DAHLGREN DIVISION
NAVAL SURFACE WARFARE CENTER
Dahlgren, Virginia 22448-5100

NSWCDD/TR-98/75

TESTING A FORTRAN 90 COMPILER USING THE NSWC FORTRAN 77 MATHEMATICS LIBRARY

BY ARMIDO DIDONATO RUSS GNOFFO

THEATER WARFARE SYSTEMS DEPARTMENT

19990308030

FEBRUARY 1999

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, search existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services. Directorate for information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
February 1999

3. REPORT TYPE AND DATES COVERED
Final

4. TITLE AND SUBTITLE
Testing a Fortran 90 Complier Using the NSWC Fortran 77 Mathematics Library

5. FUNDING NUMBERS

6. AUTHOR(s)
Armido DiDonato Russ Gnoffo

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Commander
Naval Surface Warfare Center
Dahlgren Division (Code T10)
17320 Dahlgren Road
Dahlgren, VA 22448-5100

8. PERFORMING ORGANIZATION REPORT NUMBER
NSWCDD/TR-98/75

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report describes the analysis and associated Fortran program (TEST90) that were developed to aid in establishing the validity of a new Fortran 90 mainframe compiler. The FORTRAN 77 Naval Surface Warfare Center (NSWC) Mathematics Library (MLIB) is used as a source of routines for checking the Fortran 90 compiler. At the same time, this study can be considered as an aid to determine whether MLIB can operate in a Fortran 90 environment. The inputs for the routines were chosen so that many of the different possible paths of the routines were executed. Seventy-four directly callable routines, with 293 supporting routines, were chosen for testing. All but 17, and their supporting routines, were taken from MLIB. The ones not belonging to MLIB, are double-precision versions of routines in MLIB. Thirteen hundred and twenty five numerical cases were submitted for testing. A true value for each test was obtained independently and given correctly to 35 digits by using MAPLE software. If the difference in the test output and the corresponding true value exceeds a prespecified error tolerance, an error message is printed identifying the routine and the input. Additional test cases were also prepared to check the bit and string instructions, since these do not appear in MLIB.

TEST90 has been used to test the latest Fortran 90 compilers of the CRAY EL98 and IBM PC machines. No errors were found; however, TEST90 did reveal a complex arithmetic error in an earlier version of the Cray EL98 compiler. MLIB routines ran under TEST90 without any problems on both machines.

The transportability of MLIB allows TEST90 to be used as an aid in testing Fortran 90 compilers on a variety of computers, with a single-precision word length no larger than 64 bits.

14. SUBJECT TERMS
Fortran 90 compiler TEST90 NSWC Mathematics Library MLIB

15. NUMBER OF PAGES
65

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORTS	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

FOREWORD

The work described in this report was proposed by Tom Owen and William Fairfax (T11) and carried out in the Space and Weapons Systems Analysis Division (T10) of the Theater Warfare Systems Department (T). It was sponsored by the Scientific Computing Systems and Networks Branch (T11). The program described in this report is used as an aid in testing whether the NSWC Mathematics Library can run in a Fortran 90 environment or as an aid for testing Fortran 90 compilers.

The document was reviewed by James L. Sloop, Head, Space and Weapons Systems Analysis Division, Dr. Jeffrey Blanton, Head, Space Systems Applications Branch, William Ormsby, of the Space Systems Applications Branch, Thomas Owen, Head, and William Fairfax of the Scientific Computing Systems and Networks Branch.

Approved by:

THOMAS C. PENDERGRAFT, Head
Theater Warfare Systems Department

CONTENTS

I. Introduction

The objective of the work described in this report is to use the NSWC Mathematical Library (MLIB)[1], which is written in FORTRAN 77, as an aid in testing a new Fortran 90 compiler used with the CRAY EL98 mainframe computer. Indirectly, one can view the testing as an aid in establishing whether MLIB can operate in a Fortran 90 environment. The final result is a transportable driver program, TEST90, which is not restricted to testing only the new 90-compiler. Any post Fortran 77 compiler can be tested provided it is limited to a single precision word length of 64 bits or less. This limitation is imposed because the routines in MLIB are based on a 64 bit word length with a 48 bit floating point mantissa.

TEST90 has been used to test the latest Fortran 90 compilers of the CRAY EL98 and IBM PC machines; no errors were found, however TEST90 did reveal a complex arthmetic error in an earlier Cray EL98 compiler. MLIB routines ran under TEST90 without any problems on both machines.

MLIB contains 1099 routines of which 586 are *callable* directly by the user. The rest are called *supporting* routines. We have chosen 57 of the callable routines, Group A, as a representative sample of MLIB. Eleven other callable routines are included in Group A that act as supporting routines making a total of 68. In addition, 17 double precision versions of a subset of the 57, which are not part of MLIB, were constructed using a routine called FETCH (FCH), (designed by Russ Gnoffo. See CD disk of [1]). These routines are said to belong to Group B. Thus 85 routines, making up Groups A and B, were used for testing. Supporting these routines were approximately 282 additional routines.

Thirteen hundred and twenty-five output quantities were generated and examined. The fixed input data specified for these test cases were chosen to bring into play as many of the supporting routines as possible. Of the 282 supporting routines mentioned in the previous paragraph, 238 (or 319 of the total 367) were actually executed in running the 1325 test cases.

TEST90 also calls subprograms which check that all the bit and string commands of the Fortran 90 compiler operate correctly.[1]

For the numerical testing of groups A and B, a *true result* for each output was obtained independently to 35 significant digits by using MapleV software [2]. TEST90 *checks* that each Fortran 90 test case *calculated* result agrees with the corresponding true Maple value to within a certain accuracy. In order to do this, for each single precision (double precision) output, a *prespecified tolerance* is given in terms of a positive parameter EPS (EPD) that

[1]MLIB does not use any bit, string, or I/O statements.

depends on the machine dependent parameter SPMPAR(1) (DPMPAR(1))[1, p.3-4][2]. Then if the *relative error,* which is defined in (1), is within the tolerance, that case is said to have *passed.* If the relative error exceeds the tolerance, a message is printed by TEST90 identifying the routine, the case number, and the output; that particular test case has *failed.* If no disagreement occurs for all 1325 cases, a message is printed at the end of the run that all has gone well. It should be noted that in case of a failure the test case and the subprogram of Group A or B is identified, but the specific problem with the compiler is not isolated. An independent study is required to determine what caused the failure.

Let T denote the true result (obtained from Maple), C the calculated result (one of the 1325 numerical outputs), and let, for a single precision result, $X \equiv \mathrm{SPMPAR(2)/SPMPAR(1)}$[3] and for double precision, $X \equiv \mathrm{DPMPAR(2)/DPMPAR(1)}$. Then the relative error is given by

$$\mathrm{RE} = \begin{cases} \max(|T|,\ |C|) & \text{for } |T| \le X. \\ |T| & \text{for } |C| \le X. \\ Q & \text{for } Q > 5\ \mathrm{SPMPAR(1)(DPMPAR(1))} \text{ else } 0, \end{cases} \tag{1}$$

where $Q \equiv |(T - C)/T|$.

A test result is said to *pass* if

$$\mathrm{RE} \le \mathrm{EPS} \quad \text{(or EPD for a double precision result)}, \tag{2}$$

where the machine dependent parameters EPS and EPD are defined below.

It is convenient here to introduce the following briefer notation:

CRAY:	SPMPAR(1)	$\rightarrow$	Sc,	DPMPAR(1)	$\rightarrow$	Dc,	RE	$\rightarrow$	REc
IBM PC:	SPMPAR(1)	$\rightarrow$	Spc,	DPMPAR(1)	$\rightarrow$	Dpc,	RE	$\rightarrow$	REpc
Machine:	SPMPAR(1)	$\rightarrow$	Sm,	DPMPAR(1)	$\rightarrow$	Dm,	RE	$\rightarrow$	REm

where Machine refers to the computer which is used with the compiler being tested.

The quantities EPS and EPD are defined in terms of pre-computed elements SE and SF, where SE is given by

$$\mathrm{SE} = \begin{cases} \mathrm{REc/Sc}, & \text{if } \mathrm{REc} > 5\,\mathrm{Sc}; \quad \text{else } 1.25 \quad \text{(single precision).} \\ \mathrm{REc/Dc}, & \text{if } \mathrm{REc} > 5\,\mathrm{Dc}; \quad \text{else } 1.25 \quad \text{(double precision).} \\ 1, & \text{if } T = C = 0. \end{cases} \tag{3}$$

[2]SPMPAR(1) (DPMPAR(1)) denotes the smallest positive value such that $1 + \mathrm{SPMPAR(DPMPAR(1))} > 1$.
CRAY : SPMPAR(1) (DPMPAR(1)) $= 2^{-47}\ (2^{-95})$. PC : SPMPAR(1) (DPMPAR(1)) $= 2^{-23}\ (2^{-52})$.

[3]SPMPAR(2) (DPMPAR(2)), is the smallest positive single precision (double precision) number in the machine.
CRAY : SPMPAR(2) = DPMPAR(2) $= 2^{-8188}$; PC : SPMPAR(2) (DPMPAR(2)) $= 2^{-126}\ (2^{-1021})$.

Note that the 1325 SE values depend only on the 64 bit CRAY.

In order to define SF (safety factor) for a particular output, it is necessary to determine whether *rounding error* $\mathcal{R}$ or *truncation error* $\mathcal{T}$ is the major contributor to the relative error. For example, in the inversion of the Hilbert matrix $\mathcal{R}$ plays the dominate role, whereas in an application of Runge Kutta, $\mathcal{T}$ is the main contributor. With this in mind, each of the 1325 cases is classified as either an $\mathcal{R}$ or a $\mathcal{T}$ type. Note that this classification is independent of the machine. Thus let

$$\mathrm{Z} = \begin{cases} \mathrm{SE} * \mathrm{Spc(Dpc)}, & \text{if REpc is of type } \mathcal{R} \quad \text{(single (double) precision)} \\ \mathrm{SE} * \mathrm{Sc(Dc)}, & \text{if REpc is of type } \mathcal{T} \quad \text{(single (double) precision).} \end{cases} \tag{4}$$

Using the above parameters, we have

$$\mathrm{SF} = \begin{cases} 5, & \text{if REpc} < 5\,\mathrm{Spc(Dpc)} \text{ (single(double) precision)} \\ \mathrm{Max}[\,3, 5\,\mathrm{REpc/Z}\,], & \text{otherwise.} \end{cases} \tag{5}$$

Hence SF is computed using the 32-bit PC, as well as SE which was computed using the CRAY. This approach allows TEST90 to be used on computers with word lengths from 32 to 64 bits.

Using SE and SF, from (3) and (5), the tolerance parameters EPS and EPD are given by

$$\mathrm{EPS} = \begin{cases} \mathrm{Max}[\,5 * \mathrm{Sm},\ \mathrm{SF} * \mathrm{SE} * \mathrm{Sm}\,], & \text{if REpc is of type } \mathcal{R} \\ \mathrm{Max}[\,5 * \mathrm{Sm},\ \mathrm{SF} * \mathrm{SE} * \mathrm{Sc}\,], & \text{if REpc is of type } \mathcal{T}. \end{cases} \tag{6}$$

$$\mathrm{EPD} = \begin{cases} \mathrm{Max}[\,5 * \mathrm{Dm},\ \mathrm{SF} * \mathrm{SE} * \mathrm{Dm}\,], & \text{if REpc is of type } \mathcal{R} \\ \mathrm{Max}[\,5 * \mathrm{Dm},\ \mathrm{SF} * \mathrm{SE} * \mathrm{Dc}\,], & \text{if REpc is of type } \mathcal{T}. \end{cases} \tag{7}$$

If a single precision output result is tested using (2), where EPS is given by (6), it passes if REm$\leq$EPS or if EPS $\geq 1/10$ for $\mathcal{R}$ type cases. The latter inequality holds when no digits in C are correct because of using too short a word length (An example is attempting to find the inverse elements of the tenth order Hilbert matrix in single precision on a 32 bit machine). If a double precision output is tested, using (7), it passes only if REm$\leq$EPD. All 1325 cases will pass if no compiler errors are found.

Section II lists the call lines of the Group A and the Group B routines, the associated test cases, and the corresponding 35 digit true results obtained from Maple software. Subroutines testing bit manipulation functions are described in Section III and subroutines testing string operations are explained in Section IV. Section V contains directions for using TEST90. Section VI contains an index of the routines discussed in Section II.

II. Callable Routines and True Value Maple Data

In this section we list the directly callable routines that were used, with their call line from MLIB and their page number in [1], and the routines in Group B. Also given are the test cases, and the 35 digit Maple [2] results which are considered to be the true values. Each of the subprograms is numbered in numerical order, $1 \leq \mathrm{N} \leq 74$, and on the same line in parentheses the range of each set of outputs is itemized in numerical order, so that on the first line with each subprogram we have $\underline{\mathrm{N}.\ (a_{\mathrm{N}} - b_{\mathrm{N}}) - \mathrm{Page\,M}}$, where $1 \leq a_{\mathrm{N}} \leq b_{\mathrm{N}} \leq 1325$, $a_{\mathrm{N}+1} = b_{\mathrm{N}} + 1$, and M refers to the page number in [1]. A complex number is listed as an ordered pair of real numbers, i.e., $z = a + I\,b = (a, b)$; $I = \sqrt{-1}$. If the imaginary part of a complex number is not given it is assumed to take the value zero.

1. (1–7)–Page 11

   $y = \mathbf{CBRT}(x)$ – CUBE ROOT.

   $x_1, y_1 \ = \ 1E(-200), \ .21544346900318837217592935665193505E(-66)$

   $x_2, y_2 \ = \ 5E(-29), \ .36840314986403866057798228335798072E(-9)$

   $x_3, y_3 \ = \ 1E(-3), \ .1$

   $x_4, y_4 \ = \ 0, \ 0$

   $x_5, y_5 \ = \ 1E3, \ 10$

   $x_6, y_6 \ = \ 5E29, \ .79370052598409973737585281963615413E10$

   $x_7, y_7 \ = \ 1E50, \ .46415888336127788924100763509194466E17$

2. (8–14)–Page 11

   $y = \mathbf{DCBRT}(x)$ – DOUBLE PRECISION CUBE ROOT.

   Double precision version of CBRT. Uses same input.

3. (15–18)–Page 11

   $w = \mathbf{ARTNQ}(y, x)$ – FOUR QUADRANT ARCTANGENT, $[0, 2\pi)$.

   $y_1/x_1 \ = \ 3.9/1.7, \ w_1 = 1.1597317794050322209919342332641325$

   $y_2/x_2 \ = \ 3.9/-1.7, \ w_2 = 1.9818608741847610174707091500153704$

   $y_3/x_3, \ = \ -3.9/-1.7, \ w_3 = 4.3013244329948254594545776165436354$

   $y_4/x_4, \ = \ -3.9/1.7, \ w_4 = 5.1234535277745542559333525332948733$

4. (19–22)–Page 11

   $w = \mathbf{DARTNQ}(y/x)$ – FOUR QUADRANT ARCTANGENT.

   Double precision version of ARTNQ. Uses same inputs as ARTNQ.

5. (23–30)–Page 13

   Call $\mathbf{DCSQRT}(Z, W)$ – $\sqrt{Z}$, DOUBLE PRECISION COMPLEX.

   $Z = (\text{real}, \text{imag}), \ Z_1 = (0, 1), \ Z_2 = (1, 1), \ Z_3 = (50, 25), \ Z_4 = (1E30, 1E40).$

   W_1=(.70710678118654752440084436210484905, .70710678118654752440084436210484905)

   W_2=(1.0986841134678099660398011952406784, .45508986056222734130435775782246858)

   W_3=(7.2767334511267740406133091985484853, 1.7178037486125623206928287195727927)

   W_4=(70710678122190286346.105562180133025, 70710678115119218534.240086936133420)

6. (31–35)–Page 29

   $y = \mathbf{REXP}(x) - [EXP(x) - 1]$.

   $x_1 = -50, \quad x_2 = -5, \quad x_3 = 1E(-20), \quad x_4 = 50, \quad x_5 = 85.$

   $y_1 = -.99999999999999999999998071250152036l$

   $y_2 = -.99326205300091453290336395157685158$

   $y_3 = .10000000000000000000050000000000000E(-19)$

   $y_4 = .51847055285870724640864533229334854E22$

   $y_5 = .82230127146229135103043280164077747E37$

7. (36–40)–Page 29

   $y = \mathbf{DREXP}(x) - [EXP(x) - 1]$.

   Double precision version of REXP. Uses same input.

8. (41–46)–Page 31

   $y = \mathbf{RLOG}(x) - [x - 1 - LOG(x)]$.

   $x_i = [1E(-100),\ 1E(-5),\ 1,\ 1.005,\ 50,\ 100]; \quad i = 1, \ldots, 6.$

   $y_1 = 229.25850929940456840179914546843642$

   $y_2 = 10.512935464970228420089957273421821, \qquad y_3 = 0$

   $y_4 = .12458488960926387897797540656528063E(-4)$

   $y_5 = 45.087976994571853941381249212089448$

   $y_6 = 94.394829814011908631964017090631272$

9. (47–52)–Page 32

   $y_i = \mathbf{DRLOG}(x_i) - [x - 1 - LOG(x)]$.

   Double precision version of RLOG. Uses same input.

10. (53–59)–Page 46

    $y = \mathbf{ERF}(x)$ – ERROR FUNCTION.

    $x_i = [-5,\ -1,\ .4,\ 1.3,\ 3.5,\ 6.7,\ 20]; \quad i = 1, \ldots, 7.$

    $y_1 = -.99999999999846254020557196514981166$

    $y_2 = -.84270079294971486934122063508260926$

    $y_3 = .42839235504666845510360384532017244$

    $y_4 = .93400794494065243660389332750371102$

    $y_5 = .99999925690162765858725447631624390$

    $y_6 = .99999999999999999999973382854236234l$

    $y_7 = 1.0$

11. (60–66)–Page 48

    y = **DERF**(x) – DOUBLE PRECISION ERROR FUNCTION.

    Uses same arguments as ERF.

12. (67–73)–Page 46

    y = **ERFC1**(IND, x) – e^{x^2}[1 – ERF(x)] = e^{x^2}ERFC(x).

    Uses same arguments as ERF. IND = 1.

    y_1 = .144009798674661040410589634305882 11E12

    y_2 = .50089800807622834663098245982148099E1

    y_3 = .67078778529476152332941247446792512

    y_4 = .35764266908609031764697338728600749

    y_5 = .15529365560889429740272649758187893

    y_6 = .83299189466810412994067439478728164E(–1)

    y_7 = .28174348741051319318649154534470758E(–1)

13. (74–80)–Page 49

    y = **DERFC1**(IND, x) – DOUBLE PRECISION ERFC1.

    Uses same arguments as ERF. IND = 1.

14. (81–88)–Page 45

    Call **CERF**(MO, z, w)–ERROR FUNCTION, COMPLEX ARGUMENT.

    z_i = [(.4, .3), (3.4, 2.3), (5.3, 3.7), (6.2, .009)]; i = 1, . . . , 4; MO = 0.

    w_1 = (.46443721297956140209712031572604685, .29443980776931707303974297404552757)

    w_2 = (1.0002253313625125573357082824478311, –.12334728086728759257490539663821289E(–3))

    w_3 = (1.0000000252590040600802224870937450, .41332248953387653013128019073968635E(–7))

    w_4 = (.99999999999999999819476945857815225, .20489319058601197920097004639286141E(–18))

15. (89–96)–Page 47

    Call **DCERF**(MO, Z, W)–CERF, DOUBLE PRECISION COMPLEX.

    Uses same arguments as CERF. MO=0.

16. (97–104)–Page 46

    Call **CERFC**(MO, z, w) – e^{z^2}ERFC(z) WITH COMPLEX ARGUMENT.

    Uses same arguments as CERF. MO=1, Re(z)$\geq$ 0.

    w_1 = (.63299603234343978026485063394958694, –.17020263553343032296655297457404080)

    w_2 = (.11438075139307192540234735774672486, –.73122766392201685536727808849318310E(–1))

    w_3 = (.71800086151584466507931675846826108E(–1), –.48959427476482652124106743611245310E(–1))

    w_4 = (.89857930230247263981968919036797811E(–1), –.12724736894709053219621092389506531E(–3))

17. (105−112)−Page 48

    Call **DCERFC(MO, Z, W)**−CERFC(z), DOUBLE PRECISION.

    Uses same arguments as CERF. MO=1, Re(Z)≥ 0.

18. (113−184)−Page 88

    Call **BRATIO**(a, b, x, y, w, w1, IERR)−INCOMPLETE BETA FUNCTION .

    $x + y = 1$, Output : w, w1 ($w + w1 = 1$), IERR(= 0),

    A=a_i=[.25, 1.25, 7.25, 20.25], B=b_j=[.25, 2.25, 8.25, 25.25], X=x_k=[.25, .5, .75, .99] All combinations of the elements of A,B,X are used. Data below shown as BRAT(A[i], B[j], X[k])= w, or BRAT(B[j], A[i], y[k])=w1,where the first is listed if w≤ w1 otherwise the second is listed. For i=1, j=2, k=3, Call **BRATIO**(.25, 2.25, .75, w, w1, IERR) so that BRAT(2.25,.25,.25) = w1 = .73514100840602161576356511063802129E(−2). In addition, another 8 values of w = BRAT(a, b, x) are included at the end of the listing that follows, making a total of 72 BRAT cases.

BRAT(A[1], B[1], X[1])	=	.39775677831735583689234908829962206
BRAT(A[1], B[1], X[2])	=	.50000000000000000000000000000000000
BRAT(B[1], A[1], 1-X[3])	=	.39775677831735583689234908829962206
BRAT(B[1], A[1], 1-X[4])	=	.17081535717490444338991136891551120
BRAT(B[2], A[1], 1-X[1])	=	.14085505922511297496662594047200120
BRAT(B[2], A[1], 1-X[2])	=	.42344141898912087162604411421830524E(−1)
BRAT(B[2], A[1], 1-X[3])	=	.73514100840602161576356511063802129E(−2)
BRAT(B[2], A[1], 1-X[4])	=	.45719954438368161042018130444696542E(−5)
BRAT(B[3], A[1], 1-X[1])	=	.12277951728998734846248338369272288E(−1)
BRAT(B[3], A[1], 1-X[2])	=	.28782107488638388447298067656873723E(−3)
BRAT(B[3], A[1], 1-X[3])	=	.73070314067824048001705945939654844E(−6)
BRAT(B[3], A[1], 1-X[4])	=	.17832580880707298509927490339357089E(−17)
BRAT(B[4], A[1], 1-X[1])	=	.44807240299498264499828954500884385E(−4)
BRAT(B[4], A[1], 1-X[2])	=	.10005293450945211288323071593641690E(−8)
BRAT(B[4], A[1], 1-X[3])	=	.18833241696028164854032908626610601E(−16)
BRAT(B[4], A[1], 1-X[4])	=	.77705810900429982378507276583122027E(−52)
BRAT(A[2], B[1], X[1])	=	.42842807196177999860791418123947654E(−1)
BRAT(A[2], B[1], X[2])	=	.11862011824909340596883700951819210
BRAT(A[2], B[1], X[3])	=	.24732925056146632607609324152470353
BRAT(B[1], A[2], 1-X[4])	=	.34094562129787317503712966284227090

BRAT(A[2], B[2], X[1])	=	.38001107976129694504077120479083834
BRAT(B[2], A[2], 1-X[2])	=	.27117207094945604358130220571091526
BRAT(B[2], A[2], 1-X[3])	=	.60588505752236891712369301632731374E(−1)
BRAT(B[2], A[2], 1-X[4])	=	.45403258833349311699534203586891981E(−4)
BRAT(B[3], A[2], 1-X[1])	=	.13405072148901327755227909052879592
BRAT(B[3], A[2], 1-X[2])	=	.53935152322007520804276981962034880E(−2)
BRAT(B[3], A[2], 1-X[3])	=	.19290796077732735634114110543000211E(−4)
BRAT(B[3], A[2], 1-X[4])	=	.60089970920153482181852795569159478E(−16)
BRAT(B[4], A[2], 1-X[1])	=	.12649438096455270664540720009991680E(−2)
BRAT(B[4], A[2], 1-X[2])	=	.52923753445854760109332516271631398E(−7)
BRAT(B[4], A[2], 1-X[3])	=	.14588815980030198715137476274492143E(−14)
BRAT(B[4], A[2], 1-X[4])	=	.78497514038461340403536969387757035E(−50)
BRAT(A[3], B[1], X[1])	=	.32053821989521731672299996038770368E(−5)
BRAT(A[3], B[1], X[2])	=	.62820068537400843086996184454438728E(−3)
BRAT(A[3], B[1], X[3])	=	.17690074829443825633183038465251117E(−1)
BRAT(A[3], B[1], X[4])	=	.44191160942382014481057048512463266
BRAT(A[3], B[2], X[1])	=	.39791669199364443677769395264851494E(−3)
BRAT(A[3], B[2], X[2])	=	.40112235501938455812919777329719793E(−1)
BRAT(A[3], B[2], X[3])	=	.41006899961171290768594879542371616
BRAT(B[2], A[3], 1-X[4])	=	.12264603413482860075721589941537032E(−2)
BRAT(A[3], B[3], X[1])	=	.35094060735746471970058683739185098E(−1)
BRAT(B[3], A[3], 1-X[2])	=	.39702207630401328146312869741167449
BRAT(B[3], A[3], 1-X[3])	=	.95100701659741858342243382133621038E(−2)
BRAT(B[3], A[3], 1-X[4])	=	.12539562113523425469799050241820693E(−12)
BRAT(B[4], A[3], 1-X[1])	=	.33218601813701179608821314194707665
BRAT(B[4], A[3], 1-X[2])	=	.48873314409464396177573572535759741E(−3)
BRAT(B[4], A[3], 1-X[3])	=	.12965691544661661958807016151086044E(−9)
BRAT(B[4], A[3], 1-X[4])	=	.34207572438829445357106628963957989E(−44)
BRAT(A[4], B[1], X[1])	=	.22686183750632833038237745473990532E(−13)
BRAT(A[4], B[1], X[2])	=	.37515536937999003719100338783203475E(−7)
BRAT(A[4], B[1], X[3])	=	.21915408366712445080427980539350023E(−3)
BRAT(A[4], B[1], X[4])	=	.29032021519120007197335268397917479
BRAT(A[4], B[2], X[1])	=	.18565198028188161033737070811254010E(−10)
BRAT(A[4], B[2], X[2])	=	.14489894766588339444424096839148527E(−4)

BRAT(A[4], B[2], X[3])	=	.25011190127138313243352465595362702E(−1)
BRAT(B[2], A[4], 1-X[4])	=	.10108292471982372254346080172398906E(−1)
BRAT(A[4], B[3], X[1])	=	.11872583678129764648897029511175989E(−6)
BRAT(A[4], B[3], X[2])	=	.10179807647530308112604293623106432E(−1)
BRAT(B[3], A[4], 1-X[3])	=	.33872352758566424339682566036491872
BRAT(B[3], A[4], 1-X[4])	=	.86292559297135160668356606705617935E(−10)
BRAT(A[4], B[4], X[1])	=	.23422666139670676884195587790023536E(−2)
BRAT(B[4], A[4], 1-X[2])	=	.22696748842771075618260174519028746
BRAT(B[4], A[4], 1-X[3])	=	.64552803333196107540883097672076182E(−5)
BRAT(B[4], A[4], 1-X[4])	=	.52248939289048511271402156079156800E(−38)
BRAT(40.25, 45.25, .25)	=	.57379676252380811202923922414822492E(−5)
BRAT(45.25, 40.25, .50)	=	.29337857806185349044329069800003140
BRAT(45.25, 40.25, .25)	=	.18931660667787807019263365418922239E(−7)
BRAT(45.25, 40.25, .15)	=	.19972678050801983781084463824470771E(−15)
BRAT(100.25, 201.25, .325)	=	.39598657181697412939230979816035992
BRAT(100.25, 201.25, .33)	=	.46864593257685334023219947633433660
BRAT(201.25, 100.25, .665)	=	.45800835607995502969459981579906802
BRAT(201.25, 100.25, .66)	=	.38641541908994611668339046999369105

19. (185−256)−Page 89

    Call **DBRAT**(a, b, x, y, w, w1, IERR)−DOUBLE PRECISION BRATIO.

    Uses same arguments as BRATIO.

20. (257−274)−Page 92

    Call **BSSLJ**(z, n, w)−J BESSEL FUNCTION COMPLEX ARGUMENT, INTEGER ORDER..

    z= (0, 1), (5, 3), (10, −17.1), n = 0, 5, 10.

    z =(0, 1)

    w(0) =(1.2660658777520083355982446252147175, 0)

    w(5) =(0, .27146315595697187518107390515377734E(−3))

    w(10)=(−.27529480398368736252357102010027635E(−9), 0)

    z =(5, 3)

    w(0) =(−.82671066540395971142023564909882222, 3.2432199411359523046181066510440041)

    w(5) =(.22330911878482278054181870553277152, .98508733694458286449225062826498287)

    w(10)=(−.88641555041832623396338227470429441E(−4), −.86353649183906788526302676193032605E(−2)) z =(10, 17.1)

w(0) =(−2293359.1837843503944023401281591159, −727820.96263506654809560905605979146)

w(5) =(−823249.19677847131208538619627261896, 1112946.0824808045050358986659018367)

w(10)=(−2898.9586233545119417282522764953009, 263560.38741949811021174069631545538)

21. (275−292)

    Call **DBSLJ**(z, n, w)−J BESSEL FUNCTION COMPLEX ARGUMENT, INTEGER ORDER..

    Double precision version of BSSLJ obtained from FCH. Uses same arguments.

22. (293−310)−Page 92

    Call **CBSSLJ**(z, ν, w)−J BESSEL FUNCTION COMPLEX ARGUMENT AND ORDER..

    $\nu_i = (1, 1), (5, 3), (10, -17.25);\quad z_j = (0, 1), (5, 3), (10, -17.1)$

    $w_{i,j} = w(\nu_i, z_j);\quad i, j = 1, \ldots, 3.$

    $w_{1,1}$ = (.14642708861596860306081675249533466, .52737913663990901653557434964427650E(−1))

    $w_{2,1}$ = (.46218361676493820924597648673675168E(−5), .25456792856760833967909202786165000E(−5))

    $w_{3,1}$ = (−5575074.6647956050067554961459728768, −5357286.9085164482330012058102014785)

    $w_{1,2}$ = (−.83227019009068576300571777487790209, −.11619615843724546691699138790274513)

    $w_{2,2}$ = (.24429431173142058417192829176498112, −.44252772655967690910905720945575648E(−1))

    $w_{3,2}$ =(7083286.9137785574760158909682753024, 4045803.5624379514501179491119759740)

    $w_{1,3}$ = (−2931371.2484970823912168697786184327, 10881912.336502746795574579146898874)

    $w_{2,3}$ =(17579849.767255547205734221645520626, 125607208.16444575961851583496229498)

    $w_{3,3}$ = (.14990141503715764888724742102752180, .62701690587612355833398868063692519E(−1))

23. (311−328)−Page 91

    Call **DBSSLJ**(Z, NU, W)−DOUBLE PRECISION VERSION OF CBSSLJ .

    Z = (0, 1), (5, 3), (10, −17.1), NU = (1, 1), (5, 3), (10, −17.25).

24. (329−346)−Page 99

    Call **BSSLK**(MO, z, n, w)−K BESSEL FUNCTION COMPLEX ARGUMENT, INTEGER ORDER.

    Same arguments for n as BSSLJ, z = (0, 4), (5, 3), (10, −17.1), MO = 0.

    z =(0, 4)

    w(0) = (.26610451105001945410223079731868354E(−1), .62384146252142305380166549056420011)

    w(5) = (−.20748123413740288540578776799012713, −1.2501204889606838069891853734751489) w(10) = (−280.12098217552229101545025349837889, .30636898683602146453207605247735934E(−3))

z =(5, 3)

w(0) = (−.34121248887629061450927984359655133E(−2), .40776998107085526974196201704735308E(−3))

w(5) = (−.10382903044547709062774499608837364E(−1), .16466569738005823397449971863723252E(−1))

w(10) = (2.5529708335454787244882116901948488, .23322059884261042141739974174297953)

z =(10, −17.1)

w(0) = (.42104750178273550442610856630426844E(−5), −.12027131043135942524217995028976991E(−4))

w(5) = (.13496231935937735062830185967188919E(−4), −.11451721272657599924162664202357980E(−4))

w(10) = (.31396305911482406685020468010650756E(−4), .37919833264167302402012353819984313E(−4))

25. (347−364)

Call **DBSSLK**(MO, z, n, w)−K BESSEL FUNCTION COMPLEX ARGUMENT, INTEGER ORDER.

Double precision version of BSSLK obtained from FCH. Uses same arguments.

26. (365−382)−Page 99

Call **CBESK**(z, ν , w)−K BESSEL FUNCTION COMPLEX ARGUMENT AND COMPLEX ORDER.

Uses same arguments as CBSSLJ− $w_{i,j} = w(\nu_i, z_j)$.

$w_{1,1}$ = (.40647504293828635018310643245122930, −2.5249418710952837862596837388948793)

$w_{2,1}$ = (8075.4464054492489165942412865748102, −15508.059687733317145710512692794138)

$w_{3,1}$ = (−.31358215995107745014533341187714286E(−8), −.97567321129228537792167706290504644E(−9))

$w_{1,2}$ = (−.37126892365307062099384875555422001E(−2), −.77069030224451568902197301642716214E(−4))

$w_{2,2}$ = (−.88484788284610544522205958222540880E(−2), −.29005444533438071634841325572526054E(−1))

$w_{3,2}$ = (.47411031953562064823534039929015534E(−8), −.94186612231164769089342007910597252E(−9))

$w_{1,3}$ = (.43341207234568979066457083646363663E(−5), −.11417790579579744437339079985538951E(−4))

$w_{2,3}$ = (.72716607990176812148274009930771236E(−5), −.39571763205722002654735170788575282E(−5))

$w_{3,3}$ = (−.11431483104556780068789373397193914E(−2), −.86272020002795326178608532104303599E(−4))

27. (383−400)−Page 98

Call **DBESK**(Z, NU, W)−DOUBLE PRECISION VERSION OF CBESK.

Uses same arguments as CBESK.

28. (401−404)−Page 151

x = **ZEROIN**(F, a, b, AERR, RERR)−ZEROES OF CONTINUOUS FUNCTIONS.

F(ν , r) = I(ν , r) − K(ν , r) I, K BESSEL FCNS WITH ν , r REAL. Find ν = x such that $F(x,r) = 0$ at $r = 1.25,\ 3.5,\ 5.,10.5$ with AERR = RERR = 10SPMPAR(1), $a = 0,\ b = 20$.

$$\begin{aligned}
F(x,\,1.25) &= 0, \quad \nu = x = 1.35695776014489693928788620754022 74\\
F(x,\,3.5) &= 0, \quad \nu = x = 4.78595560261083083809031984115866 56\\
F(x,\,5.0) &= 0, \quad \nu = x = 7.05476622941454263817579893987287 54\\
F(x,\,10.5) &= 0, \quad \nu = x = 15.3602234276808937891169040530968 03
\end{aligned}$$

29. (405–408)–Page 151

X = **DZERO**(F, a, b, AERR, RERR)–DOUBLE PRECISION VERSION OF ZEROIN. Same input arguments as ZEROIN except AERR=RERR=100 DPMPAR(1), Double precision: DBSSLI and DBSSLK.

30. (409–502)–Page 157

Call **DRPOLY**(A, n, ZR, ZI, NUM, WK, IWK)–DOUBLE PRECISION ROOTS OF POLYNOMIALS.

ZR(J&(J+1)) $\Rightarrow$ ZR(J + 1) = ZR(J), ZI(J + 1) = –ZI(J). ZI = 0, if not listed.

$P1 = 3x^4 - 16x^3 - 3x^2 + 13x + 16$

$A(1) = 3,\ A(2) = -16,\ A(3) = -3,\ A(4) = 13,\ A(5) = 16.$

$$\begin{aligned}
ZR(1\&2) &= -.662358978622373012980454427239048 67,\\
ZI(1\&2) &= \pm.562279512062301243899182144909373 06,\\
ZR(3) &= 1.32471795724474602596090885447809 73, \quad ZR(4) = 16/3.
\end{aligned}$$

$P2 = x^3 - 111x^2 + 1130x - 3000$

$A(1) = 1,\ A(2) = -111,\ A(3) = 1130,\ A(4) = -3000$

$ZR(1) = 5,\ ZR(2) = 6,\ ZR(3) = 100.$

$P3 = x^{20} + 10^3x^{19} + 10^5x^{18} + 10^7x^{17} + 10^8x^{16} + 10^{10}x^{15} + 10^{11}x^{14} + 10^{12}x^{13} + 10^{13}x^{12} + 10^{14}x^{11} + 10^{16}x^{10} + 10^{17}x^9 + 10^{17}x^8 + 10^{18}x^7 + 10^{19}x^6 + 10^{19}x^5 + 10^{19}x^4 + 10^{20}x^3 + 10^{20}x^2 + 10^{19}x + 10^{19}$

$$\begin{aligned}
ZR(1) &= -901.231012995215875932266184282919 90\\
ZR(2\&3) &= -49.8178965341880883285324369040581 18\\
ZI(2\&3) &= \pm 87.2351772340392354328080022052275 38\\
ZR(4) &= -13.0644440762489522025318094033990 11\\
ZR(5) &= -10.2123532452303662245314150019886 74\\
ZR(6\&7) &= -5.61464409523763967140471974553667 75\\
ZI(6\&7) &= \pm 16.3455953248019594953792049630559 60\\
ZR(8) &= -4.38967155207497707383081644161093 63
\end{aligned}$$

ZR(9)	=	−2.2712434624302353069288227177846490
ZR(10)	=	−1.1133698400337566260605048816290935
ZR(11&12)	=	−.42765931463811996386647708443938082E(−4)
ZI(11&12)	=	±.31768590601815950904846274988714440
ZR(13&14)	=	+1.1266346074981447400220207122339536
ZI(13&14)	=	±1.9262965672364648702480715208645235
ZR(15&16)	=	+2.2576700255688602943167522772773882
ZI(15&16)	=	±3.8726756637371263811836888992538821
ZR(17&18)	=	+5.8633008008649406000116272590630581
ZI(17&18)	=	±31.270717298930054322838041312035918
ZR(19&20)	=	+12.326025547042327860657919413394970
ZI(19&20)	=	±9.5493492083624067302039056516068169

$P4 = \Pi_{k=1}^{n}(x-k) = \sum_{j=0}^{n} A(j+1)\,x^{n-j},\ n = 20.$

$A(1) = 1,\ A(2) = -210,\ A(3) = 20615,\ A(4) = -1256850,\ A(5) = 53327946,$

$A(6) = -1672280820,\ A(7) = 40171771630,\ A(8) = -756111184500,\ A(9) = 11310276995381,$

$A(10) = -135585182899530,\ A(11) = 1307535010540395,\ A(12) = -10142299865511450,$

$A(13) = 63030812099294896,\ A(14) = -311333643161390640,\ A(15) = 1206647803780373360,$

$A(16) = -3599979517947607200,\ A(17) = 8037811822645051776,$

$A(18) = -12870931245150988800,\ A(19) = 13803759753640704000,$

$A(20) = -8752948036761600000,\ A(21) = 2432902008176640000$

$ZR(k) = k,\ ZI(k) = 0, \quad k = 1, \ldots, 20.$

31. (503−514)−Page 157

Call **DCPOLY**(AR, AI, n, ZR, ZI, NUM, DWK)−DRPOLY, EXCEPT COEFFICIENTS ARE COMPLEX.

$PC1 = (Z + A)^6,\ A = 1 + 7.3\,I.$

$AR(1) = 1,\ AR(2) = 6,\ AI(2) = 43.8,\ AR(3) = -784.35,\ AI(3) = 219,$

$AR(4) = -3177.4,\ AI(4) = -7342.34,\ AR(5) = 37816.2615,\ AI(5) = -22903.020,$

$AR(6) = 82003.3230,\ AI(6) = 101262.27558,\ AR(7) = -109535.214789,\ AI(7) = 116647.75558$

$ZR(k) = -1,\ ZI(k) = -7.3, \quad k = 1, \ldots, 6.$

32. (515−524)−Page 175

Call **SAXPY**(n, a, X, kx, Y, ky)−VECTOR ADDITION.

$n = 5,\ a = 100,\ kx = 2,\ ky = -2,\ X = (1,0,2,0,3,0,4,0,5,0),\ Y = (11,0,10,9,0,8,0,7,0)$

Output : $Y = (511,0,410,0,309,0,208,0,107,0)$

33. (525–534)

Call **DAXPY**(n, a, X, kx, Y, ky)–VECTOR ADDITION.

Double precision version of SAXPY obtained from FCH. Uses same arguments.

34. (535–634)–Page 216

Call **KROUT**(MO, n, m, A, ka, B, kb, IERR, INDEX, TEMP)–INVERT MATRIX

$\mathrm{MO} = 0,\ \mathrm{n} = 10,\ \mathrm{m} = -1,\ \mathrm{A} =$ Hilbert matrix of order 10, $\mathrm{ka} = 10$.

Inverse stored in A. Elements of inverse given in APPENDIX A.

35. (635–734)

Call **DKROUT**(MO, n, m, A, ka, B, kb, IERR, INDEX, TEMP)–INVERT MATRIX.

Double precision version of KROUT obtained from FCH. Uses same arguments.

36. (735–835)–Page 217

Call **MSLV**(MO, n, m, A, ka, B, kb, D, RCOND, IERR, IWK, WK).

Same arguments as KROUT. Output: Inverse of A stored in A.

$\mathrm{D} =$ Determinant of $\mathrm{A} = .21641792264314918690605949836507259\mathrm{E}(-52)$

37. (836–936)–Page 217

Call **DMSLV**(MO, n, m, A, ka, B, kb, D, RCOND, IERR, IWK, WK)

Double precision version of MSLV. Same arguments as KROUT. Output: Inverse of A stored in A.

38. (937–942)–Page 221

Call **SLVMP**(MO, n, A, ka, b, X, WK, IWK, IND)–SOLUTION OF REAL EQUATIONS WITH ITERATIVE IMPROVEMENT.

$\mathrm{MO} = 0,\ \mathrm{n} = 3,\ \mathrm{ka} = 3.$ Call **SLVMP** again with $\mathrm{MO} = 1$.

SLVMP solves $AX = b = b_1$ with $\mathrm{MO} = 0$. Then another call is made to SLVMP with MO=1 and it solves $AX = b = b_2$.

$$\mathrm{A} = \begin{pmatrix} 1 & 5 & 2 \\ 6 & 3 & 7 \\ 4 & 8 & 5 \end{pmatrix} \qquad b_1 = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad b_2 = \begin{pmatrix} \pi \\ \exp(-1) \\ \ln(2) \end{pmatrix}$$

X (with b_1) $= 4/3,\ 1/3,\ -1$

X (with b_2) $= -5.3340450253088614142013505296964594,$

$-.18671846323305651879169987951202407,\ 4.7046149975319686233112466552680414$

39. (943–948)

Call **DSLVMP**(MO, n, A, ka, b, X, WK, IWK, IND)

SOLUTION OF REAL EQUATIONS WITH ITERATIVE IMPROVEMENT.

Double precision version of SLVMP obtained from FCH. Uses same arguments.

40. (949–983)–Page 353

Call **EIGV**(IBAL, A, ka, n, WR, WI, ZR, ZI, IERR)

FINDS EIGENVALUES AND EIGENVECTORS OF A MATRIX.

IBAL=1, Input matrix A, ka=No. of rows of A, n=order of A, (ZR(K,J), ZI(K,J))=the Kth eigenvector of the Jth eigenvalue of A. Output requires IERR=0. Let H(5) denote the Hilbert matrix of order 5 (n=5). First the eigenvalues of $A = H(5) \times H(5)$ are given. Then the eigenvalues and eigenvectors for A=H(5) are listed.

$A = H(5) \times H(5)$

$WR(1) = .10810475610875574307108389406252299E(-10)$

$WR(2) = .93573610968340105856166349581076979E(-7)$

$WR(3) = .13013086513839305374962217453885253E(-3)$

$WR(4) = .43486520331705900754379338104236462E(-1)$

$WR(5) = 2.4556478684714427530446185082147609$

$A = H(5)$

$WR(1) = .32879287721718629571150047605447314E(-5)$

$ZR(1,1) = .12197366118335218941776511729954141E(-1)$

$ZR(2,1) = -.23054351082321566090721878767252246,\ \ ZR(3,1) = 1$

$ZR(4,1) = -1.5156978981503881831443663648351046$

$ZR(5,1) = .74332785300803664576594824839796720$

$WR(2) = .30589804015119172687949784069272283E(-3)$

$ZR(1,2) = .70670226210875245446950011934428519E(-1)$

$ZR(2,2) = -.64833602593662613024121970477768598,\ \ ZR(3,2) = 1$

$ZR(4,2) = .34917863233062409987909081721800737$

$ZR(5,2) = -.83554293387428295255819834635554228$

$WR(3) = .11407491623419806559451458866589345E(-1)$

$ZR(1,3) = -1.7783959618547884961815728224945915$

$ZR(2,3) = 6.0114771493213747655493913824738878,\ \ ZR(3,3) = 1$

$ZR(4,3) = -2.5700750831853313392911158332983055$

$ZR(5,3) = -4.6922210619044074735319696492611082$

WR(4) = .20853421861101333590500251006882006

ZR(1,4) = −1.4165794178621925206014926444904337

ZR(2,4) = .64939656106525325387143805590856927, ZR(3,4) = 1

ZR(4,4) = 1.0447810384175732744990567015288115

ZR(5,4) = 1.0097363119375138848812509913358092

WR(5) = 1.5670506910982307955330110055207246

ZR(1,5) = 2.3877691630491462137700003189820182

ZR(2,5) = 1.3862597946271820256958630223217197, ZR(3,5) = 1

ZR(4,5) = .78810960688381844435363097568288750

ZR(5,5) = .65247760861091767322074653366065188

41. (984−1018)−Page 357

Call **DEIGV**(IBAL, A, ka, n, WR, WI, ZR, ZI, IERR)−FINDS DOUBLE PRECISION EIGENVALUES AND EIGENVECTORS OF A MATRIX.

Same input as EIGV.

42. (1019−1026)−Page 377

Call **LLSQ**(m, n, A, ka, B, kb, L, WK, IWK, IERR)−LEAST SQUARES (LSQ) SOLUTION OF A LINEAR SYSTEM OF EQUATIONS.

Rank of input matrix A=n. L=No. of LSQ problems (No. of columns of B). WK and IWK=work space. Output X stored in B. Norm stored in B(n+1, J), J= 1, ... , L. Output IERR must be 0.

$$\mathrm{A} = \mathrm{A}_1 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 4 \\ 1 & 0 & 0 \\ 1 & -1 & 1 \end{pmatrix} \qquad \mathrm{B} = \begin{pmatrix} 3 \\ 10 \\ 3 \\ 9/10 \end{pmatrix} \qquad \mathrm{X} = \mathrm{X}_1 = \begin{pmatrix} 327/200 \\ 301/200 \\ 49/40 \end{pmatrix}$$

$$\mathrm{A} = \mathrm{A}_2 = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 0 \\ 1 & -1 \end{pmatrix} \qquad \mathrm{B} = \begin{pmatrix} 3 \\ 10 \\ 3 \\ 9/10 \end{pmatrix} \qquad \mathrm{X} = \mathrm{X}_2 = \begin{pmatrix} 143/50 \\ 273/100 \end{pmatrix}$$

$$A = A_3 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 10 \\ 10 & 11 & 12 \end{pmatrix} \qquad B = \begin{pmatrix} 3 \\ 10 \\ 3 \\ 9/10 \end{pmatrix} \qquad X = X_3 = \begin{pmatrix} -119/15 \\ 499/60 \\ -4/5 \end{pmatrix}$$

$\| (A_3 X_3 - B) \|_{NORM} = 46.705$

43. (1027–1034)–Page 385
Call **DLLSQ**(m, n, A, ka, B, kb, L, WK, IWK, IERR)–DOUBLE PRECISION LEAST SQUARES (LSQ) SOLUTION OF A LINEAR SYSTEM OF EQUATIONS.
Same input as LLSQ.

44. (1035–1037)–Page 409
Call **OPTF**(F, n, RERR, ITER, X, FVAL, IND, WK)
FINDS LOCAL MINIMUM OF F AT X= (X(1), X(2), ... , X(n)).
RERR=Relative accuracy of F, FVAL=F(X), ITER=No. of iterations, WK=work space.
Output: IND=1 OR 2. F obtained from user subroutine Call **F**(n, X, FVAL).
$F = F_1 = x^2 - 2x + y^2 + 2y$, Initial guess $= X(1) = -5,\ X(2) = 5$.
Output : $X(1) = 1,\ X(2) = -1,\ FVAL = -2$.
$F = F_2 = x^3 + y^2 - 3x$, Initial guess $= X(1) = X(2) = 0$.
Output : $X(1) = 1,\ X(2) = 0,\ FVAL = -2$.
$F = F_3 = 2x^2 + 2y^2 + z^2 - 2xy - 2yz - 4x + 4y - 2z + 3$,
Initial guess $= X(1) = X(2) = X(3) = 0$.
Output : $X(1) = 1,\ X(2) = 0,\ X(3) = 1,\ FVAL = 0$.

45. (1038–1040)
Call **DOPTF**(F, n, RERR, ITER, X, FVAL, IND, WK)
Double precision version of OPTF obtained from FCH. Uses same arguments.

46. (1041–1059)–Page 413
Call **SMPLX**(A, B, C, ka, m, n, IND, IB, X, z, ITER, MIT, NLE, NGE, BI, WK, IWK)
Find X when z=C·X is maximized subject to A·X($\leq, \geq, =$)B. DIM(A) $= m \times n$, DIM(B) = m, DIM(C)=n. IND = 0 $\Rightarrow$ SMPLX selects starting input, IB not used, ITER=No. of iterations, MIT=Max No. of ITER's allowed. NLE(NGE)= No. of $\leq$ ($\geq$).
Output: X=$\{x_i\}$, z. Four CASES

$$A = A_1 = \begin{pmatrix} -1 & 1 \\ 1 & 4 \\ 2 & 1 \\ 3 & -4 \end{pmatrix} \qquad B = B_1 = \begin{pmatrix} \leq 5 \\ \leq 45 \\ \leq 27 \\ \leq 24 \end{pmatrix} \qquad C = C_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$\begin{aligned} -X(1) + X(2) &\leq 5 \\ X(1) + 4X(2) &\leq 45 \\ 2X(1) + X(2) &\leq 27 \\ 3X(1) - 4X(2) &\leq 24 \end{aligned} \qquad \text{NLE} = 4, \qquad z = \text{MAX}\{X(1) + X(2)\} = 9 + 9 = 18.$$

$$A = A_2 = \begin{pmatrix} 5 & 2 & 4 \\ 1 & 1 & 2 \\ 4 & 3 & 6 \\ 1 & 1 & 1 \end{pmatrix} \qquad B = B_2 = \begin{pmatrix} \leq 20 \\ \leq 8 \\ \geq 5 \\ = 6 \end{pmatrix} \qquad C = C_2 = \begin{pmatrix} 2 \\ 1 \\ -3 \end{pmatrix}$$

$$z = \text{MAX}\{2X(1) + X(2) - 3X(3)\} = 2 \times 8/3 + 10/3 + 0 = 26/3.$$

$$A = A_3 = \begin{pmatrix} 200 & -10 & 2000 \\ 10 & 1 & 100 \\ -1 & .2 & 10 \end{pmatrix} \qquad B = B_3 = \begin{pmatrix} \leq 0 \\ \leq 40 \\ \geq 1 \end{pmatrix} \qquad C = C_3 = \begin{pmatrix} 2 \\ .1 \\ 10 \end{pmatrix}$$

$$z = \text{MAX}\{2X(1) + .1X(2) + 10X(3)\} = 2 \times 4/3 + .1 \times 80/3 + 0 = 16/3.$$

$$A = A_4 = \begin{pmatrix} 0 & 0 & 3 & 0 & 1 & 1 & 0 \\ 0 & 1 & 2 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{pmatrix} \qquad B = B_4 = \begin{pmatrix} = 2 \\ = 20 \\ = 10 \\ = 3 \end{pmatrix} \qquad C = C_4 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -3 \\ 1 \\ -1 \\ 3 \end{pmatrix}$$

$$\begin{aligned} z = \text{MAX}\{X(1) + X(2) + X(3) - 3X(4) + X(5) - X(6) + 3X(7)\} = \\ 10 + \quad 20 + \quad 0 - \quad 3 \times 0 + \quad 2 - \quad 0 + \quad 3 \times 3 \ = 41. \end{aligned}$$

47. (1060–1078)
Call **DSMPLX**(A, B, C, ka, m, n, IND, IB, X, z, ITER, MIT, NLE, NGE, BI, WK, IWK)
Double precision version of SMPLX using FCH. Uses same arguments.

48. (1079–1086)–Page 413
Call **SSPLX**(TA, ITA, JTA, B, C, m, n, IND, IB, X, z, ITER, MIT, NLE, NGE, BI, WK, IWK)

Generates the same results as SMPLX. Used when A is sparse. TA, ITA, JTA specify A^T (TRANSPOSE) in sparse format–Page295. For CASE 4 of SMPLX above

$TA = TA_4 = (1, 1, 1, 1, 1, 3, 2, 1, 1, -1, 1, 1, 1, 1, -1, 1, 1, 1)^T$

$JTA = JTA_4 = (1, 10, 2, 9, 3, 8, 9, 11, 4, 9, 5, 8, 6, 8, 10, 11, 7, 11)^T$

$ITA = ITA_4 = (1, 3, 5, 9, 11, 13, 17, 19)^T, \quad m = 11, n = 7$

$B = B_4 = (0, 0, 0, 0, 0, 0, 0, 2, 20, 10, 3)^T, \ NLE = 0, NGE = 7$

49. (1087–1094)

Call **DSSPLX**(TA, ITA, JTA, B, C, m, n, IND, IB, X, z, ITER, MIT, NLE, NGE, BI, WK, IWK)

Double precision version of SSPLX obtained by FCH. Uses same arguments.

50. (1095–1099)–Page 433

Call **CHEBY**(a, b, F, G, PHI, EPS, ITER, MIT, L, m, P, Q, ERR, IERR, WK)

MINIMAXFIT.

CHEBY is a double precision routine. $a \le X \le b$, $EPS = 10^{-4}$, $MIT = 40$, WK=work space, G=F(X), PHI=Argument of polynomials. Output: P, Q; ERR specifies relative agreement between F AND P/Q at critical points. IERR should be 0.

CASE: Fit f(X,K), THE ELLIPTIC INTEGRAL OF THE FIRST KIND (Page 107), where K=.9. PHI= X^2, F(X)=f(X,K)/X on $[a = 0,\ b = \pi/2]$. The final result $X \times P/Q$ is compared with f(X,K) at $X = \pi/10,\ \pi/5,\ \pi/4,\ 2\pi/5,\ \pi/2$. L=5 and m=6. The elliptic integral is evaluated by Call **DELLPI** –Page 109.

$f(X_1, K) = .31841393498182425425662894958775021$

$f(X_2, K) = .66408837881648915721400693335979825$

$f(X_3, K) = .85794019788551098164818816682426150$

$f(X_4, K) = 1.6019861670355365511633360018405787$

$f(X_5, K) = 2.2805491384227702046137519445555304$

51. (1100–1104)–Page 447

Call **HTRP**(n, X, Y, A, WK, IERR)–HERMITE INTERPOLATION.

n=10 for case below. x_j and the associated function and the first derivative values y_j, y_j' are stored in $X = (X(2j-1) = x_j,\ X(2j) = x_j), Y = (Y(2j-1) = y_j,\ Y(2j) = y_j')$, $j = 1, \ldots, 5$.

CASE : $y = EXP(x)$, $x_j = .1,\ .2,\ .3,\ .4,\ .5$.

$$
\begin{aligned}
y_1 &= y_1' = \mathrm{EXP}(.1) = 1.1051709180756476248117078264902467 \\
y_2 &= y_2' = \mathrm{EXP}(.2) = 1.2214027581601698339210719946396742 \\
y_3 &= y_3' = \mathrm{EXP}(.3) = 1.3498588075760031039837443133280073 \\
y_4 &= y_4' = \mathrm{EXP}(.4) = 1.4918246976412703178248529528372223 \\
y_5 &= y_5' = \mathrm{EXP}(.5) = 1.6487212707001281468486507878141636
\end{aligned}
$$

The interpolation polynomial obtained from HTRP with its coefficients stored in A is converted by using the routine PCOEFF(α,n,X,A,C,T) –Page 449, ($\alpha = 0$, T=work space) from $A(1) + \sum_{j=1}^{n-1} A(j+1)(x-x_1)\cdots(x-xj)$ to $\sum_{j=0}^{n-1} C(j+1)x^j$. The latter is used for five arguments and compared with the MAPLE values given below:

$$
\begin{aligned}
\mathrm{EXP}(.15) &= 1.1618342427282831226166202143316552 \\
\mathrm{EXP}(.25) &= 1.2840254166877414840734205680624365 \\
\mathrm{EXP}(.35) &= 1.4190675485932572482703956619398724 \\
\mathrm{EXP}(.45) &= 1.5683121854901688111795997746932154 \\
\mathrm{EXP}(.55) &= 1.7332530178673952368219167671373288
\end{aligned}
$$

52. (1105–1109)

Call **DHTRP**(n, X, Y, A, WK, IERR)–HERMITE INTERPOLATION.

Double precision version of HTRP obtained from FCH. Uses same arguments.

53. (1110–1127)–Page 451

Call **PFIT**(n, m, X, Y, A, RNORM, PHI, WK, IERR)–LSQ POLYNOMIAL FIT.

Given $(x_i,\ y_i)$, $i = 1, \ldots, m$, PFIT fits a unique polynomial of degree n by minimizing $\mathrm{RNORM}^2 = \sum_{i=1}^{m}[P(x_i) - y_i]^2$. The polynomial coefficients are stored in A with $P(X) = \sum_{j=0}^{n} A(j+1)X^j$. PHI and WK are work spaces.

CASE : $n = 6$, $m = 10$, $y = \mathrm{LOG}(x)$, $X(i) = i/2$, $Y(i) = \mathrm{LOG}(X(i))$, $i = 1, \ldots, m$

$$
\begin{aligned}
X(1) &= .5, & Y(1) &= -.69314718055994530941723212145817657 \\
X(2) &= 1.0, & Y(2) &= 0.0 \\
X(3) &= 1.5, & Y(3) &= .40546510810816438197801311546434914 \\
X(4) &= 2.0, & Y(4) &= .69314718055994530941723212145817657 \\
X(5) &= 2.5, & Y(5) &= .91629073187415506518352721176801107 \\
X(6) &= 3.0, & Y(6) &= 1.0986122886681096913952452369225257 \\
X(7) &= 3.5, & Y(7) &= 1.2527629684953679956881206219850032 \\
X(8) &= 4.0, & Y(8) &= 1.3862943611198906188344642429163531 \\
X(9) &= 4.5, & Y(9) &= 1.5040773967762740733732583523868748 \\
X(10) &= 5.0, & Y(10) &= 1.6094379124341003746007593332261876
\end{aligned}
$$

$$
\begin{aligned}
A(1) &= -1.9800785652725251593569509944240339\\
A(2) &= 3.4868393335634220135011658377975746\\
A(3) &= -2.2320116785443095629139842118019607\\
A(4) &= .91658519020745719115335408583374835\\
A(5) &= -.21880135089709283048347902999887716\\
A(6) &= .27681436863144786664714633861698448E(-1)\\
A(7) &= -.14317677628543903352379990489323698E(-2)
\end{aligned}
$$

RNORM = .40491051629081667615526368118077578E(−2)

54. (1128−1145)

Call **DPFIT**(n, m, X, Y, A, RNORM, PHI, WK, IERR)

Double precision version of PFIT obtained from FCH. Uses same arguments.

55. (1146−1179)−Page 455

Call **CBSPL**(X, Y, A, B, C, n, I, J, W, V, IERR)−CUBIC SPLINE.

Input values: $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_n)$. $I = J = 2$, $W = V = 0$.

Output: Arrays A, B, C which contain the coefficients of the spline $F_i(x)$ where $F_i(x) = y_i + a_i(x - x_i) + b_i(x - x_i)^2 + c_i(x - x_i)^3$, $i = 1, \dots, n-1$.

The Maple result gives $F_i(x) = \sum_{j=0}^{3} d_{i,j}\, x^j$. Hence for comparison one has

$$
\begin{aligned}
d_{i,0} &= ((-c_i\, x_i + b_i)\, x_i - a_i)\, x_i + y_i\\
d_{i,1} &= (3\, c_i\, x_i - 2\, b_i)\, x_i + a_i\\
d_{i,2} &= b_i - 3\, c_i\, x_i\\
d_{i,3} &= c_i
\end{aligned}
$$

Maple results for fitting y=f(x) on [0, 2], where $f(x) = \sinh(x)/H - x^3/12$, $H = \sinh(2) = 3.6268604078470187676682139828012617$, $n = 9$, are :

$$
\begin{aligned}
y_1, &= y(x_1 = 0.00), &&= 0\\
y_2, &= y(x_2 = .250), &&= .68348327325323456517151982082545372E(-1)\\
y_3, &= y(x_3 = .500), &&= .13326002526399426077516293656568618\\
y_4, &= y(x_4 = .750), &&= .19157338375063409925452199590007833\\
y_5, &= y(x_5 = 1.00), &&= .24069380349860936645415534327974183\\
y_6, &= y(x_6 = 1.25), &&= .27892161687257251424544092568240252\\
y_7, &= y(x_7 = 1.50), &&= .30583613391569788177127850212405712\\
y_8 &= y(x_8, = 1.75), &&= .32276004168403583070008267835042497\\
y_9, &= y(x_9 = 2.00), &&= .33333333333333333333333333333333333
\end{aligned}
$$

$$
\begin{aligned}
d_{1,0} &= 0 \\
d_{1,1} &= .27571464554407011163764157784986716 \\
d_{1,2} &= 0 \\
d_{1,3} &= -.37141379884420569104538392314970775E(-1), \quad x < .25 \\
d_{2,0} &= -.45374977511776094409446680123941891E(-4) \\
d_{2,1} &= .27625914527421142477055493801135446 \\
d_{2,2} &= -.21779989205652525316534406459492108E(-2) \\
d_{2,3} &= -.34237381323666899062333804787038494E(-1)), \quad x < .5 \\
d_{3,0} &= -.79273026732965099034843331664170571E(-3) \\
d_{3,1} &= .28074327701311867414618885783046105 \\
d_{3,2} &= -.11146262398379751282921280284162377E(-1) \\
d_{3,3} &= -.28258539005123899894821911694896383E(-1), \quad x < .75 \\
d_{4,0} &= -.47903004880775018439174234311353492E(-2) \\
d_{4,1} &= .29673355789611007756046481828843562 \\
d_{4,2} &= -.32466636909034955835289227561461808E(-1) \\
d_{4,3} &= -.18782817000388253427102824016096636E(-1), \quad x < 1. \\
d_{5,0} &= -.18184195832899588642442061780969739E(-1) \\
d_{5,1} &= .33691524393057633795603873333793879 \\
d_{5,2} &= -.72648322943501216230863142610964978E(-1) \\
d_{5,3} &= -.53889216555661666285781856662622460E(-2), \quad x < 1.25 \\
d_{6,0} &= -.54909328726213634961930986334089364E(-1) \\
d_{6,1} &= .42505556287453004912281215226542589 \\
d_{6,2} &= -.14316057809866418516428187775295466 \\
d_{6,3} &= .13414346385810625087000143704935002E(-1), \quad x < 1.5 \\
d_{7,0} &= -.13243024816007326936780977901232400 \\
d_{7,1} &= .58009740174224931793456973762189515 \\
d_{7,2} &= -.24652180401047703103878693465726750 \\
d_{7,3} &= .36383507699546813059112378572560075E(-1), \quad x < 1.75 \\
d_{8,0} &= -.33409108003353744375636485161886800 \\
d_{8,1} &= .92580168495390218831494986209025630 \\
d_{8,2} &= -.44406710870285009982757557721061673 \\
d_{8,3} &= .74011184783808349971262596201769454E(-1), \quad x \geq 1.75
\end{aligned}
$$

A comparison is also made between the generated spline F_i and f(x) at x=.1 and 1.6. Maple gives:

$$\begin{aligned} F_1(.1) &= .27534323174522590594659619392671745E(-1) \\ f(.1) &= .27534699553492088299860036105729450E(-1) \\ F_7(1.6) &= .31365662389804818615833155109330953 \\ f(1.6) &= .31365960437678305898304387504099094 \end{aligned}$$

56. (1180–1213)
Call **DCBSPL**(X, Y, A, B, C, n, I, J, W, V, IERR)–CUBIC SPLINE.
Double precision version of CBSPL obtained from FCH. Uses same arguments.

57. (1214–1214)–Page 465
IntF = **CSINT**(X, Y, A, B, C, N, α, β)–INTEGRAL OF A SPLINE .
X, Y, A, B, C, N, f(x), F(x) are given above in CBSPL. Considering the integral of f(x)(F(x)) from 0 to 2 requires $\alpha = 0$, $\beta = 2$. It is denoted by Intf(0,2)(IntF(0,2)). Thus Intf(0,2)= (COSH(2) − 1)/H − 1/3. A comparison is made with

$$\begin{aligned} \text{Intf}(0,2) &= .42826082262243155478612494927146026 \text{ True value} \\ \text{IntF}(0,2) &= .42825482879644533223662485594509265 \text{ Maple spline result.} \end{aligned}$$

58. (1215–1215)
DIntF = **DCSINT**(X, Y, A, B, C, N, α, β)–INTEGRAL OF A SPLINE .
Double precision version of CSINT obtained from FCH. Uses same arguments.

59. (1216–1221)–Page 525
Call **SFVAL**(n, X, Y, Z, m, XI, YI, ZI, IADJ, IEND, DZ, IERR)
SURFACE INTERPOLATION.
Given surface points (x_i, y_i), i = 1, … , n, stored in X and Y; a triangular mesh is generated by TRMESH(n,X,Y,IADJ,IEND,IERR)–Page 523 which stores the results in IADJ and IEND. Let f(x,y) specify a surface. The numerical partial derivatives $F_x(x_i, y_i)$ and $F_y(x_i, y_i)$ required by SFVAL, are generated by GRADL(n,X,Y,Z,IADJ, IEND,DZ,IERR)–Page 524. Its results are stored in DZ(k,n), where the first row (k=1) contains the partials with respect to x and the second row (k=2) the partials with respect to y. The interpolation is to be carried out at the points $(\bar{x}_j, \bar{y}_j)$, j = 1, … , m stored in XI AND YI. The interpolated values of f, $F(\bar{x}_j, \bar{y}_j)$ are stored in ZI.
Let f(x, y)=Circular coverage function, evaluated by Call **CIRCV**(R, D, I) – Page125 with $I = 1; x_i = R,\ y_i = D;\ R, D = (i-1)/8,\ i = 1, \ldots, n = 80$. (6400 points)

XI = (.0625, 7.5625, 1.0625, 2.0625, 5.0625, 8.0625), YI = (0, 9.0625, 1.0625, 2.0625, 4.0625, 3.0625). The values to compare are given by:

f(.0625, 0) = .19512188925245272899573409173264047E(−2)
F(.0625, 0) = .199726939136652E(−2)
f(7.5625, 9.0625) = .59324966894443808464458766563315521E(−1)
F(7.5625, 9.0626) = .593298922404450E(−1)
f(1.0625, 1.0625) = .28253832617966374058497114901082367
F(1.0625, 1.0625) = .282556449740250
f(2.0625, 2.0625) = .39990590720072733397670102612204638
F(2.0625, 2.0625) = .399917529883358
f(5.0625, 4.0625) = .81301989279372316483739066020196562
F(5.0625, 4.0625) = .812995939477743
f(8.0625, 3.0625) = .99999952719147575100918544926929062
F(8.0625, 3.0625) = .999999539276169

60. (1222−1227)
Call **DSFVAL**(n, X, Y, Z, m, XI, YI, ZI, IADJ, IEND, DZ, IERR)
SURFACE INTERPOLATION.
Double precision version of SFVAL obtained from FCH. Uses same arguments.

61. (1228−1233)−Page 531
Call **QAGS**(f, a, b, AERR, RERR, z, ERROR, NUM, IERR, L, m, n, IWK, WK)
EVALUATION OF INTEGRALS.
z contains the output value for the estimate of the integral of f from a to b. ERROR=the estimated absolute error in z; NUM = no. of points at which f was evaluated. IERR should be zero. IWK(L) and WK(m) are work spaces. n= no. of subintervals used. CASE: Circular coverage function tested with the input taken from XI and YI of SFVAL above. AERR=RERR=50 SPMPAR(1). Uses BESI−Page 97.

62. (1234−1239)−Page 534
Call **DQAGS**(F, a, b, AERR, RERR, z, ERROR, NUM, IERR, L, m, n, IWK, WK)
EVALUATION OF INTEGRALS.
Double precision version of QAGS with same CASE. AERR=RERR=50 DPMPAR(1). Uses double precision version of BESI obtained from FCH.

63. (1240−1245)−Page 533
x = **QSUBA**(F,a,b,RERR,MCOUNT,ERROR,IERR)

EVALUATION OF INTEGRALS .

Variables that also appear in QAGS have the same meaning, except for ERROR which specifies the relative error here and RERR = max[50SPMPAR(1), 1E(−14)]. Input MCOUNT specifies the maximum no. of times F is to be evaluated. CASE: Same as used above in QAGS.

64. (1246−1251)

X = **DQSUBA**(F, a, b, RERR, MCOUNT, ERROR, IERR)

EVALUATION OF INTEGRALS.

Double precision version of QSUBA obtained from FCH. Uses same arguments. Because of the limited accuracy of the modified Bessel function I_0 which appears in F, RERR=max[50DPMPAR(1), 1E-14].

65. (1252−1252)−Page 539

Call **QAGI**(F, a, MO, AERR, RERR, z, ERROR, NUM, IERR, L, m, n, IWK, WK)

EVALUATION OF INTEGRALS OVER INFINITE INTERVALS.

For MO= −1, $z = \int_{-\infty}^{a} F(x)dx$; for MO = 1, $z = \int_{a}^{\infty} F(x)dx$; for MO = 2, $z = \int_{-\infty}^{\infty} F(x)dx$.

Except for a, variables that also appear in QAGS have the same meaning.

CASE: With $b = \sqrt{8}$,

$$z \simeq \mathrm{ZTRUE} = \int_0^\infty e^{-t^2} Y_{2/3}(b\,t)dt = -\frac{\sqrt{\pi}}{2e}\left[\tan(\pi/3)I_{1/3}(1) + \frac{1}{\pi\cos(\pi/3)}K_{1/3}(1)\right],$$

see [3, Page 487, Eq. (11.4.30)]. Hence a=0, MO = 1, AERR=RERR=100 SPMPAR(1). F gives the integrand. Uses Bessel function Y by Call **CBSSLY**−page 95.

ZTRUE = −.69218599490499706343637364910052890 (From MAPLE).

66. (1253−1253)−Page 540

Call **DQAGI**(F, a, MO, AERR, RERR, z, ERROR, NUM, IERR, L, m, n, IWK, WK)

Double Precision version of QAGI; same input is used with DBSSLY.

67. (1254−1263)−Page 551

Call **ODE**(F, n, Y, T, TOUT, RERR, AERR, IND, WK, IWK)

ADAPTIVE ADAMS DE INITIAL VALUE SOLVER.

ODE solves the set of n DE's $y'(t) = f(t, y(t))$ numerically to obtain $y(t)$ on T=t=a to TOUT=t=b, b > a, with a set of given initial conditions y(a). Y contains $(y_1(t), \ldots, y_n(t))$. F computes $(y_1'(t), \ldots, y_n'(t))$ using the DE's and subroutine F(t,Y,DY). Two CASES are carried out. RERR = AERR = 10 SPMPAR(1), IND = −1.

CASE1: $\bar{y}\,\bar{y}'' - (\bar{y}')^2 - \bar{y}^2\log(\bar{y}) = 0$; $\bar{y}(0) = \bar{y}'(0) = 1$; DE $\Rightarrow \bar{y}''(0) = 1$, [4, p. 571, 6.113]. Hence for $y_1(0) = Y(1) = 1$, $y_1'(0) = y_2(0) = Y(2) = 1$, $DY(2) = Y(2)^2/Y(1) + Y(1)\log(Y(1)$. The solution of the DE's is $\bar{y}(t) = e^{\sinh(t)}$, $\bar{y}'(t) = \bar{y}(t)\cosh(t)$.

Output(1): With $a = 0$, $b = 1$:

$$Y(1) = \bar{y}(1) = e^{\sinh(1)} = 3.2387945031585805075005735281910145$$
$$Y(2) = \bar{y}'(1) = Y(1)\cosh(1) = 4.9977210779700644805224989588330930.$$

Output(2): With $a = 0$, $b = 2$:

$$Y(1) = \bar{y}(2) = e^{\sinh(2)} = 37.594599428049235605895004431737778 2$$
$$Y(2) = \bar{y}'(2) = Y(1)\cosh(2) = 141.43823997622198995416647345901998.$$

CASE2: $((\bar{y}')^2 + 1)\,\bar{y}''' - 3\,\bar{y}'\,(\bar{y}'')^2 = 0$; $\bar{y}(1) = 3$, $\bar{y}'(1) = 0$, $\bar{y}''(1) = -1/2$;
DE $\Rightarrow \bar{y}'''(1) = 0$, [4, p.602, 7.11]. $DY(3) = 3Y(2)Y(3)^2/[1 + Y(2)^2]$.
The solution is given by $\bar{y}(t) = 1 + \sqrt{(3-t)\,(1+t)}$, $\bar{y}'(t) = (1-t)/(\bar{y}-1)$,
$\bar{y}''(t) = -1/(\bar{y}-1) - (\bar{y}')^2/(\bar{y}-1)$. $y_1' = y_2$, $y_2' = y_3$, $y_3' = f = 3\,y_2\,y_3^2/(y_2^2 + 1)$.

Output(1): With $a = 1$, $b = 2$:

$$Y(1) = y_1(2) = \bar{y}(2) = 1 + \sqrt{3} = 2.7320508075688772935274463415058724$$
$$Y(2) = y_1'(2) = \bar{y}'(2) = -1/\sqrt{(3)} = -.57735026918962576450914878050195746$$
$$Y(3) = y_1''(2) = \bar{y}''(2) = -4/(3\sqrt{3}) = -.76980035891950101934553170733594327.$$

Output(2): With $a = 1$, $b = 2.5$:

$$Y(1) = y_1(2.5) = \bar{y}(2.5) = 1 + \sqrt{7}/2 = 2.3228756555322952952508078768196302$$
$$Y(2) = y_1'(2.5) = \bar{y}'(2.5) = -3/\sqrt{7} = -1.1338934190276816816435496087025402$$
$$Y(3) = y_1''(2.5) = \bar{y}''(2.5) = -32/7^{3/2} = -1.7278375908993244672663613084991088.$$

68. (1264–1273)
Call **DODE**(F, n, Y, T, TOUT, RERR, AERR, IND, WK, IWK)
Double precision version of ODE obtained by FCH. Uses same arguments except for AERR=RERR=10 DPMPAR(1).

69. (1274–1283)–Page 563
Call **GERK**(F, n, Y, T, TOUT, RERR, AERR, IND, GERROR, WK, IWK)
ADAPTIVE RKF INITIAL VALUE DE SOLVER.
GERROR(i) is an estimate of the error at Y(i), $i = 1, \ldots, n$. The remaining variables play the same role as in ODE. The same test cases used in ODE are used with GERK.

70. (1284–1293)
Call **DGERK**(F, n, Y, T, TOUT, RERR, AERR, IND, GERROR, WK, IWK)
Double precision version of GERK obtained from FCH. Uses same arguments.

71. (1294–1301)–Page 571

Call **RK**(n, T, h, A, F)–FOURTH ORDER RUNGE-KUTTA .

Solves the initial value problem specified in ODE by using the Runge-Kutta fourth order procedure. T contains the initial value of $t = t_0$. The increment by which t is increased at each call of RK is stored in h. $A(1), \ldots, A(n)$ contain the values $y_1(t_0), \ldots, y_n(t_0)$. $A(n+1), \ldots, A(2n)$ contain the values $y_1'(t_0), \ldots, y_n'(t_0)$. When RK is called the values $y_1(t_0+h), \ldots, y_N(t_0+h)$ and $y_1'(t_0+h), \ldots, y_n'(t_0+h)$ are computed and stored in $A(1), \ldots, A(2n)$. $A(2n+1), \ldots, A(3n)$ are used as work space. $F(t, Z)$ is a user defined subroutine which computes the y_i'. Two CASES are carried out.

CASE1: $9\,\overline{y}^2\,\overline{y}''' - 45\,\overline{y}\,\overline{y}'\,\overline{y}'' + 40\,(\overline{y}')^3 = 0$; $\overline{y}(0) = 1$, $\overline{y}'(0) = -3/2$, $\overline{y}''(0) = 0$;
DE $\Rightarrow \overline{y}'''(0) = 15.$ [4, p. 602, 7.9].
Hence for $y_1(0) = A(1) = 1$, $y_1'(0) = y_2(0) = A(2) = -3/2$,
$y_1''(0) = y_2'(0) = A(3) = 0$, $Z(3) = [45\,Z(1)\,Z(2)\,Z(3) - 40\,Z(2)^3]/[9\,Z(1)^2]$.
The solution is: $\overline{y}(t) = 1/(1 + t + 5t^2/4)^{3/2}$,
$\overline{y}'(t) = -(3/2)\overline{y}^{5/3}(1 + 5t/2)$, $\overline{y}''(t) = 5(\overline{y}')^2/(3\overline{y}) - (15/4)\overline{y}^{5/3}$.
The integration is carried out from 0 to 2 with h=1/32768, where
$y_1(2) = \overline{y}(2) = (1/8)^{3/2}$, $y_1'(2) = \overline{y}'(2) = -9\,y_1(2)/8$, $y_1''(2) = \overline{y}''(2) = 105(1/8)^{3.5}$.

$$\begin{aligned} \text{Output}: A(1) = \overline{y}(2) &= .44194173824159220275052772631553065E(-1) \\ A(2) = A(4) = \overline{y}'(2) &= -.49718445552179122809434369210497198E(-1) \\ A(3) = A(5) = \overline{y}''(2) &= .72506066430261220763758455098641747E(-1) \end{aligned}$$

CASE2: $\overline{y}'' - t\,\overline{y} = 0$; $\overline{y}(0) = 1$, $\overline{y}'(0) = 0$, DE $\Rightarrow \overline{y}''(0) = 0$.
The solution, derived in Appendix B, is given by

$$\overline{y}(t) = \frac{\pi}{3^{1/3}\,\Gamma(1/3)}\,[\sqrt{3}\,\mathrm{Ai}(t) + \mathrm{Bi}(t)], \quad \overline{y}'(t) = \frac{\pi}{3^{1/3}\,\Gamma(1/3)}\,\frac{2\,t}{\sqrt{3}}\,I_{2/3}(\frac{2\,t^{3/2}}{3}). \tag{8}$$

Ai and Bi are Airy functions [3, p.446] evaluated from Call **AI** and Call **BI**–Page (101-102). $I_{2/3}(t)$ represents a Modified Bessel Function which can be evaluated from Call **BESI**–Page 97. Z(2)=t Z(1).

Output: $A(1) = \overline{y}(2) = 2.7308830178901459635915275691024881$
$A(2) = A(3) = \overline{y}'(2) = 3.2595163616105247767976271320619734$

72. (1302–1309)

Call **DRK**(n, T, h, A, F)–FOURTH ORDER RUNGE-KUTTA .

Double precision version of RK obtained from FCH. Uses same arguments.

73. (1310–1317)–Page 573

Call **RK8**(n, T, h, Y, DY, WK, F)–EIGHTH ORDER RUNGE-KUTTA .

Solves the initial value problems specified in RK by using the eighth order Runge-Kutta procedure. T contains the initial value of $t = t_0$. The increment by which t is increased at each call of RK8 is denoted by h. $Y(1), \dots, Y(n)$ contain the values $y_1(t_0), \dots, y_n(t_0)$. $DY(1), \dots, DY(n)$ contain the values $y'_1(t_0), \dots, y'_n(t_0)$. When RK8 is called the values $y_1(t_0 + h), \dots, y_N(t_0 + h)$ and $y'_1(t_0 + h), \dots, y'_n(t_0 + h)$ are computed and stored in $Y(1), \dots, Y(n)$, $DY(1), \dots, DY(n)$. WK is a work space, and $F(t, Z)$ is a user defined subroutine which computes the elements of DY from the differential equations. The test cases are the same as those used in RK above.

74. (1318–1325)

Call **DRK8**(n, T, h, Y, DY, WK, F)–EIGHTH ORDER RUNGE-KUTTA .

Double precision version of RK8 obtained from FCH. Uses same arguments.

III. Testing Of Bit Operations

The following intrinsic bit functions were tested:

1. BTEST 2. IBITS 3. IBCLR 4. IBSET 5. ISHFTC
6. ISHFT 7. IAND 8. MVBITS 9. IOR 10. IEOR

In case an error occurs, the error message will be listed with an identifying symbol $< KX >$, where K denotes an integer and X a letter. The symbol will also be included in a comment statement of the code to help locate the section of the code that was operating when the error occured.

Let MSIZE denote the number of bits in an integer word. Let $\text{ISIZE} = \min(\text{MSIZE}, 128)$. If word size exceeds 128 bits only the first 128 bits are tested. Let I be an integer with ISIZE alternating zero and one bits with bit zero set to one, and let J denote its complement. Tests 2, 3, and 4 are only done if ISIZE is even.

1.The following test is done twice–once with I and once with J.

For each bit position K (K= $0, \ldots$, ISIZE),
let N be the value of the bit at K, and let M be the value of the bit at K detected by BTEST.
Verify that $M = N$ and that I is unchanged.$< 1A >$

Use IBCLR to *clear* the bit at K in I. Use IBITS to check that it is cleared and I is unchanged.$< 1B >$
Also use IBITS to ascertain that the result is unchanged except at K.$< 1B >$

Test IBSET as was done with IBCLR.$< 1C >$

2. For each $L(L = 0, 1, 2, \ldots, \text{ISIZE})$
If L is odd let $M = J$
if L is even let $M = I$

for each II $(\text{II} = L, L + 1, \ldots, \text{ISIZE})$ and $\text{II} \neq 0$
let $K = \text{ISHFTC}(I, L, \text{II})$ (left circular shift)
compute K using BTEST, IBCLR, IBSET
if the two results differ print an error message$< 2A >$
let $K = \text{ISHFTC}(I, -L, \text{II})$ (right circular shift) and repeat above tests$< 2B >$

let $K = \text{ISHFTC}(M, 1, \text{ISIZE})$ and let MM = complement of M
$K \neq MM$ print error message$< 2C >$
let K =ISHFTC(M,–1,ISIZE)

if K $\neq$ MM print error message< 2D >

3. Let $\langle \text{McJ} \rangle$ denote the M corresponding bits of J

   let KI = I and let KJ = J

   for each L (L = 0, 1, ... , ISIZE − 1)

   if L $\neq$ 0 then use IBCLR to:

   clear the right most L bits of KI

   clear the left most L bits of KJ

   $K = \text{ISHFT}(\text{I1}, L) = \overbrace{11\cdots111}^{\text{ISIZE}-L}\overbrace{00\cdots000}^{L}$ I1 contains all one bits

   $K = \text{IAND}(K, I) = \overbrace{\text{xxx}\cdots\text{xxx}}^{\langle(\text{ISIZE}-L)\text{cI}\rangle}\overbrace{00\cdots000}^{L} = \text{KI}$

   if K $\neq$ KI print error message< 3A >

   $K = \text{ISHFT}(\text{I1}, -L) = \overbrace{00\cdots000}^{L}\overbrace{11\cdots111}^{\text{ISIZE}-L}$

   $K = \text{IAND}(K, J) = \overbrace{00\cdots000}^{L}\overbrace{\text{xxx}\cdots\text{xxx}}^{\langle(\text{ISIZE}-L)\text{cJ}\rangle} = \text{KJ}$

   if K $\neq$ KJ print error message< 3B >

4. For each L (L = 0, 1, ... , ISIZE − 1)

   For each N (N = 0, ... , ISIZE − L)

   K = J

   N bits are moved from I to K beginning with bit position L in both I and K

   call MVBITS(I,L,N,K,L)

   $K = \overbrace{\text{xx}\cdots\text{xxx}}^{\langle\text{McJ}\rangle}\overbrace{\text{xx}\cdots\text{xxx}}^{\langle\text{NcI}\rangle}\overbrace{\text{xx}\cdots\text{xxx}}^{\langle\text{LcJ}\rangle}, \quad M + N + L = \text{ISIZE}$

   KI = IEOR(K, I), KJ = IEOR(K, J)

   $\text{KI} = \overbrace{11\cdots111}^{M}\overbrace{00\cdots000}^{N}\overbrace{11\cdots111}^{L}$

   $\text{KJ} = \overbrace{00\cdots000}^{M}\overbrace{11\cdots111}^{N}\overbrace{00\cdots000}^{L}$

   if IEOR(KI, KJ) $\neq$ I1 (all ones) print an error message< 4A >

   if IAND(KI, KJ) $\neq$ 0 print an error message< 4B >

   KI = IOR(K, I), KJ = IOR(K, J)

   $\text{KI} = \overbrace{11\cdots111}^{M}\overbrace{\text{xx}\cdots\text{xxx}}^{\langle\text{NcI}\rangle}\overbrace{11\cdots111}^{L}$

$$KJ = \overbrace{xx\cdots xxx}^{\langle McJ\rangle}\overbrace{11\cdots 111}^{N}\overbrace{xx\cdots xxx}^{\langle LcJ\rangle}$$

if $IAND(KI, KJ) \neq K$ print an error message< 4C >

IV. Testing Of String Operations

The following intrinsic string functions were tested:

1. ACHAR	2. CHAR	3. IACHAR	4. ICHAR	5. ADJUSTL
6. ADJUSTR	7. LEN	8. LEN_TRIM	9. INDEX	10. LLT
11. LLE	12. LGE	13. LGT	14. REPEAT	15. TRIM
16. SCAN	17. VERIFY			

In case an error occurs, the error message will be listed with an identifying symbol < KX >, where K denotes an integer and X a letter. The symbol will also be included in a comment statement of the code to help locate the section of the code that was operating when the error occurred.

Let AA = 'abc$\cdots$ xyz' BB = 'ABC$\cdots$ XYZ'

CC = '1234567890'$-$ = \~! @ # $ % ^ & *() _'

ALL(I+1:I+1)=CHAR(I) $I = 0, 1, \ldots, 127$

1. For $I = 0, 1, \ldots, 127$

   C = ACHAR(I) D = CHAR(I)

   if $C \neq D$ print error message< 1A >

   J = IACHAR(D) K = ICHAR(D)

   if $J \neq I$ print error message< 1B >

   if $K \neq I$ print error message< 1C >

2. ABC = AA//BB//CC

   for $I = 0, 1, \ldots, 72$

   if $I < 72$ then

   ABCX = ABCY = ABC

   if $I \geq 1$ ABCX(1 : I) =' $\underline{b}$' $\underline{b} \equiv$ a blank

   ABCX = ADJUSTL(ABCX)

   ABCX has I blanks on the right; its length (minus trailing blanks) is $72 - I$.

   J=LEN_TRIM(ABCX) K=LEN(ABCX(1:J))

   if $J \neq 72 - I$ print error message< 2A >

   if $J \neq K$ print error message< 2B >

in ABCY move character positions I+1,... ,72 into character
positions $1, 2, \ldots, 72 - \mathrm{I}$, then blank out positions $72 - \mathrm{I} + 1, \ldots, 72$
if ABCX $\neq$ ABCY print error message< 2C >

if $\mathrm{I} > 1$, then
ABCX = ABCY = ABC
$\mathrm{ABCX(I:72)} =' \underline{b}'$
ABCX = ADJUSTR(ABCX)
in ABCY move character positions $1, 2, \ldots, \mathrm{I} - 1$ to character
positions $74 - \mathrm{I}, \ldots, 72$ and blank out positions $1, 2, \ldots, 73 - \mathrm{I}$
if (ABCX $\neq$ ABCY) print error message< 2D >

3. DD = CC DD(1:1) = '$\underline{b}$' ABC = DD//DD//DD DD(2:2) = '$\underline{b}$'
the arguments of INDEX are INDEX(string, substring, back). If back is missing it is equivalent to back = .FALSE.
when string is shorter than substring INDEX = 0
J = INDEX(DD(2 : 72), ABC)
K = INDEX(DD(2 : 72), ABC, .FALSE.) N = INDEX(DD(2 : 72), ABC, .TRUE.)
if J $\neq$ K print error message< 3A >
if J $\neq$ 0 print error message< 3B >
if N $\neq$ 0 print error message< 3C >
For each I ($\mathrm{I} = 0, 1, \ldots, 71$)
For each II ($\mathrm{II} = 1, 2, \ldots, \mathrm{I} + 1$)
J = INDEX(ABC, DD(II:I))
K = INDEX(ABC, DD(II:I), .FALSE.)
N = INDEX(ABC, DD(II:I)), .TRUE.)
if J $\neq$ K print error message< 3D >

if II > I then
DD(II:I) is empty, hence
if back = .FALSE. then INDEX= 1
if back = .TRUE. INDEX = 73.
therefore if J $\neq$ 1 print error message< 3E >
if N $\neq$ 73 print error message< 3F >

if II = 1 and I > 1 then

DD(II:I) begins with 2 blanks, hence DD(II:I) is not in ABC, INDEX = 0

if $J \neq 0$ print error message< 3G >

if $N \neq 0$; print error message< 3H >

if $II = 2$ and $I \neq 2$ then

DD(II:I) begins with '$\underline{b}$3', this substring doesn't occur in ABC, since in ABC '3' is always preceded by '2'

if $J \neq 0$ print error message< 3I >

if $N \neq 0$ print error message< 3J >

else

$II \leq I$ and $(II > 1$ or $I \leq 1)$ and $(II \neq 2$ or $I = 2)$

therefore $II = I = 1$ or $II = I = 2$ or $2 < II \leq I$

if $II = 2$ then $DD(II:I) = ' \underline{b}'$ occurs at ABC(1:1), ABC(27:27), ABC(53:53)

print error message if $J \neq 1$ < 3K >

print error message if $N \neq 53$ < 3L >

if $II \neq 2$ then DD(II:I) occurs at ABC(M:M), where $M = II$, $II + 26$, $II + 52$

print error message if $J \neq II$ < 3K >

print error message if $N \neq II + 52$. < 3L >

4. For each J $(J = 1, 2, \ldots, 128)$

For each I $(I = 1, 2, \ldots, 128)$

let II = ALL(I : I) let JJ = ALL(J : J)

if LLT(II, JJ) is TRUE and $I \geq J$ print error message< 4A >

if LLE(II, JJ) is TRUE and $I > J$ print error message< 4B >

if LGE(II, JJ) is TRUE and $I < J$ print error message< 4C >

if LGT(II, JJ) is TRUE and $I \leq J$ print error message< 4D >

5. For each I $(I = 1, 2, \ldots, 8)$

For each K $(K = 0, 1, \ldots, 8)$

DD = AA set last K characters of DD to blank $M = 26 - K$

if TRIM(DD) $\neq$ DD(1 : M) print error message< 5A >

if LEN(TRIM(DD)) $\neq$ M) print error message< 5A >

if $K > 0$ then

let ABCX = AA(1 : I)//AA(1 : I)$\cdots$ K times

let ABCY = REPEAT(AA(1 : I), K)

if ABCX $\neq$ ABCY print error message< 5B >

6. $ABC = AA//BB$

    For each I $(I = 1, 2, \ldots, 52)$

    $K = SCAN(ABC(I{:}52), AA)$

    $M = SCAN(ABC(I{:}52), AA, .FALSE.)$

    $N = SCAN(ABC(I{:}52), AA, .TRUE.)$

    $M(N)$ = the position number of the left most (right most) character in $ABC(I{:}52)$ that is in AA

    if $K \neq M$ print error message< 6A >

    if $I \leq 26$ then

    if $M \neq 1$ print error message< 6B >

    if $N \neq 26 - (I - 1)$ print error message< 6C >

    if $I > 26$ then

    if $M \neq 0$ print error message< 6D >

    if $N \neq 0$ print error message< 6E >

    $K = VERIFY(ABC(I{:}52), AA)$

    $M = VERIFY(ABC(I{:}52), AA, .FALSE.)$

    $N = VERIFY(ABC(I{:}52), AA, .TRUE.)$

    $M(N)$ = the position number of the left most (right most) character in $ABC(I{:}52)$ that is not in AA

    if $K \neq M$ print error message< 6F >

    if $N \neq 52 - (I - 1)$ print error message< 6G >

    if $I \leq 26$ then

    if $M \neq 27 - (I - 1)$ print error message< 6H >

    if $I > 26$ then

    if $M \neq 1$ print error message< 6I >

V. How To Use TEST90

TEST90 consists of four Fortran 90 source files–TEST90.F, LIB.F, LIBS.F, and LIBD.F. TEST90.F contains the driver program, the bit and character test routines, the block data routines containing the safety factors (SF) and stored epsilons (SE)[4], and some miscellaneous routines. It was successfully tested on Fortran 90 compilers of the CRAY EL98 and the IBM PC.

LIB.F contains 74 basic testing routines and some supporting software. Each is used to test one of the 74 subprograms in Section II. The supporting routines to LIB.F are in LIBS.F and LIBD.F. The routines tested in LIBS.F are all MLIB routines. The routines tested in LIBD.F are all double precision routines created by FCH.

The first routine in LIBS.F is IPMPAR. The machine constants in IPMPAR must be set for the particular compiler on the particular machine being used. Instructions are given in the in-line documentation of IPMPAR for defining the constants that are needed. If constants are not provided for the machine being used then consult the Fortran compiler manual for that machine or the appendix of [1]. These constants in LIBS.F were originally set for the 64 bit CRAY EL98.[5]

To use TEST90 one needs to compile and link the four Fortran 90 source files–TEST90.F, LIB.F, LIBS.F, and LIBD.F and run the resulting executable code.

If a routine of LIB.F reveals an error, a print out of the case involved gives the true and calculated values, the relative error EPS or EPD, and the case number $N(1 \leq N \leq 1325)$. If a routine in LIB.F encounters no errors, then 'x ok' is printed where x is the routine name.

If a bit or sting error occurs, a print out gives the intrinsic function with the input, the true and calculated values, and the error message symbol $< KX >$ [6]. If no bit errors occur then 'bit functions ok' is printed. If no string errors occur then 'string functions ok' is printed. If no errors are discovered using LIB.F and all the bit function and string function tests, then 'everything ok' is printed.

[4] Also listed in Appendix C

[5] Since none of the routines in MLIB use integers that exceed 46 bits either compiler option (46 or 64 bits) may be specified when using the CRAY EL98.

[6] See Sections III and IV.

VI. Index

An integer without parentheses in this index gives the number of the page where a callable routine is discussed. An integer in parentheses refers to the corresponding page number in [1]. The seventeen double precision directly callable routines obtained by using FCH are identified by a superscript *. The eleven callable routines used as supporting routines are marked by the superscript **.

VII. Summary

The objective of this work was to use the NSWC Mathematical Library (MLIB)[1] as an aid in testing a new Fortran 90 compiler used with the CRAY EL98 mainframe computer. One can also view the testing as an aid in establishing whether MLIB can operate in a Fortran 90 environment. The final result is a transportable driver program, TEST90, which is not restricted to testing only the new 90-compiler. Any post Fortran 77 compiler can be tested provided it is limited to a single precision word length of 64 bits or less. This limitation is imposed because the routines in MLIB are based on a 64 bit word length with a 48 bit floating point mantissa.

TEST90 has been used to test the latest Fortran 90 compilers of the CRAY EL98 and IBM PC machines; no errors were found, however TEST90 did reveal a complex arthmetic error in an earlier version of the Cray EL98 compiler. MLIB routines ran under TEST90 without any problems on both machines.

VIII. References

1. Morris, Jr, Alfred H., *NSWC LIBRARY OF MATHEMATICS SUBROUTINES,* NSWCDD/TR-92/425, Naval Surface Warfare Center, Dahlgren Division, Dahlgren, VA 22448, Jan 1993. Also CD disk to be issued.

2. Monagan, M. B., etc., *Maple V, Programming Guide,* Springer-Verlag NY., 1996.

3. Abramowitz, M., Stegun, I., *HANDBOOK OF MATHEMATICAL FUNCTIONS,* Dover Publications, N.Y., 1965.

4. Kamke, E. *DIFFERENTIALGLEICHUNGEN,* 3rd Edition, Chelsea Publishing, NY, 1959.

APPENDIX A

INVERSE OF THE HILBERT MATRIX OF ORDER 10

APPENDIX A

INVERSE OF THE HILBERT MATRIX OF ORDER 10

This appendix contains the elements of the inverse of the tenth order Hilbert matrix. The matrix is given by rows. The jth row is ended with ■ (j).

100, -4950, 79200, -600600, 2522520, -6306300, 9609600, -8751600, 4375800, -923780■ **(1)**

-4950, 326700, -5880600, 47567520, -208107900,
535134600, -832431600, 770140800, -389883780, 83140200■ **(2)**

79200, -5880600, 112907520, -951350400, 4281076800,
-11237826600, 17758540800, -16635041280, 8506555200, -1829084400■ **(3)**

-600600, 47567520, -951350400, 8245036800, -37875637800,
101001700800, -161602721280, 152907955200, -78843164400 ,17071454400■ **(4)**

2522520, -208107900, 4281076800, -37875637800, 176752976400,
-477233036280, 771285715200, -735869534400, 382086104400, -83223340200■ **(5)**

-6306300, 535134600, -11237826600, 101001700800, -477233036280,
1301544644400, -2121035716800, 2037792556800, -1064382719400, 233025352560■ **(6)**

9609600, -832431600, 17758540800, -161602721280, 771285715200,
-2121035716800, 3480673996800, -3363975014400, 1766086882560, -388375587600■ **(7)**

-8751600, 770140800, -16635041280, 152907955200, -735869534400,
2037792556800, -3363975014400, 3267861442560, -1723286307600, 380449555200■ **(8)**

4375800, -389883780, 8506555200, -78843164400, 382086104400,
-1064382719400, 1766086882560, -1723286307600, 912328045200, -202113826200■ **(9)**

-923780, 83140200, -1829084400, 17071454400, -83223340200,
233025352560, -388375587600, 380449555200, -202113826200, 44914183600■ **(10)**

APPENDIX B

DERIVATION OF EQUATION (8)

APPENDIX B

DERIVATION OF EQUATION (8)

The objective here is to derive the solution of the Airy differential equation (DE) with specified initial values, where

$$y''(t) - t\,y(t) = 0; \quad y(0) = 1, \; y'(0) = 0. \tag{B-1}$$

Linearly independent solutions of the DE in (B-1) or (8) are the Airy functions [B1, p.446] denoted by Ai(t) and Bi(t). Thus

$$y = c_1\,\mathrm{Ai}(t) + c_2\,\mathrm{Bi}(t).$$

The initial value conditions yield

$$c_1\,\mathrm{Ai}(0) + c_2\,\mathrm{Bi}(0) = 1, \qquad c_1\,\mathrm{Ai}'(0) + c_2\,\mathrm{Bi}'(0) = 0. \tag{B-2}$$

The solution of (B-2) is given by

$$c_1 = \mathrm{Bi}'(0)/W, \qquad c_2 = -\mathrm{Ai}'(0)/W,$$

where W denotes the Wronskian of (B-1),

$$W = \mathrm{Ai}(0)\,\mathrm{Bi}'(0) - \mathrm{Ai}'(0)\,\mathrm{Bi}(0).$$

From [B1, p.446], we have

$$\mathrm{Ai}(0) = \mathrm{Bi}(0)/\sqrt{3} = 3^{-2/3}/\Gamma(2/3), \quad \mathrm{Ai}'(0) = -\mathrm{Bi}'(0)/\sqrt{3} = -3^{-1/3}/\Gamma(1/3).$$

Therefore using 6.1.17 of [B1, p.256] $W = 1/\pi$; $y(t)$, $y'(t)$, after some algebra, are given by

$$y(t) = \frac{\pi}{3^{1/3}\Gamma(1/3)}[\sqrt{3}\,\mathrm{Ai}(t) + \mathrm{Bi}(t)], \quad y'(t) = \frac{\pi}{3^{1/3}\Gamma(1/3)}[\sqrt{3}\,\mathrm{Ai}'(t) + \mathrm{Bi}'(t)]. \tag{B-3}$$

The quantities Ai′(t) and Bi′(t) can be expressed in terms of Modified Bessel functions [B1, p.447] as

$$\mathrm{Ai}'(t) = \frac{t}{3}\left[I_{2/3}(\frac{2\,t^{3/2}}{3}) - I_{-2/3}(\frac{2\,t^{3/2}}{3})\right], \quad \mathrm{Bi}'(t) = \frac{t}{\sqrt{3}}\left[I_{2/3}(\frac{2\,t^{3/2}}{3}) + I_{-2/3}(\frac{2\,t^{3/2}}{3})\right],$$

which gives, using (B-3),

$$y'(t) = \frac{\pi}{3^{1/3}\,\Gamma(1/3)}\frac{2\,t}{\sqrt{3}}\,I_{2/3}(\frac{2\,t^{3/2}}{3}).$$

B1-Abramowitz, M., Stegun, I., *HANDBOOK OF MATHEMATICAL FUNCTIONS,* Dover Publications, N.Y., 1965.

APPENDIX C

RELATIVE ERROR TYPE($\mathcal{R}$ or $\mathcal{T}$), SE AND SF FACTORS

APPENDIX C

RELATIVE ERROR TYPE($\mathcal{R}$ or $\mathcal{T}$), SE AND SF FACTORS

This appendix contains for each case of Section II: its number, the relative error type (either $\mathcal{R}$ or $\mathcal{T}$), the SE factor, and the SF factor (see (3) and (5) of Section I).

1	$\mathcal{R}$	4.75E01	5.00E00	2	$\mathcal{R}$	1.22E01	3.33E00	3	$\mathcal{R}$	1.25E00	5.00E00
4	$\mathcal{R}$	1.00E00	5.00E00	5	$\mathcal{R}$	1.25E00	5.00E00	6	$\mathcal{R}$	1.16E01	3.29E00
7	$\mathcal{R}$	1.25E00	3.63E01	8	$\mathcal{R}$	8.61E01	3.00E00	9	$\mathcal{R}$	1.03E01	4.07E00
10	$\mathcal{R}$	1.25E00	5.00E00	11	$\mathcal{R}$	1.00E00	5.00E00	12	$\mathcal{R}$	1.25E00	5.00E00
13	$\mathcal{R}$	1.09E01	3.56E00	14	$\mathcal{R}$	2.13E01	3.00E00	15	$\mathcal{R}$	1.25E00	5.00E00
16	$\mathcal{R}$	1.25E00	5.00E00	17	$\mathcal{R}$	1.25E00	5.00E00	18	$\mathcal{R}$	1.25E00	5.00E00
19	$\mathcal{R}$	1.25E00	5.00E00	20	$\mathcal{R}$	1.25E00	5.00E00	21	$\mathcal{R}$	1.25E00	5.00E00
22	$\mathcal{R}$	1.25E00	5.00E00	23	$\mathcal{R}$	1.25E00	5.00E00	24	$\mathcal{R}$	1.25E00	5.00E00
25	$\mathcal{R}$	1.25E00	5.00E00	26	$\mathcal{R}$	1.25E00	5.00E00	27	$\mathcal{R}$	1.25E00	5.00E00
28	$\mathcal{R}$	1.25E00	5.00E00	29	$\mathcal{R}$	1.25E00	5.00E00	30	$\mathcal{R}$	1.25E00	5.00E00
31	$\mathcal{R}$	1.25E00	5.00E00	32	$\mathcal{R}$	1.25E00	5.00E00	33	$\mathcal{R}$	1.25E00	5.00E00
34	$\mathcal{R}$	1.25E00	5.00E00	35	$\mathcal{R}$	1.25E00	5.00E00	36	$\mathcal{R}$	1.25E00	5.00E00
37	$\mathcal{R}$	1.25E00	5.00E00	38	$\mathcal{R}$	1.25E00	5.00E00	39	$\mathcal{R}$	1.25E00	5.00E00
40	$\mathcal{R}$	1.25E00	5.00E00	41	$\mathcal{R}$	1.25E00	5.00E00	42	$\mathcal{R}$	1.25E00	5.00E00
43	$\mathcal{R}$	1.00E00	5.00E00	44	$\mathcal{R}$	1.46E02	3.00E00	45	$\mathcal{R}$	1.25E00	5.00E00
46	$\mathcal{R}$	1.25E00	5.00E00	47	$\mathcal{R}$	1.25E00	5.00E00	48	$\mathcal{R}$	1.25E00	5.00E00
49	$\mathcal{R}$	1.00E00	5.00E00	50	$\mathcal{R}$	6.43E01	1.49E01	51	$\mathcal{R}$	1.25E00	5.00E00
52	$\mathcal{R}$	1.25E00	5.00E00	53	$\mathcal{R}$	1.25E00	5.00E00	54	$\mathcal{R}$	1.25E00	5.00E00
55	$\mathcal{R}$	1.25E00	5.00E00	56	$\mathcal{R}$	1.25E00	5.00E00	57	$\mathcal{R}$	1.25E00	5.00E00
58	$\mathcal{R}$	1.25E00	5.00E00	59	$\mathcal{R}$	1.25E00	5.00E00	60	$\mathcal{R}$	1.25E00	5.00E00
61	$\mathcal{R}$	1.25E00	5.00E00	62	$\mathcal{R}$	1.25E00	5.00E00	63	$\mathcal{R}$	1.25E00	5.00E00
64	$\mathcal{R}$	1.25E00	5.00E00	65	$\mathcal{R}$	1.25E00	5.00E00	66	$\mathcal{R}$	1.25E00	5.00E00
67	$\mathcal{R}$	1.25E00	5.00E00	68	$\mathcal{R}$	1.25E00	5.00E00	69	$\mathcal{R}$	1.25E00	5.00E00
70	$\mathcal{R}$	1.25E00	5.00E00	71	$\mathcal{R}$	1.25E00	5.00E00	72	$\mathcal{R}$	1.25E00	5.00E00
73	$\mathcal{R}$	1.25E00	5.00E00	74	$\mathcal{R}$	1.25E00	5.00E00	75	$\mathcal{R}$	1.25E00	5.00E00
76	$\mathcal{R}$	1.25E00	5.00E00	77	$\mathcal{R}$	1.25E00	5.00E00	78	$\mathcal{R}$	1.25E00	5.00E00
79	$\mathcal{R}$	1.25E00	5.00E00	80	$\mathcal{R}$	1.25E00	5.00E00	81	$\mathcal{R}$	1.25E00	5.00E00

82 $\mathcal{R}$ 1.25E00 5.00E00	83 $\mathcal{R}$ 1.25E00 5.00E00	84 $\mathcal{R}$ 5.94E00 5.00E00
85 $\mathcal{R}$ 1.25E00 5.00E00	86 $\mathcal{R}$ 3.24E01 3.00E00	87 $\mathcal{R}$ 1.25E00 5.00E00
88 $\mathcal{R}$ 1.11E01 5.00E00	89 $\mathcal{R}$ 1.25E00 5.00E00	90 $\mathcal{R}$ 1.25E00 5.00E00
91 $\mathcal{R}$ 1.25E00 5.00E00	92 $\mathcal{R}$ 1.25E00 4.35E01	93 $\mathcal{R}$ 1.25E00 5.00E00
94 $\mathcal{R}$ 2.60E01 3.33E00	95 $\mathcal{R}$ 1.25E00 5.00E00	96 $\mathcal{R}$ 1.90E01 4.04E00
97 $\mathcal{R}$ 1.25E00 5.00E00	98 $\mathcal{R}$ 1.25E00 5.00E00	99 $\mathcal{R}$ 1.25E00 5.00E00
100 $\mathcal{R}$ 1.25E00 5.00E00	101 $\mathcal{R}$ 1.25E00 5.00E00	102 $\mathcal{R}$ 1.25E00 5.00E00
103 $\mathcal{R}$ 1.25E00 5.00E00	104 $\mathcal{R}$ 1.25E00 5.00E00	105 $\mathcal{R}$ 1.25E00 5.00E00
106 $\mathcal{R}$ 1.25E00 5.00E00	107 $\mathcal{R}$ 1.25E00 5.00E00	108 $\mathcal{R}$ 1.25E00 5.00E00
109 $\mathcal{R}$ 1.25E00 5.00E00	110 $\mathcal{R}$ 1.25E00 5.00E00	111 $\mathcal{R}$ 1.25E00 5.00E00
112 $\mathcal{R}$ 4.99E01 5.00E00	113 $\mathcal{R}$ 1.25E00 5.00E00	114 $\mathcal{R}$ 1.25E00 5.00E00
115 $\mathcal{R}$ 1.25E00 5.00E00	116 $\mathcal{R}$ 5.12E00 5.00E00	117 $\mathcal{R}$ 7.10E00 5.00E00
118 $\mathcal{R}$ 1.25E00 5.00E00	119 $\mathcal{R}$ 1.25E00 5.00E00	120 $\mathcal{R}$ 3.92E01 3.00E00
121 $\mathcal{R}$ 1.25E00 5.00E00	122 $\mathcal{R}$ 9.33E00 5.00E00	123 $\mathcal{R}$ 7.83E00 5.00E00
124 $\mathcal{R}$ 1.75E02 3.00E00	125 $\mathcal{R}$ 1.25E00 5.00E00	126 $\mathcal{R}$ 1.25E00 2.17E01
127 $\mathcal{R}$ 1.25E00 4.44E01	128 $\mathcal{R}$ 4.75E02 5.00E00	129 $\mathcal{R}$ 1.25E00 5.00E00
130 $\mathcal{R}$ 1.25E00 5.00E00	131 $\mathcal{R}$ 9.60E00 5.00E00	132 $\mathcal{R}$ 5.87E00 5.00E00
133 $\mathcal{R}$ 1.25E00 5.00E00	134 $\mathcal{R}$ 1.25E00 5.00E00	135 $\mathcal{R}$ 1.25E00 5.00E00
136 $\mathcal{R}$ 3.97E01 3.00E00	137 $\mathcal{R}$ 1.25E00 5.00E00	138 $\mathcal{R}$ 1.25E00 5.00E00
139 $\mathcal{R}$ 1.25E00 5.00E00	140 $\mathcal{R}$ 1.61E02 3.00E00	141 $\mathcal{R}$ 1.25E00 5.00E00
142 $\mathcal{R}$ 5.63E00 5.00E00	143 $\mathcal{R}$ 1.25E00 5.00E00	144 $\mathcal{R}$ 5.00E02 5.00E00
145 $\mathcal{R}$ 1.25E00 5.00E00	146 $\mathcal{R}$ 7.00E00 5.00E00	147 $\mathcal{R}$ 1.25E00 5.00E00
148 $\mathcal{R}$ 1.02E01 5.00E00	149 $\mathcal{R}$ 7.98E00 5.00E00	150 $\mathcal{R}$ 1.25E00 5.00E00
151 $\mathcal{R}$ 1.25E00 5.00E00	152 $\mathcal{R}$ 3.90E01 3.00E00	153 $\mathcal{R}$ 1.25E00 5.00E00
154 $\mathcal{R}$ 1.25E00 5.00E00	155 $\mathcal{R}$ 1.25E00 5.00E00	156 $\mathcal{R}$ 1.71E02 3.00E00
157 $\mathcal{R}$ 1.25E00 5.00E00	158 $\mathcal{R}$ 1.80E01 3.00E00	159 $\mathcal{R}$ 1.62E01 3.00E00
160 $\mathcal{R}$ 4.96E02 5.00E00	161 $\mathcal{R}$ 1.25E00 5.00E00	162 $\mathcal{R}$ 6.36E00 5.00E00
163 $\mathcal{R}$ 1.25E00 5.00E00	164 $\mathcal{R}$ 1.29E01 3.00E00	165 $\mathcal{R}$ 1.25E00 2.50E01
166 $\mathcal{R}$ 1.25E00 5.00E00	167 $\mathcal{R}$ 1.25E00 5.00E00	168 $\mathcal{R}$ 3.94E01 3.00E00
169 $\mathcal{R}$ 1.46E01 4.80E00	170 $\mathcal{R}$ 1.25E00 5.00E00	171 $\mathcal{R}$ 7.38E00 5.00E00
172 $\mathcal{R}$ 1.78E02 3.00E00	173 $\mathcal{R}$ 9.17E00 8.41E00	174 $\mathcal{R}$ 1.43E01 3.41E00
175 $\mathcal{R}$ 1.06E01 3.18E00	176 $\mathcal{R}$ 4.91E02 5.00E00	177 $\mathcal{R}$ 1.25E00 6.32E01
178 $\mathcal{R}$ 1.25E00 5.00E00	179 $\mathcal{R}$ 1.25E00 8.03E01	180 $\mathcal{R}$ 7.45E01 3.00E00
181 $\mathcal{R}$ 1.25E00 5.00E00	182 $\mathcal{R}$ 1.25E00 5.00E00	183 $\mathcal{R}$ 6.55E00 5.00E00

184 $\mathcal{R}$ 1.25E00 5.00E00	185 $\mathcal{R}$ 1.25E00 5.00E00	186 $\mathcal{R}$ 1.25E00 5.00E00
187 $\mathcal{R}$ 1.25E00 5.00E00	188 $\mathcal{R}$ 7.32E00 5.00E00	189 $\mathcal{R}$ 9.76E00 5.00E00
190 $\mathcal{R}$ 1.11E01 5.00E00	191 $\mathcal{R}$ 1.25E00 5.00E00	192 $\mathcal{R}$ 7.09E01 3.00E00
193 $\mathcal{R}$ 1.25E00 5.00E00	194 $\mathcal{R}$ 1.36E01 5.00E00	195 $\mathcal{R}$ 5.22E00 5.00E00
196 $\mathcal{R}$ 2.74E02 3.00E00	197 $\mathcal{R}$ 6.13E00 5.00E00	198 $\mathcal{R}$ 1.77E01 3.00E00
199 $\mathcal{R}$ 2.51E01 3.00E00	200 $\mathcal{R}$ 8.55E02 3.00E00	201 $\mathcal{R}$ 1.25E00 5.00E00
202 $\mathcal{R}$ 6.32E00 5.00E00	203 $\mathcal{R}$ 1.01E01 5.00E00	204 $\mathcal{R}$ 7.33E00 5.00E00
205 $\mathcal{R}$ 1.25E00 5.00E00	206 $\mathcal{R}$ 1.25E00 5.00E00	207 $\mathcal{R}$ 1.25E00 5.00E00
208 $\mathcal{R}$ 7.19E01 5.00E00	209 $\mathcal{R}$ 1.25E00 5.00E00	210 $\mathcal{R}$ 5.07E00 5.00E00
211 $\mathcal{R}$ 1.25E00 3.16E01	212 $\mathcal{R}$ 2.65E02 3.00E00	213 $\mathcal{R}$ 1.25E00 2.16E01
214 $\mathcal{R}$ 9.01E00 3.00E00	215 $\mathcal{R}$ 3.10E01 5.00E00	216 $\mathcal{R}$ 8.12E02 3.00E00
217 $\mathcal{R}$ 9.52E00 5.00E00	218 $\mathcal{R}$ 1.87E01 5.00E00	219 $\mathcal{R}$ 6.18E00 5.00E00
220 $\mathcal{R}$ 6.79E00 5.00E00	221 $\mathcal{R}$ 1.29E01 5.00E00	222 $\mathcal{R}$ 1.25E00 5.00E00
223 $\mathcal{R}$ 1.25E00 5.00E00	224 $\mathcal{R}$ 7.41E01 3.00E00	225 $\mathcal{R}$ 1.25E00 5.00E00
226 $\mathcal{R}$ 1.25E00 5.00E00	227 $\mathcal{R}$ 1.25E00 5.00E00	228 $\mathcal{R}$ 2.76E02 5.00E00
229 $\mathcal{R}$ 1.25E00 5.00E00	230 $\mathcal{R}$ 1.25E00 2.40E01	231 $\mathcal{R}$ 1.25E00 1.26E02
232 $\mathcal{R}$ 8.70E02 3.00E00	233 $\mathcal{R}$ 1.25E00 2.76E01	234 $\mathcal{R}$ 7.94E00 5.00E00
235 $\mathcal{R}$ 1.39E01 5.00E00	236 $\mathcal{R}$ 2.07E01 5.00E00	237 $\mathcal{R}$ 1.49E01 4.47E00
238 $\mathcal{R}$ 1.25E00 5.00E00	239 $\mathcal{R}$ 1.25E00 5.00E00	240 $\mathcal{R}$ 6.72E01 5.00E00
241 $\mathcal{R}$ 1.46E01 5.00E00	242 $\mathcal{R}$ 1.25E00 5.00E00	243 $\mathcal{R}$ 1.25E00 5.00E00
244 $\mathcal{R}$ 2.48E02 3.00E00	245 $\mathcal{R}$ 1.58E01 5.00E00	246 $\mathcal{R}$ 1.25E00 2.20E01
247 $\mathcal{R}$ 1.25E00 1.25E02	248 $\mathcal{R}$ 8.08E02 3.00E00	249 $\mathcal{R}$ 6.65E00 2.85E01
250 $\mathcal{R}$ 6.82E00 5.62E00	251 $\mathcal{R}$ 1.02E01 3.09E00	252 $\mathcal{R}$ 5.00E00 3.06E01
253 $\mathcal{R}$ 8.21E00 5.00E00	254 $\mathcal{R}$ 1.25E00 5.00E00	255 $\mathcal{R}$ 1.25E00 5.00E00
256 $\mathcal{R}$ 1.25E00 5.00E00	257 $\mathcal{R}$ 1.25E00 5.00E00	258 $\mathcal{R}$ 1.00E00 5.00E00
259 $\mathcal{R}$ 1.00E00 5.00E00	260 $\mathcal{R}$ 1.25E00 5.00E00	261 $\mathcal{R}$ 1.25E00 5.00E00
262 $\mathcal{R}$ 1.00E00 5.00E00	263 $\mathcal{R}$ 1.25E00 5.00E00	264 $\mathcal{R}$ 1.25E00 5.00E00
265 $\mathcal{R}$ 7.84E00 5.00E00	266 $\mathcal{R}$ 1.25E00 5.00E00	267 $\mathcal{R}$ 6.61E01 3.00E00
268 $\mathcal{R}$ 1.25E00 5.00E00	269 $\mathcal{R}$ 1.25E00 5.00E00	270 $\mathcal{R}$ 1.66E01 5.00E00
271 $\mathcal{R}$ 1.25E00 5.00E00	272 $\mathcal{R}$ 1.25E00 2.39E01	273 $\mathcal{R}$ 1.67E02 5.00E00
274 $\mathcal{R}$ 1.25E00 5.00E00	275 $\mathcal{R}$ 1.25E00 5.00E00	276 $\mathcal{R}$ 1.00E00 5.00E00
277 $\mathcal{R}$ 1.00E00 5.00E00	278 $\mathcal{R}$ 1.25E00 5.00E00	279 $\mathcal{R}$ 1.25E00 5.00E00
280 $\mathcal{R}$ 1.00E00 5.00E00	281 $\mathcal{R}$ 1.25E00 5.00E00	282 $\mathcal{R}$ 1.25E00 5.00E00
283 $\mathcal{R}$ 1.01E01 5.00E00	284 $\mathcal{R}$ 5.08E00 5.00E00	285 $\mathcal{R}$ 1.92E02 5.00E00

286	$\mathcal{R}$	1.25E00	5.00E00	287	$\mathcal{T}$	5.40E14	5.59E00	288	$\mathcal{T}$	2.43E15	4.89E00
289	$\mathcal{R}$	7.64E00	5.00E00	290	$\mathcal{R}$	1.25E00	2.64E01	291	$\mathcal{R}$	4.24E02	4.30E00
292	$\mathcal{R}$	1.25E00	5.00E00	293	$\mathcal{R}$	1.25E00	5.00E00	294	$\mathcal{R}$	1.25E00	5.00E00
295	$\mathcal{R}$	6.60E00	5.14E00	296	$\mathcal{R}$	1.25E00	5.00E00	297	$\mathcal{R}$	1.20E01	1.21E01
298	$\mathcal{R}$	1.33E01	3.29E00	299	$\mathcal{R}$	1.25E00	5.00E00	300	$\mathcal{R}$	1.25E00	5.00E00
301	$\mathcal{R}$	1.25E00	5.00E00	302	$\mathcal{R}$	1.55E01	5.00E00	303	$\mathcal{R}$	1.25E00	6.91E01
304	$\mathcal{R}$	2.75E01	3.00E00	305	$\mathcal{R}$	1.25E00	7.73E01	306	$\mathcal{R}$	5.40E00	6.90E00
307	$\mathcal{R}$	3.24E01	3.00E00	308	$\mathcal{R}$	1.82E01	3.00E00	309	$\mathcal{R}$	5.84E00	1.33E01
310	$\mathcal{R}$	4.68E01	3.04E00	311	$\mathcal{R}$	1.20E01	5.00E00	312	$\mathcal{R}$	1.25E00	5.00E00
313	$\mathcal{R}$	8.25E00	4.00E00	314	$\mathcal{R}$	8.99E00	5.00E00	315	$\mathcal{R}$	3.76E01	4.10E00
316	$\mathcal{R}$	1.64E01	4.06E00	317	$\mathcal{R}$	1.68E01	5.00E00	318	$\mathcal{R}$	1.07E02	3.00E00
319	$\mathcal{R}$	1.25E00	5.00E00	320	$\mathcal{R}$	2.82E01	5.00E00	321	$\mathcal{R}$	2.37E01	3.50E00
322	$\mathcal{R}$	3.21E01	6.30E00	323	$\mathcal{R}$	7.80E01	3.16E00	324	$\mathcal{R}$	3.08E01	5.00E00
325	$\mathcal{R}$	1.53E01	2.00E01	326	$\mathcal{R}$	4.22E01	3.00E00	327	$\mathcal{R}$	2.62E02	4.15E00
328	$\mathcal{R}$	4.05E02	4.91E00	329	$\mathcal{R}$	1.25E00	1.25E02	330	$\mathcal{R}$	1.25E00	5.00E00
331	$\mathcal{R}$	6.02E00	6.97E00	332	$\mathcal{R}$	5.60E00	5.00E00	333	$\mathcal{R}$	1.25E00	5.00E00
334	$\mathcal{R}$	1.52E06	3.97E00	335	$\mathcal{R}$	5.15E00	5.00E00	336	$\mathcal{R}$	7.18E00	5.00E00
337	$\mathcal{R}$	1.25E00	5.00E00	338	$\mathcal{R}$	1.25E00	5.00E00	339	$\mathcal{R}$	1.25E00	5.00E00
340	$\mathcal{R}$	3.00E01	5.00E00	341	$\mathcal{R}$	2.27E01	7.69E00	342	$\mathcal{R}$	7.61E00	5.00E00
343	$\mathcal{R}$	7.91E00	7.88E00	344	$\mathcal{R}$	9.99E00	4.90E00	345	$\mathcal{R}$	1.26E01	3.80E00
346	$\mathcal{R}$	8.85E00	6.64E00	347	$\mathcal{T}$	7.28E13	3.19E00	348	$\mathcal{T}$	8.63E11	5.00E00
349	$\mathcal{T}$	4.05E12	5.00E00	350	$\mathcal{T}$	1.30E10	5.00E00	351	$\mathcal{T}$	4.99E11	5.00E00
352	$\mathcal{T}$	7.41E17	1.72E01	353	$\mathcal{T}$	5.91E12	5.00E00	354	$\mathcal{T}$	4.83E12	5.00E00
355	$\mathcal{T}$	2.17E12	5.00E00	356	$\mathcal{T}$	1.75E12	5.00E00	357	$\mathcal{T}$	1.95E12	5.00E00
358	$\mathcal{T}$	2.05E12	5.00E00	359	$\mathcal{T}$	6.23E14	7.36E00	360	$\mathcal{T}$	7.09E14	4.56E00
361	$\mathcal{T}$	1.05E14	8.52E00	362	$\mathcal{T}$	5.60E14	6.23E00	363	$\mathcal{T}$	7.42E13	1.44E01
364	$\mathcal{T}$	7.04E12	5.53E01	365	$\mathcal{R}$	1.35E01	5.00E00	366	$\mathcal{R}$	1.25E00	5.00E00
367	$\mathcal{R}$	1.25E00	5.00E00	368	$\mathcal{R}$	1.25E00	5.00E00	369	$\mathcal{R}$	1.25E00	5.46E01
370	$\mathcal{R}$	3.53E01	8.93E00	371	$\mathcal{R}$	8.42E00	5.00E00	372	$\mathcal{R}$	2.06E01	3.00E00
373	$\mathcal{R}$	1.50E01	3.00E00	374	$\mathcal{R}$	1.02E01	5.00E00	375	$\mathcal{R}$	7.07E00	1.74E01
376	$\mathcal{R}$	5.14E01	4.34E00	377	$\mathcal{R}$	1.06E01	4.46E00	378	$\mathcal{R}$	1.25E00	5.00E00
379	$\mathcal{R}$	1.36E01	5.00E00	380	$\mathcal{R}$	1.25E00	5.00E00	381	$\mathcal{R}$	7.69E00	9.56E00
382	$\mathcal{R}$	2.38E02	3.00E00	383	$\mathcal{R}$	1.60E01	5.00E00	384	$\mathcal{R}$	1.19E01	5.00E00
385	$\mathcal{R}$	2.99E01	5.00E00	386	$\mathcal{R}$	1.25E00	5.00E00	387	$\mathcal{R}$	4.16E01	3.00E00

388	$\mathcal{R}$	1.92E02 3.00E00	389	$\mathcal{R}$	1.25E00 5.00E00	390	$\mathcal{R}$	6.97E01 3.00E00
391	$\mathcal{R}$	5.30E00 5.00E00	392	$\mathcal{R}$	1.25E00 5.00E00	393	$\mathcal{R}$	1.34E01 4.98E00
394	$\mathcal{R}$	2.37E02 3.00E00	395	$\mathcal{R}$	1.25E00 6.69E01	396	$\mathcal{R}$	1.25E00 5.00E00
397	$\mathcal{R}$	1.25E00 5.00E00	398	$\mathcal{R}$	1.25E00 3.86E01	399	$\mathcal{R}$	6.32E01 3.00E00
400	$\mathcal{R}$	4.54E02 1.91E01	401	$\mathcal{R}$	1.25E00 5.00E00	402	$\mathcal{R}$	1.25E00 5.00E00
403	$\mathcal{R}$	1.25E00 5.00E00	404	$\mathcal{R}$	1.25E00 5.00E00	405	$\mathcal{T}$	1.23E13 5.00E00
406	$\mathcal{T}$	2.64E14 5.01E00	407	$\mathcal{T}$	1.75E14 4.84E00	408	$\mathcal{T}$	7.86E13 4.66E00
409	$\mathcal{R}$	1.25E00 5.00E00	410	$\mathcal{R}$	1.25E00 5.00E00	411	$\mathcal{R}$	1.25E00 5.00E00
412	$\mathcal{R}$	1.25E00 5.00E00	413	$\mathcal{R}$	1.25E00 5.00E00	414	$\mathcal{R}$	1.00E00 5.00E00
415	$\mathcal{R}$	1.25E00 5.00E00	416	$\mathcal{R}$	1.00E00 5.00E00	417	$\mathcal{R}$	1.25E00 5.00E00
418	$\mathcal{R}$	1.00E00 5.00E00	419	$\mathcal{R}$	1.25E00 5.00E00	420	$\mathcal{R}$	1.00E00 5.00E00
421	$\mathcal{R}$	1.25E00 5.00E00	422	$\mathcal{R}$	1.00E00 5.00E00	423	$\mathcal{R}$	1.25E00 5.00E00
424	$\mathcal{R}$	1.00E00 5.00E00	425	$\mathcal{R}$	1.25E00 5.00E00	426	$\mathcal{R}$	1.25E00 5.00E00
427	$\mathcal{R}$	1.25E00 5.00E00	428	$\mathcal{R}$	1.25E00 5.00E00	429	$\mathcal{R}$	1.25E00 5.00E00
430	$\mathcal{R}$	1.00E00 5.00E00	431	$\mathcal{R}$	1.25E00 5.00E00	432	$\mathcal{R}$	1.00E00 5.00E00
433	$\mathcal{R}$	1.25E00 5.00E00	434	$\mathcal{R}$	1.25E00 5.00E00	435	$\mathcal{R}$	1.25E00 5.00E00
436	$\mathcal{R}$	1.25E00 5.00E00	437	$\mathcal{R}$	1.25E00 5.00E00	438	$\mathcal{R}$	1.00E00 5.00E00
439	$\mathcal{R}$	1.25E00 5.00E00	440	$\mathcal{R}$	1.00E00 5.00E00	441	$\mathcal{R}$	1.25E00 5.00E00
442	$\mathcal{R}$	1.00E00 5.00E00	443	$\mathcal{R}$	1.80E03 3.00E00	444	$\mathcal{R}$	1.25E00 5.00E00
445	$\mathcal{R}$	1.80E03 3.00E00	446	$\mathcal{R}$	1.25E00 5.00E00	447	$\mathcal{R}$	1.25E00 5.00E00
448	$\mathcal{R}$	8.83E00 5.00E00	449	$\mathcal{R}$	1.25E00 5.00E00	450	$\mathcal{R}$	8.83E00 5.00E00
451	$\mathcal{R}$	1.25E00 5.00E00	452	$\mathcal{R}$	1.25E00 5.00E00	453	$\mathcal{R}$	1.25E00 5.00E00
454	$\mathcal{R}$	1.25E00 5.00E00	455	$\mathcal{R}$	1.25E00 5.00E00	456	$\mathcal{R}$	1.25E00 5.00E00
457	$\mathcal{R}$	1.25E00 5.00E00	458	$\mathcal{R}$	1.25E00 5.00E00	459	$\mathcal{R}$	1.25E00 5.00E00
460	$\mathcal{R}$	1.25E00 5.00E00	461	$\mathcal{R}$	1.25E00 5.00E00	462	$\mathcal{R}$	1.25E00 5.00E00
463	$\mathcal{R}$	4.40E01 3.00E00	464	$\mathcal{R}$	1.00E00 5.00E00	465	$\mathcal{R}$	6.63E03 3.00E00
466	$\mathcal{R}$	1.00E00 5.00E00	467	$\mathcal{R}$	7.63E05 2.35E01	468	$\mathcal{R}$	1.00E00 5.00E00
469	$\mathcal{R}$	1.50E07 1.60E01	470	$\mathcal{R}$	1.00E00 5.00E00	471	$\mathcal{R}$	1.62E08 1.05E01
472	$\mathcal{R}$	1.00E00 5.00E00	473	$\mathcal{R}$	1.13E09 7.64E00	474	$\mathcal{R}$	1.00E00 5.00E00
475	$\mathcal{R}$	5.76E09 6.60E00	476	$\mathcal{R}$	1.00E00 5.00E00	477	$\mathcal{R}$	2.39E10 6.83E00
478	$\mathcal{R}$	1.00E00 5.00E00	479	$\mathcal{R}$	8.58E10 7.81E00	480	$\mathcal{R}$	1.00E00 5.00E00
481	$\mathcal{R}$	2.63E11 9.14E00	482	$\mathcal{R}$	1.00E00 5.00E00	483	$\mathcal{R}$	6.52E11 1.06E01
484	$\mathcal{R}$	1.00E00 5.00E00	485	$\mathcal{R}$	1.25E12 1.26E01	486	$\mathcal{R}$	1.00E00 5.00E00
487	$\mathcal{R}$	1.82E12 1.48E01	488	$\mathcal{R}$	1.00E00 5.00E00	489	$\mathcal{R}$	1.96E12 1.85E01

490 $\mathcal{R}$ 1.00E00 5.00E00	491 $\mathcal{R}$ 1.52E12 2.37E01	492 $\mathcal{R}$ 1.00E00 5.00E00
493 $\mathcal{R}$ 8.06E11 3.28E01	494 $\mathcal{R}$ 1.00E00 5.00E00	495 $\mathcal{R}$ 2.61E11 5.45E01
496 $\mathcal{R}$ 1.00E00 5.00E00	497 $\mathcal{R}$ 3.39E10 1.49E02	498 $\mathcal{R}$ 1.00E00 5.00E00
499 $\mathcal{R}$ 5.77E09 1.92E02	500 $\mathcal{R}$ 1.00E00 5.00E00	501 $\mathcal{R}$ 1.92E09 5.72E01
502 $\mathcal{R}$ 1.00E00 5.00E00	503 $\mathcal{R}$ 1.15E04 4.38E00	504 $\mathcal{R}$ 1.05E03 1.33E01
505 $\mathcal{R}$ 5.56E03 3.67E00	506 $\mathcal{R}$ 6.27E02 1.29E01	507 $\mathcal{R}$ 3.56E02 1.25E01
508 $\mathcal{R}$ 2.31E02 4.16E00	509 $\mathcal{R}$ 8.20E01 6.90E01	510 $\mathcal{R}$ 1.36E02 3.00E00
511 $\mathcal{R}$ 4.20E01 4.25E02	512 $\mathcal{R}$ 8.11E01 2.92E01	513 $\mathcal{R}$ 1.75E04 3.00E00
514 $\mathcal{R}$ 4.07E02 6.23E01	515 $\mathcal{R}$ 1.25E00 5.00E00	516 $\mathcal{R}$ 1.00E00 5.00E00
517 $\mathcal{R}$ 1.25E00 5.00E00	518 $\mathcal{R}$ 1.00E00 5.00E00	519 $\mathcal{R}$ 1.25E00 5.00E00
520 $\mathcal{R}$ 1.00E00 5.00E00	521 $\mathcal{R}$ 1.25E00 5.00E00	522 $\mathcal{R}$ 1.00E00 5.00E00
523 $\mathcal{R}$ 1.25E00 5.00E00	524 $\mathcal{R}$ 1.00E00 5.00E00	525 $\mathcal{R}$ 1.25E00 5.00E00
526 $\mathcal{R}$ 1.00E00 5.00E00	527 $\mathcal{R}$ 1.25E00 5.00E00	528 $\mathcal{R}$ 1.00E00 5.00E00
529 $\mathcal{R}$ 1.25E00 5.00E00	530 $\mathcal{R}$ 1.00E00 5.00E00	531 $\mathcal{R}$ 1.25E00 5.00E00
532 $\mathcal{R}$ 1.00E00 5.00E00	533 $\mathcal{R}$ 1.25E00 5.00E00	534 $\mathcal{R}$ 1.00E00 5.00E00
535 $\mathcal{R}$ 1.71E10 3.00E00	536 $\mathcal{R}$ 2.82E10 3.00E00	537 $\mathcal{R}$ 3.58E10 3.00E00
538 $\mathcal{R}$ 4.14E10 3.00E00	539 $\mathcal{R}$ 4.55E10 3.00E00	540 $\mathcal{R}$ 4.87E10 3.00E00
541 $\mathcal{R}$ 5.12E10 3.00E00	542 $\mathcal{R}$ 5.32E10 3.00E00	543 $\mathcal{R}$ 5.48E10 3.00E00
544 $\mathcal{R}$ 5.62E10 3.00E00	545 $\mathcal{R}$ 2.84E10 3.00E00	546 $\mathcal{R}$ 3.49E10 3.00E00
547 $\mathcal{R}$ 3.93E10 3.00E00	548 $\mathcal{R}$ 4.24E10 3.00E00	549 $\mathcal{R}$ 4.47E10 3.00E00
550 $\mathcal{R}$ 4.64E10 3.00E00	551 $\mathcal{R}$ 4.77E10 3.00E00	552 $\mathcal{R}$ 4.87E10 3.00E00
553 $\mathcal{R}$ 4.95E10 3.00E00	554 $\mathcal{R}$ 5.02E10 3.00E00	555 $\mathcal{R}$ 3.63E10 3.00E00
556 $\mathcal{R}$ 3.96E10 3.00E00	557 $\mathcal{R}$ 4.17E10 3.00E00	558 $\mathcal{R}$ 4.30E10 3.00E00
559 $\mathcal{R}$ 4.40E10 3.00E00	560 $\mathcal{R}$ 4.46E10 3.00E00	561 $\mathcal{R}$ 4.51E10 3.00E00
562 $\mathcal{R}$ 4.54E10 3.00E00	563 $\mathcal{R}$ 4.56E10 3.00E00	564 $\mathcal{R}$ 4.58E10 3.00E00
565 $\mathcal{R}$ 4.21E10 3.00E00	566 $\mathcal{R}$ 4.29E10 3.00E00	567 $\mathcal{R}$ 4.33E10 3.00E00
568 $\mathcal{R}$ 4.34E10 3.00E00	569 $\mathcal{R}$ 4.33E10 3.00E00	570 $\mathcal{R}$ 4.32E10 3.00E00
571 $\mathcal{R}$ 4.30E10 3.00E00	572 $\mathcal{R}$ 4.29E10 3.00E00	573 $\mathcal{R}$ 4.27E10 3.00E00
574 $\mathcal{R}$ 4.25E10 3.00E00	575 $\mathcal{R}$ 4.65E10 3.00E00	576 $\mathcal{R}$ 4.54E10 3.00E00
577 $\mathcal{R}$ 4.44E10 3.00E00	578 $\mathcal{R}$ 4.35E10 3.00E00	579 $\mathcal{R}$ 4.27E10 3.00E00
580 $\mathcal{R}$ 4.20E10 3.00E00	581 $\mathcal{R}$ 4.14E10 3.00E00	582 $\mathcal{R}$ 4.09E10 3.00E00
583 $\mathcal{R}$ 4.04E10 3.00E00	584 $\mathcal{R}$ 3.99E10 3.00E00	585 $\mathcal{R}$ 4.99E10 3.00E00
586 $\mathcal{R}$ 4.73E10 3.00E00	587 $\mathcal{R}$ 4.52E10 3.00E00	588 $\mathcal{R}$ 4.36E10 3.00E00
589 $\mathcal{R}$ 4.22E10 3.00E00	590 $\mathcal{R}$ 4.11E10 3.00E00	591 $\mathcal{R}$ 4.01E10 3.00E00

592 $\mathcal{R}$ 3.92E10 3.00E00	593 $\mathcal{R}$ 3.85E10 3.00E00	594 $\mathcal{R}$ 3.78E10 3.00E00
595 $\mathcal{R}$ 5.27E10 3.00E00	596 $\mathcal{R}$ 4.88E10 3.00E00	597 $\mathcal{R}$ 4.59E10 3.00E00
598 $\mathcal{R}$ 4.36E10 3.00E00	599 $\mathcal{R}$ 4.17E10 3.00E00	600 $\mathcal{R}$ 4.02E10 3.00E00
601 $\mathcal{R}$ 3.89E10 3.00E00	602 $\mathcal{R}$ 3.79E10 3.00E00	603 $\mathcal{R}$ 3.69E10 3.00E00
604 $\mathcal{R}$ 3.61E10 3.00E00	605 $\mathcal{R}$ 5.49E10 3.00E00	606 $\mathcal{R}$ 5.00E10 3.00E00
607 $\mathcal{R}$ 4.64E10 3.00E00	608 $\mathcal{R}$ 4.36E10 3.00E00	609 $\mathcal{R}$ 4.13E10 3.00E00
610 $\mathcal{R}$ 3.95E10 3.00E00	611 $\mathcal{R}$ 3.80E10 3.00E00	612 $\mathcal{R}$ 3.67E10 3.00E00
613 $\mathcal{R}$ 3.56E10 3.00E00	614 $\mathcal{R}$ 3.47E10 3.00E00	615 $\mathcal{R}$ 5.68E10 3.00E00
616 $\mathcal{R}$ 5.10E10 3.00E00	617 $\mathcal{R}$ 4.67E10 3.00E00	618 $\mathcal{R}$ 4.35E10 3.00E00
619 $\mathcal{R}$ 4.09E10 3.00E00	620 $\mathcal{R}$ 3.89E10 3.00E00	621 $\mathcal{R}$ 3.72E10 3.00E00
622 $\mathcal{R}$ 3.57E10 3.00E00	623 $\mathcal{R}$ 3.45E10 3.00E00	624 $\mathcal{R}$ 3.34E10 3.00E00
625 $\mathcal{R}$ 5.83E10 3.00E00	626 $\mathcal{R}$ 5.18E10 3.00E00	627 $\mathcal{R}$ 4.70E10 3.00E00
628 $\mathcal{R}$ 4.34E10 3.00E00	629 $\mathcal{R}$ 4.06E10 3.00E00	630 $\mathcal{R}$ 3.83E10 3.00E00
631 $\mathcal{R}$ 3.64E10 3.00E00	632 $\mathcal{R}$ 3.49E10 3.00E00	633 $\mathcal{R}$ 3.35E10 3.00E00
634 $\mathcal{R}$ 3.24E10 3.00E00	635 $\mathcal{R}$ 7.31E10 9.97E00	636 $\mathcal{R}$ 1.29E11 9.83E00
637 $\mathcal{R}$ 1.73E11 9.74E00	638 $\mathcal{R}$ 2.09E11 9.65E00	639 $\mathcal{R}$ 2.38E11 9.62E00
640 $\mathcal{R}$ 2.63E11 9.56E00	641 $\mathcal{R}$ 2.85E11 9.49E00	642 $\mathcal{R}$ 3.03E11 9.46E00
643 $\mathcal{R}$ 3.19E11 9.43E00	644 $\mathcal{R}$ 3.34E11 9.37E00	645 $\mathcal{R}$ 1.28E11 9.92E00
646 $\mathcal{R}$ 1.70E11 9.76E00	647 $\mathcal{R}$ 2.03E11 9.65E00	648 $\mathcal{R}$ 2.29E11 9.60E00
649 $\mathcal{R}$ 2.51E11 9.53E00	650 $\mathcal{R}$ 2.70E11 9.46E00	651 $\mathcal{R}$ 2.86E11 9.41E00
652 $\mathcal{R}$ 3.00E11 9.36E00	653 $\mathcal{R}$ 3.12E11 9.33E00	654 $\mathcal{R}$ 3.23E11 9.28E00
655 $\mathcal{R}$ 1.72E11 9.82E00	656 $\mathcal{R}$ 2.02E11 9.71E00	657 $\mathcal{R}$ 2.26E11 9.60E00
658 $\mathcal{R}$ 2.45E11 9.54E00	659 $\mathcal{R}$ 2.62E11 9.44E00	660 $\mathcal{R}$ 2.75E11 9.40E00
661 $\mathcal{R}$ 2.87E11 9.34E00	662 $\mathcal{R}$ 2.97E11 9.31E00	663 $\mathcal{R}$ 3.06E11 9.26E00
664 $\mathcal{R}$ 3.14E11 9.23E00	665 $\mathcal{R}$ 2.07E11 9.78E00	666 $\mathcal{R}$ 2.28E11 9.66E00
667 $\mathcal{R}$ 2.45E11 9.55E00	668 $\mathcal{R}$ 2.59E11 9.45E00	669 $\mathcal{R}$ 2.70E11 9.40E00
670 $\mathcal{R}$ 2.80E11 9.33E00	671 $\mathcal{R}$ 2.88E11 9.29E00	672 $\mathcal{R}$ 2.95E11 9.25E00
673 $\mathcal{R}$ 3.02E11 9.19E00	674 $\mathcal{R}$ 3.07E11 9.18E00	675 $\mathcal{R}$ 2.36E11 9.73E00
676 $\mathcal{R}$ 2.50E11 9.59E00	677 $\mathcal{R}$ 2.60E11 9.53E00	678 $\mathcal{R}$ 2.69E11 9.44E00
679 $\mathcal{R}$ 2.77E11 9.35E00	680 $\mathcal{R}$ 2.83E11 9.31E00	681 $\mathcal{R}$ 2.89E11 9.24E00
682 $\mathcal{R}$ 2.94E11 9.19E00	683 $\mathcal{R}$ 2.98E11 9.16E00	684 $\mathcal{R}$ 3.02E11 9.11E00
685 $\mathcal{R}$ 2.60E11 9.71E00	686 $\mathcal{R}$ 2.68E11 9.56E00	687 $\mathcal{R}$ 2.74E11 9.46E00
688 $\mathcal{R}$ 2.79E11 9.37E00	689 $\mathcal{R}$ 2.83E11 9.31E00	690 $\mathcal{R}$ 2.87E11 9.24E00
691 $\mathcal{R}$ 2.90E11 9.19E00	692 $\mathcal{R}$ 2.93E11 9.14E00	693 $\mathcal{R}$ 2.95E11 9.12E00

694 $\mathcal{R}$ 2.97E11 9.09E00	695 $\mathcal{R}$ 2.81E11 9.67E00	696 $\mathcal{R}$ 2.83E11 9.54E00
697 $\mathcal{R}$ 2.85E11 9.43E00	698 $\mathcal{R}$ 2.87E11 9.34E00	699 $\mathcal{R}$ 2.88E11 9.28E00
700 $\mathcal{R}$ 2.89E11 9.23E00	701 $\mathcal{R}$ 2.91E11 9.15E00	702 $\mathcal{R}$ 2.92E11 9.11E00
703 $\mathcal{R}$ 2.93E11 9.07E00	704 $\mathcal{R}$ 2.93E11 9.06E00	705 $\mathcal{R}$ 2.99E11 9.64E00
706 $\mathcal{R}$ 2.96E11 9.52E00	707 $\mathcal{R}$ 2.95E11 9.40E00	708 $\mathcal{R}$ 2.93E11 9.33E00
709 $\mathcal{R}$ 2.92E11 9.26E00	710 $\mathcal{R}$ 2.92E11 9.18E00	711 $\mathcal{R}$ 2.91E11 9.14E00
712 $\mathcal{R}$ 2.91E11 9.08E00	713 $\mathcal{R}$ 2.90E11 9.06E00	714 $\mathcal{R}$ 2.90E11 9.02E00
715 $\mathcal{R}$ 3.15E11 9.60E00	716 $\mathcal{R}$ 3.08E11 9.49E00	717 $\mathcal{R}$ 3.03E11 9.39E00
718 $\mathcal{R}$ 2.99E11 9.31E00	719 $\mathcal{R}$ 2.96E11 9.23E00	720 $\mathcal{R}$ 2.94E11 9.16E00
721 $\mathcal{R}$ 2.92E11 9.10E00	722 $\mathcal{R}$ 2.90E11 9.07E00	723 $\mathcal{R}$ 2.89E11 9.01E00
724 $\mathcal{R}$ 2.87E11 9.00E00	725 $\mathcal{R}$ 3.28E11 9.60E00	726 $\mathcal{R}$ 3.18E11 9.47E00
727 $\mathcal{R}$ 3.10E11 9.38E00	728 $\mathcal{R}$ 3.05E11 9.26E00	729 $\mathcal{R}$ 3.00E11 9.19E00
730 $\mathcal{R}$ 2.96E11 9.13E00	731 $\mathcal{R}$ 2.92E11 9.10E00	732 $\mathcal{R}$ 2.89E11 9.06E00
733 $\mathcal{R}$ 2.87E11 9.00E00	734 $\mathcal{R}$ 2.85E11 8.96E00	735 $\mathcal{R}$ 1.86E11 3.00E00
736 $\mathcal{R}$ 3.23E11 3.00E00	737 $\mathcal{R}$ 4.29E11 3.00E00	738 $\mathcal{R}$ 5.13E11 3.00E00
739 $\mathcal{R}$ 5.82E11 3.00E00	740 $\mathcal{R}$ 6.39E11 3.00E00	741 $\mathcal{R}$ 6.88E11 3.00E00
742 $\mathcal{R}$ 7.29E11 3.00E00	743 $\mathcal{R}$ 7.65E11 3.00E00	744 $\mathcal{R}$ 7.96E11 3.00E00
745 $\mathcal{R}$ 3.23E11 3.00E00	746 $\mathcal{R}$ 4.22E11 3.00E00	747 $\mathcal{R}$ 4.98E11 3.00E00
748 $\mathcal{R}$ 5.58E11 3.00E00	749 $\mathcal{R}$ 6.08E11 3.00E00	750 $\mathcal{R}$ 6.49E11 3.00E00
751 $\mathcal{R}$ 6.84E11 3.00E00	752 $\mathcal{R}$ 7.13E11 3.00E00	753 $\mathcal{R}$ 7.39E11 3.00E00
754 $\mathcal{R}$ 7.62E11 3.00E00	755 $\mathcal{R}$ 4.29E11 3.00E00	756 $\mathcal{R}$ 4.98E11 3.00E00
757 $\mathcal{R}$ 5.51E11 3.00E00	758 $\mathcal{R}$ 5.93E11 3.00E00	759 $\mathcal{R}$ 6.28E11 3.00E00
760 $\mathcal{R}$ 6.56E11 3.00E00	761 $\mathcal{R}$ 6.81E11 3.00E00	762 $\mathcal{R}$ 7.02E11 3.00E00
763 $\mathcal{R}$ 7.20E11 3.00E00	764 $\mathcal{R}$ 7.35E11 3.00E00	765 $\mathcal{R}$ 5.14E11 3.00E00
766 $\mathcal{R}$ 5.59E11 3.00E00	767 $\mathcal{R}$ 5.93E11 3.00E00	768 $\mathcal{R}$ 6.21E11 3.00E00
769 $\mathcal{R}$ 6.44E11 3.00E00	770 $\mathcal{R}$ 6.63E11 3.00E00	771 $\mathcal{R}$ 6.79E11 3.00E00
772 $\mathcal{R}$ 6.92E11 3.00E00	773 $\mathcal{R}$ 7.04E11 3.00E00	774 $\mathcal{R}$ 7.15E11 3.00E00
775 $\mathcal{R}$ 5.83E11 3.00E00	776 $\mathcal{R}$ 6.08E11 3.00E00	777 $\mathcal{R}$ 6.28E11 3.00E00
778 $\mathcal{R}$ 6.44E11 3.00E00	779 $\mathcal{R}$ 6.57E11 3.00E00	780 $\mathcal{R}$ 6.68E11 3.00E00
781 $\mathcal{R}$ 6.77E11 3.00E00	782 $\mathcal{R}$ 6.85E11 3.00E00	783 $\mathcal{R}$ 6.92E11 3.00E00
784 $\mathcal{R}$ 6.98E11 3.00E00	785 $\mathcal{R}$ 6.40E11 3.00E00	786 $\mathcal{R}$ 6.49E11 3.00E00
787 $\mathcal{R}$ 6.57E11 3.00E00	788 $\mathcal{R}$ 6.63E11 3.00E00	789 $\mathcal{R}$ 6.68E11 3.00E00
790 $\mathcal{R}$ 6.72E11 3.00E00	791 $\mathcal{R}$ 6.76E11 3.00E00	792 $\mathcal{R}$ 6.79E11 3.00E00
793 $\mathcal{R}$ 6.82E11 3.00E00	794 $\mathcal{R}$ 6.84E11 3.00E00	795 $\mathcal{R}$ 6.88E11 3.00E00

796 $\mathcal{R}$ 6.84E11 3.00E00	797 $\mathcal{R}$ 6.81E11 3.00E00	798 $\mathcal{R}$ 6.79E11 3.00E00
799 $\mathcal{R}$ 6.77E11 3.00E00	800 $\mathcal{R}$ 6.76E11 3.00E00	801 $\mathcal{R}$ 6.75E11 3.00E00
802 $\mathcal{R}$ 6.74E11 3.00E00	803 $\mathcal{R}$ 6.74E11 3.00E00	804 $\mathcal{R}$ 6.73E11 3.00E00
805 $\mathcal{R}$ 7.30E11 3.00E00	806 $\mathcal{R}$ 7.14E11 3.00E00	807 $\mathcal{R}$ 7.02E11 3.00E00
808 $\mathcal{R}$ 6.93E11 3.00E00	809 $\mathcal{R}$ 6.85E11 3.00E00	810 $\mathcal{R}$ 6.79E11 3.00E00
811 $\mathcal{R}$ 6.74E11 3.00E00	812 $\mathcal{R}$ 6.70E11 3.00E00	813 $\mathcal{R}$ 6.66E11 3.00E00
814 $\mathcal{R}$ 6.63E11 3.00E00	815 $\mathcal{R}$ 7.66E11 3.00E00	816 $\mathcal{R}$ 7.40E11 3.00E00
817 $\mathcal{R}$ 7.20E11 3.00E00	818 $\mathcal{R}$ 7.05E11 3.00E00	819 $\mathcal{R}$ 6.92E11 3.00E00
820 $\mathcal{R}$ 6.82E11 3.00E00	821 $\mathcal{R}$ 6.74E11 3.00E00	822 $\mathcal{R}$ 6.66E11 3.00E00
823 $\mathcal{R}$ 6.60E11 3.00E00	824 $\mathcal{R}$ 6.55E11 3.00E00	825 $\mathcal{R}$ 7.97E11 3.00E00
826 $\mathcal{R}$ 7.62E11 3.00E00	827 $\mathcal{R}$ 7.36E11 3.00E00	828 $\mathcal{R}$ 7.15E11 3.00E00
829 $\mathcal{R}$ 6.99E11 3.00E00	830 $\mathcal{R}$ 6.85E11 3.00E00	831 $\mathcal{R}$ 6.73E11 3.00E00
832 $\mathcal{R}$ 6.63E11 3.00E00	833 $\mathcal{R}$ 6.55E11 3.00E00	834 $\mathcal{R}$ 6.47E11 3.00E00
835 $\mathcal{R}$ 6.77E11 5.00E00	836 $\mathcal{R}$ 5.63E10 1.38E01	837 $\mathcal{R}$ 9.81E10 1.38E01
838 $\mathcal{R}$ 1.31E11 1.38E01	839 $\mathcal{R}$ 1.57E11 1.38E01	840 $\mathcal{R}$ 1.78E11 1.38E01
841 $\mathcal{R}$ 1.96E11 1.38E01	842 $\mathcal{R}$ 2.11E11 1.38E01	843 $\mathcal{R}$ 2.24E11 1.38E01
844 $\mathcal{R}$ 2.35E11 1.38E01	845 $\mathcal{R}$ 2.45E11 1.38E01	846 $\mathcal{R}$ 9.84E10 1.38E01
847 $\mathcal{R}$ 1.29E11 1.38E01	848 $\mathcal{R}$ 1.52E11 1.38E01	849 $\mathcal{R}$ 1.71E11 1.38E01
850 $\mathcal{R}$ 1.87E11 1.38E01	851 $\mathcal{R}$ 2.00E11 1.38E01	852 $\mathcal{R}$ 2.11E11 1.38E01
853 $\mathcal{R}$ 2.20E11 1.38E01	854 $\mathcal{R}$ 2.28E11 1.38E01	855 $\mathcal{R}$ 2.35E11 1.38E01
856 $\mathcal{R}$ 1.31E11 1.38E01	857 $\mathcal{R}$ 1.53E11 1.37E01	858 $\mathcal{R}$ 1.69E11 1.38E01
859 $\mathcal{R}$ 1.83E11 1.38E01	860 $\mathcal{R}$ 1.94E11 1.38E01	861 $\mathcal{R}$ 2.03E11 1.38E01
862 $\mathcal{R}$ 2.11E11 1.38E01	863 $\mathcal{R}$ 2.17E11 1.38E01	864 $\mathcal{R}$ 2.23E11 1.38E01
865 $\mathcal{R}$ 2.29E11 1.37E01	866 $\mathcal{R}$ 1.58E11 1.37E01	867 $\mathcal{R}$ 1.72E11 1.37E01
868 $\mathcal{R}$ 1.83E11 1.38E01	869 $\mathcal{R}$ 1.92E11 1.37E01	870 $\mathcal{R}$ 2.00E11 1.37E01
871 $\mathcal{R}$ 2.06E11 1.37E01	872 $\mathcal{R}$ 2.11E11 1.37E01	873 $\mathcal{R}$ 2.16E11 1.37E01
874 $\mathcal{R}$ 2.20E11 1.37E01	875 $\mathcal{R}$ 2.23E11 1.37E01	876 $\mathcal{R}$ 1.80E11 1.37E01
877 $\mathcal{R}$ 1.88E11 1.37E01	878 $\mathcal{R}$ 1.95E11 1.37E01	879 $\mathcal{R}$ 2.00E11 1.37E01
880 $\mathcal{R}$ 2.04E11 1.37E01	881 $\mathcal{R}$ 2.08E11 1.37E01	882 $\mathcal{R}$ 2.11E11 1.37E01
883 $\mathcal{R}$ 2.14E11 1.37E01	884 $\mathcal{R}$ 2.17E11 1.37E01	885 $\mathcal{R}$ 2.19E11 1.37E01
886 $\mathcal{R}$ 1.98E11 1.37E01	887 $\mathcal{R}$ 2.01E11 1.37E01	888 $\mathcal{R}$ 2.04E11 1.37E01
889 $\mathcal{R}$ 2.07E11 1.36E01	890 $\mathcal{R}$ 2.09E11 1.36E01	891 $\mathcal{R}$ 2.10E11 1.37E01
892 $\mathcal{R}$ 2.12E11 1.36E01	893 $\mathcal{R}$ 2.13E11 1.37E01	894 $\mathcal{R}$ 2.14E11 1.37E01
895 $\mathcal{R}$ 2.15E11 1.37E01	896 $\mathcal{R}$ 2.14E11 1.36E01	897 $\mathcal{R}$ 2.13E11 1.36E01

898 $\mathcal{R}$ 2.13E11 1.36E01	899 $\mathcal{R}$ 2.12E11 1.37E01	900 $\mathcal{R}$ 2.12E11 1.36E01
901 $\mathcal{R}$ 2.12E11 1.36E01	902 $\mathcal{R}$ 2.12E11 1.36E01	903 $\mathcal{R}$ 2.12E11 1.36E01
904 $\mathcal{R}$ 2.12E11 1.36E01	905 $\mathcal{R}$ 2.12E11 1.36E01	906 $\mathcal{R}$ 2.27E11 1.36E01
907 $\mathcal{R}$ 2.23E11 1.36E01	908 $\mathcal{R}$ 2.20E11 1.36E01	909 $\mathcal{R}$ 2.17E11 1.36E01
910 $\mathcal{R}$ 2.15E11 1.36E01	911 $\mathcal{R}$ 2.14E11 1.36E01	912 $\mathcal{R}$ 2.13E11 1.36E01
913 $\mathcal{R}$ 2.12E11 1.35E01	914 $\mathcal{R}$ 2.11E11 1.36E01	915 $\mathcal{R}$ 2.10E11 1.36E01
916 $\mathcal{R}$ 2.39E11 1.36E01	917 $\mathcal{R}$ 2.31E11 1.36E01	918 $\mathcal{R}$ 2.26E11 1.36E01
919 $\mathcal{R}$ 2.22E11 1.36E01	920 $\mathcal{R}$ 2.18E11 1.36E01	921 $\mathcal{R}$ 2.15E11 1.36E01
922 $\mathcal{R}$ 2.13E11 1.36E01	923 $\mathcal{R}$ 2.11E11 1.35E01	924 $\mathcal{R}$ 2.09E11 1.36E01
925 $\mathcal{R}$ 2.08E11 1.35E01	926 $\mathcal{R}$ 2.49E11 1.36E01	927 $\mathcal{R}$ 2.39E11 1.36E01
928 $\mathcal{R}$ 2.31E11 1.36E01	929 $\mathcal{R}$ 2.25E11 1.36E01	930 $\mathcal{R}$ 2.21E11 1.35E01
931 $\mathcal{R}$ 2.17E11 1.35E01	932 $\mathcal{R}$ 2.13E11 1.35E01	933 $\mathcal{R}$ 2.10E11 1.36E01
934 $\mathcal{R}$ 2.08E11 1.35E01	935 $\mathcal{R}$ 2.06E11 1.35E01	936 $\mathcal{R}$ 2.12E11 1.36E01
937 $\mathcal{R}$ 1.25E00 5.00E00	938 $\mathcal{R}$ 1.25E00 5.00E00	939 $\mathcal{R}$ 1.25E00 5.00E00
940 $\mathcal{R}$ 1.25E00 5.00E00	941 $\mathcal{R}$ 5.36E00 4.97E00	942 $\mathcal{R}$ 1.25E00 5.00E00
943 $\mathcal{R}$ 1.25E00 5.00E00	944 $\mathcal{R}$ 1.25E00 5.00E00	945 $\mathcal{R}$ 1.25E00 5.00E00
946 $\mathcal{R}$ 1.25E00 5.00E00	947 $\mathcal{R}$ 2.21E01 3.00E00	948 $\mathcal{R}$ 1.25E00 5.00E00
949 $\mathcal{R}$ 1.25E00 5.00E00	950 $\mathcal{R}$ 7.19E00 7.04E00	951 $\mathcal{R}$ 2.19E03 3.58E00
952 $\mathcal{R}$ 2.32E06 3.00E00	953 $\mathcal{R}$ 4.90E09 3.00E00	954 $\mathcal{R}$ 8.30E00 5.00E00
955 $\mathcal{R}$ 1.25E00 5.00E00	956 $\mathcal{R}$ 1.25E00 5.00E00	957 $\mathcal{R}$ 1.25E00 5.00E00
958 $\mathcal{R}$ 5.08E00 5.00E00	959 $\mathcal{R}$ 1.25E00 5.00E00	960 $\mathcal{R}$ 1.25E00 5.00E00
961 $\mathcal{R}$ 1.25E00 5.00E00	962 $\mathcal{R}$ 1.25E00 5.00E00	963 $\mathcal{R}$ 1.25E00 5.00E00
964 $\mathcal{R}$ 1.25E00 5.00E00	965 $\mathcal{R}$ 1.25E00 5.00E00	966 $\mathcal{R}$ 3.36E01 3.00E00
967 $\mathcal{R}$ 6.35E01 3.69E00	968 $\mathcal{R}$ 4.39E01 6.27E00	969 $\mathcal{R}$ 1.25E00 5.00E00
970 $\mathcal{R}$ 7.00E00 3.10E01	971 $\mathcal{R}$ 4.52E01 5.21E00	972 $\mathcal{R}$ 4.92E02 3.00E00
973 $\mathcal{R}$ 2.10E02 3.37E00	974 $\mathcal{R}$ 1.51E02 3.00E00	975 $\mathcal{R}$ 1.25E00 5.00E00
976 $\mathcal{R}$ 1.21E03 3.00E00	977 $\mathcal{R}$ 4.19E02 3.00E00	978 $\mathcal{R}$ 1.87E04 3.21E00
979 $\mathcal{R}$ 1.01E03 3.00E00	980 $\mathcal{R}$ 2.50E02 3.00E00	981 $\mathcal{R}$ 1.25E00 5.00E00
982 $\mathcal{R}$ 1.43E02 3.00E00	983 $\mathcal{R}$ 2.56E02 3.00E00	984 $\mathcal{R}$ 6.52E00 5.00E00
985 $\mathcal{R}$ 1.72E01 3.55E00	986 $\mathcal{R}$ 1.17E03 6.56E00	987 $\mathcal{R}$ 1.26E06 3.00E00
988 $\mathcal{R}$ 2.19E08 2.17E02	989 $\mathcal{R}$ 5.74E00 5.00E00	990 $\mathcal{R}$ 1.25E00 5.00E00
991 $\mathcal{R}$ 1.25E00 5.00E00	992 $\mathcal{R}$ 1.25E00 5.00E00	993 $\mathcal{R}$ 5.71E00 5.00E00
994 $\mathcal{R}$ 1.25E00 5.00E00	995 $\mathcal{R}$ 1.25E00 5.00E00	996 $\mathcal{R}$ 1.25E00 5.00E00
997 $\mathcal{R}$ 6.16E00 5.00E00	998 $\mathcal{R}$ 1.25E00 5.00E00	999 $\mathcal{R}$ 6.70E00 5.00E00

1000 $\mathcal{R}$ 1.25E00 5.00E00	1001 $\mathcal{R}$ 2.94E01 5.00E00	1002 $\mathcal{R}$ 6.86E01 3.00E00
1003 $\mathcal{R}$ 8.05E01 3.00E00	1004 $\mathcal{R}$ 1.25E00 5.00E00	1005 $\mathcal{R}$ 7.78E01 3.00E00
1006 $\mathcal{R}$ 5.29E01 3.00E00	1007 $\mathcal{R}$ 2.04E02 3.13E00	1008 $\mathcal{R}$ 7.43E01 9.76E00
1009 $\mathcal{R}$ 3.62E01 1.62E01	1010 $\mathcal{R}$ 1.25E00 5.00E00	1011 $\mathcal{R}$ 2.06E02 2.41E01
1012 $\mathcal{R}$ 6.76E01 2.61E01	1013 $\mathcal{R}$ 2.57E04 3.00E00	1014 $\mathcal{R}$ 1.61E03 3.00E00
1015 $\mathcal{R}$ 5.56E02 3.00E00	1016 $\mathcal{R}$ 1.25E00 5.00E00	1017 $\mathcal{R}$ 3.45E02 3.00E00
1018 $\mathcal{R}$ 5.78E02 3.00E00	1019 $\mathcal{R}$ 1.25E00 5.00E00	1020 $\mathcal{R}$ 1.25E00 5.00E00
1021 $\mathcal{R}$ 1.25E00 5.00E00	1022 $\mathcal{R}$ 1.25E00 5.00E00	1023 $\mathcal{R}$ 1.25E00 5.00E00
1024 $\mathcal{R}$ 1.66E01 4.69E00	1025 $\mathcal{R}$ 3.66E01 4.74E00	1026 $\mathcal{R}$ 2.03E02 4.57E00
1027 $\mathcal{R}$ 1.25E00 5.00E00	1028 $\mathcal{R}$ 1.25E00 5.00E00	1029 $\mathcal{R}$ 1.25E00 5.00E00
1030 $\mathcal{R}$ 1.25E00 5.00E00	1031 $\mathcal{R}$ 1.25E00 5.00E00	1032 $\mathcal{R}$ 1.76E01 5.00E00
1033 $\mathcal{R}$ 3.75E01 5.00E00	1034 $\mathcal{R}$ 2.09E02 3.00E00	1035 $\mathcal{R}$ 1.05E03 5.00E00
1036 $\mathcal{R}$ 1.25E00 2.40E01	1037 $\mathcal{R}$ 6.72E02 3.00E00	1038 $\mathcal{R}$ 9.21E02 5.00E00
1039 $\mathcal{R}$ 1.25E00 1.54E02	1040 $\mathcal{T}$ 1.43E13 3.88E02	1041 $\mathcal{R}$ 1.25E00 5.00E00
1042 $\mathcal{R}$ 1.25E00 5.00E00	1043 $\mathcal{R}$ 1.25E00 5.00E00	1044 $\mathcal{R}$ 1.25E00 5.00E00
1045 $\mathcal{R}$ 1.25E00 5.00E00	1046 $\mathcal{R}$ 1.25E00 5.00E00	1047 $\mathcal{R}$ 1.00E00 5.00E00
1048 $\mathcal{R}$ 1.25E00 5.00E00	1049 $\mathcal{R}$ 1.25E00 5.00E00	1050 $\mathcal{R}$ 1.25E00 5.00E00
1051 $\mathcal{R}$ 1.00E00 5.00E00	1052 $\mathcal{R}$ 1.25E00 5.00E00	1053 $\mathcal{R}$ 1.25E00 5.00E00
1054 $\mathcal{R}$ 1.25E00 5.00E00	1055 $\mathcal{R}$ 1.00E00 5.00E00	1056 $\mathcal{R}$ 1.00E00 5.00E00
1057 $\mathcal{R}$ 1.25E00 5.00E00	1058 $\mathcal{R}$ 1.00E00 5.00E00	1059 $\mathcal{R}$ 1.25E00 5.00E00
1060 $\mathcal{R}$ 1.25E00 5.00E00	1061 $\mathcal{R}$ 1.25E00 5.00E00	1062 $\mathcal{R}$ 1.25E00 5.00E00
1063 $\mathcal{R}$ 1.25E00 5.00E00	1064 $\mathcal{R}$ 1.25E00 5.00E00	1065 $\mathcal{R}$ 1.25E00 5.00E00
1066 $\mathcal{R}$ 1.00E00 5.00E00	1067 $\mathcal{R}$ 1.25E00 5.00E00	1068 $\mathcal{R}$ 1.25E00 5.00E00
1069 $\mathcal{R}$ 1.25E00 5.00E00	1070 $\mathcal{R}$ 1.00E00 5.00E00	1071 $\mathcal{R}$ 1.25E00 5.00E00
1072 $\mathcal{R}$ 1.25E00 5.00E00	1073 $\mathcal{R}$ 1.25E00 5.00E00	1074 $\mathcal{R}$ 1.00E00 5.00E00
1075 $\mathcal{R}$ 1.00E00 5.00E00	1076 $\mathcal{R}$ 1.25E00 5.00E00	1077 $\mathcal{R}$ 1.00E00 5.00E00
1078 $\mathcal{R}$ 1.25E00 5.00E00	1079 $\mathcal{R}$ 1.25E00 5.00E00	1080 $\mathcal{R}$ 1.25E00 5.00E00
1081 $\mathcal{R}$ 1.25E00 5.00E00	1082 $\mathcal{R}$ 1.00E00 5.00E00	1083 $\mathcal{R}$ 1.00E00 5.00E00
1084 $\mathcal{R}$ 1.25E00 5.00E00	1085 $\mathcal{R}$ 1.00E00 5.00E00	1086 $\mathcal{R}$ 1.25E00 5.00E00
1087 $\mathcal{R}$ 1.25E00 5.00E00	1088 $\mathcal{R}$ 1.25E00 5.00E00	1089 $\mathcal{R}$ 1.25E00 5.00E00
1090 $\mathcal{R}$ 1.00E00 5.00E00	1091 $\mathcal{R}$ 1.00E00 5.00E00	1092 $\mathcal{R}$ 1.25E00 5.00E00
1093 $\mathcal{R}$ 1.00E00 5.00E00	1094 $\mathcal{R}$ 1.25E00 5.00E00	1095 $\mathcal{T}$ 3.75E18 6.41E00
1096 $\mathcal{T}$ 8.54E18 4.59E00	1097 $\mathcal{T}$ 4.42E19 5.05E00	1098 $\mathcal{T}$ 2.29E19 3.59E00
1099 $\mathcal{T}$ 2.15E20 5.21E00	1100 $\mathcal{R}$ 1.25E00 5.00E00	1101 $\mathcal{R}$ 1.25E00 5.00E00

1102 $\mathcal{R}$ 1.25E00 5.00E00	1103 $\mathcal{R}$ 1.25E00 5.00E00	1104 $\mathcal{R}$ 6.92E00 5.00E00
1105 $\mathcal{T}$ 1.35E13 5.00E00	1106 $\mathcal{T}$ 2.26E12 5.00E00	1107 $\mathcal{T}$ 2.06E12 5.00E00
1108 $\mathcal{T}$ 1.03E13 5.00E00	1109 $\mathcal{T}$ 7.59E14 4.78E00	1110 $\mathcal{R}$ 2.32E01 3.41E01
1111 $\mathcal{R}$ 8.59E00 5.00E00	1112 $\mathcal{R}$ 2.06E01 5.00E00	1113 $\mathcal{R}$ 4.12E01 5.00E00
1114 $\mathcal{R}$ 6.11E01 5.00E00	1115 $\mathcal{R}$ 8.00E01 3.00E00	1116 $\mathcal{R}$ 9.65E01 3.00E00
1117 $\mathcal{R}$ 1.10E02 5.00E00	1118 $\mathcal{T}$ 4.59E10 5.00E00	1119 $\mathcal{T}$ 1.71E11 5.01E00
1120 $\mathcal{T}$ 8.26E11 5.00E00	1121 $\mathcal{T}$ 3.17E11 5.00E00	1122 $\mathcal{T}$ 1.41E11 4.99E00
1123 $\mathcal{T}$ 1.85E11 5.00E00	1124 $\mathcal{T}$ 5.68E10 4.99E00	1125 $\mathcal{T}$ 1.65E11 5.01E00
1126 $\mathcal{T}$ 8.92E10 4.99E00	1127 $\mathcal{T}$ 1.65E10 5.07E00	1128 $\mathcal{R}$ 1.68E02 3.00E00
1129 $\mathcal{R}$ 1.25E00 3.43E01	1130 $\mathcal{R}$ 1.20E01 5.97E00	1131 $\mathcal{R}$ 2.69E01 4.66E00
1132 $\mathcal{R}$ 4.15E01 3.88E00	1133 $\mathcal{R}$ 5.54E01 3.40E00	1134 $\mathcal{R}$ 6.66E01 3.09E00
1135 $\mathcal{R}$ 7.57E01 3.00E00	1136 $\mathcal{T}$ 1.29E25 5.01E00	1137 $\mathcal{T}$ 4.82E25 5.00E00
1138 $\mathcal{T}$ 2.32E26 5.01E00	1139 $\mathcal{T}$ 8.92E25 5.00E00	1140 $\mathcal{T}$ 3.96E25 5.00E00
1141 $\mathcal{T}$ 5.20E25 5.00E00	1142 $\mathcal{T}$ 1.60E25 5.00E00	1143 $\mathcal{T}$ 4.64E25 5.01E00
1144 $\mathcal{T}$ 2.51E25 5.00E00	1145 $\mathcal{T}$ 4.64E24 5.00E00	1146 $\mathcal{R}$ 1.00E00 5.00E00
1147 $\mathcal{R}$ 1.25E00 5.00E00	1148 $\mathcal{R}$ 2.00E00 5.00E00	1149 $\mathcal{R}$ 2.27E02 3.00E00
1150 $\mathcal{R}$ 1.48E01 4.07E02	1151 $\mathcal{R}$ 1.25E00 5.00E00	1152 $\mathcal{R}$ 1.82E03 7.09E00
1153 $\mathcal{R}$ 1.03E02 8.47E00	1154 $\mathcal{R}$ 1.14E03 5.34E00	1155 $\mathcal{R}$ 1.42E01 6.65E00
1156 $\mathcal{R}$ 2.78E02 1.39E01	1157 $\mathcal{R}$ 7.19E00 1.03E02	1158 $\mathcal{R}$ 2.05E03 3.00E00
1159 $\mathcal{R}$ 1.14E02 3.00E00	1160 $\mathcal{R}$ 1.23E03 3.00E00	1161 $\mathcal{R}$ 8.31E02 3.00E00
1162 $\mathcal{R}$ 1.23E03 3.00E00	1163 $\mathcal{R}$ 1.97E02 3.00E00	1164 $\mathcal{R}$ 8.73E02 3.00E00
1165 $\mathcal{R}$ 3.71E03 3.00E00	1166 $\mathcal{R}$ 1.65E03 3.00E00	1167 $\mathcal{R}$ 4.83E02 3.00E00
1168 $\mathcal{R}$ 1.07E03 3.00E00	1169 $\mathcal{R}$ 2.78E03 3.00E00	1170 $\mathcal{R}$ 1.09E03 3.00E00
1171 $\mathcal{R}$ 4.59E02 3.00E00	1172 $\mathcal{R}$ 6.61E02 3.00E00	1173 $\mathcal{R}$ 9.09E02 3.00E00
1174 $\mathcal{R}$ 3.46E02 3.00E00	1175 $\mathcal{R}$ 1.93E02 3.00E00	1176 $\mathcal{R}$ 2.05E02 3.00E00
1177 $\mathcal{R}$ 2.06E02 3.00E00	1178 $\mathcal{T}$ 1.92E09 4.97E00	1179 $\mathcal{T}$ 1.34E09 4.98E00
1180 $\mathcal{R}$ 1.00E00 5.00E00	1181 $\mathcal{R}$ 1.25E00 5.00E00	1182 $\mathcal{R}$ 1.00E00 5.00E00
1183 $\mathcal{R}$ 3.61E02 3.00E00	1184 $\mathcal{R}$ 2.00E04 3.00E00	1185 $\mathcal{R}$ 4.34E01 3.00E00
1186 $\mathcal{R}$ 1.90E04 3.00E00	1187 $\mathcal{R}$ 1.18E03 3.00E00	1188 $\mathcal{R}$ 1.16E04 3.00E00
1189 $\mathcal{R}$ 1.72E02 3.00E00	1190 $\mathcal{R}$ 7.19E03 3.00E00	1191 $\mathcal{R}$ 1.47E03 3.00E00
1192 $\mathcal{R}$ 2.83E03 3.44E00	1193 $\mathcal{R}$ 1.43E02 4.15E00	1194 $\mathcal{R}$ 1.26E03 5.19E00
1195 $\mathcal{R}$ 6.41E02 6.96E00	1196 $\mathcal{R}$ 1.98E03 4.98E00	1197 $\mathcal{R}$ 3.25E02 4.53E00
1198 $\mathcal{R}$ 1.48E03 4.23E00	1199 $\mathcal{R}$ 6.31E03 4.03E00	1200 $\mathcal{R}$ 2.52E03 3.00E00
1201 $\mathcal{R}$ 7.25E02 3.00E00	1202 $\mathcal{R}$ 1.58E03 3.00E00	1203 $\mathcal{R}$ 4.08E03 3.00E00

1204	$\mathcal{R}$	1.30E03	3.72E00	1205	$\mathcal{R}$	5.41E02	3.81E00	1206	$\mathcal{R}$	7.66E02	3.90E00
1207	$\mathcal{R}$	1.03E03	4.02E00	1208	$\mathcal{R}$	2.51E02	6.54E00	1209	$\mathcal{R}$	1.36E02	6.71E00
1210	$\mathcal{R}$	1.41E02	6.85E00	1211	$\mathcal{R}$	1.41E02	6.86E00	1212	$\mathcal{T}$	5.41E23	5.00E00
1213	$\mathcal{T}$	3.76E23	5.01E00	1214	$\mathcal{T}$	1.97E09	5.01E00	1215	$\mathcal{T}$	5.54E23	5.00E00
1216	$\mathcal{T}$	3.32E12	5.00E00	1217	$\mathcal{T}$	1.17E10	4.99E00	1218	$\mathcal{T}$	9.03E09	5.01E00
1219	$\mathcal{T}$	4.09E09	4.99E00	1220	$\mathcal{T}$	4.15E09	4.99E00	1221	$\mathcal{T}$	1.70E06	5.00E00
1222	$\mathcal{T}$	9.35E26	5.00E00	1223	$\mathcal{T}$	3.29E24	5.00E00	1224	$\mathcal{T}$	2.54E24	5.00E00
1225	$\mathcal{T}$	1.15E24	5.01E00	1226	$\mathcal{T}$	1.17E24	4.99E00	1227	$\mathcal{T}$	4.79E20	5.00E00
1228	$\mathcal{R}$	1.25E00	5.00E00	1229	$\mathcal{R}$	1.37E01	5.00E00	1230	$\mathcal{R}$	5.31E00	5.00E00
1231	$\mathcal{R}$	5.63E00	5.00E00	1232	$\mathcal{R}$	2.15E01	5.00E00	1233	$\mathcal{R}$	1.20E01	5.00E00
1234	$\mathcal{R}$	1.25E00	5.00E00	1235	$\mathcal{T}$	3.07E13	5.00E00	1236	$\mathcal{T}$	8.35E09	5.00E00
1237	$\mathcal{T}$	1.76E11	5.00E00	1238	$\mathcal{T}$	8.85E12	5.00E00	1239	$\mathcal{T}$	3.49E11	6.30E02
1240	$\mathcal{R}$	1.25E00	5.00E00	1241	$\mathcal{R}$	1.79E01	5.00E00	1242	$\mathcal{R}$	5.31E00	5.00E00
1243	$\mathcal{R}$	7.50E00	5.00E00	1244	$\mathcal{R}$	1.29E01	5.00E00	1245	$\mathcal{R}$	1.15E01	7.81E00
1246	$\mathcal{T}$	1.19E08	5.00E00	1247	$\mathcal{T}$	3.08E13	5.00E00	1248	$\mathcal{T}$	1.19E10	5.00E00
1249	$\mathcal{T}$	1.52E11	5.00E00	1250	$\mathcal{T}$	8.77E12	5.00E00	1251	$\mathcal{T}$	5.59E11	5.00E00
1252	$\mathcal{R}$	6.65E04	3.00E00	1253	$\mathcal{R}$	4.59E06	3.00E00	1254	$\mathcal{R}$	1.05E01	1.05E01
1255	$\mathcal{R}$	1.36E01	1.38E01	1256	$\mathcal{R}$	1.78E02	5.83E00	1257	$\mathcal{R}$	2.13E02	6.45E00
1258	$\mathcal{R}$	1.25E00	2.45E01	1259	$\mathcal{R}$	3.72E01	7.59E00	1260	$\mathcal{R}$	2.66E01	9.02E00
1261	$\mathcal{R}$	8.61E00	1.27E01	1262	$\mathcal{R}$	3.00E01	1.81E01	1263	$\mathcal{R}$	6.66E01	1.54E01
1264	$\mathcal{R}$	8.97E02	3.00E00	1265	$\mathcal{R}$	1.47E03	3.00E00	1266	$\mathcal{R}$	3.11E03	3.17E00
1267	$\mathcal{R}$	4.11E03	3.28E00	1268	$\mathcal{R}$	8.35E01	5.35E00	1269	$\mathcal{R}$	8.92E02	6.02E00
1270	$\mathcal{R}$	5.27E02	7.81E00	1271	$\mathcal{R}$	1.10E02	1.71E01	1272	$\mathcal{R}$	8.95E02	1.10E01
1273	$\mathcal{R}$	1.54E03	1.15E01	1274	$\mathcal{R}$	1.36E02	5.00E00	1275	$\mathcal{R}$	1.94E02	5.00E00
1276	$\mathcal{R}$	3.10E02	3.00E00	1277	$\mathcal{R}$	3.53E02	3.00E00	1278	$\mathcal{R}$	4.54E01	5.00E00
1279	$\mathcal{R}$	1.82E01	5.00E00	1280	$\mathcal{R}$	2.99E01	5.00E00	1281	$\mathcal{R}$	8.35E01	5.00E00
1282	$\mathcal{R}$	7.67E01	5.00E00	1283	$\mathcal{R}$	2.33E02	3.00E00	1284	$\mathcal{T}$	2.41E14	5.97E00
1285	$\mathcal{T}$	3.61E14	5.85E00	1286	$\mathcal{T}$	1.04E15	3.60E00	1287	$\mathcal{T}$	1.27E15	3.13E00
1288	$\mathcal{T}$	1.33E13	2.66E01	1289	$\mathcal{T}$	1.17E14	3.00E00	1290	$\mathcal{T}$	2.41E14	4.03E00
1291	$\mathcal{T}$	7.94E12	5.25E01	1292	$\mathcal{T}$	2.88E14	4.44E00	1293	$\mathcal{T}$	8.37E14	4.50E00
1294	$\mathcal{T}$	1.89E04	3.73E05	1295	$\mathcal{T}$	2.03E04	1.10E05	1296	$\mathcal{T}$	2.16E04	8.54E04
1297	$\mathcal{T}$	2.03E04	1.10E05	1298	$\mathcal{T}$	2.16E04	8.54E04	1299	$\mathcal{T}$	1.15E03	5.00E00
1300	$\mathcal{T}$	7.19E02	5.00E00	1301	$\mathcal{T}$	7.19E02	5.00E00	1302	$\mathcal{T}$	2.40E10	6.96E05
1303	$\mathcal{T}$	5.98E09	1.59E06	1304	$\mathcal{T}$	8.07E09	3.05E05	1305	$\mathcal{T}$	5.98E09	1.59E06

1306 $\mathcal{T}$ 8.07E09 3.05E05	1307 $\mathcal{T}$ 2.64E17 4.99E00	1308 $\mathcal{T}$ 1.43E17 4.99E00
1309 $\mathcal{T}$ 1.43E17 4.99E00	1310 $\mathcal{T}$ 1.89E04 4.01E05	1311 $\mathcal{T}$ 2.03E04 1.23E05
1312 $\mathcal{T}$ 2.16E04 7.53E04	1313 $\mathcal{T}$ 2.03E04 1.23E05	1314 $\mathcal{T}$ 2.16E04 7.53E04
1315 $\mathcal{T}$ 2.15E02 5.00E00	1316 $\mathcal{T}$ 2.10E02 5.00E00	1317 $\mathcal{T}$ 2.10E02 5.00E00
1318 $\mathcal{T}$ 1.92E04 8.70E11	1319 $\mathcal{T}$ 2.04E04 4.65E11	1320 $\mathcal{T}$ 2.17E04 1.14E11
1321 $\mathcal{T}$ 2.04E04 4.65E11	1322 $\mathcal{T}$ 2.17E04 1.14E11	1323 $\mathcal{T}$ 5.32E09 5.00E00
1324 $\mathcal{T}$ 7.49E09 5.00E00	1325 $\mathcal{T}$ 7.49E09 5.00E00	

DISTRIBUTION

	COPIES
DOD ACTIVITIES (CONUS)	
DEFENSE TECHNICAL INFORMATION CENTER 8725 JOHN J KINGMAN SUITE 0944 FT BELVOIR VA 22060-6218	2
ATTN CODE A76 (TECHNICAL LIBRARY) COMMANDING OFFICER CSSDD NSWC 6703 W HIGHWAY 98 PANAMA CITY FL 32407-7001	1
ATTN MICHAEL ALBIN CHIEF ANGLO/AMERICAN ACQUISITIONS DIVISION LIBRARY OF CONGRESS 101 INDEPENDENCE AVENUE SE WASHINGTON DC 20540-4170	4

INTERNAL

B60	(TECHNICAL LIBRARY)	3
T10	J SLOOP	1
T10	A DIDONATO	2
T11	T OWEN	1
T11	W FAIRFAX	2
T12	J BLANTON	1
T12	W ORMSBY	1
T12	R GNOFFO	2