02189568,0.01062446,0.00587872,0.00251540,0.00053208)
      DATA (NERR1= 2,25HARGUMENT IS OUT OF RANGE.),(NERR2= 3,49HORDER RE
     1QUESTED WAS TOO LARGE FOR GIVEN ARGUMENT.),(NERR3= 4,37HARGUMENT T
     200 LARGE FOR I(X) AND K(X).),(NERR4= 5,64HNEGATIVE ARGUMENTS FOR Y
     3(X) OR K(X) MAY PRODUCE COMPLEX RESULTS.),(NERR5= 6,34HPARAMETER K
     4ODE INCORRECT FOR CALL.)
      IERR=1
      IF (KODE.LT.10) GO TO 600
      SIGN = 1.0
      KO = IABS(NO) + 1
      XSQFR = 0.25*X*X
C     ---------
C     INITIAL CHECK OF ORDER-ARGUMENT RANGE TO DETERMINE IF ORDER IS OUT
C     OF RANGE FOR THE GIVEN ARGUMENT.
C
C     ---------
      IF (X) 3,27,3
    3 XCHK = ABS(X)
      IF (XCHK-1100.0) 4,18,608
    4 IF (XCHK-0.025) 5,5,6
    5 LARGOR = IFIX(600.0*XCHK + 70.0)
      GO TO 24
    6 IF (XCHK-0.20) 7,7,8
    7 LARGOR = IFIX(140.0*XCHK + 83.0)
      GO TO 24
    8 IF (XCHK-1.0) 9,9,10
    9 LARGOR = IFIX(42.0*XCHK + 102.0)
      GO TO 24
   10 IF (XCHK-20.0) 11,11,12
   11 LARGOR = IFIX(((0.02*XCHK-0.86)*XCHK+17.15)*XCHK+124.0)
      GO TO 24
   12 IF (XCHK-100.0) 13,13,14
   13 LARGOR = IFIX(2.75*XCHK + 228.0)
      GO TO 24
   14 IF (XCHK-400.0) 16,16,18
   16 LARGOR = IFIX(1.67*XCHK + 336.0)
      GO TO 24
   18 LARGOR = IFIX(1.33*XCHK + 470.0)
   24 IF (IABS(NO)-LARGOR) 25,25,606
   25 XX=X
C     ---------
C     DETERMINE WHICH SET OF FUNCTIONS IS TO BE CALCULATED.
C     ---------
   27 MASK1 = KODE/10
      IF (MASK1-2) 30,31,600
   30 MASK2 = KODE-10
      GO TO 32
   31 MASK2 = KODE-20
   32 IF (MASK2-2) 34,36,600
   34 IF (MASK2) 600,37,42
C     ---------
C     CHECK FUNCTIONS J(X) AND I(X) FOR ZERO ARGUMENT.
```

```
C     ---------
   36 IF (X) 604,38,59
   37 IF (X) 58,38,59
   38 IF (NO) 54,40,54
   40 T1(1) = 1.0
      RSLT1 = 1.0
      IF (MASK2.EQ.0) RETURN
C     ---------
C     CHECK FUNCTIONS Y(X) AND K(X) FOR ZERO ARGUMENT.
C     ---------
   42 IF (X) 604,44,59
   44 IF (MASK1.EQ.2) GO TO 50
   46 DO 48 IK=1,KO
   48 T2(IK) = -1.0E322
      RSLT2 = -1.0E322
      RETURN
   50 DO 52 IK=1,KO
   52 T2(IK) = 1.0E322
      RSLT2 = 1.0E322
      RETURN
C     ---------
C     FILL OUT ARRAY FOR J(X) OR I(X) WHEN (NO.NE.0).
C     ---------
   54 DO 56 IK=2,KO
   56 T1(IK) = 0.0
      RSLT1 = 0.0
      T1(1) = 1.0
      IF (MASK2.EQ.0) RETURN
      GO TO 44
   58 X = ABS(X)
   59 MO = IABS(NO)
      IMO = MO
      IF (X-1.0) 60,71,71
   60 IF (MASK1.EQ.2) GO TO 175
C     ---------
C     USE SERIES TO DETERMINE J(N) AND J(N-1) WHEN ARGUMENT IS SMALL,
C     THEN USE RECURRENCE TO DETERMINE REMAINING FUNCTION VALUES.
C     ---------
      IF (MO.GT.1) GO TO 61
      AX = 1.0
      A1 = 1.0
      XORD = MO
      ILOOP = 1
      GO TO 63
   61 A0 = 1.0D00
      IEND = MO-1
      DO 62 IK=1,IEND
      A0 = A0*DBLE(FLOAT(IK))
   62 CONTINUE
      A1 = 1.0D00/A0
      AX = 1.0D00/(A0*MO)
      ILOOP = 1
      XORD = MO
   63 SUMJN = AX
```

```
      DO 66 IK=1,200
      XK = IK - 1
      FA = -XSQFR/((XK+1.0)*(XORD+XK+1.0))
      TERM = AX*FA
      SUMJN = SUMJN + TERM
      IF (SUMJN - 0.0) 64,65,64
   64 CKSUM = ABS(TERM/SUMJN)
      IF (CKSUM - 1.0E-11) 67,67,65
   65 AX = TERM
   66 CONTINUE
   67 IF (ILOOP.GT.1) GO TO 68
      T1(KO) = SUMJN*(X/2.0)**MO
      IF (MO.EQ.0) GO TO 83
      ILOOP = 2
      AX = A1
      XORD = MO-1
      GO TO 63
   68 T1(KO-1) = SUMJN*(X/2.0)**(MO-1)
      IF (KO.LE.2) GO TO 83
      IEND = KO-2
      DO 69 IK=1,IEND
      NK = KO-IK
      T1(NK-1) = 2.0*(NK-1)*T1(NK)/X - T1(NK+1)
   69 CONTINUE
      GO TO 83
   71 IF (MASK2.EQ.0) GO TO 74
C     ---------
C     DETERMINE STARTING LOCATION OF RECURRENCE IF Y(X) OR K(X)
C     ARE TO BE FOUND.
C     ---------
      JO = 2*IFIX(X)
      IF (IMO - JO) 72,73,73
   72 IMO = JO
   73 IMO = IMO + 51
      GO TO 78
C     ---------
C     DETERMINE STARTING LOCATION FOR RECURRENCE OF J(X).
C     ---------
   74 JO = 2*IFIX(X)
      IF (IMO - JO) 75,76,76
   75 IMO = JO
   76 IMO = IMO + 11
C     ---------
C     INITIALIZE VALUES FOR J(X) AND Y(X)
C     ---------
   78 T1(IMO) = 0.0
      T1(IMO-1) = 1.0E-200
      IF (MASK1.EQ.2) GO TO 151
      F = 2*(IMO-1)
      IMO = IMO - 3
      I2 = IMO
   79 F = F - 2.0
C     ---------
C     RECURRENCE USED FOR FUNCTION VALUES.
```

```
C     VARIABLE SUM IS USED TO DETERMINE ADJUSTMENT FACTOR
C     ON RECURRED VALUES.
C     ---------
      T1(I2+1) = F/X*T1(I2+2) - T1(I2+3)
      IF (I2) 80,81,80
   80 I2 = I2-1
      GO TO 79
   81 SUM = T1(1)
      DO 82 J=3,IMO,2
   82 SUM = SUM + 2.0*T1(J)
      F = 1.0/SUM
   83 IF (NO) 86,84,84
   84 IF (XX) 90,32,92
   86 IF (XX) 92,32,90
   90 SIGN = -SIGN
   92 IF (MASK2.EQ.0) GO TO 93
      GO TO 300
   93 IF (X - 1.0) 96,94,94
   94 DO 95 J=1,KO
   95 T1(J) = T1(J)*F
   96 IF (MO.EQ.0) GO TO 98
      DO 97 J=1,KO,2
   97 T1(J+1) = T1(J+1)*SIGN
   98 RSLT1 = T1(KO)
      X = XX
      RETURN
C     ---------
C     INITIALIZE STARTING VALUES FOR I(X) AND K(X) RECURRENCE.
C     ---------
  151 IF (X-600.0) 152,152,602
  152 F = 2*(IMO-1) - 2
      IMO = IMO - 3
      I2 = IMO
  153 T1(I2+1) = F/X*T1(I2+2) + T1(I2+3)
      IF (I2) 154,155,154
  154 I2 = I2-1
      F=F-2.
      GO TO 153
  155 SUM = T1(1)
      DO 170 J=2,IMO
  170 SUM = SUM + 2.0*T1(J)
      F = 1.0/SUM*EXP(X)
      IF (XX) 171,32,172
  171 SIGN = -SIGN
  172 DO 173 J=1,KO,2
      T1(J) = T1(J)*F
  173 T1(J+1) = T1(J+1)*F*SIGN
      RSLT1 = T1(KO)
      IF (MASK2.NE.0) GO TO 400
      X = XX
      RETURN
  175 IF (MO.GT.1) GO TO 177
      AX = 1.0
      A1 = 1.0
```

```
      XORD = MO
      ILOOP = 1
      GO TO 180
  177 A0 = 1.0D00
      IEND = MO - 1
      DO 178 IK=1,IEND
      A0 = A0*DBLE(FLOAT(IK))
  178 CONTINUE
      A1 = 1.0D00/A0
      AX = 1.0D00/(A0*MO)
      XORD = MO
      ILOOP = 1
  180 SUMIN = AX
      DO 182 IK=1,200
      XK = IK-1
      FA = XSQFR/((XK+1.0)*(XORD+XK+1.0))
      TERM = AX*FA
      SUMIN = SUMIN + TERM
      CKSUM = ABS(TERM/SUMIN)
      IF (CKSUM -1.0E-11) 184,184,181
  181 AX = TERM
  182 CONTINUE
  184 IF (ILOOP.GT.1) GO TO 185
      T1(KO) = SUMIN*(X/2.0)**MO
      IF (MO.EQ.0) GO TO 188
      ILOOP = 2
      AX = A1
      XORD = MO-1
      GO TO 180
  185 T1(KO-1) = SUMIN*(X/2.0)**(MO-1)
      IF (KO.LE.2) GO TO 188
      IEND = KO-2
      DO 187 IK=1,IEND
      NK = KO-IK
      T1(NK-1) = 2.0*(NK-1)*T1(NK)/X + T1(NK+1)
  187 CONTINUE
  188 IF (XX) 189,32,190
  189 SIGN = -SIGN
  190 IF (MO.EQ.0) GO TO 194
      DO 192 J=1,KO,2
  192 T1(J+1) = T1(J+1)*SIGN
  194 RSLT1 = T1(KO)
      IF (MASK2.NE.0) GO TO 400
      X = XX
      RETURN
C     ---------
C     EVALUATE Y0 AND Y1 TO START RECURRENCE.
C     ---------
  300 IF (X-1.0) 320,301,301
  320 XOVTWO = X/2.0
      XOVTHR = X/3.0
      T2(1) = (2.0/PI)*ALOG(XOVTWO)*T1(1) + C(1)
      AA = XOVTHR *XOVTHR
      SUM = ((((((C(7)*AA)+C(6))*AA+C(5))*AA+C(4))*AA+C(3))*AA+C(2))*AA
```

```
      T2(1) = T2(1) + SUM
      GO TO 321
  301 DO 302 J=1,IMO
  302 T1(J) = T1(J)*F
      SUMJ1 = 0.0
      SUMJ2 = 0.0
      IF (IMO.LE.80) GO TO 305
      IF (KO - JO) 303,304,304
  303 KEND = JO/2
      GO TO 306
  304 KEND = KO/2
      GO TO 306
  305 KEND = IMO/2
  306 DO 307 N=1,KEND,2
      XN = N
  307 SUMJ1 = SUMJ1 + T1(2*N+1)/XN
      DO 308 N=2,KEND,2
      XN = N
  308 SUMJ2 = SUMJ2 + T1(2*N+1)/XN
      SUMJN = 2.0*(SUMJ2-SUMJ1)
      T2(1) = 2.0/PI*(T1(1)*(ALOG(X/2.0) + EULER) - SUMJN)
  321 IF (MO.GT.0) GO TO 309
      RSLT1 = T1(1)
      RSLT2 = T2(1)
      X = XX
      RETURN
  309 T2(2) = (T1(2)*T2(1) - 2.0/(PI*X))/T1(1)
      IF (MO.EQ.1) GO TO 311
      NORD = KO-1
      DO 310 N=2,NORD
      XN = N-1
      T2(N+1) = (2.0*XN)/X*T2(N) - T2(N-1)
  310 CONTINUE
  311 DO 312 J=2,KO,2
  312 T2(J) = T2(J)*SIGN
      RSLT2 = T2(KO)
      IF (MASK2.EQ.1) GO TO 315
      DO 313 J=1,KO,2
  313 T1(J+1) = T1(J+1)*SIGN
      RSLT1 = T1(KO)
  315 X = XX
      RETURN
  400 IF (X-2.0) 401,401,402
  401 XOVTWO = X/2.0
      T2(1) = -(ALOG(XOVTWO)*T1(1) + A(1))
      AA= XOVTWO*XOVTWO
      SUM = ((((((A(7)*AA)*AA+A(6))*AA+A(5))*AA+A(4))*AA+A(3))*AA+A(2))*
     1AA
      T2(1) = T2(1)+SUM
      GO TO 403
  402 BB = 2.0/X
      T2(1) = B(1)
      SUM = ((((((B(7)*BB)*BB-B(6))*BB+B(5))*BB-B(4))*BB+B(3))*BB-B(2))*
     1BB
```

```
      T2(1) = (T2(1)+SUM)/(SQRT(X)*EXP(X))                              BES 4330
  403 T2(2) = (1.0/X - T2(1)*T1(2))/T1(1)                               BES 4340
      NORD = KO - 1                                                     BES 4350
      DO 405 N=2,NORD                                                   BES 4360
      XN = N-1                                                          BES 4370
      T2(N+1) = (2.0*XN)/X*T2(N) + T2(N-1)                              BES 4380
  405 CONTINUE                                                          BES 4390
      RSLT2 =T2(KO)                                                     BES 4400
      X = XX                                                            BES 4410
      RETURN                                                            BES 4420
  600 CALL ERRCHK(34,NERR5)                                             BES 4430
      IERR = 6                                                          BES 4440
      RETURN                                                            BES 4450
  602 CALL ERRCHK(37,NERR3)                                             BES 4460
      X = XX                                                            BES 4470
      IERR = 4                                                          BES 4480
      RETURN                                                            BES 4490
  604 CALL ERRCHK(64,NERR4)                                             BES 4500
      IERR = 5                                                          BES 4510
      RETURN                                                            BES 4520
  606 CALL ERRCHK(49,NERR2)                                             BES 4530
      IFRR = 3                                                          BES 4540
      RETURN                                                            BES 4550
  608 CALL ERRCHK(25,NERR1)                                             BES 4560
      IERR = 2                                                          BES 4570
      RETURN                                                            BES 4580
      END                                                               BES 4590
```

APPENDIX B

Control Data 3600 Version of BES

APPENDIX B

Control Data 3600 Version of BES

Subroutine BES for this machine is a single precision routine. Due to the large exponent range ($10^{-307} \le |N| \le 10^{307}$) and mantissa (36 bits) this provides sufficient accuracy for the range of arguments and orders shown. The range restrictions on the argument (discussed in Section 2.2) for the CDC 3600 are:

$$\text{for}\quad J_n(x),\quad -1200 \le x \le 1200$$
$$Y_n(x),\quad 0 \le x \le 1200$$
$$I_n(x),\quad -650 \le x \le 650$$
$$K_n(x),\quad 0 \le x \le 650.$$

In addition, the order restrictions are dependent on the argument and these restrictions are:

(1) for $0.0 \le |x| \le 0.025$,
$|NO| \le [830.0\,|x| + 60.0]$

(2) for $0.025 < |x| \le 0.250$,
$|NO| \le [160.0\,|x| + 77.0]$

(3) for $0.250 < |x| \le 1.0$,
$|NO| \le [38.0\,|x| + 110.0]$

(4) for $1.0 < |x| \le 20.0$,
$|NO| \le [0.02\,|x|^3 - 0.86\,|x|^2 + 17.15\,|x| + 135.0]$

(5) for $20.0 < |x| \le 200.0$,
$|NO| \le [2.3\,|x| + 250.0]$

(6) for $200.0 < |x| \le 500.0$,
$|NO| \le [1.5\,|x| + 410.0]$

(7) for $500.0 < |x| \le 650.0$,
$|NO| \le [|x| + 650.0]$

(8) for $650.0 < |x| \le 1000.0$,
$|NO| \le [1.35\,|x| + 480.0]$

(9) for $1000.0 < |x| \le 1200.0$,
$|NO| \le [1.25\,|x| + 590.0]$.

For the CDC 3600, the recurrence starting value ξ (see Section 3.4) has the value $\xi = 1.0 \times 10^{-200}$. It should be noted that the function values returned for zero arguments are embedded values in the routine.

The length of the CDC 3600 version of BES is 2707_8 locations (1479_{10} locations). Timing information for BES is shown in Table I. The time involved to compute $Y_n(x)$ is longer than the other functions since the $J_n(x)$ functions are necessary in the $Y_n(x)$ evaluations (see Section 3.4) and must be determined first.

Since many arguments were tested in the validation of the routine, only a sample of output from test results is shown. Table II(A, B, C, D, E, F) is concerned with the interlinked Wronskian tests that were described in Section 6.2. The true Wronskian value, W_t, is shown at the top of each table. Each table then gives the calculated Wronskian value, W, and the absolute value of the relative error between the two Wronskian values.

In all of the ranges prescribed by the restrictions on argument and order, the results of these tests showed the relative error R to be such that $R \leq 3.0 \times 10^{-10}$. Accuracy of the functions $J_n(x)$, $I_n(x)$, and $Y_n(x)$ is at least nine significant places. The principal restriction on accuracy for these functions is the amount of precision available with the computer. From the results of the Wronskian tests performed we might suspect that many function values for $K_n(x)$ are of better accuracy than that described in Section 3.5. However, as a result of the absolute error described there, we can only guarantee the accuracy of the polynomial approximation for this function.

TABLE I

CDC 3600 Timing of BES

Function Type	KODE	Argument	Max. Order Requested	Time (millisec)
J(x)	10	0.001	60	12
↓	↓	1.0	100	13
		128.0	100	24
		256.0	100	46
		512.0	100	80
		1024.0	100	156
Y(x)	11	0.001	60	70
↓	↓	1.0	100	27
		128.0	100	45
		256.0	100	75
		512.0	100	135
		1024.0	100	255
J(x) & Y(x)	12	Timing was essentially identical to that for Y(x) alone		
I(x)	20	0.001	60	12
↓	↓	1.0	100	17
		128.0	100	27
		256.0	100	49
		512.0	100	95
K(x)	21	0.001	60	12
↓	↓	1.0	100	18
		128.0	100	31
		256.0	100	52
		512.0	100	95
I(x) & K(x)	22	0.001	60	17
↓	↓	1.0	100	25
		128.0	100	38
		256.0	100	64
		512.0	100	104

TABLE II (A)

Wronskian Test for J(x) and Y(x)

Argument = 1.0000000-003, Wronskian Value = 6.3661977235+002

N	W(J(X),Y(X))	ABS(REL. ERROR)
0	6,3661977235+002	0,0000000+000
1	6,3661977236+002	2,3406639-011
2	6,3661977236+002	2,3406639-011
3	6,3661977236+002	2,3406639-011
4	6,3661977240+002	7,0220068-011
5	6,3661977240+002	7,0220068-011
6	6,3661977239+002	4,6813379-011
7	6,3661977239+002	4,6813379-011
8	6,3661977240+002	7,0220068-011
9	6,3661977239+002	4,6813379-011
10	6,3661977239+002	4,6813379-011
11	6,3661977239+002	4,6813379-011
12	6,3661977239+002	4,6813379-011
13	6,3661977235+002	0,0000000+000
14	6,3661977235+002	0,0000000+000
15	6,3661977239+002	4,6813379-011
16	6,3661977239+002	4,6813379-011
17	6,3661977242+002	9,3626757-011
18	6,3661977242+002	9,3626757-011
19	6,3661977242+002	9,3626757-011
20	6,3661977242+002	9,3626757-011
21	6,3661977242+002	9,3626757-011
22	6,3661977240+002	7,0220068-011
23	6,3661977240+002	7,0220068-011
24	6,3661977244+002	1,1703345-010
25	6,3661977244+002	1,1703345-010
26	6,3661977242+002	9,3626757-011
27	6,3661977240+002	7,0220068-011
28	6,3661977242+002	9,3626757-011
29	6,3661977236+002	2,3406689-011
30	6,3661977239+002	4,6813379-011
31	6,3661977239+002	4,6813379-011
32	6,3661977240+002	7,0220068-011
33	6,3661977244+002	1,1703345-010
34	6,3661977244+002	1,1703345-010
35	6,3661977246+002	1,6384682-010
36	6,3661977244+002	1,1703345-010
37	6,3661977245+002	1,4044014-010
38	6,3661977248+002	1,8725351-010
39	6,3661977246+002	1,6384682-010
40	6,3661977245+002	1,4044014-010
41	6,3661977248+002	1,8725351-010
42	6,3661977248+002	1,8725351-010
43	6,3661977249+002	2,1065020-010
44	6,3661977249+002	2,1065020-010
45	6,3661977251+002	2,3406639-010
46	6,3661977253+002	2,5747358-010
47	6,3661977254+002	2,8088027-010
48	6,3661977253+002	2,5747358-010

TABLE II (B)

Wronskian Test for J(x) and Y(x)

Argument = 1.0000000+002; Wronskian Value = 6.3661977235-003

N	W(J(X),Y(X))	ABS(REL, ERROR)
0	6,3661977235-003	0,0000000+000
1	6,3661977234-003	1,7857887-011
2	6,3661977233-003	3,5715773-011
3	6,3661977237-003	1,7857887-011
4	6,3661977237-003	1,7857887-011
5	6,3661977233-003	3,5715773-011
6	6,3661977237-003	1,7857887-011
7	6,3661977234-003	1,7857887-011
8	6,3661977235-003	0,0000000+000
9	6,3661977235-003	0,0000000+000
10	6,3661977234-003	1,7857887-011
11	6,3661977234-003	1,7857887-011
12	6,3661977237-003	1,7857887-011
13	6,3661977237-003	1,7857887-011
14	6,3661977235-003	0,0000000+000
15	6,3661977235-003	0,0000000+000
16	6,3661977235-003	0,0000000+000
17	6,3661977235-003	0,0000000+000
18	6,3661977234-003	1,7857887-011
19	6,3661977235-003	0,0000000+000
20	6,3661977234-003	1,7857887-011
21	6,3661977235-003	0,0000000+000
22	6,3661977235-003	0,0000000+000
23	6,3661977233-003	3,5715773-011
24	6,3661977237-003	1,7857887-011
25	6,3661977237-003	1,7857887-011
26	6,3661977235-003	0,0000000+000
27	6,3661977235-003	0,0000000+000
28	6,3661977235-003	0,0000000+000
29	6,3661977233-003	3,5715773-011
30	6,3661977233-003	3,5715773-011
31	6,3661977233-003	3,5715773-011
32	6,3661977233-003	3,5715773-011
33	6,3661977233-003	3,5715773-011
34	6,3661977233-003	3,5715773-011
35	6,3661977233-003	3,5715773-011
36	6,3661977233-003	3,5715773-011
37	6,3661977234-003	1,7857887-011
38	6,3661977231-003	7,1431547-011
39	6,3661977231-003	7,1431547-011
40	6,3661977231-003	7,1431547-011
41	6,3661977231-003	8,9289434-011
42	6,3661977231-003	7,1431547-011
43	6,3661977231-003	7,1431547-011
44	6,3661977232-003	5,3573660-011
45	6,3661977233-003	3,5715773-011
46	6,3661977231-003	7,1431547-011
47	6,3661977232-003	5,3573660-011
48	6,3661977233-003	3,5715773-011

TABLE II (C)

Wronskian Test for J(x) and Y(x)

Argument = 1.2000000+003, Wronskian Value = 5.3051647698-004

N	W(J(X),Y(X))	ABS(REL. ERROR)
0	5,3051647698-004	0,0000000+000
1	5,3051647698-004	0,0000000+000
2	5,3051647698-004	0,0000000+000
3	5,3051647698-004	0,0000000+000
4	5,3051647698-004	0,0000000+000
5	5,3051647698-004	0,0000000+000
6	5,3051647700-004	5,3573660-011
7	5,3051647699-004	2,6786830-011
8	5,3051647698-004	0,0000000+000
9	5,3051647699-004	2,6786830-011
10	5,3051647700-004	5,3573660-011
11	5,3051647699-004	2,6786830-011
12	5,3051647699-004	2,6786830-011
13	5,3051647699-004	2,6786830-011
14	5,3051647700-004	5,3573660-011
15	5,3051647699-004	2,6786830-011
16	5,3051647699-004	2,6786830-011
17	5,3051647698-004	0,0000000+000
18	5,3051647698-004	0,0000000+000
19	5,3051647699-004	2,6786830-011
20	5,3051647698-004	0,0000000+000
21	5,3051647699-004	2,6786830-011
22	5,3051647699-004	2,6786830-011
23	5,3051647698-004	0,0000000+000
24	5,3051647699-004	2,6786830-011
25	5,3051647699-004	2,6786830-011
26	5,3051647699-004	2,6786830-011
27	5,3051647698-004	0,0000000+000
28	5,3051647699-004	2,6786830-011
29	5,3051647698-004	0,0000000+000
30	5,3051647699-004	2,6786830-011
31	5,3051647699-004	2,6786830-011
32	5,3051647699-004	2,6786830-011
33	5,3051647698-004	0,0000000+000
34	5,3051647698-004	0,0000000+000
35	5,3051647699-004	2,6786830-011
36	5,3051647699-004	2,6786830-011
37	5,3051647698-004	0,0000000+000
38	5,3051647699-004	2,6786830-011
39	5,3051647699-004	2,6786830-011
40	5,3051647699-004	2,6786830-011
41	5,3051647698-004	0,0000000+000
42	5,3051647698-004	0,0000000+000
43	5,3051647698-004	0,0000000+000
44	5,3051647699-004	2,6786830-011
45	5,3051647699-004	2,6786830-011
46	5,3051647699-004	2,6786830-011
47	5,3051647699-004	2,6786830-011
48	5,3051647699-004	2,6786830-011

TABLE II (D)

Wronskian Test for I(x) and K(x)

Argument = 1.0000000-003, Wronskian Value = 1.0000000000+003

N	W(K(X),I(X))	ABS(REL, ERROR)
0	1,0000000000+003	0,0000000+000
1	1,0000000000+003	1,4901161-011
2	1,0000000000+003	0,0000000+000
3	1,0000000000+003	0,0000000+000
4	1,0000000000+003	4,4703484-011
5	1,0000000000+003	1,4901161-011
6	1,0000000000+003	0,0000000+000
7	1,0000000001+003	5,9604645-011
8	1,0000000000+003	1,4901161-011
9	1,0000000000+003	2,9802322-011
10	1,0000000001+003	7,4505806-011
11	1,0000000001+003	7,4505806-011
12	1,0000000001+003	1,1920929-010
13	1,0000000001+003	5,9604645-011
14	1,0000000001+003	5,9604645-011
15	1,0000000001+003	5,9604645-011
16	1,0000000001+003	5,9604645-011
17	1,0000000000+003	4,4703484-011
18	1,0000000001+003	7,4505806-011
19	1,0000000001+003	5,9604645-011
20	1,0000000001+003	5,9604645-011
21	1,0000000001+003	8,9406967-011
22	1,0000000001+003	1,0430813-010
23	1,0000000001+003	8,9406967-011
24	1,0000000001+003	1,1920929-010
25	1,0000000001+003	1,1920929-010
26	1,0000000001+003	8,9406967-011
27	1,0000000001+003	8,9406967-011
28	1,0000000001+003	1,0430813-010
29	1,0000000001+003	1,0430813-010
30	1,0000000001+003	1,1920929-010
31	1,0000000001+003	1,0430813-010
32	1,0000000001+003	1,1920929-010
33	1,0000000001+003	1,0430813-010
34	1,0000000001+003	8,9406967-011
35	1,0000000001+003	7,4505806-011
36	1,0000000001+003	1,0430813-010
37	1,0000000001+003	8,9406967-011
38	1,0000000001+003	1,3411045-010
39	1,0000000001+003	1,6391277-010
40	1,0000000001+003	1,4901161-010
41	1,0000000001+003	1,1920929-010
42	1,0000000001+003	1,6391277-010
43	1,0000000002+003	1,9371510-010
44	1,0000000001+003	1,6391277-010
45	1,0000000002+003	2,0861626-010
46	1,0000000002+003	1,7881393-010
47	1,0000000002+003	1,9371510-010
48	1,0000000002+003	1,7881393-010

TABLE II (E)

Wronskian Test for I(x) and K(x)

Argument = 1.0000000+002, Wronskian Value = 1.0000000000-002

N	W(K(X),I(X))	ABS(REL. ERROR)
0	1,0000000000-002	0,0000000+000
1	1,0000000000-002	0,0000000+000
2	1,0000000000-002	0,0000000+000
3	1,0000000000-002	0,0000000+000
4	1,0000000000-002	0,0000000+000
5	1,0000000000-002	0,0000000+000
6	9,9999999998-003	2,2737368-011
7	9,9999999998-003	2,2737368-011
8	9,9999999998-003	2,2737368-011
9	9,9999999998-003	2,2737368-011
10	9,9999999998-003	2,2737368-011
11	9,9999999998-003	2,2737368-011
12	9,9999999998-003	2,2737368-011
13	9,9999999995-003	4,5474735-011
14	9,9999999995-003	4,5474735-011
15	9,9999999998-003	2,2737368-011
16	9,9999999995-003	4,5474735-011
17	9,9999999995-003	4,5474735-011
18	9,9999999995-003	4,5474735-011
19	9,9999999993-003	6,8212103-011
20	9,9999999993-003	6,8212103-011
21	9,9999999993-003	6,8212103-011
22	9,9999999993-003	6,8212103-011
23	9,9999999993-003	6,8212103-011
24	9,9999999995-003	4,5474735-011
25	9,9999999995-003	4,5474735-011
26	9,9999999995-003	4,5474735-011
27	9,9999999995-003	4,5474735-011
28	9,9999999998-003	2,2737368-011
29	9,9999999998-003	2,2737368-011
30	9,9999999995-003	4,5474735-011
31	9,9999999993-003	6,8212103-011
32	9,9999999995-003	4,5474735-011
33	9,9999999995-003	4,5474735-011
34	9,9999999995-003	4,5474735-011
35	9,9999999998-003	2,2737368-011
36	9,9999999998-003	2,2737368-011
37	9,9999999995-003	4,5474735-011
38	1,0000000000-002	0,0000000+000
39	9,9999999998-003	2,2737368-011
40	9,9999999998-003	2,2737368-011
41	9,9999999998-003	2,2737368-011
42	9,9999999995-003	4,5474735-011
43	9,9999999993-003	6,8212103-011
44	9,9999999995-003	4,5474735-011
45	9,9999999995-003	4,5474735-011
46	9,9999999995-003	4,5474735-011
47	9,9999999993-003	6,8212103-011
48	9,9999999998-003	2,2737368-011

TABLE II (F)

Wronskian Test for I(x) and K(x)

Argument = 6.5000000+002, Wronskian Value = 1.5384615385-003

N	W(K(X),I(X))	ABS(REL, ERROR)
0	1,5384615385-003	0,0000000+000
1	1,5384615385-003	0,0000000+000
2	1,5384615384-003	1,8474111-011
3	1,5384615384-003	3,6948222-011
4	1,5384615384-003	1,8474111-011
5	1,5384615384-003	1,8474111-011
6	1,5384615384-003	1,8474111-011
7	1,5384615384-003	1,8474111-011
8	1,5384615385-003	0,0000000+000
9	1,5384615384-003	1,8474111-011
10	1,5384615384-003	3,6948222-011
11	1,5384615384-003	3,6948222-011
12	1,5384615384-003	1,8474111-011
13	1,5384615384-003	1,8474111-011
14	1,5384615384-003	3,6948222-011
15	1,5384615384-003	1,8474111-011
16	1,5384615384-003	1,8474111-011
17	1,5384615384-003	1,8474111-011
18	1,5384615384-003	1,8474111-011
19	1,5384615384-003	1,8474111-011
20	1,5384615385-003	0,0000000+000
21	1,5384615384-003	1,8474111-011
22	1,5384615385-003	0,0000000+000
23	1,5384615385-003	1,8474111-011
24	1,5384615385-003	1,8474111-011
25	1,5384615385-003	1,8474111-011
26	1,5384615385-003	0,0000000+000
27	1,5384615385-003	0,0000000+000
28	1,5384615385-003	0,0000000+000
29	1,5384615385-003	0,0000000+000
30	1,5384615385-003	0,0000000+000
31	1,5384615385-003	0,0000000+000
32	1,5384615385-003	1,8474111-011
33	1,5384615385-003	1,8474111-011
34	1,5384615385-003	0,0000000+000
35	1,5384615385-003	0,0000000+000
36	1,5384615385-003	0,0000000+000
37	1,5384615385-003	0,0000000+000
38	1,5384615385-003	1,8474111-011
39	1,5384615385-003	0,0000000+000
40	1,5384615385-003	0,0000000+000
41	1,5384615385-003	0,0000000+000
42	1,5384615385-003	0,0000000+000
43	1,5384615385-003	0,0000000+000
44	1,5384615384-003	1,8474111-011
45	1,5384615384-003	1,8474111-011
46	1,5384615385-003	0,0000000+000
47	1,5384615385-003	0,0000000+000
48	1,5384615385-003	0,0000000+000

APPENDIX C

Control Data 6600 Version of BES

APPENDIX C

Control Data 6600 Version of BES

Subroutine BES for this machine is a single precision routine. Due to the large exponent range ($10^{-295} \le |N| \le 10^{322}$) and mantissa (48 bits) this provides sufficient accuracy for the range of arguments and orders shown. Also, since the exponent range is nonsymmetric, the range restrictions on the argument (Section 2.2) for the 6600 are:

for	$J_n(x)$,	$-1100 \le x \le 1100$
	$Y_n(x)$,	$0 \le x \le 1100$
	$I_n(x)$,	$-600 \le x \le 600$
	$K_n(x)$,	$0 \le x \le 600.$

In addition, the order restrictions are dependent on the argument and these restrictions are:

(1) for $0.0 \le |x| \le 0.025$,
$|NO| \le [600.0\ |x| + 70.0]$

(2) for $0.025 < |x| \le 0.20$,
$|NO| \le [140.0\ |x| + 83.0]$

(3) for $0.20 < |x| \le 1.0$,
$|NO| \le [42.0\ |x| + 102.0]$

(4) for $1.0 < |x| \le 20.0$,
$|NO| \le [0.02\ |x|^3 - 0.86\ |x|^2 + 17.15\ |x| + 124.0]$

(5) for $20.0 < |x| \le 100.0$,
$|NO| \le [2.75\ |x| + 228.0]$

(6) for $100.0 < |x| \le 400.0$,
$|NO| \le [1.67\ |x| + 336.0]$

(7) for $400 < |x| \le 1100.0$,
$|NO| \le [1.33\ |x| + 470.0]$.

For the CDC 6600, the recurrence starting value ξ (see Section 3.4) has the value $\xi = 1.0 \times 10^{-200}$. It should be noted that the function values returned for zero arguments are embedded values in the routine.

The length of the CDC 6600 version of BES is 2010_8 locations (1032_{10} locations). Timing information for BES is shown in Table I. The time involved to compute $Y_n(x)$ is longer than the other functions since the $J_n(x)$ functions are necessary in the $Y_n(x)$ evaluations (see Section 3.4) and must be determined first.

Since many arguments were tested in the validation of the routine, only a sample of output from test results is shown. Table II (A, B, C, D, E, F) is concerned with the interlinked Wronskian tests that were described in Section 6.2. The true Wronskian value, W_t, is shown at the top of each table. Each table then gives the calculated Wronskian value, W, and the absolute value of the relative error between the two Wronskian values.

In all of the ranges prescribed by the restrictions on argument and order, the results of these tests showed the relative error R to be such that $R \leq 3.0 \times 10^{-13}$. Accuracy of the functions $J_n(x)$, $Y_n(x)$, and $I_n(x)$ is at least 12 significant places. The principal restriction on accuracy for these functions is the amount of precision available with the computer. From the results of the Wronskian tests performed we might suspect that many function values for $K_n(x)$ are of better accuracy than that described in Section 3.5. However, as a result of the absolute error described there, we can only guarantee the accuracy of the polynomial approximation for this function.

TABLE I

CDC 6600 Timing of BES

Function Type	KODE	Argument	Max. Order Requested	Time (millisec)
J(x)	10	0.001	60	4
↓	↓	1.0	100	4
		128.0	100	6
		256.0	100	12
		512.0	100	20
		1024.0	100	36
Y(x)	11	0.001	60	4
↓	↓	1.0	100	6
		128.0	100	8
		256.0	100	16
		512.0	100	28
		1024.0	100	50
J(x) & Y(x)	12	Timing was essentially identical to that for Y(x) alone		
I(x)	20	0.001	60	2
↓	↓	1.0	100	4
		128.0	100	6
		256.0	100	12
		512.0	100	20
K(x)	21	0.001	60	2
↓	↓	1.0	100	4
		128.0	100	8
		256.0	100	12
		512.0	100	20
I(x) & K(x)	22	0.001	60	6
↓	↓	1.0	100	6
		128.0	100	8
		256.0	100	12
		512.0	100	22

TABLE II (A)

Wronskian Test for J(x) and Y(x)

Argument = 5.0000000E-03, Wronskian Value = 1.273239544735E+02

N	W(J(X),Y(X))	ABS(REL. ERROR)	J(X)	Y(X)
0	1.273239544735E+02	7.1431547E-15	9.999937500090E-01	-3.446792364684E+00
2	1.273239544735E+02	0.	3.124993489586E-06	-5.092990011159E+04
4	1.273239544735E+02	3.5715773E-15	1.627602132161E-12	-4.889250037719E+10
6	1.273239544735E+02	1.0714732E-14	3.390838986350E-19	-1.564558708269E+17
8	1.273239544735E+02	0.	3.784419262429E-26	-1.051383076463E+24
10	1.273239544735E+02	7.1431547E-15	2.628069264069E-33	-1.211193063769E+31
12	1.273239544735E+02	1.4286309E-14	1.244351086077E-40	-2.131699523079E+38
14	1.273239544735E+02	2.5001041E-14	4.273183948639E-48	-5.320721544507E+45
16	1.273239544735E+02	2.8572619E-14	1.112808374507E-55	-1.787762324354E+53
18	1.273239544735E+02	1.7857887E-14	2.272893009748E-63	-7.780341254200E+60
20	1.273239544735E+02	3.2144196E-14	3.738310988412E-71	-4.257402569538E+68
22	1.273239544735E+02	2.8572619E-14	5.057239022131E-79	-2.860974437100E+76
24	1.273239544735E+02	2.8572619E-14	5.726040683957E-87	-2.316244844332E+84
26	1.273239544735E+02	3.5715773E-14	5.505800451918E-95	-2.223595002219E+92
28	1.273239544735E+02	3.5715773E-14	4.551759705534-103	-2.497541860242+100
30	1.273239544735E+02	4.2858928E-14	3.269942362661-111	-3.244806333026+108
32	1.273239544735E+02	4.6430505E-14	2.060195566164-119	-4.828271756408+116
34	1.273239544735E+02	4.6430505E-14	1.147613407119-127	-8.157847859947+124
36	1.273239544735E+02	6.0716815E-14	5.692526875976-136	-1.553254215724+133
38	1.273239544735E+02	6.4288392E-14	2.530461826675-144	-3.310295352598+141
40	1.273239544735E+02	7.1431547E-14	1.013806829510-152	-7.849372272084+149
42	1.273239544735E+02	7.8574702E-14	3.679612592533-161	-2.059675268094+158
44	1.273239544735E+02	6.7859969E-14	1.215516822907-169	-5.951637612485+166
46	1.273239544735E+02	6.7859969E-14	3.670038738197-178	-1.885478783455+175
48	1.273239544735E+02	6.0716815E-14	1.016743893539-186	-6.522248169180+183
50	1.273239544735E+02	7.1431547E-14	2.593734435266-195	-2.454452417704+192
52	1.273239544735E+02	7.5003124E-14	6.112684877594-204	-1.001416581414+201
54	1.273239544735E+02	7.1431547E-14	1.334880525814-212	-4.415846536982+209
56	1.273239544735E+02	7.5003124E-14	2.708767311559-221	-2.098410265375+218
58	1.273239544735E+02	7.5003124E-14	5.120930356979-230	-1.071700086459+227
60	1.273239544735E+02	8.5717856E-14	9.041190633412-239	-5.867772291572+235
62	1.273239544735E+02	8.9289434E-14	1.494115326352-247	-3.436167442010+244
64	1.273239544735E+02	9.2861011E-14	2.316026988642-256	-2.147467197574+253
66	1.273239544735E+02	1.0357574E-13	3.374165202928-265	-1.429354162342+262
68	1.273239544735E+02	1.0714732E-13	4.628738493266-274	-1.011296654037+271
70	1.273239544735E+02	1.0714732E-13	5.989568458950-283	-7.592006220662+279

TABLE II (B)

Wronskian Test for K(x) and Y(x)

Argument = 2.0000000E+02, Wronskian Value = 3.183098861838E-03

N	W(J(X),Y(X))	ABS(REL. ERROR)	J(X)	Y(X)
0	3.183098861838E-03	4.3598356E-15	-1.543743993051E-02	-5.426577524983E-02
20	3.183098861838E-03	8.7196712E-15	3.745093871090E-02	-4.238574289323E-02
40	3.183098861838E-03	0.	-3.193299329803E-02	4.721236385570E-02
60	3.183098861838E-03	4.3598356E-15	3.415650000123E-02	4.658442831622E-02
80	3.183098861838E-03	0.	-1.395009114461E-02	5.725740582833E-02
100	3.183098861838E-03	4.3598356E-15	9.333214186607E-03	-5.990294357227E-02
120	3.183098861838E-03	4.3598356E-15	-4.331910558266E-02	-4.584989654395E-02
140	3.183098861838E-03	0.	4.970931161697E-02	-4.456348248711E-02
160	3.183098861838E-03	8.7196712E-15	-6.237734851090E-02	-3.758953367824E-02
180	3.183098861838E-03	8.7196712E-15	3.971943875705E-02	-7.557346150550E-02
200	3.183098861838E-03	8.7196712E-15	7.648760893097E-02	-1.324833973407E-01
220	3.183098861838E-03	4.3598356E-14	1.092007784858E-04	-3.187038303258E+01
240	3.183098861838E-03	8.7196712E-15	1.923842162395E-09	-1.247477370831E+06
260	3.183098861838E-03	2.6159014E-14	1.683848978223E-15	-1.137953620968E+12
280	3.183098861838E-03	1.7439342E-14	1.339798069630E-22	-1.212435761530E+19
300	3.183098861838E-03	2.1799178E-14	1.394118395466E-30	-1.021109739783E+27
320	3.183098861838E-03	1.3079507E-14	2.437257529640E-39	-5.228299172841E+35
340	3.183098861838E-03	1.7439342E-14	8.625601061854E-49	-1.342154113691E+45
360	3.183098861838E-03	1.3079507E-14	7.148802920207E-59	-1.487525741497E+55
380	3.183098861838E-03	1.3079507E-14	1.561126009224E-69	-6.310490209254E+65
400	3.183098861838E-03	4.3598356E-15	9.906636976730E-81	-9.275431234715E+76
420	3.183098861838E-03	0.	1.984820746808E-92	-4.342324643433E+88
440	3.183098861838E-03	3.0518849E-14	1.348425690592-104	-6.023209779112+100
460	3.183098861838E-03	6.5397534E-14	3.305782443772-117	-2.324436024429+113
480	3.183098861838E-03	9.1556548E-14	3.089472062197-130	-2.361199515955+126
500	3.183098861838E-03	1.1771556E-13	1.155689238211-143	-6.010348501162+139
520	3.183098861838E-03	9.5916384E-14	1.807717165201-157	-3.668416238189+153
540	3.183098861838E-03	1.3951474E-13	1.229938181859-171	-5.159548385572+167
560	3.183098861838E-03	1.1335573E-13	3.772662534661-186	-1.613036737817+182
580	3.183098861838E-03	9.1556548E-14	5.390271644012-201	-1.084677149400+197
600	3.183098861838E-03	6.1037699E-14	3.696364517353-216	-1.522300687694+212

TABLE II (C)

Wronskian Test for J(x) and Y(x)

Argument = 1.1000000E+03, Wronskian Value = 5.787452476069E-04

N	W(J(X),Y(X))	ABS(REL. ERROR)	J(X)	Y(X)
0	5.787452476069E-04	1.1989548E-14	2.265627601561E-02	-8.089397067848E-03
50	5.787452476069E-04	5.9947740E-15	-1.687927418515E-02	-1.715908200502E-02
100	5.787452476069E-04	1.1989548E-14	-1.169985477788E-02	-2.107759515997E-02
150	5.787452476069E-04	1.7984322E-14	2.148906893374E-02	1.106445736868E-02
200	5.787452476069E-04	2.3979096E-14	1.391179656327E-02	-1.987503529531E-02
250	5.787452476069E-04	3.5968644E-14	2.429231181991E-02	-2.044706617427E-03
300	5.787452476069E-04	4.1963418E-14	-2.452368143035E-02	3.716037582446E-04
350	5.787452476069E-04	4.7958192E-14	-1.849206087983E-02	1.639643335413E-02
400	5.787452476069E-04	5.9947740E-14	-1.423638317036E-02	-2.045977825769E-02
450	5.787452476069E-04	5.9947740E-14	-9.316185299129E-03	2.339774124839E-02
500	5.787452476069E-04	7.1937288E-14	-1.672474733776E-02	1.923616165890E-02
550	5.787452476069E-04	7.1937288E-14	1.155444656806E-02	-2.312513537727E-02
600	5.787452476069E-04	7.1937288E-14	-7.891283007186E-03	-2.506468222866E-02
650	5.787452476069E-04	5.9947740E-14	2.672047170156E-02	1.845253395872E-03
700	5.787452476070E-04	1.0191116E-13	3.290900670808E-03	-2.719251010547E-02
750	5.787452476070E-04	1.0790593E-13	2.812132122804E-02	-5.850857063021E-04
800	5.787452476070E-04	1.1989548E-13	-5.575037787750E-03	-2.849806739415E-02
850	5.787452476070E-04	1.3787980E-13	2.964913484770E-02	-5.719239396019E-03
900	5.787452476070E-04	1.0790593E-13	8.137637219940E-03	-3.066516795050E-02
950	5.787452476070E-04	1.0191116E-13	1.205297042146E-03	3.386089080305E-02
1000	5.787452476069E-04	5.9947740E-14	-3.263155660891E-02	1.800782532282E-02

TABLE II (D)

Wronskian Test for I(x) and K(x)

Argument = 5.0000000E-03, Wronskian Value = 2.000000000000E+02

N	W(K(X),I(X))	ABS(REL. ERROR)	I(X)	K(X)
0	2.000000000000E+02	9.0949470E-15	1.000006250009E+00	5.414288976226E+00
2	2.000000000000E+02	1.3642421E-14	3.125006510420E-06	7.999950001932E+04
4	2.000000000000E+02	4.5474735E-15	1.627606201172E-12	7.679984000030E+10
6	2.000000000000E+02	4.5474735E-15	3.390845041425E-19	2.457596928004E+17
8	2.000000000000E+02	4.5474735E-15	3.784424518570E-26	1.651505725442E+24
10	2.000000000000E+02	4.5474735E-15	2.628072250513E-33	1.902534973196E+31
12	2.000000000000E+02	0.	1.244352282569E-40	3.348461975610E+38
14	2.000000000000E+02	4.5474735E-15	4.273187509627E-48	8.357761821697E+45
16	2.000000000000E+02	9.0949470E-15	1.112809192749E-55	2.808208152103E+53
18	2.000000000000E+02	9.0949470E-15	2.272894505073E-63	1.222132247704E+61
20	2.000000000000E+02	2.2737368E-14	3.738313213597E-71	6.687507918240E+68
22	2.000000000000E+02	2.2737368E-14	5.057241770631E-79	4.494005461847E+76
24	2.000000000000E+02	1.8189894E-14	5.726043546978E-87	3.638346916071E+84
26	2.000000000000E+02	1.8189894E-14	5.505811000904E-95	3.492813115359E+92
28	2.000000000000E+02	1.3642421E-14	4.551761667500-103	3.923127763821+100
30	2.000000000000E+02	2.2737368E-14	3.269943681186-111	5.096927672125+108
32	2.000000000000E+02	2.2737368E-14	2.060196346542-119	7.584228481576+116
34	2.000000000000E+02	2.7284841E-14	1.147613816981-127	1.281431259905+125
36	2.000000000000E+02	2.2737368E-14	5.692528799128-136	2.439845145264+133
38	2.000000000000E+02	2.7284841E-14	2.530462637721-144	5.199798023778+141
40	2.000000000000E+02	3.1832315E-14	1.013807138597-152	1.232976118079+150
42	2.000000000000E+02	3.1832315E-14	3.679613572188-161	3.235329359130+158
44	2.000000000000E+02	3.6379788E-14	1.215517160550-169	9.348807782428+166
46	2.000000000000E+02	3.1832315E-14	3.670039714271-178	2.961702324606+175
48	2.000000000000E+02	2.7284841E-14	1.016744152912-186	1.024512074183+184
50	2.000000000000E+02	1.8189894E-14	2.593735070985-195	3.855443858490+192
52	2.000000000000E+02	2.2737368E-14	6.112686319265-204	1.573021102132+201
54	2.000000000000E+02	2.7284841E-14	1.334880829196-212	6.936393884039+209
56	2.000000000000E+02	3.1832315E-14	2.708767905587-221	3.296174387830+218
58	2.000000000000E+02	3.6379788E-14	5.120931441922-230	1.683422190065+227
60	2.000000000000E+02	4.0927262E-14	9.041192486116-239	9.217073209301+235
62	2.000000000000E+02	5.0022209E-14	1.494115622803-247	5.397518090113+244
64	2.000000000000E+02	5.4569682E-14	2.316027434032-256	3.373232916569+253
66	2.000000000000E+02	5.4569682E-14	3.374165832436-265	2.245223836122+262
68	2.000000000000E+02	5.4569682E-14	4.628739331806-274	1.588540773092+271
70	2.000000000000E+02	6.8212103E-14	5.989569513452-283	1.192549332400+280

TABLE II (E)

Wronskian Test for I(x) and K(x)

Argument = 2.0000000+02, Wronskian Value = 5.000000000000E-03

N	W(K(X),I(X))	ABS(REL. ERROR)	I(X)	K(X)
0	5.000000000000E-03	5.5511151E-15	2.039687173410E+85	1.225682053676E-88
20	5.000000000000E-03	1.1102230E-14	7.491067663771E+84	3.320755266225E-88
40	5.000000000000E-03	5.5511151E-15	3.748237630587E+83	6.540293470468E-87
60	5.000000000000E-03	5.5511151E-15	2.630647948131E+81	9.102584071607E-85
80	5.000000000000E-03	5.5511151E-15	2.711154304903E+78	8.561642529290E-82
100	5.000000000000E-03	5.5511151E-15	4.352750449729E+74	5.137138007720E-78
120	5.000000000000E-03	5.5511151E-15	1.165863786098E+70	1.838749232904E-73
140	5.000000000000E-03	2.2204460E-14	5.611324513984E+64	3.649901086122E-68
160	5.000000000000E-03	2.7755576E-14	5.240997906477E+58	3.724805658898E-62
180	5.000000000000E-03	2.2204460E-14	1.026029226836E+52	1.811091851370E-55
200	5.000000000000E-03	0.	4.540059132268E+44	3.093704530876E-48
220	5.000000000000E-03	0.	4.882600242180E+36	3.444230598441E-40
240	5.000000000000E-03	1.6653345E-14	1.367468950317E+28	1.170380768046E-31
260	5.000000000000E-03	1.1102230E-14	1.064490542220E+19	1.431929577702E-22
280	5.000000000000E-03	0.	2.448166312209E+09	5.935439432460E-13
300	5.000000000000E-03	1.6653345E-14	1.761089738435E-01	7.874382834573E-03
320	5.000000000000E-03	2.7755576E-14	4.178691011789E-12	3.170841055949E+08
340	5.000000000000E-03	3.3306691E-14	3.436299635602E-23	3.688705119516E+19
360	5.000000000000E-03	4.4408921E-14	1.025428919531E-34	1.183998919396E+31
380	5.000000000000E-03	4.4408921E-14	1.158937213971E-46	1.004683919794E+43
400	5.000000000000E-03	2.7755576E-14	5.162267204284E-59	2.165780121882E+55
420	5.000000000000E-03	2.2204460E-14	9.404734939821E-72	1.142864317684E+68
440	5.000000000000E-03	5.5511151E-14	7.254161760288E-85	1.426087667931E+81
460	5.000000000000E-03	6.1062266E-14	2.446649542569E-98	4.074204371501E+94
480	5.000000000000E-03	4.4408921E-14	3.718715848379-112	2.585672813845+108
500	5.000000000000E-03	4.9960036E-14	2.620015353282-126	3.543782699550+122
520	5.000000000000E-03	4.4408921E-14	8.785916544272-141	1.021461491770+137
540	5.000000000000E-03	4.4408921E-14	1.437536461361-155	6.040095529966+151
560	5.000000000000E-03	4.9960036E-14	1.174698557638-170	7.157928913317+166
580	5.000000000000E-03	5.5511151E-14	4.900499561367-186	1.663048018530+182
600	5.000000000000E-03	5.5511151E-14	1.065471922489-201	7.419898280631+197

TABLE II (F)

Wronskian Test for I(x) and K(x)

Argument = 6.0000000E+02, Wronskian Value = 1.66666666667E-03

N	W(K(X),I(X))	ABS(REL. ERROR)	I(X)	K(X)
0	1.66666666667E-03	4.1633363E-15	6.146305403939+258	1.355828563158-262
50	1.66666666667E-03	8.3266727E-15	7.648983914180+257	1.085706392829-261
100	1.66666666667E-03	1.6653345E-14	1.495622591663+255	5.496006644940-259
150	1.66666666667E-03	1.2490009E-14	4.793324651775+250	1.686621119392-254
200	1.66666666667E-03	4.1633363E-15	2.694359673627+244	2.934164781167-248
250	1.66666666667E-03	4.1633363E-15	2.904790408028+236	2.648145689243-240
300	1.66666666667E-03	0.	6.682884412500+226	1.115320790384-230
350	1.66666666667E-03	2.0816682E-14	3.697165347932+215	1.946939467188-219
400	1.66666666667E-03	4.1633363E-15	5.588826052356+202	1.240645489145-206
450	1.66666666667E-03	8.3266727E-15	2.634946131482+188	2.530095711822-192
500	1.66666666667E-03	4.5796700E-14	4.428547127266+172	1.445585425736-176
550	1.66666666667E-03	4.1633363E-14	3.029272831987+155	2.027863022241-159
600	1.66666666667E-03	2.0816682E-14	9.599844458885+136	6.138178350561-141
650	1.66666666667E-03	2.0816682E-14	1.597542283628+117	3.538143272522-121
700	1.66666666667E-03	1.6653345E-14	1.574361742644E+96	3.444736151003-100
750	1.66666666667E-03	2.4980018E-14	1.030380219362E+74	5.052301471974E-78
800	1.66666666667E-03	2.0816682E-14	4.993264280272E+50	1.001348862057E-54
850	1.66666666667E-03	2.4980018E-14	1.986000159019E+26	2.419784741795E-30
900	1.66666666667E-03	8.3266727E-15	7.144923836214E+00	6.469629861884E-05
950	1.66666666667E-03	2.0816682E-14	2.548228954630E-26	1.746287853531E+22
1000	1.66666666667E-03	2.0816682E-14	9.821753767667E-54	4.365273708539E+49

APPENDIX D

Control Cards for Using BES on the CDC 3600

APPENDIX D

Control Cards for Using BES on the CDC 3600

BES is maintained on an auxiliary library tape for the convenience of the Control Data 3600 users at Sandia Laboratories, Albuquerque, New Mexico. The tape is labeled 36-00001 and is in HI (556 BPI) density. Questions concerning the availability of BES on the Control Data 3600 at Sandia Laboratories, Livermore, California, should be directed to the Numerical Applications Division 8321.

Two control cards, EQUIP and LIBRARY, are required for using the auxiliary library tape. The EQUIP card may immediately precede the LIBRARY card or may appear at the beginning of the job. The LIBRARY card must precede the first binary deck or first LOAD card (for an execution in which the auxiliary library is needed). If the job includes a compilation, then the LIBRARY card should appear between the SCOPE and LOAD cards.

A complete typical example follows:

```
7
9JOB,...
7
9FTN,L,X
  .
  .
  .
        SCOPE
7
9EQUIP,72=(36-00001),HI,RO
7
9LIBRARY,72
7
9LOAD
7
9RUN
```

All library routines required by a program must be available on a single library tape. Auxiliary library tape 36-00001 contains routines which are likely to be used in connection with the mathematical library routines. In particular, the tape includes the standard Control Data 3600 FORTRAN routines (as modified by Sandia), the SCORS SC4020 plot package, and a few other special Sandia routines (GOFU, ROMBERG, DATE, ANDGEN, and UDGEN).

APPENDIX E

Control Cards for Using BES on the CDC 6600

APPENDIX E

Control Cards for Using BES on the CDC 6600

BES is maintained in a library file for the convenience of the Control Data 6600 users at Sandia Laboratories, Albuquerque, New Mexico. The name of the file is MATHLIB. Questions concerning the availability of BES on the Control Data 6600 at Sandia Laboratories, Livermore, California, should be directed to the Numerical Applications Division 8321.

One control card, COLLECT, is required for using the mathematical library file. The COLLECT processor operates on one relocatable binary file and from one to six library files. The library files are searched for routines which contain entry points matching external references in the relocatable binary file. Such routines are added to the relocatable binary file.

A complete typical example follows:

```
JOB CARD
ACCOUNT CARD
FUN,S.
COLLECT,LGO,MATHLIB.
REDUCE.
LGO.
7/8/9 punch in column 1
Program
7/8/9 punch in column 1
Data
6/7/8/9 punch in column 1
```

In the above example, external references in LGO are satisfied, if possible, by selectively adding routines to LGO from MATHLIB. Additional information on the COLLECT processor with examples is contained in UR0004/6600.[11]

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
G. W. McClure, 5232
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

L. E. Lamkin, Actg., 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
J. C. King, 8310
J. W. Weihe, 8320
J. N. Rogers, 8321
A. G. Schuknecht, 8322
G. W. Anderson, Jr., 8330
J. L. Wirth, 8340
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
J. H. Scott, Actg., 9250
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
M. B. Moore, 9414
J. L. Tischhauser, 9420
A. R. Iacoletti, 9421
D. A. Young, 9422
R. D. Halbgewachs, 9422 (50)
D. K. Robbins, 9424
B. T. Fox, 9427
P. Van Delinder, 9427
For: COSMIC (3)
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)
J. L. Gardner, 3421
C. H. Sproul, 3428-1 (65)
B. F. Hefley, 8232 (5)