THE UNIVERSITY OF WISCONSIN

madison, wisconsin

UNITED STATES ARMY

MATHEMATICS RESEARCH CENTER

MRC Computer Program # 2

Interval Arithmetic Package

(INTERVAL)

Allen Reiter

Title: Interval Arithmetic Package (Interval)		
Programmer: Allen Reiter	Date: June 1965	MRC Program # 2
Program Language: FORTRAN-63/CODAP	Computer: CDC 1604	
Other Info: COOP Organ. Code - WISC		

1.0 Purpose

To provide means for automatic error control by the computer by performing arithmetic operations on intervals containing the exact value, rather than on (approximate) real numbers of finite precision. This package is designed for use with FORTRAN '63, and enables the user to treat intervals as ordinary variables.

2.0 Usage

An interval number is a pair of real numbers (a,b), with $a \leq b$. For a full description of the rules governing interval arithmetic, see [1]. Herein we describe how the user can use this package within the scope of the FORTRAN '63 compiler.

The user designates certain variables to be interval-valued by the use of TYPE - designation statements (see FORTRAN '63 reference manual, Chapter 5). Interval numbers are of type 5. Thus the statement

TYPE INT5 (2) X, Y, MU

designates the variables X, Y, and MU as being interval-valued. The complier automatically assigns two consecutive storage cells (corresponding to the left end-point and the right end-point respectively of the interval) to each of the variables. The user then refers to all interval variables as he might to an ordinary (real-valued) variable, and uses them in arithmetic assignment statements just as if they were ordinary numbers. This usage, however, is subject to certain restrictions.

We give some examples. Let A, B, C, D all be of type INT5; let E, F, and G be of type REAL, and let K and L be of type INTEGER.

(a) The statement

A = C + D - 3*(K/B - L*E)

causes the right-hand side of the equation to be evaluated in interval arithmetic, and the interval thus computed is stored in A.

(b) The statement

A = E*F + K

causes the right-hand side to be evaluated in real arithmetic. This result is is then converted to interval form (i.e. an interval with both end-points being equal) and stored in A.

(c) The statement

E = L*A

will compute the interval L*A, find the mid-point of this interval, and store this in the (REAL) cell E.

(d) The statement

$$K = L * A$$

computes the interval L * A, finds its mid-point, truncates this value to an integer, and stores it in K.

Thus the three types INTEGER, REAL, and INT5 may be freely mixed. No other types may appear in a statement which contains a type INT5 variable, with one important exception.

(e) The statement

$$A = (a, b)$$

where a and b are floating-point constants such that $b \geq a$ will assign to A the interval (a, b). This works because the FORTRAN '63 compiler thinks that (a, b) is a complex number, and generates the proper calling sequence to the INTERVAL subroutine. See also the section on input-output.

(f) Arithmetic operations. Addition, subtraction, and multiplication in interval mode are straight-forward, governed by the rules described in [1]. Division likewise is straight-forward, except that an attempt to divide by an interval containing zero will produce an error indication (see section on errors.) Exponentiation, however, has several cases.

Let A and B be of type INT5, F and G of type REAL, and K and L of type INTEGER. The expression

$$A ** B$$

is an interval which is exactly the set of all values of x^y, where x ranges over A and y ranges over B. An error indication will be given if the left end-point of A is not positive. The expressions

$$F ** B \quad \text{and} \quad K ** B$$

where F and K are positive also give the range respectively of F^y and K^y as y ranges over B. The expression

$$A ** L ; \quad L \geq 0$$

will give the product of A with itself taken L times; if zero is interior to A, then A ** L is an interval much wider than the range of values of x^L for x in A, and in any case always contains the latter. No error indication can be triggered by this expression. The expression

$$A ** F$$

where F is an integer in floating-point form ($|F| \leq 2^{35}-1$) will give the exact range of values of x^F. No error indication will result if $F > 0$, even though the left end-point of A is negative. If F is negative and A contains zero, the error indicator will be triggered; as long as A does not contain zero, the exponentiation will take place. If F is not an exact integer (or if $|F|>2^{35}-1$), then A ** F is defined only as long as the left end-point of A is positive. The expression

$$A ** (-L), \quad L > 0$$

is also defined only if

$$A ** L$$

does not contain zero, and is equal to 1/(A ** L).

(g) In addition to the arithmetic functions, the user is provided with the following elementary transcendental functions: LOGINT, EXPINT, SININT, COSINT, and ATANINT. The usage for all of them is the same; for example

CALL LOGINT (ARGUMENT, ANSWER)

will assign to the interval ANSWER the range of values of LOG(x) as x ranges over the interval ARGUMENT. LOGINT will produce an error indication if ARGUMENT has a non-positive left end-point; the other routines have no error indications.

2.3 External subroutines.

INTERVAL makes use of the following subroutines, all of which are part of the FORTRAN '63 system library:

	DLOG	DEXP
DSIN	DCOS	DATAN

2.7 Error processing.

The user has to reserve one cell of labeled common, by using the statement

COMMON/ OVERFLOW/ INDCATOR

for communication with the INTERVAL package. If an error condition is triggered by an attempt to perform an illegal operation (such as, for instance, attempting to divide by an interval containing zero) the contents of the cell INDCATOR are set to 1. The result of an illegal operation is not defined. Control flow follows the normal sequence: it is therefore up to the user to do something about the error condition, if it exists.

2.10 Input and Output.

FORTRAN '63, while making excellent provisions for evaluating TYPE OTHER arithmetic statements, has entirely neglected the question of getting the values of TYPE OTHER variables in/out of the computer. Fortunately, in our case, interval numbers formally are identical to complex numbers; that is, they both are of the form (a, b). The user can therefore use the same input-output format as he would for TYPE COMPLEX numbers.

Example. Suppose X is an array of interval numbers, and it is desired to print out the first 12 values of X, four intervals per line. This can be accomplished by the statements

```
      PRINT 1, (X(I), I = 1,12)
1     FORMAT(4C(F12.6, F12.6)).
```

There is an alternate method of doing this, which at the same time provides the means for getting at either end-point of an interval number separately. Consider the group of statements

```
      TYPE INT5 (2) X
      DIMENSION X(12), XX(2,12)
      EQUIVALENCE (X,XX).
```

Here XX is an ordinary (TYPE REAL) variable; and for any index J the value of XX(1, J) corresponds to the value of the left end-point of X(J), while XX(2, J) corresponds to the right end-point of X(J). Thus, to print out the array X in the same format as above, the user can also give the command

```
      PRINT 2, (XX(1,I),XX(2,I), I = 1,12)
2     FORMAT ( 8F12.6)
```

All the above comments also apply, of course, to input statements as well.

2.13 Accuracy. Exact, and then some - see section on method.

2.14 Cautions to user.

a) This program, like all MRC programs, is not designed for general utility. Rather, the motivation is to see how a certain scheme can be implemented on the computer, and also to test its usefulness in the context of various applications. The program is not guaranteed to be completely debugged. Also, it is anticipated that the user may have to change the program substantially to suit his own needs. For this purpose an appendix describing how the program works is included.

b) It is well known that interval arithmetic is not distributive, and the user is cautioned that the way he writes a given statement may greatly affect the width of a computed interval. In general, it is advisable that the user obtain at all times a CODAP - language listing of the compiled version of his program, and wade through it to determine that FORTRAN '63 does indeed give the user what he thinks he should get. (For conventions regarding QnQ subroutines, see first of all the FORTRAN '63 reference manual, p. E-1 ff.)

c) Integers in fixed-point form must not exceed $2^{35}-1$ in absolute value.

2.17 References.

[1] Ramon E. Moore, "The Automatic Analysis and Control of Error in Digital Computation Based on the Use of Interval Numbers". Error in Digital Computation, Volume I, (ed. L. B. Rall), John Wiley & Sons, Inc., 1965.

[2] FORTRAN '63 Reference Manual for the 1604/1604-A. Control Data Corp., Pub. No. 60052900, Revision A, June 1964.

3.0 Method.

Arithmetic operations on interval numbers are performed such that the resulting interval is guaranteed to contain the exact answer. To accomplish this, the computed result for the left end-point is rounded "down", while that for the right end-point is rounded "up". The actual arithmetic operations are performed in fixed point, simulating the hardware floating-point. For increased accuracy, rounding is performed after normalization (unlike the hadrware floating-point). The elementary functions are computed using the library double-precision routines, with the results rounded up or down to single precision as necessary. The whole process is quite slow - no timing figures are available at this moment.

4.0 Acknowledgement.

The author wishes to express his gratitude to Dr. R. E. Moore and Dr. H. J. Wertz for some very valuable suggestions, and especially to Mr. Augustine Chai for his aid in writing and debugging this package.

APPENDIX

Description of the program.

1. INTERVAL makes use of three arithmetic subroutines which simulate the hardware floating-point operations. These are SPOPSADD for simulated addition, SPOPSMUL for multiplication, and SPOPSDIV for division. Each subroutine expects, upon entry, to find its two arguments in the A and Q registers (for SPOPSDIV the dividend is in the A - register, while the divisor is in the Q- register). Upon exit, the lower bound for the result is in the Q-register, while A contains the upper bound. These routines are described in Fig. 1-4.

2. Exponentiation, as already indicated, has several subcases. The most general case, that of B ** C where C is not an integer, is handled simply by computing the answer as EXPINT (C * LOGINT(B)). If the exponent is an integer in fixed point, exponentiation is handled by repeated multiplications in interval arithmetic (see figure 5). For exponents that are integers in floating-point form, repeated multiplication can still be used if the base does not contain zero. If the base does contain zero, we have to distinguish between even and odd exponents. For even exponents, the answer is given by the interval (0, M) where M is the larger in absolute value of the two end-points of the base raised to the proper power. For odd exponents, the answer is given by (N, M), where N is the power of the left end-point, and M the power of the right end-point. (N must be computed so that the result is rounded down, while M is rounded up.) The respective powers are again computed using repeated multiplication, this time on the "real" end-points. See figure 5.

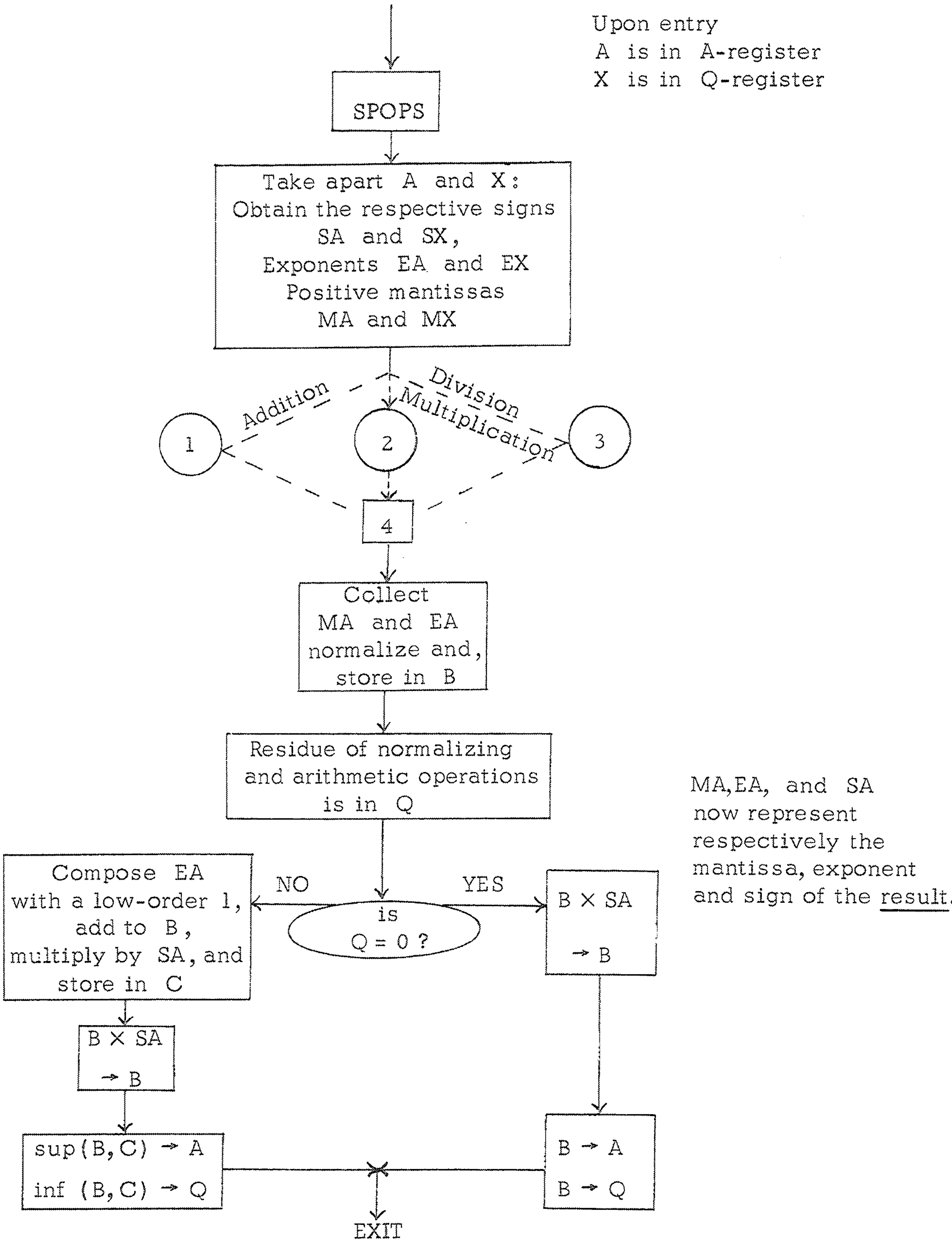

Figure 1 Flow Chart For SPOPS

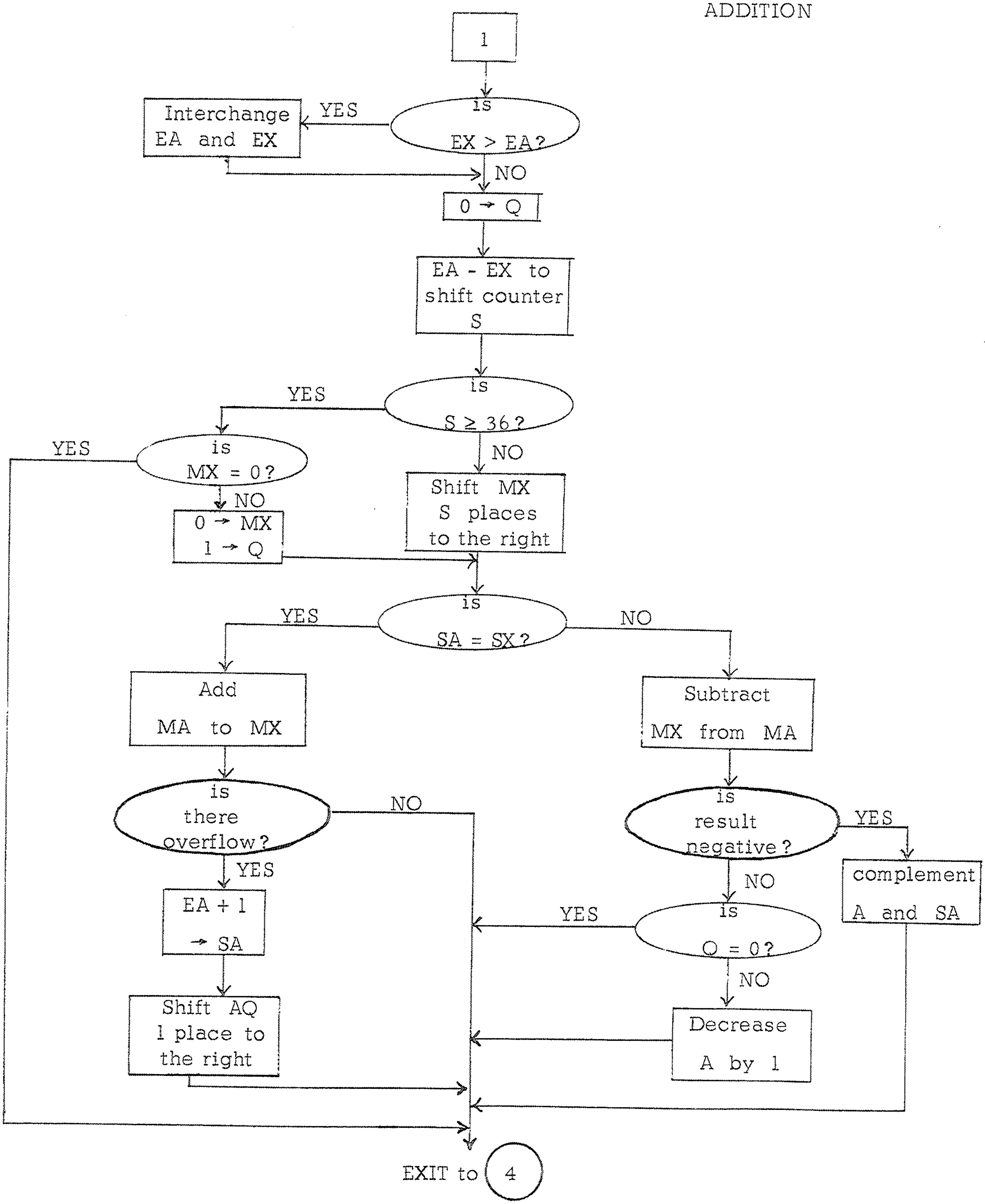

Figure 2 Flow Chart For Addition

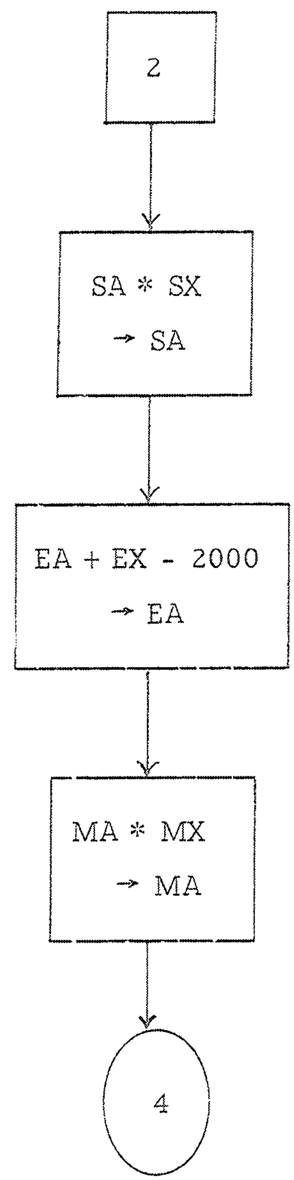

Figure 3 Flow Chart For Multiplication

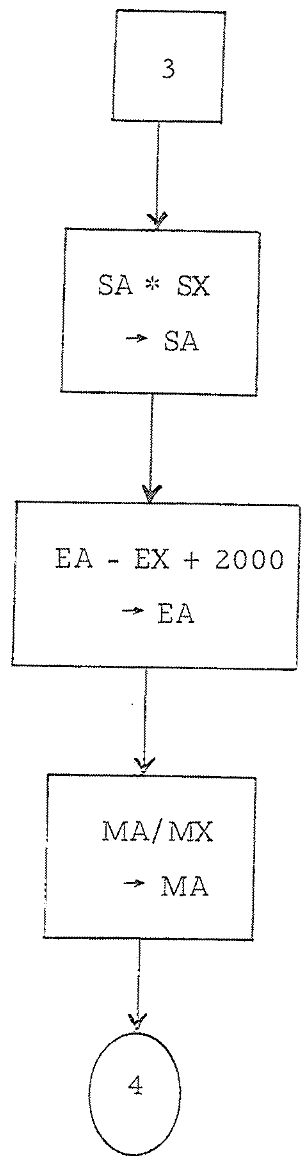

Figure 4 Flow Chart For Division

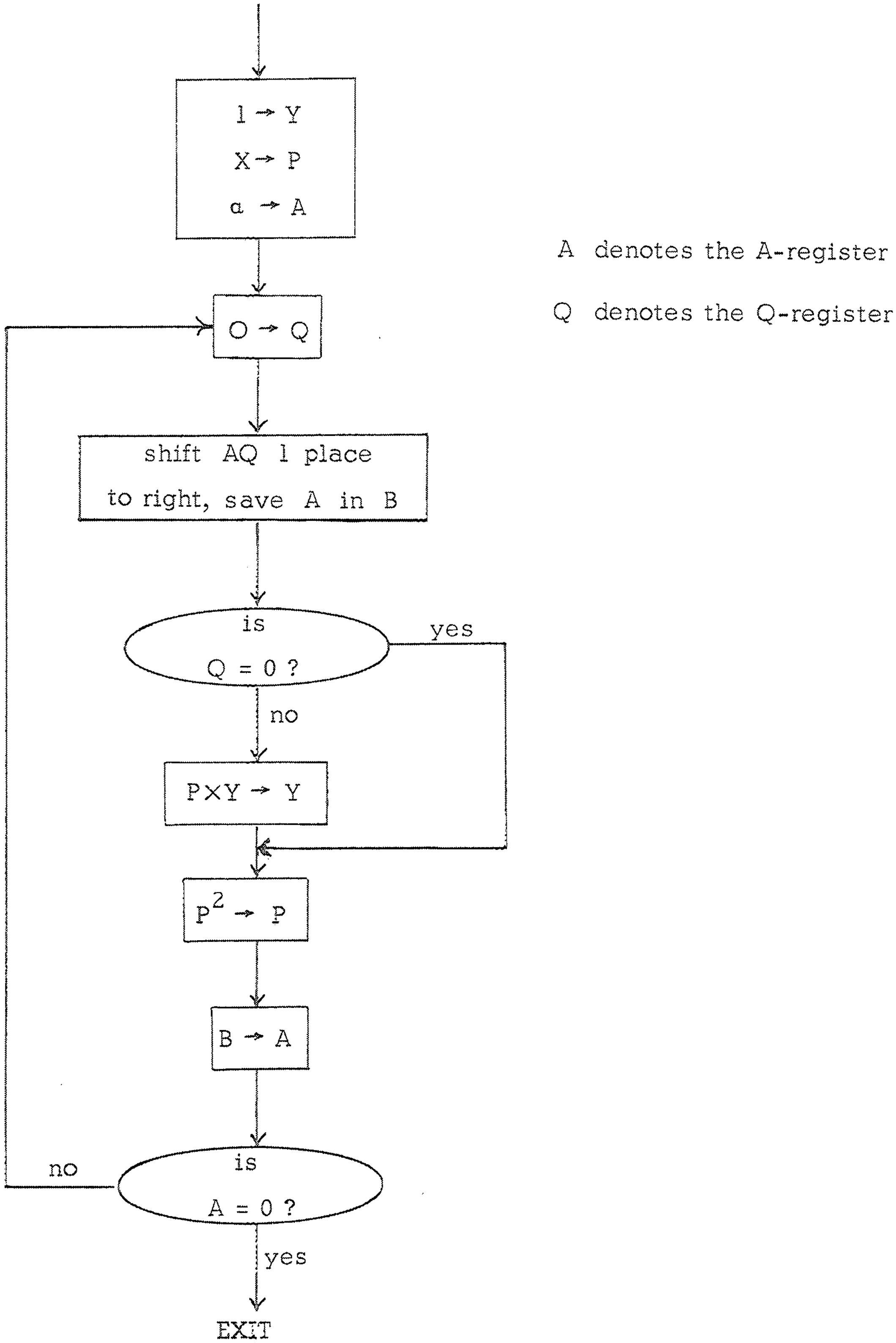

Figure 5 Algorithm for evaluating $Y = X^{\alpha}$ by repeated multiplications

```
         IDENT       INTERVAL          INTERVAL ARITHMETIC PACKAGE     INTVL001
         EXT         ERRORINT                                          INTVL002
         EXT         LOGINT                                            INTVL003
         EXT         EXPINT                                            INTVL004
         EXT         SPOPSADD                                          INTVL005
         EXT         SPOPSMUL                                          INTVL006
         EXT         SPOPSDIV                                          INTVL007
A        BSS         2                                                 INTVL008
B        BSS         2                 ARGUMENTS                       INTVL009
TEMP     BSS         4                                                 INTVL010
         ENTRY       Q1Q00530                                          INTVL011
Q1Q00530 BSS         0                                                 INTVL012
         ENTRY       Q1Q00550          LOAD ACC WITH INTERVAL          INTVL013
Q1Q00550 SLJ         **                                                INTVL014
         RTJ         IN1                                               INTVL015
         LDA     7   A                                                 INTVL016
         STA         =SEL                                              INTVL017
         LDA     7   A+1                                               INTVL018
Q005     STA         =SER                                              INTVL019
         SLJ         EXIT                                              INTVL020
         ENTRY       Q1Q01530                                          INTVL021
Q1Q01530 BSS         0                                                 INTVL022
         ENTRY       Q1Q01550          LOAD NEGATIVE WITH INTERVAL     INTVL023
Q1Q01550 SLJ         **                                                INTVL024
         RTJ         IN1                                               INTVL025
         LAC     7   A+1                                               INTVL026
         STA         EL                                                INTVL027
         LAC     7   A                                                 INTVL028
         SLJ         Q005                                              INTVL029
         ENTRY       Q1Q00500          LDA WITH INTEGER                INTVL030
Q1Q00500 SLJ         **                                                INTVL031
         RTJ         IN1                                               INTVL032
         LDA     7   A                                                 INTVL033
Q054     RTJ         FLOAT                                             INTVL034
Q05      STA         =SEL                                              INTVL035
```

```
         SLJ        Q005                                             INTVL036
         ENTRY      Q1Q01500         LAC   WTH INTEGER               INTVL037
Q1Q01500 SLJ        **                                               INTVL038
         RTJ        IN1                                              INTVL039
         LAC      7 A                                                INTVL040
         SLJ        Q054                                             INTVL041
         ENTRY      Q1Q00510         LDA WITH REAL                   INTVL042
Q1Q00510 SLJ        **                                               INTVL043
         RTJ        IN1                                              INTVL044
         LDA      7 A                                                INTVL045
         SLJ        Q05                                              INTVL046
         ENTRY      Q1Q01510         LAC WITH REAL                   INTVL047
Q1Q01510 SLJ        **                                               INTVL048
         RTJ        IN1                                              INTVL049
         LAC      7 A                                                INTVL050
         SLJ        Q05                                              INTVL051
         ENTRY      Q1Q02500          ADD INTEGER                    INTVL052
Q1Q02500 SLJ        **                                               INTVL053
         RTJ        IN1                                              INTVL054
         LDA     7  A                                                INTVL055
ADINT    RTJ        FLOAT                                            INTVL056
ADREAL   STA        =SSAVA                                           INTVL057
+        LDQ        EL                                               INTVL058
         RTJ        SPOPSADD                                         INTVL059
         STQ        EL                                               INTVL060
         LDA        SAVA                                             INTVL061
+        LDQ        ER                                               INTVL062
         RTJ        SPOPSADD                                         INTVL063
         SLJ        Q005                                             INTVL064
         ENTRY      Q1Q03500          SUBTRACT INTEGER               INTVL065
Q1Q03500 SLJ        **                                               INTVL066
         RTJ        IN1                                              INTVL067
         LAC     7  A                                                INTVL068
         SLJ        ADINT                                            INTVL069
         ENTRY      Q1Q02510          ADD REAL                       INTVL070
```

```
Q1Q02510 SLJ        **                                        INTVL071
         RTJ        IN1                                       INTVL072
         LDA     7  A                                         INTVL073
         SLJ        ADREAL                                    INTVL074
         ENTRY      Q1Q03510          SUBTRACT REAL           INTVL075
Q1Q03510 SLJ        **                                        INTVL076
         RTJ        IN1                                       INTVL077
         LAC     7  A                                         INTVL078
         SLJ        ADREAL                                    INTVL079
         ENTRY      Q1Q02530                                  INTVL080
Q1Q02530 BSS        0                                         INTVL081
         ENTRY      Q1Q02550          ADD INTERVAL            INTVL082
Q1Q02550 SLJ        **                                        INTVL083
         RTJ        IN1                                       INTVL084
AD55     LDA     7  A                                         INTVL085
+        LDQ        EL                                        INTVL086
         RTJ        SPOPSADD                                  INTVL087
         STQ        EL                                        INTVL088
         LDA     7  A+1                                       INTVL089
AD5      LDQ        ER                                        INTVL090
         RTJ        SPOPSADD                                  INTVL091
         SLJ        Q005                                      INTVL092
         ENTRY      Q1Q03530                                  INTVL093
Q1Q03530 BSS        0                                         INTVL094
         ENTRY      Q1Q03550          SUBTRACT INTERVAL       INTVL095
Q1Q03550 SLJ        **                                        INTVL096
         RTJ        IN1                                       INTVL097
SUB55    LAC     7  A+1                                       INTVL098
+        LDQ        EL                                        INTVL099
         RTJ        SPOPSADD                                  INTVL100
         STQ        EL                                        INTVL101
         LAC     7  A                                         INTVL102
         SLJ        AD5                                       INTVL103
         ENTRY      Q1Q02050          ADD TO INTEGER          INTVL104
Q1Q02050 SLJ        **                                        INTVL105
```

```
         RTJ        IN1                                           INTVL106
         RTJ        FLOAT                                         INTVL107
Q205     STA        EL                                            INTVL108
         STA        ER                                            INTVL109
         SLJ        AD55                                          INTVL110
         ENTRY      Q1Q03050          SUBTRACT FROM INTEGER       INTVL111
Q1Q03050 SLJ        **                                            INTVL112
         RTJ        IN1                                           INTVL113
         RTJ        FLOAT                                         INTVL114
Q305     STA        EL                                            INTVL115
         STA        ER                                            INTVL116
         SLJ        SUB55                                         INTVL117
         ENTRY      Q1Q02150          ADD TO REAL                 INTVL118
Q1Q02150 SLJ        **                                            INTVL119
         RTJ        IN1                                           INTVL120
         SLJ        Q205                                          INTVL121
         ENTRY      Q1Q03150          SUBTRACT FROM REAL          INTVL122
Q1Q03150 SLJ        **                                            INTVL123
         RTJ        IN1                                           INTVL124
         SLJ        Q305                                          INTVL125
         ENTRY      Q1Q04500                                      INTVL126
Q1Q04500 SLJ        **                                            INTVL127
         RTJ        IN1                                           INTVL128
         LDA     7  A                                             INTVL129
         RTJ        FLOAT                                         INTVL130
Q0315    STA        =SDL                                          INTVL131
         STA        =SDR                                          INTVL132
MULT     LDA        DL                                            INTVL133
+        LDQ        EL                                            INTVL134
         RTJ        SPOPSMUL                                      INTVL135
         STA        =STMAX                                        INTVL136
         STQ        =STMIN                                        INTVL137
         LDA        DL                                            INTVL138
+        LDQ        ER                                            INTVL139
         RTJ        SPOPSMUL                                      INTVL140
```

```
         RTJ        MAXMIN                                              INTVL141
+        LDA        DR                                                  INTVL142
+        LDQ        EL                                                  INTVL143
         RTJ        SPOPSMUL                                            INTVL144
         RTJ        MAXMIN                                              INTVL145
+        LDA        DR                                                  INTVL146
+        LDQ        ER                                                  INTVL147
         RTJ        SPOPSMUL                                            INTVL148
         RTJ        MAXMIN                                              INTVL149
+        LDA        TMIN                                                INTVL150
         STA        EL                                                  INTVL151
         LDA        TMAX                                                INTVL152
         SLJ        Q005                                                INTVL153
         ENTRY      Q1Q04510           MULTIPLY BY REAL                 INTVL154
Q1Q04510 SLJ        **                                                  INTVL155
         RTJ        IN1                                                 INTVL156
+        LDA     7  A                                                   INTVL157
         SLJ        Q0315                                               INTVL158
         ENTRY      Q1Q04530                                            INTVL159
Q1Q04530 BSS        0                                                   INTVL160
         ENTRY      Q1Q04550           MULTIPLY BY INTERVAL             INTVL161
Q1Q04550 SLJ        **                                                  INTVL162
         RTJ        IN1                                                 INTVL163
Q0455    LDA     7  A                                                   INTVL164
         STA        =SDL                                                INTVL165
         LDA     7  A+1                                                 INTVL166
         STA        =SDR                                                INTVL167
         SLJ        MULT                                                INTVL168
         ENTRY      Q1Q04050           MULTIPLY INTEGER BY INTERVAL     INTVL169
Q1Q04050 SLJ        **                                                  INTVL170
         RTJ        IN1                                                 INTVL171
         RTJ        FLOAT                                               INTVL172
SMUL     STA        EL                                                  INTVL173
         STA        ER                                                  INTVL174
         SLJ        Q0455                                               INTVL175
```

```
         ENTRY      Q1Q04150               MULTIPLY REAL BY INTERVAL      INTVL176
Q1Q04150 SLJ        **                                                    INTVL177
         RTJ        IN1                                                   INTVL178
+        SLJ        SMUL                                                  INTVL179
         ENTRY      Q1Q05530                                              INTVL180
Q1Q05530 BSS        0                                                     INTVL181
         ENTRY      Q1Q05550               DIVIDE BY INTERVAL             INTVL182
Q1Q05550 SLJ        **                                                    INTVL183
         RTJ        IN1                                                   INTVL184
Q555     LDA     7  A                      TEST FOR DIVISION BY ZERO      INTVL185
         AJP        ERRDIV                                                INTVL186
         AJP     2  Q5551                                                 INTVL187
         LDA     7  A+1                                                   INTVL188
         AJP        ERRDIV                                                INTVL189
         AJP     2  ERRDIV                                                INTVL190
Q5551    LDA        F1                                                    INTVL191
+        LDQ     7  A+1                                                   INTVL192
         RTJ        SPOPSDIV                                              INTVL193
         STQ        DL                                                    INTVL194
         LDA        F1                                                    INTVL195
+        LDQ     7  A                                                     INTVL196
         RTJ        SPOPSDIV                                              INTVL197
         STA        DR                                                    INTVL198
         SLJ        MULT                                                  INTVL199
         ENTRY      Q1Q05500               DIVIDE BY INTEGER              INTVL200
Q1Q05500 SLJ        **                                                    INTVL201
         RTJ        IN1                                                   INTVL202
         LDA     7  A                                                     INTVL203
         RTJ        FLOAT                                                 INTVL204
Q055     STA        DL                                                    INTVL205
         AJP        ERRDIV                                                INTVL206
         LDA        F1                                                    INTVL207
+        LDQ        DL                                                    INTVL208
         RTJ        SPOPSDIV                                              INTVL209
         STQ        DL                                                    INTVL210
```

```
         STA        DR                                        INTVL211
         SLJ        MULT                                      INTVL212
         ENTRY      Q1Q05510       DIVIDE BY REAL             INTVL213
Q1Q05510 SLJ        **                                        INTVL214
         RTJ        IN1                                       INTVL215
         LDA     7  A                                         INTVL216
         SLJ        Q055                                      INTVL217
         ENTRY      Q1Q05150       DIVIDE REAL BY INTERVAL    INTVL218
Q1Q05150 SLJ        **                                        INTVL219
         RTJ        IN1                                       INTVL220
Q515     STA        EL                                        INTVL221
         STA        ER                                        INTVL222
         SLJ        Q555                                      INTVL223
         ENTRY      Q1Q05050       DIVIDE INTEGER BY INTERVAL INTVL224
Q1Q05050 SLJ        **                                        INTVL225
         RTJ        IN1                                       INTVL226
         RTJ        FLOAT                                     INTVL227
+        SLJ        Q515                                      INTVL228
ERRDIV   RTJ        ERRORINT                                  INTVL229
+        SLJ        EXIT                                      INTVL230
         ENTRY      Q1Q10050       STORE INTEGER              INTVL231
Q1Q10050 SLJ        **                                        INTVL232
         RTJ        IN1                                       INTVL233
+        RTJ        FLOAT                                     INTVL234
Q100     STA     7  A                                         INTVL235
         STA     7  A+1                                       INTVL236
         SLJ        EXIT                                      INTVL237
         ENTRY      Q1Q10150       STORE REAL                 INTVL238
Q1Q10150 SLJ        **                                        INTVL239
         RTJ        IN1                                       INTVL240
+        SLJ        Q100                                      INTVL241
         ENTRY      Q1Q10550       STORE INTERVAL             INTVL242
Q1Q10550 SLJ        **                                        INTVL243
         RTJ        IN1                                       INTVL244
+        LDA        EL                                        INTVL245
```

```
         STA    7  A                                                    INTVL246
         LDA       ER                                                   INTVL247
         STA    7  A+1                                                  INTVL248
         SLJ       EXIT                                                 INTVL249
         ENTRY     Q1Q10510          STORE INTERVAL IN REAL             INTVL250
Q1Q10510 SLJ       **                -----TAKES AVERAGE                 INTVL251
         RTJ       IN1                                                  INTVL252
         LDA       EL                                                   INTVL253
         FAD       ER                                                   INTVL254
         FDV       =02002400000000000                                   INTVL255
         STA    7  A                                                    INTVL256
         SLJ       EXIT                                                 INTVL257
         ENTRY     Q1Q10500          STORE INTERVAL IN INTEGER          INTVL258
Q1Q10500 SLJ       **                ----- TAKES AVERAGE                INTVL259
         RTJ       IN1                                                  INTVL260
         LDA       EL                                                   INTVL261
         FAD       ER                                                   INTVL262
         FDV       =02002400000000000                                   INTVL263
         RTJ       FIX                                                  INTVL264
+        STA     7 A                                                    INTVL265
         SLJ       EXIT                                                 INTVL266
         ENTRY     Q0Q06500          COMPLEMENT ACC                     INTVL267
Q0Q06500 SLJ       **                                                   INTVL268
         LAC       EL                                                   INTVL269
         LQC       ER                                                   INTVL270
         STQ       EL                                                   INTVL271
         STA       ER                                                   INTVL272
         SLJ       Q0Q06500                                             INTVL273
         ENTRY     Q2Q07055,Q2Q07155,Q2Q07555  INTERVAL EXPONENT - IREATINTVL274
Q2Q07055 BSS       0                           SET OF POWERS            INTVL275
Q2Q07155 BSS       0                                                    INTVL276
Q2Q07555 SLJ       **                                                   INTVL277
         RTJ       IN2                                                  INTVL278
         LDA    7  A+1                                                  INTVL279
         STA       TEMP+3                                               INTVL280
```

```
Q071     LDA    7  A                                                   INTVL281
         STA       TEMP+2               EXPONENT TO TEMP+2             INTVL282
         LDA    7  B                                                   INTVL283
         AJP       ERREXP               NEGATIVE BASE                  INTVL284
         AJP    3  ERREXP                                              INTVL285
         STA       B                                                   INTVL286
         LDA    7  B+1                                                 INTVL287
Q072     STA       B+1                                                 INTVL288
         RTJ       LOGINT                                              INTVL289
+        ZRO    7  B                                                   INTVL290
         ZRO       TEMP                 TEMP=LOG(BASE)                 INTVL291
+        RTJ       Q1Q00550             LDA(TEMP/2)                    INTVL292
         ZRO       TEMP+2                                              INTVL293
+        RTJ       Q1Q04550             FMU(TEMP)                      INTVL294
         ZRO       TEMP                 TEMP*EXPONENT TO ACC           INTVL295
+        RTJ       Q1Q10550                                            INTVL296
         ZRO       TEMP                 TEMP=EXPONENT * LOG(BASE)      INTVL297
-        RTJ       EXPINT                                              INTVL298
+        ZRO       TEMP                                                INTVL299
         ZRO       TEMP+2               TEMP+2=EXPF(TEMP)              INTVL300
         RTJ       Q1Q00550                                            INTVL301
         ZRO       TEMP+2               TEMP+2 TO ACC                  INTVL302
         SLJ       EX2                                                 INTVL303
ERREXP   RTJ       ERRORINT                                            INTVL304
+        SLJ       EX2                                                 INTVL305
         ENTRY     Q2Q07115,Q2Q07515,Q2Q07015                          INTVL306
Q2Q07015 BSS       0                                                   INTVL307
Q2Q07115 BSS       0                                                   INTVL308
Q2Q07515 SLJ       **                   REAL TO INTERVAL POWER         INTVL309
         RTJ       IN2                                                 INTVL310
         LDA    7  B                                                   INTVL311
RTIP     AJP       ERREXP                                              INTVL312
         AJP    3  ERREXP                                              INTVL313
         STA       B                                                   INTVL314
         LDQ    7  A                                                   INTVL315
```

```
         STQ        TEMP+2                                               INTVL316
         LDQ     7  A+1                                                  INTVL317
         STQ        TEMP+3                                               INTVL318
         SLJ        Q072                                                 INTVL319
         ENTRY      Q2Q07005,Q2Q07105,Q2Q07505                           INTVL320
Q2Q07005 BSS        0                                                    INTVL321
Q2Q07105 BSS        0                                                    INTVL322
Q2Q07505 SLJ        **                    INTEGER TO INTERVAL POWER      INTVL323
         RTJ        IN2                                                  INTVL324
         LDA     7  B                                                    INTVL325
         RTJ        FLOAT                                                INTVL326
         SLJ        RTIP                                                 INTVL327
         ENTRY      Q2Q07050,Q2Q07150,Q2Q07550  INTEGER EXPONENTS        INTVL328
Q2Q07050 BSS        0                     -TREATED AS INTERVAL PRODUCTS  INTVL329
Q2Q07150 BSS        0                       TAKEN N TIMES                INTVL330
Q2Q07550 SLJ        **                                                   INTVL331
         RTJ        IN2                                                  INTVL332
         LDA     7  A                     J=EXPONENT                     INTVL333
         RTJ        TESTEXP                                              INTVL334
REMULTI  LDQ     7  B                                                    INTVL335
         STQ        TEMP                                                 INTVL336
         LDQ     7  B+1                                                  INTVL337
         STQ        TEMP+1                Y=BASE                         INTVL338
         LDQ        F1                                                   INTVL339
         STQ        TEMP+2                                               INTVL340
         STQ        TEMP+3                P=1                            INTVL341
LOOP     ENQ        0                                                    INTVL342
         LRS        1                                                    INTVL343
         STA        =SJ                   J=J/2                          INTVL344
         QJP        PASS                  SKIP IF LOW-ORDER BIT=0        INTVL345
+        RTJ        Q1Q00550                                             INTVL346
         ZRO        TEMP                  LDA (TEMP)                     INTVL347
+        RTJ        Q1Q04550                                             INTVL348
         ZRO        TEMP+2                FMU(TEMP+2)                    INTVL349
+        RTJ        Q1Q10550              STA(TEMP+2)                    INTVL350
```

```
         ZRO        TEMP+2                 P=P*Y                         INTVL351
PASS     RTJ        Q1Q00550               LDA(TEMP)                     INTVL352
         ZRO        TEMP                                                 INTVL353
         RTJ        Q1Q04550               FMU(TEMP)                     INTVL354
         ZRO        TEMP                                                 INTVL355
         RTJ        Q1Q10550               STA(TEMP)                     INTVL356
         ZRO        TEMP                   Y=Y*Y                         INTVL357
         LDA        J                                                    INTVL358
         AJP     1  LOOP                   END IF J=0                    INTVL359
ENDEXP   LDQ        SEXP                                                 INTVL360
         QJP     2  EXPOS                                                INTVL361
         RTJ        Q1Q00510                                             INTVL362
         ZRO        F1                     LDA   1.                      INTVL363
         RTJ        Q1Q05550                                             INTVL364
         ZRO        TEMP+2                 FDV B                         INTVL365
         SLJ        EX2                                                  INTVL366
EXPOS    RTJ        Q1Q00550               LDA B                         INTVL367
         ZRO        TEMP+2                                               INTVL368
         SLJ        EX2                    EXIT                          INTVL369
TESTEXP  SLJ        **                     CHECK FOR NEGATIVE EXPONENT   INTVL370
         AJP      2 POSEXP                                               INTVL371
         ENQ        -1                     -1 FOR NEGATIVE EXPONENT      INTVL372
         SCM        ALSEV                  COMPLEMENT A                  INTVL373
         STQ        =SSEXP                                               INTVL374
         SLJ        TESTEXP                                              INTVL375
POSEXP   ENQ        1                      +1 FOR POSITIVE EXPONENT      INTVL376
         SLJ        *-1                                                  INTVL377
         ENTRY      Q2Q07553                                             INTVL378
Q2Q07553 BSS        0                                                    INTVL379
         ENTRY      Q2Q07051,Q2Q07151,Q2Q07551   REAL EXPONENT - SET OF PINTVL380
Q2Q07051 BSS        0                                                    INTVL381
Q2Q07151 BSS        0                                                    INTVL382
Q2Q07551 SLJ        **                                                   INTVL383
         RTJ        IN2                                                  INTVL384
+        LDA     7  A                                                    INTVL385
```

```
         RTJ          FIX              FIX EXPONENT                      INTVL386
+        STA          =SK1             TEST FOR INTEGER                  INTVL387
         RTJ          FLOAT                                              INTVL388
+        FSB     7    A                                                  INTVL389
         AJP          WHOLE            EXACT INTEGER IF A=0              INTVL390
         LDA     7    A                                                  INTVL391
         STA          TEMP+3                                             INTVL392
         SLJ          Q071                                               INTVL393
WHOLE    LDA          K1                                                 INTVL394
         RTJ          TESTEXP                                            INTVL395
         LDQ     7    B                                                  INTVL396
         QJP     2    REMULTI          IF B DOES NOT CONTAIN ZERO,       INTVL397
         LDQ     7    B+1              ALGORITHM IS SAME AS INTEGER EXPINTVL398
         QJP     3    REMULTI                                            INTVL399
         STA          K1                                                 INTVL400
         ENQ          1                                                  INTVL401
         LDL          K1               TEST FOR EVEN OR ODD              INTVL402
         AJP          EVEN                                               INTVL403
         LDA          K1                                                 INTVL404
         LDQ     7    B                ODD EXPONENT                      INTVL405
         RTJ          REPEAT1          OBTAIN LEFT END-POINT BY          INTVL406
+        STQ          TEMP+1           REPEATED MULTIPLICATIONS          INTVL407
         SLJ          REPEAT2                                            INTVL408
+        LDQ          TEMP+1           LOWER RESULT                      INTVL409
         STQ          TEMP+2                                             INTVL410
         LDQ     7    B+1                                                INTVL411
UPPER    LDA          K1               OBTAIN RIGHT ENDPOINT             INTVL412
         RTJ          REPEAT1                                            INTVL413
+        STA          TEMP+1           UPPER RESULT                      INTVL414
         SLJ          REPEAT2                                            INTVL415
+        LDQ          TEMP+1                                             INTVL416
         STQ          TEMP+3                                             INTVL417
         SLJ          ENDEXP                                             INTVL418
EVEN     ENQ          0                EVEN EXPONENT                     INTVL419
         STQ          TEMP+2           SET LEFT END-POINT TO ZERO        INTVL420
```

```
         LAC     7  B                                              INTVL421
         LDQ     7  B+1            FIND BIGGER(IN ABSOLUTE         INTVL422
         STQ        TEMP+3         VALUE) OF THE TWO END-POINTS    INTVL423
+        THS        TEMP+3                                         INTVL424
         STA        TEMP+3                                         INTVL425
         LDQ        TEMP+3                                         INTVL426
         SLJ        UPPER                                          INTVL427
REPEAT1  SLJ        **                                             INTVL428
         STQ        TEMP           Y=BASE                          INTVL429
         LDQ        F1                                             INTVL430
         STQ        TEMP+1         P=1.                            INTVL431
LOOPLOOP ENQ        0                                              INTVL432
         LRS        1              J=J/2                           INTVL433
         STA        J                                              INTVL434
         QJP        REPEAT2        SKIP MULTIPLICATION OF P BY Y   INTVL435
         LDA        TEMP           IF LOW-ORDER BIT OF J IS ZERO   INTVL436
         LDQ        TEMP+1                                         INTVL437
         RTJ        SPOPSMUL       P=P*Y                           INTVL438
+        SLJ        REPEAT1        EXIT TO STA/STQ                 INTVL439
REPEAT2  LDA        TEMP                                           INTVL440
         LDQ        TEMP                                           INTVL441
         RTJ        SPOPSMUL       Y=Y*Y                           INTVL442
+        STA        TEMP                                           INTVL443
         LDA        J                                              INTVL444
         AJP     1  LOOPLOOP       END IF J=0                      INTVL445
         LDA        REPEAT1        SET UP EXIT                     INTVL446
         ARS        24                                             INTVL447
         INA        1                                              INTVL448
         SAU        *+1                                            INTVL449
+        SLJ        **             EXIT                            INTVL450
IN1      SLJ        **             INITIALIZATION FOR Q1Q          INTVL451
         SIU     1  END1                                           INTVL452
         LIU     1  IN1                                            INTVL453
         INI     1  -1                                             INTVL454
         SIU     1  *+1            ADD. OF CALL TO IN1             INTVL455
```

```
         SIU    1  EXIT           SET UP EXIT                 INTVL456
+        LIU    1  **                                         INTVL457
         INI    1  -1             ADD. OF CALL TO Q1Q         INTVL458
         STA       =SSAVA                                     INTVL459
         LDA    1  0                                          INTVL460
         STA       A                                          INTVL461
         INA       1                                          INTVL462
         STA       A+1                                        INTVL463
         LDA       SAVA           RESTORE A                   INTVL464
END1     ENI    1  **             AND I1                      INTVL465
         SLJ       IN1            EXIT                        INTVL466
IN2      SLJ       **             INITIALIZATION FOR Q2Q      INTVL467
         SIU    1  END2                                       INTVL468
         LIU    1  IN2            ADD.+1 OF CALL TO IN2       INTVL469
         INI    1  -1                                         INTVL470
         SIU    1  *+1                                        INTVL471
         STA       SAVA           SAVE A                      INTVL472
+        LIU    1  **             ADD.+1 OF CALL TO Q2Q       INTVL473
         SIU    1  EX2                                        INTVL474
         INI    1  -1                                         INTVL475
         LDA    1  0              CALL TO Q2Q                 INTVL476
         STA       A              EXPONENT                    INTVL477
         INA       1                                          INTVL478
         STA       A+1                                        INTVL479
         INI    1  -1                                         INTVL480
         LDA    1  0                                          INTVL481
         STA       B              BASE                        INTVL482
         INA       1                                          INTVL483
         STA       B+1                                        INTVL484
         LDA       SAVA           RESTORE A                   INTVL485
END2     ENI    1  **             RESTORE I1                  INTVL486
         SLJ       IN2                                        INTVL487
EXIT     SLJ       **             EXIT FOR Q1Q-S              INTVL488
EX2      SLJ       **             EXIT FOR Q2Q-S              INTVL489
FIX      SLJ       **                                         INTVL490
```

```
         AJP       3 FIXNEG           EXIT FOR NEGATIVE                   INTVL491
         FAD         HEXP             POSITION COEFFICIENT IN A           INTVL492
         SCL         HEXP             CLEAR HIGH BITS                     INTVL493
         SLJ         FIX                                                  INTVL494
FIXNEG   FSB         HEXP                                                 INTVL495
         SST         HEXP                                                 INTVL496
         SLJ         FIX                                                  INTVL497
HEXP     OCT         2044400000000000                                     INTVL498
FLOAT    SLJ         **               FLOAT FIXED-POINT NUMBER            INTVL499
         AJP       3 FLN                                                  INTVL500
         SST         MONE             PUT IN EXPONENT                     INTVL501
         FAD         PZERO            NORMALIZE                           INTVL502
         SLJ         FLOAT                                                INTVL503
FLN      SCL         MONE             PUT IN NEGATIVE EXPONENT            INTVL504
         FAD         PZERO                                                INTVL505
         SLJ         FLOAT                                                INTVL506
MAXMIN   SLJ         **                                                   INTVL507
+        THS         =STMAX           TEST FOR MAXIMUM                    INTVL508
         STA         TMAX             A.GT.TMAX                           INTVL509
         STQ         =SSAVQ                                               INTVL510
         LDA         SAVQ                                                 INTVL511
+        THS         =STMIN           TEST FOR MINUMUM                    INTVL512
         SLJ         MAXMIN                                               INTVL513
         STA         TMIN             A .LE. TMIN                         INTVL514
         SLJ         MAXMIN                                               INTVL515
ALSEV    OCT         -0                                                   INTVL516
MONE     OCT         2044000000000000                                     INTVL517
PZERO    OCT         0                                                    INTVL518
F1       OCT         2001400000000000                                     INTVL519
         END                                                              INTVL520
         IDENT       ERRORINT         ERROR ROUTINE FOR THE               INTVL521
         ENTRY       ERRORINT         INTERVAL ARITHMETIC PACKAGE         INTVL522
OVERFLOW BLOCK       1                                                    INTVL523
         COMMON      INDCATOR         OVERFLOW INDICATOR CELL             INTVL524
         ORGR        *                                                    INTVL525
```

```
ERRORINT SLJ       **                                                 INTVL526
         ENQ       1                                                  INTVL527
         STQ       INDCATOR               SET INDCATOR TO +1 FOR EXISTING INTVL528
         SLJ       ERRORINT                ERROR CONDITION            INTVL529
         END                                                          INTVL530
      SUBROUTINE LOGINT(ARG,ANS)                                      INTVL531
      DIMENSION ARG(2),ANS(2)                                         INTVL532
      TYPE DOUBLE X,PEP,MEP,PI2,LI,RI,AL,AR,T1,T2                     INTVL533
      DATA(PEP=1.000000000015D),(MEP=.999999999985D),(EP=16574000000000INTVL534
     10B),(PI2=6.2831853071795864769252D),(PI3=9.42477796076938D)     INTVL535
     2(PI=3.14159265358980)                                           INTVL536
      IF(ARG(1))1,1,2                                                 INTVL537
    1 CALL ERRORINT                                                   INTVL538
      RETURN                                                          INTVL539
    2 X=ARG(1)                                                        INTVL540
      LI=DLOG(X)                                                      INTVL541
      X=ARG(2)                                                        INTVL542
      RI=DLOG(X)                                                      INTVL543
    3 IF(LI)4,5,6                                                     INTVL544
    4 ANS(1)=LI*PEP                                                   INTVL545
      GO TO 7                                                         INTVL546
    5 ANS(1)=-EP                                                      INTVL547
      GO TO 7                                                         INTVL548
    6 ANS(1)=LI*MEP                                                   INTVL549
      GO TO 10                                                        INTVL550
    7 IF(RI)8,9,10                                                    INTVL551
    8 ANS(2)=RI*MEP                                                   INTVL552
      RETURN                                                          INTVL553
    9 ANS(2)=EP                                                       INTVL554
      RETURN                                                          INTVL555
   10 ANS(2)=RI*PEP                                                   INTVL556
      RETURN                                                          INTVL557
      ENTRY EXPINT                                                    INTVL558
      X=ARG(1)                                                        INTVL559
      ANS(1)=MEP*DEXP(X)                                              INTVL560
```

```
      X=ARG(2)                                                          INTVL561
      ANS(2)=PEP*DEXP(X)                                                INTVL562
      RETURN                                                            INTVL563
      ENTRY COSINT                                                      INTVL564
      AL=ARG(1)                                                         INTVL565
      AR=ARG(2)                                                         INTVL566
  100 K=AL/PI2                                                          INTVL567
      IF(AL)101,102,102                                                 INTVL568
  101 K=K-1                                                             INTVL569
  102 AL=AL-K*PI2                                                       INTVL570
      AR=AR-K*PI2                                                       INTVL571
      T1=AL-PI                                                          INTVL572
      IF(T1)110,110,120                                                 INTVL573
  110 T1=AR-PI2                                                         INTVL574
      IF(T1)112,111,111                                                 INTVL575
  111 ANS(1)=-1.                                                        INTVL576
      ANS(2)=1.                                                         INTVL577
      RETURN                                                            INTVL578
  112 T1=AR-PI                                                          INTVL579
      IF(T1)113,114,114                                                 INTVL580
  113 LI=DCOS(AR)                                                       INTVL581
      RI=DCOS(AL)                                                       INTVL582
      GO TO 3                                                           INTVL583
  114 ANS(1)=-1.                                                        INTVL584
      T1=DCOS(AL)                                                       INTVL585
      T2=DCOS(AR)                                                       INTVL586
      IF(T1-T2)115,115,116                                              INTVL587
  115 RI=T2                                                             INTVL588
      GO TO 7                                                           INTVL589
  116 RI=T1                                                             INTVL590
      GO TO 7                                                           INTVL591
  120 T1=AR-PI3                                                         INTVL592
      IF(T1)121,111,111                                                 INTVL593
  121 T1=AR-PI2                                                         INTVL594
      IF(T1)122,123,123                                                 INTVL595
```

```
  122 LI=DCOS(AL)                                                          INTVL596
      RI=DCOS(AR)                                                          INTVL597
      GO TO 3                                                              INTVL598
  123 ANS(2)=1.                                                            INTVL599
      T1=DCOS(AL)                                                          INTVL600
      T2=DCOS(AR)                                                          INTVL601
      IF(T1-T2)124,124,125                                                 INTVL602
  124 LI=T1                                                                INTVL603
      GO TO 126                                                            INTVL604
  125 LI=T2                                                                INTVL605
  126 IF(LI)127,128,129                                                    INTVL606
  127 ANS(1)=PEP*LI                                                        INTVL607
      RETURN                                                               INTVL608
  128 ANS(1)=-EP                                                           INTVL609
      RETURN                                                               INTVL610
  129 ANS(1)=MEP*LI                                                        INTVL611
      RETURN                                                               INTVL612
      ENTRY SININT                                                         INTVL613
      AL=1.5707963267-ARG(2)                                               INTVL614
      AR=1.5707963268-ARG(1)                                               INTVL615
      GO TO 100                                                            INTVL616
      ENTRY ATANINT                                                        INTVL617
      X=ARG(1)                                                             INTVL618
      LI=DATAN(X)                                                          INTVL619
      X=ARG(2)                                                             INTVL620
      RI=DATAN(X)                                                          INTVL621
      GO TO 3                                                              INTVL622
      END                                                                  INTVL623
         IDENT      SPOPS                SPECIAL OPERATIONS FOR THE        INTVL624
INIT     SLJ        **                   INTERVAL PACKAGE.  ARGUMENTS AREINTVL625
         STQ        =SX                  IN A AND Q.  ANSWERS ARE-         INTVL626
         AJP     3  NEGA                 UPPER LIMIT IN A,                 INTVL627
         LDQ        =01                  LOWER LIMIT IN Q.                 INTVL628
RET1     STQ        =SSA                 STORE SIGN(A)                     INTVL629
         LQC        M12                                                    INTVL630
```

```
          STL        =SMA                  MANTISSA(A)                      INTVL631
          ARS        36                                                     INTVL632
          INA        -2000B                                                 INTVL633
          AJP        *+2                                                    INTVL634
          AJP      2 *+1                   IF NEGATIVE - INCREASE BY 1      INTVL635
          INA        1                                                      INTVL636
          STA        =SEA                                                   INTVL637
          LDA        X                                                      INTVL638
          AJP      3 NEGX                                                   INTVL639
          LDQ        =01                                                    INTVL640
RET2      STQ        =SSX                  STORE SIGN(X)                    INTVL641
          LQC        M12                                                    INTVL642
          STL        =SMX                  MANTISSA(X)                      INTVL643
          ARS        36                                                     INTVL644
          INA        -2000B                                                 INTVL645
          AJP        *+2                                                    INTVL646
          AJP      2 *+1                   IF NEGATIVE - INCREASE BY 1      INTVL647
          INA        1                                                      INTVL648
          STA        =SEX                                                   INTVL649
          SLJ        INIT                  --RETURN--                       INTVL650
NEGA      SCM        ALSEV                 COMPLEMENT A IF NEGATIVE         INTVL651
          LQC        =01                   -1 TO SIGN(A)                    INTVL652
          SLJ        RET1                                                   INTVL653
NEGX      SCM        ALSEV                 COMPLEMENT X IF NEGATIVE         INTVL654
          LQC        =01                   -1 TO SIGN(X)                    INTVL655
          SLJ        RET2                                                   INTVL656
M12       OCT        7777000000000000                                       INTVL657
ALSEV     OCT        -0                                                     INTVL658
N12 .     OCT        777777777777                                           INTVL659
N13       OCT        4000777777777777                                       INTVL660
HERD      SLJ        **                    -COLLECT BITS AND PIECES-        INTVL661
          STA        =STEMP                SAVE A                           INTVL662
+         LDA        EA                    RE-NORMALIZE EXPONENT WRT. 2000  INTVL663
          AJP        *+2                                                    INTVL664
          AJP      2 *+1                                                    INTVL665
```

```
          INA         -1                                               INTVL666
          INA         2000B                                            INTVL667
          ALS         36                                               INTVL668
          SCL         N13                                              INTVL669
          STA         EA                                               INTVL670
          SST         TEMP             FLOAT A                         INTVL671
          QJP         NOCARRY          EXECT IF Q=0                    INTVL672
          FAD         =00              NORMALIZE                       INTVL673
          STA         =SSAVA                                           INTVL674
          MUI         SA               MULTIPLY BY SIGN                INTVL675
          STA         =SSAVB                                           INTVL676
          LDA         EA                                               INTVL677
          INA         1                EXPONENT WITH LOW-ORDER 1       INTVL678
          FAD         SAVA             ADD TO CHOPPED NUMBER FOR UPPER INTVL679
          MUI         SA               MULTIPLY BY SIGN                INTVL680
          AJP     3   FLIP                                             INTVL681
RET3      LDQ         SAVB                                             INTVL682
          SLJ         HERD                                             INTVL683
FLIP      LRS         48               A TO Q                          INTVL684
          LDA         SAVB             CHOPPED NUMBER TO A             INTVL685
          SLJ         HERD                                             INTVL686
NOCARRY   MUI         SA               EXACT NUMBER TO A               INTVL687
          FAD         =00                                              INTVL688
          STA         SAVB             AND Q                           INTVL689
          SLJ         RET3                                             INTVL690
          ENTRY       SPOPSADD         SP. ADDITION                    INTVL691
SPOPSADD  SLJ         **                                               INTVL692
          RTJ         INIT             GO TO INITIALIZE                INTVL693
          LDA         EA                                               INTVL694
          SUB         EX               EA-EX                           INTVL695
          AJP     3   SWITCH           SWITCH IF EX.GT.EA              INTVL696
RET4      ENQ         0                                                INTVL697
+         THS         =O110                                            INTVL698
          SLJ         SKIPS                                            INTVL699
REASON    SAU         *+1              SHIFT ADDRESS=EA-EX             INTVL700
```

```
         LDA        MX                                                INTVL701
         LRS        **              LINE UP MX WITH MA                INTVL702
 RET8    STA        MX                                                INTVL703
         LDA        SA              COMPARE SIGNS OF X AND A          INTVL704
         ADD        SX                                                INTVL705
         AJP        SUBTRACT        SUBTRACT IF DIFFERENT SIGNS       INTVL706
         LDA        MA                                                INTVL707
         ADD        MX                                                INTVL708
         STA        =STEMP          SAVE SUM                          INTVL709
         SCL        N12             TEST FOR OVERFLOW                 INTVL710
         AJP        NOVER           EXIT FOR NO OVERFLOW              INTVL711
         LDA        EA                                                INTVL712
         INA        1                                                 INTVL713
         STA        EA                                                INTVL714
         LDA        TEMP            AND SHIFT RIGHT BY 1              INTVL715
         LRS        1                                                 INTVL716
RET6     RTJ        HERD            GO TO COLLECT                     INTVL717
+        SLJ        SPOPSADD        --EXIT--                          INTVL718
SKIPS    LDA        MX                                                INTVL719
         AJP        RET8                                              INTVL720
         ENQ        2                                                 INTVL721
         ENA        0                                                 INTVL722
         SLJ        RET8                                              INTVL723
NOVER    LDA        TEMP            RELOAD ACC                        INTVL724
         SLJ        RET6                                              INTVL725
SUBTRACT LDA        MA              SUBTRACT MX FROM MA               INTVL726
         SUB        MX              -NO OVERFLOW POSSIBLE             INTVL727
         AJP    3   NEGATE         IF NEGATIVE - COMPLEMENT, SIGN     INTVL728
         QJP        RET6            NO RESIDUE-EXIT                   INTVL729
         INA        -1              NEGATIVE RESIDUE-DECREASE MANTISINTVL730
         STQ        =SSTORE                                           INTVL731
+        AJP    2   *+1                                               INTVL732
         ENA        0                                                 INTVL733
         LQC        STORE                                             INTVL734
         SIU    1   GOOUT                                             INTVL735
```

```
+        ENI      1 83                                    INTVL736
         SIU     1  *+1                                   INTVL737
+        SCQ     1  **                                    INTVL738
         LRS        11                                    INTVL739
         STA        TEMP                                  INTVL740
         LDA        EA                                    INTVL741
         INA     1  -72                                   INTVL742
         STA        EA                                    INTVL743
GOOUT    ENI     1  **                                    INTVL744
         SLJ        NOVER                                 INTVL745
NEGATE   LQC        SA                                    INTVL746
         STQ        SA        CHANGE SIGN OF RESULT       INTVL747
         SCM        ALSEV     COMPLEMENT ACC              INTVL748
         ENQ        0         NO RESIDUE IN Q POSSIBLE    INTVL749
         SLJ        RET6                                  INTVL750
SWITCH   STA        TEMP      INTERCHANGE X AND A         INTVL751
         LDA        EA                                    INTVL752
         LDQ        EX                                    INTVL753
         STA        EX                                    INTVL754
         STQ        EA                                    INTVL755
         LDA        MA                                    INTVL756
         LDQ        MX                                    INTVL757
         STQ        MA                                    INTVL758
         STA        MX                                    INTVL759
         LDA        SX                                    INTVL760
         LDQ        SA                                    INTVL761
         STA        SA                                    INTVL762
         STQ        SX                                    INTVL763
         LAC        TEMP      CHANGE SIGN OF (EA-EX)      INTVL764
         SLJ        RET4                                  INTVL765
         ENTRY      SPOPSMUL  SP.MULTIPLICATION           INTVL766
SPOPSMUL SLJ        **                                    INTVL767
         RTJ        INIT                                  INTVL768
         LDA        EA                                    INTVL769
         ADD        EX        ADD EXPONENTS               INTVL770
```

```
         STA      EA              NEW EXPONENT                          INTVL771
         LDA      SA                                                    INTVL772
         MUI      SX                                                    INTVL773
         STA      SA              NEW SIGN=SA*SX                        INTVL774
         LDA      MX                                                    INTVL775
         ALS      11              LEFT-JUSTIFY MULTIPLIER               INTVL776
         MUF      MA              FORM PRODUCT-NO OVERFLOW POSSIBLINTVL777
         RTJ      ADJUST                                                INTVL778
+        RTJ      HERD                                                  INTVL779
+        SLJ      SPOPSMUL        --EXIT                                INTVL780
         ENTRY    SPOPSDIV        SP.DIVISION                           INTVL781
SPOPSDIV SLJ      **                                                    INTVL782
         RTJ      INIT                                                  INTVL783
         LDA      EA                                                    INTVL784
         SUB      EX              NEW EXPONENT=EA-EX+2000               INTVL785
         INA      1                                                     INTVL786
         STA      EA                                                    INTVL787
         LDA      SA              NEW SIGN=SA*SX                        INTVL788
         MUI      SX                                                    INTVL789
         STA      SA                                                    INTVL790
         LDA      MX              LEFT-JUSTIFY DIVISOR                  INTVL791
         ALS      11                                                    INTVL792
         STA      TEMP                                                  INTVL793
         LDA      MA                                                    INTVL794
         ENQ      0                                                     INTVL795
         LRS      1                                                     INTVL796
         DVF      TEMP            DIVIDE-NO OVERFLOW POSSIBLE           INTVL797
         RTJ      ADJUST                                                INTVL798
+        RTJ      HERD                                                  INTVL799
+        SLJ      SPOPSDIV                                              INTVL800
MASKMM   OCT      377777777777                                          INTVL801
ADJUST   SLJ      **                                                    INTVL802
         STA      TEMP                                                  INTVL803
         SCL      MASKMM                                                INTVL804
         AJP    1 ADJUSTA                                               INTVL805
```

```
         RSO     EA                                                INTVL806
         LDA     TEMP                                              INTVL807
         LLS     1                                                 INTVL808
         SLJ     ADJUST                                            INTVL809
ADJUSTA  LDA     TEMP                                              INTVL810
         SLJ     ADJUST                                            INTVL811
         END                                                       INTVL812
```