12 MAY REC'D

4000

NRL Report 7997

AB

FØRTRAN Subroutines to Evaluate, in Single or Double Precision, Bessel Functions of the First and Second Kinds for Complex Arguments

8100

LAWRENCE FLAX

Acoustics Division

and

JANET P. MASON

Research Computation Center

April 27, 1976

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release: distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
NRL Report 7997

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
FØRTRAN SUBROUTINES TO EVALUATE, IN SINGLE OR DOUBLE PRECISION, BESSEL FUNCTIONS OF THE FIRST AND SECOND KINDS FOR COMPLEX ARGUMENTS

5. TYPE OF REPORT & PERIOD COVERED
A final report on one phase of the NRL problem.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Lawrence Flax and Janet P. Mason

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Naval Research Laboratory
Washington, D.C. 20375

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
NRL probelm S01-04
Project RR 131-03-41

11. CONTROLLING OFFICE NAME AND ADDRESS
Department of the Navy
Office of Naval Research
Arlington, Va. 22217

12. REPORT DATE
April 27, 1976

13. NUMBER OF PAGES
29

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Complex arguments
Bessel functions
Computer programs

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Subroutines have been written in FØRTRAN for the CDC 3600/3800 to evaluate Bessel functions of the first and second kinds for complex arguments. These routines will compute $J_n(x + iy)$ and $Y_n(x + iy)$, where $x \gtrless 0$, $y \gtrless 0$, and $i = \sqrt{-1}$. Using the identity

$$2/[\pi(x + iy)] = J_{n+1}(x + iy)\, Y_n(x + iy) - J_n(x + iy)\, Y_{n+1}(x + iy)$$

as a check, the single-precision version will generate results that are accurate to eight figures or more

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. Continued

for arguments equal to (±20.0 ±20.0i) and the double-precision version will generate results that are accurate for seven figures or more for arguments equal to (±100.0 ± 20.0i).

CONTENTS

1.0 IDENTIFICATION 1

2.0 PURPOSE 2

3.0 USAGE 2

4.0 METHOD OR ALGORITHM 11

5.0 SOURCE LANGUAGE LISTING 13

6.0 COMPARISON 23

7.0 TEST METHOD AND RESULTS 23

FØRTRAN SUBROUTINES TO EVALUATE, IN SINGLE OR DOUBLE PRECISION, BESSEL FUNCTIONS OF THE FIRST AND SECOND KINDS FOR COMPLEX ARGUMENTS

1.0 IDENTIFICATION

1.1 Title
BESSEL FUNCTIONS OF THE 1ST AND 2ND KINDS, CMPLX. ARG.

1.2 Identification Name
C3-NRL-CMPJYBES (single precision),
C3-NRL-CMPJYBED (double precision).

1.3 Classification Code
C3-Polynomials and Special Functions, Evaluation of Special Functions

1.4 Entry Points
CMPJYBES (single precision),
CMPJYBED (double precision).

1.5 Programming Language
Language: 3600/3800 FØRTRAN.
Routine Type: Subroutines.
Operating System: DRUM SCØPE 2.1.

1.6 Computer and Configuration
CDC 3800

1.7 Contributor or Programmer
Lawrence Flax, Code 8132, Physical Acoustics Branch, Acoustics Division, and
Janet P. Mason, Code 4222.5, Software Systems and Support Branch, Research Computation Center.

1.8 Contributing Organization
NRL—Naval Research Laboratory
Washington, D.C. 20375

1.9 Program Availability
1.9.1 Submittal: Subroutine writeup, FØRTRAN source decks, source listings.
1.9.2 On File: RCC Program Library.

Note: Manuscript submitted March 17, 1976.

1.10 Verification

The subroutines were run using the following values for the argument:

x	100	100	100	100	±3	±6	0.4	4	7.5	50	20
y	22	24	26	28	±7	±2	0.3	4	5.5	4	20

The following identity was used to check the answers:

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

In addition comparisons were made with output generated by routines written elsewhere; see Sections 3.15 and 6.0 See Ref. 1.

1.11 Date

1 January 1976

2.0 PURPOSE

2.1 Description of the Routine

Subroutines have been written in FØRTRAN for the CDC 3600/3800 to evaluate Bessel functions of the first and second kinds for complex arguments. These routines will compute $J_n(x + iy)$ and $Y_n(x + iy)$, where $x \gtrless 0$, $y \gtrless 0$, and $i = \sqrt{-1}$.

2.2 Problem Background

The theory of acoustic scattering from cylinders or spheres involves Bessel functions. When absorption is included, the imaginary parts of the arguments of these functions become nonzero. See Ref. 2.

3.0 USAGE

3.1 Calling Sequence or Operational Procedure

For single precision:

```
CALL CMPJYBES (X, Y, N, XJR, XJI, XYR, XYI)
```

and for double precision:

```
CALL CMPJYBED (X, Y, N, XJR, XJI, XYR, XYI).
```

3.2 Arguments, Parameters, and/or Initial Conditions

X is the real part of the argument.
Y is the imaginary part of the argument.
N is the maximum order. TYPE INTEGER.

XJR is the name of a table that will contain the real part of $J_0(x + iy)$ to $J_n(x + iy)$.

XJI is the name of a table that will contain the imaginary part of $J_0(x + iy)$ to $J_n(X + iy)$.

XYR is the name of a table that will contain the real part of $Y_0(x + iy)$ to $Y_n(x + iy)$.

XYI is the name of a table that will contain the imaginary part of $Y_0(x + iy)$ to $Y_n(x + iy)$.

All arguments, with the exception of N, should be typed REAL for single-precision and DØUBLE for double-precision. The dimension for each of the arrays should be at least equal to the order plus 50 or to the absolute value of the argument plus 50, whichever is the larger.

3.3 Space Required (Decimal and Octal)

3.3.1 Unique Storage

		octal	(decimal) locations	
CMPJYBES	-	1253	(683)	single precision
LRGBES	-	324	(212)	
RINT	-	121	(81)	
GINT	-	124	(84)	
ØNER	-	113	(75)	
ØNEI	-	121	(81)	
TWØR	-	113	(75)	
TWØI	-	122	(82)	
CMPJYBED	-	1433	(795)	double precision
LRGBESD	-	355	(237)	
DRINT	-	144	(100)	
DGINT	-	146	(102)	
DØNER	-	130	(88)	
DØNEI	-	130	(88)	
DTWØR	-	130	(88)	
DTWØI	-	131	(89)	

3.3.2 Common Blocks

CØMMØN/LF/EN, EX, WY (EN, EX, and WY are REAL) — in LRGBES, RINT, GINT ØNER, ØNEI, TWØR, and TWØI (single precision).

CØMMØN/LF/EN, EX, WY (EN, EX, and WY are DØUBLE) — in LRGBESD, DRINT, DGINT, DØNER, DØNEI, DTWØR, and DTWØI (double precision).

3.3.3. Temporary Storage

None.

3.4 Messages and Instructions to the Operator

None.

3.5 Error Returns, Messages, and Codes
None.

3.6 Informative Messages to the User
None.

3.7 Input
See Section 3.1.

3.8 Output
See Section 3.1.

3.9 Formats
Not applicable.

3.10 External Routines and Symbols
Single-precision subroutines are

CØMPBES* MLT DVD LRGBES SQRTF	called by CMPJYBES
TWØI TWØR ØNEI ØNER GINT RINT SIMCØN MLT	called by LRGBES
SINF EXPF CØSF	called by RINT and GINT
SORTF EXPF CØSF	called by ØNER and TWØR
SQRTF SINF EXPF	called by ØNEI and TWØI.

*C3-NRL-CØMPBES comprises subroutines CØMPBES, MLT, DVD, SRT, PSUBV, and QSUBV.

Double-precision subroutines are

CØMPBESD*	called by CMPJYBED
DMLT	
DDVD	
LRGBESD	
DSQRT	
DSIMCØN	called by LRGBESD
DTWØI	
DTWØR	
DØNEI	
DØNER	
DGINT	
DRINT	
DMLT	
DEXP	called by DRINT and DGINT
DCØS	
DSIN	
DSQRT	called by DØNER and DTWØR
DEXP	
DCØS	
DSQRT	called by DØNEI and DTWØI.
DEXP	
DSIN	

C3-NRL-CØMPBES and C3-NRL-CØMPBESD are the single- and double-precision subroutines that evaluate Bessel functions of the first and second kinds for orders zero and one, for complex arguments. See Ref. 3. The following changes have been made to the routines since Ref. 3 was issued:

C3-NRL-CØMPBES

remove lines 0002000 and 0002100
replace lines 0003700 and 0003800 by

```
      PHI=ATAN2(Y, X)                                   0003700
```

replace lines 0011800 to 0012400 by

```
C       PUT INTØ FIRST QUADRANT                         0011800
    100 IF(X.GE.0.0.AND.Y.GE.0.0)NLAG=1                 0011900
        IF(X.LT.0.0.AND.Y.GE.0.0)NLAG=2                 0012000
        IF(X.LT.0.0.AND.Y.LT.0.0)NLAG=3                 0012100
        IF(X.GE.0.0.AND.Y.LT.0.0)NLAG=4                 0012200
```

*C3-NRL-CØMPBESD comprises subroutines CØMPBESD, DMLT, DDVD, DSRT, PSUBVD, AND QSUBVD.

```
      X=ABS(X)                                          0012300
      Y=ABS(Y)                                          0012400
```

insert

```
      IF(X.NE.0.0)GO TO 13                              0018410
      J0I=J1I=0.0                                       0018420
```

replace line 0018500 by

```
   13 CALL MLT (CCR, CCI, DDR, DDI, FRR, FRI)           0018500
```

replace lines 0021100 to 0022100 by

```
C     PUT BACK INTØ SECØND, THIRD, ØR
        FØURTH QUADRANT                                 0021100
      IF(NLAG.LT.4)GØ TØ 20                             0021125
      Y=-Y                                              0021150
      Y0I=-Y0I                                          0021175
      Y1I=-Y1I                                          0021200
      J0I=-J0I                                          0021225
      J1I=-J1I                                          0021250
      RETURN                                            0021275
   20 IF(NLAG.LT.3)GØ TØ 21                             0021300
      X=-X                                              0021325
      Y=-Y                                              0021350
      Y0R=Y0R+J0I+J0I                                   0021375
      Y0I=Y0I-J0R-J0R                                   0021400
      Y1R=-Y1R-J1I-J1I                                  0021425
      Y1I=-Y1I+J1R+J1R                                  0021450
      J1R=-J1R                                          0021475
      J1I=-J1I                                          0021500
      RETURN                                            0021525
   21 IF(NLAG.LT.2)RETURN                               0021550
      X=-X                                              0021575
      Y0R=Y0R+J0I+J0I                                   0021600
      Y0I=-Y0I+J0R+J0R                                  0021650
      Y1R=-Y1R-J1I-J1I                                  0021700
      Y1I=Y1I-J1R-J1R                                   0021800
      J0I=-J0I                                          0021900
      J1R=-J1R                                          0022000
      END                                               0022100

      SUBRØUTINE MLT (XA, YA, XB, YB, XC, YC)           0022200
      XX=XA*XB-YA*YB                                    0022300
      YC=XA*YB+YA*XB                                    0022410
      XC=XX                                             0022410
      END                                               0022500
```

```
      SUBRØUTINE DVD(XA, YA, XB, YB, XC, YC)          0022600
      DENØM=XB*XB+YB*YB                               0022700
      XX=(XA*XB+YA*YB)/DENØM                          0022800
      YC=(YA*XB-XA*YB)/DENØM                          0022900
      XC=XX                                           0022910
      END                                             0023000

      SUBRØUTINE SRT (XA, YA, XC, YC)                 0023100
      SS=SQRT(XA*XA+YA*YA)                            0023200
      XX=SQRT(ABS((SS+XA)/2.0))                       0023300
      IF(XA)1, 2, 1                                   0023400
    2 YC=-XX                                          0023500
      XC=XX                                           0023510
      RETURN                                          0023600
    1 YC=YA/(2.0*XX)                                  0023700
      XC=XX                                           0023710
      END                                             0023800
```

C3-NRL-CØMPBESD

replace lines 0004500 and 0004600 by

```
      PHI=DATAN2(Y, X)                                0004500
```

replace lines 0012800 to 0013400 by

```
C     PUT INTO FIRST QUADRANT                         0012800
  100 IF(X.GE.0.0D.AND.Y.GE.0.0D)NLAG=1               0012900
      IF(X.LT.0.0D.AND.Y.GE.0.0D)NLAG=2               0013000
      IF(X.LT.0.0D.AND.Y.LT.0.0D)NLAG=3               0013100
      IF(X.GE.0.0D.AND.Y.LT.0.0D)NLAG=4               0013200
      X=DABS(X)                                       0013300
      Y=DABS(Y)                                       0013400
```

replace lines 0022100 to 0023100 by

```
C     PUT BACK INTØ SECØND, THIRD, ØR FØURTH
        QUADRANT                                      0022100
      IF(NLAG.LT.4)GØ TØ 20                           0022125
      Y=-Y                                            0022150
      Y0I=-Y0I                                        0022175
      Y1I=-Y1I                                        0022200
      J0I=-J0I                                        0022225
      J1I=-J1I                                        0022250
      RETURN                                          0022275
   20 IF(NLAG.LT.3)GØ TØ 21                           0022300
      X=-X                                            0022325
      Y=-Y                                            0022350
      Y0R=Y0R+J0I+J0I                                 0022375
      Y0I=Y0I-J0R-J0R                                 0022400
```

```
      Y1R=-Y1R-J1I-J1I                                  0022425
      Y1I=-Y1I+J1R+J1R                                  0022450
      J1R=-J1R                                          0022475
      J1I=-J1I                                          0022500
      RETURN                                            0022525
21    IF(NLAG.LT.2)RETURN                               0022550
      X=-X                                              0022575
      Y0R=Y0R+J0I+J0I                                   0022600
      Y0I=-Y0I+J0R+J0R                                  0022650
      Y1R=-Y1R-J1I-J1I                                  0022700
      Y1I=Y1I-J1R-J1R                                   0022800
      J0I=-J0I                                          0022900
      J1R=-J1R                                          0023000
      END                                               0023100

      SUBRØUTINE DMLT (XA, YA, XB, YB, XC, YC)          0023200
      TYPE DØUBLE XA, YA, XB, YB, XC, YC, XX            0023300
      XX=XA*XB-YA*YB                                    0023400
      YC=XA*YB+YA*XB                                    0023500
      XC=XX                                             0023510
      END                                               0023600

      SUBRØUTINE DDVD(XA, YA, XB, YB, XC, YC)           0023700
      TYPE DØUBLE XA, YA, XB, YB, XC, YC, DENØM,
        XX                                              0023800
      DENØM=XB*XB+YB*YB                                 0023900
      XX=(XA*XB+YA*YB)/DENØM                            0024000
      YC=(YA*YB-XA*YB)/DENØM                            0024100
      XC=XX                                             0024110
      END                                               0024200

      SUBRØUTINE DSRT (XA, YA, XC, YC)                  0024300
      TYPE DØUBLE XA, YA, XC, YC, SS, XX                0024400
      SS=DSQRT(XA*XA+YA*YA)                             0024500
      XX=DSQRT(DABS((SS+XA)/2.0D))                      0024600
      IF(XA)1, 2, 1                                     0024700
2     YC=-XX                                            0024800
      XC=XX                                             0024810
      RETURN                                            0024900
1     YC=YA/(2.0D*XX)                                   0025000
      XC=XX                                             0025010
      END                                               0025100
```

SIMCØN and DSIMCØN are FØCUS library subroutines that perform numerical integration. Routines from other libraries can be substituted.

3.11 Timing

The single- and double-precision test runs shown in Section 7.0 took 5 seconds and 9 seconds respectively, exclusive of compilation time.

3.12 Accuracy

The values for $J_n(x + iy)$ and $Y_n(x + iy)$ for n = 0 or 1 and $\rho \leqslant 10$ could be checked against the National Bureau of Standards tables, Refs 4 and 5. However, for $n > 1$ or $\rho > 10$, the identity given in Section 1.11 has been used as a method of verification. As shown in the following table, up to a value of Z = 100 + 20i

$$J_{n+1}(x + iy)\, Y_n(x + iy) - J_n(x + iy)\, Y_{n+1}(x + iy)$$

agreed with $2/[\pi(x + iy)]$ to at least seven figures, using CMPJYBED; they agreed to two figures when Z = 100 + 26i.

Z Value	Agreement Using CMPJYBED* of $J_{n+1} Y_n - J_n Y_{n+1}$ vs $2/\pi(x + iy)$	Agreement Using CMPJYBES vs CMPJYBED
0.4 + 0.3i	11 figures*	10 figures
4.0 + 4.0i	9 figures	8 figures
7.5 + 5.5i	10 figures	9 figures
50.0 + 4.0i	11 figures*	10 figures
20 + 20i	8 figures	9 figures
40 + 20i	7 figures	—
60 + 20i	7 figures	—
80 + 20i	7 figures	—
100 + 20i	7 figures	—
100 + 26i	2 figures	—
±10 + 0i	10 figures	9 figures
0 ± 10i	11 figures*	9 figures

*E17.10 format is used, so agreement is probably greater.

3.13 Cautions to Users

None.

3.14 Program Deck Structure

7
9 JØB card

7
9 FTN card

main program

subroutines

CMPJYBES
LRGBES
RINT
GINT
ØNER
ØNEI
TWØR
TWØI

} -C3 NRL CMPJYBES

	SIMCØN	-FØCUS (the Control Data Corporation Users group) library routine D1 UCSD SIMCØN
	CØMPBES MLT DVD SRT PSUBV QSUBV	-C3 NRL CØMPBES

SCØPE

7
9 LØAD

7
9 RUN card

end-of-file card

7
9 JØB card

7
9 FTN card

main program subroutines	CMPJYBED LRGBESD DRINT DGINT DØNER DØNEI DTWØR DTWØI	-C3 NRL CMPJYBED
	DSIMCØN	-double-precision modification of FØCUS (the CDC Users group) library routine D1 UCSD SIMCØN
	CØMPBESD DMLT DDVD DSRT PSUBVD QSUBVD	-C3 NRL CØMPBESD

SCØPE

7
9 LØAD

7
9 RUN card

end-of-file card

3.15 References—Literature

1. M. Abramowitz and I.A. Stegun, editors, Handbook of Mathematical Functions, National Bureau of Standards Applied Mathematics Series 55, Government Printing Office, 1965.
2. R.H. Vogt, L. Flax, L.R. Dragonette, and W.G. Neubauer, "Monostatic Reflection of a Plane Wave from an Absorbing Sphere," J. Acoust. Soc. Am. 57, 558 (1975).
3. J.P. Mason, "FØRTRAN Subroutines to Evaluate, in Single or Double Precision, Bessel Functions of the First and Second Kinds for Orders Zero and One and Complex Arguments," NRL Memorandum Report 2614 (NRL Computer Bulletin 33), June 1973.
4. Table of the Bessel Functions $J_0(z)$ and $J_1(z)$ for Complex Arguments, Mathematical Tables Project, National Bureau of Standards, Columbia University Press, second edition, 1947.
5. Table of the Bessel Functions $Y_0(z)$ and $Y_1(z)$ for Complex Arguments, Computation Laboratory, National Bureau of Standards, Columbia University Press, 1950.
6. L.S. Schuetz, and L. Flax, "Acoustic Reflection from Absorbing Elastic Cylinders," to be published.

4.0 METHOD OR ALGORITHM

A. If the modulus is less than 0.5:

(1) Call CØMPBES [Ref. 3] to compute

$J_0(Z)$, $J_1(Z)$, $Y_0(Z)$, and $Y_1(Z)$.

(2) Compute $J_2(Z)$ to $J_N(Z)$ by

$$J_n(Z) = \left(\frac{Z}{2}\right)^n \sum_{k=0}^{\infty} \frac{(-Z^2/4)^k}{k!\,\Gamma(n+k+1)} .$$

(3) Use forward recursion for $Y_2(Z)$ to $Y_N(Z)$.

B. If the modulus is $\geqslant 0.5$ and $|y|$ is less than or equal to 5.0 (in subroutine CMPJYBES, or 10.0 in subroutine CMPJYBED) or if in the single-precision routine $x = 0$ and $|y| \leqslant 18.0$:

(1) Call CØMPBES [Ref. 3] to compute $J_0(Z)$, $J_1(Z)$, $Y_0(Z)$, and $Y_1(Z)$.

(2) Compute $J_2(Z)$ to $J_N(Z)$ by backward recursion.

(3) Compute $Y_2(Z)$ to $Y_N(Z)$ by forward recursion.

C. If the modulus is $\geqslant 0.5$ and $|y|$ is greater than 5.0 (in subroutine CMPJYBES, or 10.0 in subroutine CMPJYBED) (except that in the single-precision routine, if $x = 0$ and $|y| \leqslant 18.0$, method B is used):

(1) Compute $J_0(Z)$ and $J_1(Z)$ by

$$J_n(Z) = \frac{1}{\pi} \int_0^{\pi} \cos(Z \sin\theta - n\theta)\, d\theta .$$

(2) Compute $J_2(Z)$ to $J_N(Z)$ by backward recursion.

D. If the modulus $\geqslant 0.5$, $|y| > 5.0$ (in subroutine CMPJYBES, or 10.0 in subroutine CMPJYBED), and y is positive:

(1) Compute $Y_0(Z)$ and $Y_1(Z)$ by

$$Y_n(Z) = iJ_n(Z) - iH_n^{(1)}(Z) ,$$

where

$$H_n^{(1)}(Z) = \frac{-2ie^{-in\pi}}{\sqrt{\pi}} \frac{(Z/2)^n}{\Gamma(1/2 + n)} \int_0^{\infty} \frac{t^{2n}\, e^{iz\sqrt{t^2+1}}}{\sqrt{t^2 + 1}}\, dt .$$

E. If the modulus > 0.5, $|y| > 5.0$ (in subroutine CMPJYBES, or 10.0 in subroutine CMPJYBED) and y is negative:

(1) Compute $Y_0(Z)$ and $Y_1(Z)$ by

$$Y_n(Z) = -iJ_n(Z) + iH_n^{(2)}(Z) ,$$

where

$$H_n^{(2)}(Z) = \frac{2i(-1)^n}{\sqrt{\pi}} \frac{(Z/2)^n}{\Gamma(1/2 + n)} \int_0^{\infty} \frac{t^{2n}\, e^{-iz\sqrt{t^2+1}}}{\sqrt{t^2 + 1}}\, dt .$$

5.0 SOURCE LANGUAGE LISTING

```
      SUBROUTINE CMPJYBES(AR,AI,N,XJR,XJI,XYR,XYI)
C     TITLE - BESSEL FTNS OF THE 1ST AND 2ND KINDS, CMPLX. ARG.
C     IDENT NAME - C3-NRL-CMPJYBES
C     LANGUAGE - FORTRAN
C     COMPUTER - CDC-3800
C     CONTRIBUTOR - LAWRENCE FLAX, CODE 8132,PHYSICAL ACOUSTICS
C                                  BRANCH, ACOUSTICS DIVISION
C                   JANET P. MASON, CODE 4222.5, SYSTEM SOFTWARE AND
C                                   SUPPORT BRANCH, RESEARCH
C                                   COMPUTATION CENTER
C     ORGANIZATION - NRL - NAVAL RESEARCH LABORATORY
C                          WASHINGTON, D.C. 20375
C     DATE - 1 JANUARY 1976
C     PURPOSE - TO EVALUATE, IN SINGLE PRECISION, BESSEL FUNCTIONS
C               OF THE FIRST AND SECOND KINDS FOR COMPLEX ARGUMENTS AND
C               INTEGER ORDERS.
      DIMENSION XJR(1),XJI(1),XYR(1),XYI(1)
      CC=2.0
      M=N+1
      KLUE=0
      SRARG=SQRT(AR*AR+AI*AI)
      IF(SRARG.GE.0.5)GO TO 21
      CALL COMPBES(AR,AI,XJR(1),XJI(1),XJR(2),XJI(2),XYR(1),XYI(1),
     1             XYR(2),XYI(2))
      EPS=5.0E-11
      CSR=1.0
      CSI=0.0
      DFAC=1.0
      NN=N+1
      DO 12 L=1,NN
      NO=L-1
      IF(L.EQ.1)GO TO 7
      CALL MLT(CSR,CSI,AR,AI,CSR,CSI)
      CSR=0.5*CSR
      CSI=0.5*CSI
C     CHECK FOR UNDERFLOW
      IF(CSR.NE.0.0.OR. CSI.NE.0.0)GO TO 14
      ION=1
      PRINT 6,N,AR,AI,ION
   14 DFAC=DFAC*NO
C     CHECK FOR OVERFLOW
      IF EXPONENT FAULT 9,7
    9 ION=2
      PRINT 6,N,AR,AI,ION
    7 XKK=0.0
      AAR=FORTTR=1.0/DFAC
      FORTTI=0.0
      AAI=0.0
      VK=NO
      FNR=-0.25*(AR*AR-AI*AI)
      FNI=-0.5*(AR*AI)
   10 VK=VK+1.0
      XKK=XKK+1.0
      DENOM=VK*XKK
      CALL MLT(AAR,AAI,FNR,FNI,AAR,AAI)
      AAR=AAR/DENOM
      AAI=AAI/DENOM
C     CHECK FOR UNDERFLOW
      IF(AAR.NE.0.0.OR. AAI.NE.0.0)GO TO 8
      ION=3
      PRINT 6,N,AR,AI,ION
```

```
    8 TTR=AAR+FORTTR
      TTI=AAI+FORTTI
      IF(TTR.EQ.0.0.AND.TTI.EQ.0.0)GO TO 11
      CALL DVD(AAR,AAI,TTR,TTI,SAR,SAI)
      IF(ABS(SAR).LT.EPS.AND.ABS(SAI).LT.EPS)GO TO 11
      FORTTR=TTR
      FORTTI=TTI
      GO TO 10
   11 IF(L.LE.2)GO TO 12
      XJR(L)=CSR*TTR-CSI*TTI
      XJI(L)=CSI*TTR+CSR*TTI
C     CHECK FOR UNDERFLOW
      IF(XJR(L).NE.0.0.OR. XJI(L).NE.0.0)GO TO 12
      ION=4
      PRINT 6,N,AR,AI,ION
   12 CONTINUE
      GO TO 4
    6 FORMAT (/,2X,I5,2(2X,E17.10),2X,I5,/)
   21 IF(AR.EQ.0.0.AND.ABS(AI).LE.18.0.OR.AR.NE.0.0.AND.ABS(AI).LE.5.0)
     1  GO TO 16
C     FIND J AND Y VALUES FOR ORDERS ZERO AND ONE
      KLUE=1
      DO 17 J=1,2
      NORD=J-1
      CALL LRGBES(AR,AI,NORD,XJR(J),XJI(J),XYR(J),XYI(J))
   17 CONTINUE
      IF(AR.NE.0.0)GO TO 16
      XJI(1)=XJR(2)=0.0
   16 NN=SRARG+50
      IF((N+50).GT.NN)NN=N+50
      XJR(NN)=XJI(NN)=0.0
      XJR(NN-1)=XJI(NN-1)=1.0E-10
      IF(KLUE.EQ.1)GO TO 5
      CALL COMPBES(AR,AI,XJR(1),XJI(1),XJR(2),XJI(2),XYR(1),XYI(1),
     1             XYR(2),XYI(2))
    5 GDR=XJR(2)
      GDI=XJI(2)
      NM=NN-2
      DO 1 K=2,NM
      KK=NN-K
      CALL DVD(XJR(KK+1),XJI(KK+1),AR,AI,XJR(KK),XJI(KK))
      XJR(KK)=CC*KK*XJR(KK)-XJR(KK+2)
      XJI(KK)=CC*KK*XJI(KK)-XJI(KK+2)
    1 CONTINUE
      CALL DVD(GDR,GDI,XJR(2),XJI(2),RAR,RAI)
      DO 2 K=3,M
      TR=XJR(K)
      TI=XJI(K)
    2 CALL MLT(TR,TI,RAR,RAI,XJR(K),XJI(K))
   13 XJR(2)=GDR
      XJI(2)=GDI
    4 DO 3 K=3,M
      CALL DVD(XYR(K-1),XYI(K-1),AR,AI,XYR(K),XYI(K))
      XYR(K)=CC*(K-2)*XYR(K)-XYR(K-2)
    3 XYI(K)=CC*(K-2)*XYI(K)-XYI(K-2)
      END
```

```
      SUBROUTINE LRGBES(X,Y,N,A,B,C,D)
      EXTERNAL RINT,GINT,ONER,ONEI,TWOR,TWOI
      COMMON/LF/EN,EX,WY
      TEST=1.0E-8
      EX=X
      WY=Y
      EN=N
      XI=0.0
      XD=20.0
      LIM=16
      PI=3.1415926536
C     INTEGRATE FOR J VALUE
      CALL SIMCON(XI,PI,TEST,LIM,AR,NO,RD,RINT)
      A=AR/PI
      CALL SIMCON(XI,PI,TEST,LIM,AV,NI,RD,GINT)
      B=AV/PI
      IF(WY.LE.0)GO TO 5
C     INTEGRATE FOR H SUPER ONE VALUE
      CALL SIMCON(XI,XD,TEST,LIM,B1,NA,RD,ONER)
      CALL SIMCON(XI,XD,TEST,LIM,B2,NA,RD,ONEI)
      IF(N.EQ.1)GO TO 7
      TR=0.0
      TI=-2.0/PI
      GO TO 8
    7 TR=-2.0*WY/PI
      TI=2.0*EX/PI
    8 CALL MLT(TR,TI,B1,B2,HR,HI)
      C=-B+HI
      D=A-HR
      RETURN
C     INTEGRATE FOR H SUPER TWO VALUE
    5 CALL SIMCON(XI,XD,TEST,LIM,B3,NA,RD,TWOR)
      CALL SIMCON(XI,XD,TEST,LIM,B4,NA,RD,TWOI)
      IF(N.EQ.1)GO TO 10
      TR=0.0
      TI=2.0/PI
      GO TO 11
   10 TR=2.0*WY/PI
      TI=-2.0*EX/PI
   11 CALL MLT(TR,TI,B3,B4,HR,HI)
      C=B-HI
      D=-A+HR
      END
```

```
      FUNCTION RINT(T)
      COMMON/LF/EN,EX,WY
      SINT=SIN(T)
      ALPH=EX*SINT
      BETA=WY*SINT
      COSHB=(EXP(BETA)+EXP(-BETA))/2.0
      RINT=COS(EN*T)*COS(ALPH)*COSHB+SIN(ALPH)*COSHB*SIN(EN*T)
      END
```

```
      FUNCTION GINT(T)
      COMMON/LF/EN,EX,WY
      SINT=SIN(T)
      ALPH=EX*SINT
      BETA=WY*SINT
      SINHB=(EXP(BETA)-EXP(-BETA))/2.0
      GINT=COS(ALPH)*SINHB*SIN(EN*T)-SIN(ALPH)*SINHB*COS(EN*T)
      END
```

```
      FUNCTION OMER(T)
      COMMON/LF/EN,EX,WY
      ALPH=SQRT(T*T+1.0)
      TTN=T*T
      IF(EN.EQ.0.0)TTN=1.0
    2 OMER=TTN*EXP(-ALPH*WY)*COS(ALPH*EX)/ALPH
      END
```

```
      FUNCTION ONEI(T)
      COMMON/LF/EN,EX,WY
      ALPH=SQRT(T*T+1.0)
      TTN=T*T
      IF(EN.EQ.0.0)TTN=1.0
      IF(EX.NE.0.0)GO TO 2
      ONEI=0.0
      RETURN
    2 ONEI=TTN*EXP(-ALPH*WY)*SIN(ALPH*EX)/ALPH
      END
```

```
      FUNCTION TWOR(T)
      COMMON/LF/EN,EX,WY
      ALPH=SQRT(T*T+1.0)
      TTN=T*T
      IF(EN.EQ.0.0)TTN=1.0
    2 TWOR=TTN*EXP(ALPH*WY)*COS(ALPH*EX)/ALPH
      END
```

```
      FUNCTION TWOI(T)
      COMMON/LF/EN,EX,WY
      ALPH=SQRT(T*T+1.0)
      TTN=T*T
      IF(EN.EQ.0.0)TTN=1.0
      IF(EX.NE.0.0)GO TO 2
      TWOI=0.0
      RETURN
    2 TWOI=-TTN*EXP(ALPH*WY)*SIN(ALPH*EX)/ALPH
      END
```

```
      SUBROUTINE CMPJYBED(AR,AI,N,XJR,XJI,XYR,XYI)
C     TITLE - BESSEL FTNS OF THE 1ST AND 2ND KINDS, CMPLX. ARG.
C     IDENT NAME - C3-NRL-CMPJYBED
C     LANGUAGE - FORTRAN
C     COMPUTER - CDC-3800
C     CONTRIBUTOR - LAWRENCE FLAX, CODE 8132,PHYSICAL ACOUSTICS
C                                  BRANCH, ACOUSTICS DIVISION.
C                   JANET P. MASON, CODE 4222.5, SYSTEM SOFTWARE AND
C                                   SUPPORT BRANCH, RESEARCH
C                                   COMPUTATION CENTER
C     ORGANIZATION - NRL - NAVAL RESEARCH LABORATORY
C                          WASHINGTON, D.C. 20375
C     DATE - 1 JANUARY 1976
C     PURPOSE - TO EVALUATE, IN DOUBLE PRECISION, BESSEL FUNCTIONS
C               OF THE FIRST AND SECOND KINDS FOR COMPLEX ARGUMENTS AND
C               INTEGER ORDERS.
      DIMENSION XJR(1),XJI(1),XYR(1),XYI(1)
      TYPE DOUBLE AR,AI,XJR,XJI,XYR,XYI,CC,SRARG,CSR,CSI,DFAC,XKK,AAR,
     1            AAI,FORTTR,FORTTI,VK,FNR,FNI,DENOM,TTR,TTI,SAR,SAI,
     2            EPS,GDR,GDI,KD,RAR,RAI,TR,TI
      CC=2.0D
      M=N+1
      KLUE=0
      SRARG=DSQRT(AR*AR+AI*AI)
      IF(SRARG.GE.0.5D)GO TO 21
      CALL COMPBESD(AR,AI,XJR(1),XJI(1),XJR(2),XJI(2),XYR(1),XYI(1),
     1              XYR(2),XYI(2))
      EPS=5.0D-26
      CSR=1.0D
      CSI=0.0D
      DFAC=1.0D
      NN=N+1
      DO 12 L=1,NN
      NO=L-1
      IF(L.EQ.1)GO TO 7
      CALL DMLT(CSR,CSI,AR,AI,CSR,CSI)
      CSR=0.5D*CSR
      CSI=0.5D*CSI
C     CHECK FOR UNDERFLOW
      IF(CSR.NE.0.0D.OR.CSI.NE.0.0D)GO TO 14
      ION=1
      PRINT 6,N,AR,AI,ION
   14 DFAC=DFAC*NO
C     CHECK FOR OVERFLOW
      IF EXPONENT FAULT 9,7
    9 ION=2
      PRINT 6,N,AR,AI,ION
    7 XKK=0.0D
      AAR=FORTTR=1.0D/DFAC
      FORTTI=0.0D
      AAI=0.0D
      VK=NO
      FNR=-0.25D*(AR*AR-AI*AI)
      FNI=-0.5D*(AR*AI)
   10 VK=VK+1.0D
      XKK=XKK+1.0D
      DENOM=VK*XKK
      CALL DMLT(AAR,AAI,FNR,FNI,AAR,AAI)
      AAR=AAR/DENOM
      AAI=AAI/DENOM
C     CHECK FOR UNDERFLOW
      IF(AAR.NE.0.0D.OR.AAI.NE.0.0D)GO TO 8
      ION=3
      PRINT 6,N,AR,AI,ION
```

```
    8 TTR=AAR+FORTTR
      TTI=AAI+FORTTI
      IF(TTR.EQ.0.0D.AND.TTI.EQ.0.0D)GO TO 11
      CALL DDVD(AAR,AAI,TTR,TTI,SAR,SAI)
      IF(DABS(SAR).LT.EPS.AND.DABS(SAI).LT.EPS)GO TO 11
      FORTTR=TTR
      FORTTI=TTI
      GO TO 10
   11 IF(L.LE.2)GO TO 12
      XJR(L)=CSR*TTR-CSI*TTI
      XJI(L)=CSI*TTR+CSR*TTI
C     CHECK FOR UNDERFLOW
      IF(XJR(L).NE.0.0D.OR.XJI(L).NE.0.0D)GO TO 12
      ION=4
      PRINT 6,N,AR,AI,ION
   12 CONTINUE
      GO TO 4
    6 FORMAT (/,2X,I5,2(2X,E17.10),2X,I5,/)
   21 IF(DABS(AI).LE.10.0D)GO TO 16
C     FIND J AND Y VALUES FOR ORDERS ZERO AND ONE
      KLUE=1
      DO 17 J=1,2
      NORD=J-1
      CALL LRGBESD(AR,AI,NORD,XJR(J),XJI(J),XYR(J),XYI(J))
   17 CONTINUE
   16 NN=SRARG+50
      IF((N+50).GT.NN)NN=N+50
      XJR(NN)=XJI(NN)=0.0D
      XJR(NN-1)=XJI(NN-1)=1.0D-10
      IF(KLUE.EQ.1)GO TO 5
      CALL COMPBESD(AR,AI,XJR(1),XJI(1),XJR(2),XJI(2),XYR(1),XYI(1),
     1              XYR(2),XYI(2))
    5 GDR=XJR(2)
      GDI=XJI(2)
      NM=NN-2
      DO 1 K=2,NM
      KK=NN-K
      CALL DDVD(XJR(KK+1),XJI(KK+1),AR,AI,XJR(KK),XJI(KK))
      XJR(KK)=CC*KK*XJR(KK)-XJR(KK+2)
      XJI(KK)=CC*KK*XJI(KK)-XJI(KK+2)
    1 CONTINUE
      CALL DDVD(GDR,GDI,XJR(2),XJI(2),RAR,RAI)
      DO 2 K=3,M
      TR=XJR(K)
      TI=XJI(K)
    2 CALL DMLT(TR,TI,RAR,RAI,XJR(K),XJI(K))
   13 XJR(2)=GDR
      XJI(2)=GDI
    4 DO 3 K=3,M
      CALL DDVD(XYR(K-1),XYI(K-1),AR,AI,XYR(K),XYI(K))
      XYR(K)=CC*(K-2)*XYR(K)-XYR(K-2)
    3 XYI(K)=CC*(K-2)*XYI(K)-XYI(K-2)
      END
```

```
      SUBROUTINE LRGBESD(X,Y,N,A,B,C,D)
      EXTERNAL DRINT,DGINT,DONER,DONEI,DTWOR,DTWOI
      COMMON/LF/EN,EX,WY
      TYPE DOUBLE DSIMCON,XI,PI,TEST,AR,RD,DRINT,AV,DGINT,XD,B1,EN,
     1            DONER,B2,DONEI,B3,B4,DTWOR,DTWOI,EX,WY,A,B,C,D,X,Y,
     2            TR,TI,HR,HI
      TEST=1.0D-16
      EX=X
      WY=Y
      EN=N
      XI=0.0D
      XD=20.0D
      LIM=16
      PI=3.14159265358979323846264330
C     INTEGRATE FOR J VALUE
      CALL DSIMCON(XI,PI,TEST,LIM,AR,NO,RD,DRINT)
      A=AR/PI
      CALL DSIMCON(XI,PI,TEST,LIM,AV,NI,RD,DGINT)
      B=AV/PI
      IF(WY.LE.0.0D)GO TO 5
C     INTEGRATE FOR H SUPER ONE VALUE
      CALL DSIMCON(XI,XD,TEST,LIM,B1,NA,RD,DONER)
      CALL DSIMCON(XI,XD,TEST,LIM,B2,NA,RD,DONEI)
      IF(N.EQ.1)GO TO 7
      TR=0.0D
      TI=-2.0D/PI
      GO TO 8
    7 TR=-2.0D*WY/PI
      TI=2.0D*EX/PI
    8 CALL DMLT(TR,TI,B1,B2,HR,HI)
      C=-B+HI
      D=A-HR
      RETURN
C     INTEGRATE FOR H SUPER TWO VALUE
    5 CALL DSIMCON(XI,XD,TEST,LIM,B3,NA,RD,DTWOR)
      CALL DSIMCON(XI,XD,TEST,LIM,B4,NA,RD,DTWOI)
      IF(N.EQ.1)GO TO 10
      TR=0.0D
      TI=2.0D/PI
      GO TO 11
   10 TR=2.0D*WY/PI
      TI=-2.0D*EX/PI
   11 CALL DMLT(TR,TI,B3,B4,HR,HI)
      C=B-HI
      D=-A+HR
      END
```

```
      DOUBLE PRECISION FUNCTION DRINT(T)
      COMMON/LF/EN,EX,WY
      TYPE DOUBLE SINT,ALPH,BETA,COSHB,DRINT,T,EN,EX,WY
      SINT=DSIN(T)
      ALPH=EX*SINT
      BETA=WY*SINT
      COSHB=(DEXP(BETA)+DEXP(-BETA))/2.0D
      DRINT=DCOS(EN*T)*DCOS(ALPH)*COSHB+DSIN(ALPH)*COSHB*DSIN(EN*T)
      END
```

```
      DOUBLE PRECISION FUNCTION DGINT(T)
      COMMON/LF/EN,EX,WY
      TYPE DOUBLE SINT,ALPH,BETA,SINHB,DGINT,T,EN,EX,WY
      SINT=DSIN(T)
      ALPH=EX*SINT
      BETA=WY*SINT
      SINHB=(DEXP(BETA)-DEXP(-BETA))/2.0D
      DGINT=DCOS(ALPH)*SINHB*DSIN(EN*T)-DSIN(ALPH)*SINHB*DCOS(EN*T)
      END
```

```
      DOUBLE PRECISION FUNCTION DONER(T)
      COMMON/LF/EN,EX,WY
      TYPE DOUBLE DONER,T,ALPH,TSQ,TTN,EN,EX,WY
      ALPH=DSQRT(T*T+1.0D)
      TTN=T*T
      IF(EN.EQ.0.0D)TTN=1.0D
    2 DONER=TTN*DEXP(-ALPH*WY)*DCOS(ALPH*EX)/ALPH
      END
```

```
      DOUBLE PRECISION FUNCTION DONEI(T)
      COMMON/LF/EN,EX,WY
      TYPE DOUBLE DONEI,T,ALPH,TSQ,TTN,EN,EX,WY
      ALPH=DSQRT(T*T+1.0D)
      TTN=T*T
      IF(EN.EQ.0.0D)TTN=1.0D
    2 DONEI=TTN*DEXP(-ALPH*WY)*DSIN(ALPH*EX)/ALPH
      END
```

```
      DOUBLE PRECISION FUNCTION DTWOR(T)
      COMMON/LF/EN,EX,WY
      TYPE DOUBLE DTWOR,T,ALPH,TSQ,TTN,EN,EX,WY
      ALPH=DSQRT(T*T+1.0D)
      TTN=T*T
      IF(EN.EQ.0.0D)TTN=1.0D
    2 DTWOR=TTN*DEXP(ALPH*WY)*DCOS(ALPH*EX)/ALPH
      END
```

```
      DOUBLE PRECISION FUNCTION DTWOI(T)
      COMMON/LF/EN,EX,WY
      TYPE DOUBLE DTWOI,T,ALPH,TSQ,TTN,EN,EX,WY
      ALPH=DSQRT(T*T+1.0D)
      TTN=T*T
      IF(EN.EQ.0.0D)TTN=1.0D
    2 DTWOI=-TTN*DEXP(ALPH*WY)*DSIN(ALPH*EX)/ALPH
      END
```

6.0 COMPARISON

The program has been used to calculate the form function for an absorbing cylinder. Quantitative agreement between theory and experiment was found [6].

$J_n(x + iy)$ and $Y_n(x + iy)$ were generated for the values n = 12, x = ±6, and y = ±2, and the values n = 12, x + iy = ±10, and x + iy = ±10i using CMPJYBES, CMPJYBED, CØMBES (subroutine CØMBES PS582A in the CDC-6700 Math Science Library at the Naval Ship Research and Development Center, with the run being performed at NSRDC), and CØMPBE (subroutine CØMPBE (identification name-BES 4) from the Atomic Energy Commission Computing and Applied Mathematics Center, Courant Institute of Mathematical Science, New York University). For the values x + iy = ±6 ± 2i and x + iy = ±10 CMPJYBES (and CMPJYBED) results compared exactly to the results (four figures) printed by CØMBES, in all four quadrants. The values for $Y_n(x + iy)$ generated by CØMPBE in quadrants two and three did not agree with either the CØMBES results or the CMPJYBES (CMPJYBED) results. For the values of x + iy = ±10i, CMPJYBES (and CMPJYBED) results did not compare with those of CØMBES. They did however agree to nine figures (eleven in double precision) with the NBS Mathematical Tables [Ref. 5, Section 3.15]. In addition, the answers checked out using the identity mentioned in Section 1.10.

7.0 TEST METHOD AND RESULTS

Two test programs MASSNG and SPECD, which call CMPJYBES and CMPJYBED respectively, and the results as printed by these programs, follow. Note that $Y_{47}(20 + 20i)$ and $Y_{48}(20 + 20i)$ are not correct in the single-precision version. However, when values of $Y_0(20 + 20i)$ to $Y_{12}(20 + 20i)$ are printed, one would see that at n = 1 the single-precision and double-precision results agree to ten figures, and at n = 12 they agree to eight figures.

```
      PROGRAM MASSNG
      DIMENSION X(24),Y(24),XJR(250),XJI(250),XYR(250),XYI(250),N(1)
      DATA(X=0.4,4.0,7.5,50.0,20.0),(Y=0.3,4.0,5.5,4.0,20.0)
      JJ=5
      N=50
      II=1
      TP=2.0/3.1415926536
      DO 4 J=1,JJ
      DO 4 I=1,II
      U=X(J)
      V=Y(J)
      M=N(I)
      NN=M+1
      CALL CMPJYBES(U,V,M,XJR,XJI,XYR,XYI)
      DEN=U*U+V*V
      REAL=TP*U/DEN
      XMAG=-TP*V/DEN
      RHO=SQRT(DEN)
      PRINT 3,RHO,U,V,REAL,XMAG
    3 FORMAT(///,5(2X,E17.10),/)
      DO 4 L=48,49
      NORD=L-1
      BRL=XJR(L+1)*XYR(L)-XJI(L+1)*XYI(L)-
     1     XJR(L)*XYR(L+1)+XJI(L)*XYI(L+1)
      BIM=XJI(L+1)*XYR(L)+XJR(L+1)*XYI(L)-
     1     XJI(L)*XYR(L+1)-XJR(L)*XYI(L+1)
      PRINT 5,NORD,XJR(L),XJI(L),XYR(L),XYI(L),BRL,BIM
    5 FORMAT(2X,I5,6(2X,E17.10))
    4 CONTINUE
      END
```

5.0000000000-001 4.0000000000-001 3.0000000000-001 1.0185916358+000 -7.6394372684-001

47	7.5663825620-089	-1.7987786866-088	-1.3454955850+085	-3.1991732534+086	1.0185916357+000	-7.6394372684-001
48	8.7740389495-091	-5.1302309809-091	-5.6323386166+087	-3.2936545836+087	1.0185916357+000	-7.6394372684-001

5.6568542495+000 4.0000000000+000 4.0000000000+000 7.9577471548-002 -7.9577471548-002

47	3.7454216199-039	-5.2604412031-039	-6.0205463333+035	-8.5868631238+035	7.9577470375-002	-7.9577486973-002
48	3.7545047791-040	-6.1848113351-041	-1.7175109102+037	-2.9519568121+036	7.9577470373-002	-7.9577486971-002

9.3005376188+000 7.5000000000+000 5.5000000000+000 5.5198246159-002 -4.0478713849-002

47	-4.0689556705-029	-6.8376552676-029	4.5152357429+025	-7.2713544850+025	5.5198268510-002	-4.0478736338-002
48	8.0847114879-031	-7.6866876099-030	-7.4679442139+025	-8.5934960933+026	5.5198268509-002	-4.0478736338-002

5.0159744816+001 5.0000000000+001 4.0000000000+000 1.2651426319-002 -1.0121141055-003

47	3.8755527215-001	8.5720881007-002	-1.1069649929-001	3.5873900444-001	1.2651425802-002	-1.0121137930-003
48	2.8589165342-001	1.6670467217-001	-2.0977111853-001	2.6602848643-001	1.2651425803-002	-1.0121137987-003

2.8284271247+001 2.0000000000+001 2.0000000000+001 1.5915494309-002 -1.5915494309-002

47	1.1943324408-006	5.3279344452-006	-4.5587186429+009	4.7441693080+008	8.2259421577+004	2.4316730769+004
48	-9.6129549083-007	1.2695395540-006	-8.8053359625+009	1.2800796586+010	8.2259421574+004	2.4316730765+004

```
      PROGRAM SPECD
      TYPE DOUBLE XJR,XJI,XYR,XYI,X,Y,TP,U,V,DEN,REAL,XMAG,RHO,RL,BIM
      DIMENSION X(24),Y(24),XJR(250),XJI(250),XYR(250),XYI(250),N(1)
      DATA(X=0.4D,4.0D,7.5D,50.0D,20.0D),
     1     (Y=0.3D,4.0D,5.5D,4.0D,20.0D)
      JJ=5
      N=50
      II=1
      TP=2.0D/3.14159265358979323846264330
      DO 4 J=1,JJ
      DO 4 I=1,II
      U=X(J)
      V=Y(J)
      M=N(I)
      NN=M+1
      CALL CMPJYBED(U,V,M,XJR,XJI,XYR,XYI)
      DEN=U*U+V*V
      REAL=TP*U/DEN
      XMAG=-TP*V/DEN
      RHO=DSQRT(DEN)
      PRINT 3,RHO,U,V,REAL,XMAG
    3 FORMAT(///,5(2X,E17.10),/)
      DO 4 L=48,49
      NORD=L-1
      BRL=XJR(L+1)*XYR(L)-XJI(L+1)*XYI(L)-
     1     XJR(L)*XYR(L+1)+XJI(L)*XYI(L+1)
      BIM=XJI(L+1)*XYR(L)+XJR(L+1)*XYI(L)-
     1     XJI(L)*XYR(L+1)-XJR(L)*XYI(L+1)
      PRINT 5,NORD,XJR(L),XJI(L),XYR(L),XYI(L),BRL,BIM
    5 FORMAT(2X,I5,6(2X,E17.10))
    4 CONTINUE
      END
```

5,0000000000*001	3,9999999999*001	2,9999999999*001	1,0185916358*000	*7,6394372684-001		
47	7,5663825641*089	*1,7987786871*088	-1,3454955846*085	-3,1991732527*085	1,0185916358*000	*7,6394372684-001
48	8,7740389518*091	*5,1302309823*091	-5,6323386150*087	-3,2936545832*087	1,0185916358*000	*7,6394372684-001
5,6568542494*000	4,0000000000*000	4,0000000000*000	7,9577471544*002	*7,9577471544*002		
47	3,7454216201*039	*5,2604412059*039	*6,0205448944*035	*8,5868629806*035	7,9577471546*002	*7,9577471544*002
48	3,7545047805*040	*6,1848113451*041	-1,7175107249*037	-2,9910583414*036	7,9577471546*002	*7,9577471544*002
9,3005376188*000	7,5000000000*000	5,5000000000*000	5,5198246158*002	*4,0478713849*002		
47	*4,0689556696*029	*6,8376552650*029	4,5152349010*025	*7,2713508350*025	5,5198246159*002	*4,0478713849*002
48	8,0847114790*031	*7,6866876076*030	*7,4679346186*025	*8,5934922174*026	5,5198246159*002	*4,0478713849*002
5,0159744816*001	5,0000000000*001	4,0000000000*000	1,2651426319*002	*1,0121141055*003		
47	3,8755527217*001	8,5720880956*002	-1,1069650097*001	3,5873900377*001	1,2651426319*002	*1,0121141055*003
48	2,8589165364*001	1,6670467213*001	-2,0977119080*001	2,6602848651*001	1,2651426319*002	*1,0121141055*003
2,8284271247*001	2,0000000000*001	2,0000000000*001	1,5915494309*002	*1,5915494309*002		
47	1,1943324407*006	5,3279344441*006	*4,6198916796*002	1,1103762774*003	1,5915494376*002	*1,5915494175*002
48	*9,6129549057*007	1,2695395538*006	1,8724244119*003	3,5881353704*003	1,5915494376*002	*1,5915494175*002

*U.S. GOVERNMENT PRINTING OFFICE:1976 211-264/582 1-3