19.1 System Parameters

A. Purpose

These subprograms provide values of various system parameters that are needed in library subprograms and typically have different values on different computer systems. “F. Supporting Information” below has instructions for customizing these programs for your system.

B. Usage

B.1 Program Prototype

REAL R1MACH, S

DOUBLE PRECISION D1MACH, D

INTEGER I1MACH, I, J

Set J in the range $1 \leq \mathrm{J} \leq 5$ for R1MACH or D1MACH or $1 \leq \mathrm{J} \leq 16$ for I1MACH. Then, use the appropriate one of the following statements:

$$\boxed{\mathbf{S = R1MACH(J)}}$$

$$\boxed{\mathbf{D = D1MACH(J)}}$$

$$\boxed{\mathbf{I = I1MACH(J)}}$$

The results, S, D, or I are set as described in Section D.

B.2 Argument Definitions

J [in] Integer argument selecting the desired system parameter as described in Section D.

C. Examples and Remarks

The program DRMACH lists all the values obtainable from R1MACH, D1MACH, and I1MACH on the host computer system. Output is shown from several different host systems.

D. Functional Description

For the purpose of this package a model of the Fortran 77 INTEGER, REAL, and DOUBLE PRECISION number sets is characterized by a total of nine fundamental parameters.

The model of Fortran 77 numbers of type INTEGER is parameterized by two numbers, a and s, where a denotes the base (radix) of the number system and s denotes the maximum number of base a digits available to represent a Fortran integer. Thus the integers range from $-(a^s - 1)$ to $a^s - 1$.

The model of Fortran 77 numbers of type REAL is characterized by four parameters, b, t, $emin$, and $emax$, where b is the base of the fraction part, t is the number of base b digits in the fraction part, $emin$ is the minimum exponent and $emax$ is the maximum exponent. The magnitude of a floating-point number is thus of the form

$$b^e \left(c_1 b^{-1} + c_2 b^{-2} + ... + c_t b^{-t}\right),$$

where, $emin \leq e \leq emax$, and the digits c_i satisfy $0 \leq c_i \leq b - 1$. A nonzero floating-point number is normalized if and only if the digit c_1 is nonzero. We shall consider only normalized floating-point numbers, although numeric processors based on the IEEE standard also support a range of unnormalized numbers.

Fortran 77 numbers of type DOUBLE PRECISION are modeled in the same form as REAL numbers and are assumed to have the same base, b, but generally different values of t, $emin$, and $emax$.

For some computer systems no setting of these parameters will make the model system coincide exactly with the actual computer’s number set. In such cases the model parameters are selected so the model system will be as large a subset of the actual number set as possible. In particular, for Cray X/MP and Y/MP (but not T3D) systems the parameters t and $emax$ are set smaller and $emin$ is set larger than one might expect from the structure of floating-point numbers on these systems. The reasons for this are described in [1].

The values returned by this package are either prestored constants or are computed at compile-time from prestored constants by use of expressions in PARAMETER statements. Thus correct values must be determined and edited into the package whenever this package is moved to a new computer system. Correct values for many systems are present as comments in the source code.

On some systems, the Fortran compiler may not be able to compute numbers in the full range of the model, in PARAMETER statements (this disease usually strikes computation of the overflow limit). On such systems it may be necessary to restrict the model, *e.g.* by reducing *emax*.

Specification of Values Returned

J	**I1MACH (J)**
1	Standard input unit number.
2	Standard output unit number.
3	Standard punch unit number.
4	Standard error message unit number.
5	Number of bits per Fortran integer storage unit.

6 Number of characters per Fortran integer storage unit.

7 a, the base for integers.

8 s, the number of base a digits in an integer.

9 $a^s - 1$, the largest integer magnitude.

10 b, the base for floating-point numbers. Assumed the same for REAL and DOUBLE PRECISION arithmetic.

11 t, the number of base b digits for REAL arithmetic.

12 $emin$, minimum exponent for REAL arithmetic.

13 $emax$, maximum exponent for REAL arithmetic.

14 t, the number of base b digits for DOUBLE PRECISION arithmetic.

15 $emin$, minimum exponent for DOUBLE PRECISION arithmetic.

16 $emax$, maximum exponent for DOUBLE PRECISION arithmetic.

J **R1MACH(J)**

1 b^{emin-1} smallest positive normalized REAL number, (underflow limit).

2 $b^{emax}(1 - b^{-t})$, largest REAL number, (overflow limit).

3 b^{-t}, smallest relative difference between two successive nonzero REAL numbers. This is also the difference between 1.0 and the next smaller REAL number.

4 $b^{-(t-1)}$ largest relative difference between two successive nonzero REAL numbers. This is also the difference between 1.0 and the next larger REAL number.

5 $\log_{10} b$, useful in certain conversions between base b and base 10.

The values returned by D1MACH are as described above for R1MACH with REAL replaced by DOUBLE PRECISION.

Historical perspective and relations to other languages

The specifications of R1MACH, D1MACH, and I1MACH and the original implementation were developed at the AT&T Bell Laboratories, Murray Hill, New Jersey, in the 1970's to support the development of portable mathematical software, and specifically the PORT library, [2], which is a proprietary AT&T Bell Laboratories product. These three subprograms were published as a subset of Algorithm 528 in TOMS, [3], and are not proprietary. The MATH77 version has the same specification but is substantially different in its implementation from the original versions.

The attributes associated with J = 1, 2, 3, 4, and 6 in I1MACH are less relevant in Fortran 77 in the 90's than they were in Fortran 66 in the 70's. In particular, only the DNLxxx and SNLxxx subroutines of Chapter 9.3 access I1MACH(2).

Other MATH77 library subprograms use PRINT or WRITE(*,...) for printing.

Languages developed more recently than Fortran 77, such as Ada, ANSI C, and Fortran 90, provide methods within the language to obtain certain environmental parameters. Consider, for example, the underflow and overflow limits, and precision for floating-point arithmetic. Using the present package, these can be obtained for DOUBLE PRECISION arithmetic by referencing D1MACH(1), D1MACH(2), and D1MACH(4), respectively, and for REAL arithmetic by referencing R1MACH(1), R1MACH(2), and R1MACH(4). In Fortran 90 these parameters can be obtained by referencing the generic inquiry functions TINY(X), HUGE(X), and EPSILON(X), where X may be any DOUBLE PRECISION entity to obtain the values for DOUBLE PRECISION arithmetic, and any REAL entity to obtain the values for REAL arithmetic. In ANSI C these values for arithmetic of type *double* are given by the macro names DBL_MIN, DBL_MAX, and DBL_EPSILON, for type *float* there are FLT_MIN, FLT_MAX, and FLT_EPSILON, and for type *long double* LDBL_MIN, LDBL_MAX, and LDBL_EPSILON. All of these are defined in the standard header file *float.h*.

References

1. N. L. Schryer, **A Test of a Computer's Floating-Point Arithmetic Unit**. Bell Laboratory Comp. Sci. Tech. Rpt. 89, AT&T (Feb. 1981) Appendices + 31.
2. P. A. Fox, A. D. Hall, and N. L. Schryer, *The PORT mathematical subroutine library*, **ACM Trans. on Math. Software 4**, 2 (June 1978) 104–126.
3. P. A. Fox, A. D. Hall, and N. L. Schryer, *Algorithm 528: Framework for a portable library [Z]*, **ACM Trans. on Math. Software 4**, 2 (June 1978) 177–188.

E. Error Procedures and Restrictions

If the argument is outside the range $1 \leq \mathrm{J} \leq 16$ for I1MACH or outside $1 \leq \mathrm{J} \leq 5$ for R1MACH or D1MACH, an error message is printed and execution is terminated.

This package contains a partial protection against the inadvertent use of the wrong version of one of these subroutines; say using the PC version on a VAX. On the first call to any one of these subprograms, tests are done to verify that two of the stored parameter values are not grossly wrong for the current environment. These tests depend on assumptions about hardware, compilers, and linkers that may be invalidated by technological changes. Subroutines AMTEST and AMSUB1 are used to support these tests and are not intended for any other usage.

F. Supporting Information

The source language is ANSI Fortran 77. All the program units are grouped into a single file, AMACH.FOR. The filename may be different on different systems, *e.g.*, "amach.f" on UNIX systems.

One can either customize AMACH for a new system by commenting out lines defining the parameters required for the system as it is currently configured, and uncommenting lines required for the desired system, or one can use the program `m77con` described in Chapter 19.4. This requires making up a small control file `m77job`, and compiling, linking, and running `m77con`, which is self contained. To make up `m77job` for a VAX running UNIX, the control file would contain the following.

```
SET SYS = VAX
FILE amach.f
```

Running `m77con` with this control file and amach in the current directory will generate a file `amach.f` for the VAX. If one wants the extension "`.for`" change the `h.f` to `h.for`. If one wants a machine other than the VAX, choose a value for SYS (without any parenthetical remark) from the following table. `SYS=IEEE` covers any machine that uses the IEEE binary standard for floating point arithmetic. If your machine is not included in this list, either pick a machine with the same parameters for the floating point arithmetic as for a machine on this list, or enter parameters for your machine as a new option into AMACH.

```
SYS = IEEE
SYS = AMDAHL
SYS = APOLLO_10000
SYS = BUR1700
SYS = BUR5700
SYS = BUR67_7700
SYS = CDC60_7000
SYS = CONVEXC_1
SYS = CRAY1
SYS = CRAY1_SD        (Sngl prec.arith. used for dble.)
SYS = CRAY1_64        (64 bit integers)
SYS = CRAY1_SD_64     (64 bit int, SP used for DP)
SYS = CRAY_J90
SYS = CRAY_J90_SD     (Sngl prec. used for dble.)
SYS = DG_S2000
SYS = HARRIS220
SYS = HON600_6000
SYS = HON_DPS_8_70
SYS = HP700Q          (Q Precision on HP700 series)
SYS = IBM360_370
SYS = INTERDATA_8_32
SYS = PDP10_KA
SYS = PDP10_KB
SYS = PDP11
SYS = PRIME50
SYS = SEQ_BAL_8000
SYS = UNIVAC
SYS = VAX
SYS = VAX_G
SYS = ALPHA_D3
```

Designed and programmed by P.A. Fox, A.D. Hall, and N.L. Schryer, AT&T Bell Laboratories, 1978. Adapted to the JPL MATH77 library, 1984 and 1987.

Entry	**Required Files**
D1MACH	AMACH
I1MACH	AMACH
R1MACH	AMACH

DRMACH

```
      program DRMACH
c>> 1996-05-28 DRMACH Krogh Added external statement, moved formats.
c>> 1994-08-09 DRMACH WVS Removed '0' from format
c>> 1991-11-20 DRMACH CLL Edited for Fortran 90.
c>> 1989-10-30 DRMACH CLL Changed format to handle 14 digit Cray integer
c>> 1987-12-09 DRMACH   Lawson   Initial Code.
c                       Demo driver for D1MACH, I1MACH, and R1MACH
      external D1MACH, I1MACH, R1MACH
      double precision D1MACH, D1
      integer I1MACH, I1, J
      real R1MACH, R1
   10 format(15X,A)
   20 format(15X,41('-'))
   30 format(/2X,'J',6X,'I1MACH(J)',7X,'R1MACH(J)',13X,'D1MACH(J)')
   40 format(I3, I15,3X,G15.8,3X,G25.18)
   50 format(I3, I15)
```

```
c     ----------------------------------------------------------------
      print 10,'MACHINE CONSTANTS for ...                           '
      print 20
      print 30
      print*,' '
      do 60 J=1,5
        I1 = I1MACH(J)
        R1 = R1MACH(J)
        D1 = D1MACH(J)
        print 40, J,I1,R1,D1
   60 continue
      do 70 J=6,16
        print 50, J,I1MACH(J)
   70 continue
c
      end
```

Results from Various Machines

```
              MACHINE CONSTANTS for IEEE Arithmetic
              -----------------------------------------
 J     I1MACH(J)      R1MACH(J)              D1MACH(J)

 1             5   0.11754944E-37   0.222507385850720100-307
 2             6   0.34028235E+39   0.179769313486231600+309
 3             7   0.59604645E-07   0.111022302462515700E-15
 4             6   0.11920929E-06   0.222044604925031300E-15
 5            32   0.30103001       0.301029995663981200
 6             4
 7             2
 8            31
 9    2147483647
10             2
11            24
12          -125
13           128
14            53
15         -1021
16          1024
```

```
              MACHINE CONSTANTS for VAX
              -----------------------------------------
 J     I1MACH(J)      R1MACH(J)              D1MACH(J)
 1             5   0.29387359E-38   0.293873587705571877E-38
 2             6   0.17014117E+39   0.170141183460469229E+39
 3             7   0.59604645E-07   0.138777878078144568E-16
 4             6   0.11920929E-06   0.277555756156289135E-16
 5            32   0.30103001       0.301029995663981198
 6             4
 7             2
 8            31
 9    2147483647
10             2
11            24
12          -127
13           127
14            56
```

15	−127
16	127

MACHINE CONSTANTS for CRAY J90

J	I1MACH(J)	R1MACH(J)	D1MACH(J)
1	5	0.73344155−2465	0.733441547021938866−2465
2	6	0.13634352+2466	0.136343516952426991+2466
3	102	0.71054274E−14	0.504870979341447555E−28
4	6	0.14210855E−13	0.100974195868289511E−27
5	64	0.30103000	0.301029995663981195
6	8		
7	2		
8	46		
9	70368744177663		
10	2		
11	47		
12	−8188		
13	8189		
14	94		
15	−8188		
16	8189		

MACHINE CONSTANTS for UNISYS 1100

J	I1MACH(J)	R1MACH(J)	D1MACH(J)
1	5	.14693679−038	.278134232313400172−308
2	6	.17014118+039	.898846567431157951+308
3	7	.74505806−008	.867361737988403547−018
4	6	.14901161−007	.173472347597680709−017
5	36	.30103000	.301029995663981194
6	4		
7	2		
8	35		
9	34359738367		
10	2		
11	27		
12	−128		
13	127		
14	60		
15	−1024		
16	1023		