



1 SUMMARY

This function **supplies real-valued machine constants relating to the floating-point storage and arithmetic** of the machine in use.

A nonzero floating-point number is stored in the form $\pm m \beta^e$, where β is known as the base (or radix) of the arithmetic, m is the mantissa (or significand or fraction) and e is the exponent (or characteristic). The mantissa is usually normalized so that any floating-point number has a unique representation. Individual machines differ in the way that the normalization is performed. The exponent is stored as a sequence of binary digits (bits); the sign of the exponent either occupies one of these digits, or, more commonly, the actual value of the exponent is obtained by adding the stored binary representation to a fixed negative bias. The mantissa is represented as $m = \sum_{i=1}^n m_i \beta^{-i+j}$, where $0 \leq m_i < \beta$, j is usually 0 or 1 and m is usually normalized so that $m_1 > 0$. (Under special circumstances, m_1 may be zero; such circumstances are typically associated with implementations of gradual underflow on a particular machine.)

ATTRIBUTES — **Version:** 1.0.0. **Types:** FD05A, FD05AD. **Remark:** This package is also included in the HSL Archive. **Calls:** None. **Original date:** April 1988. **Origin:** N.I.M. Gould and S. Marlow, Harwell. **Licence:** A third-party licence for this package is available without charge.

2 HOW TO USE THE PACKAGE

Two versions of the function are available corresponding to the use of single and double precision arithmetic.

2.1 Argument list

The single precision version

RNUM = FD05A(INUM)

The double precision version

DNUM = FD05AD(INUM)

INUM is an INTEGER variable which must be set by the user to select which one of the real machine constants is required. Possible values of INUM and the corresponding constants are:

1. FD05A/FD05AD(1) is the smallest REAL/DOUBLE PRECISION number of the form β^i such that $1.0 + \beta^i$ and 1.0 are different stored numbers. On most machines $i = 1 - n$. This value is that returned by the Numeric Inquiry Function EPSILON in the ANSI Fortran 90 standard and is commonly called the machine precision.
2. FD05A/FD05AD(2) is the smallest REAL/DOUBLE PRECISION number of the form β^i such that $1.0 - \beta^i$ and 1.0 are different stored numbers. On most machines $i = -n$.
3. FD05A/FD05AD(3) is a close approximation to the smallest positive REAL/DOUBLE PRECISION number which may be stored on the machine. This number is normally $\beta^{e_{\min} + j - 1}$, where e_{\min} is the smallest allowable value of the exponent, but may be as small as $\beta^{e_{\min} + j - n}$ if the machine allows gradual underflow.
4. FD05A/FD05AD(4) is a close approximation to the smallest positive REAL/DOUBLE PRECISION number which may be stored on the machine to full precision, i.e., for which $m_1 > 0$. This number is normally $\beta^{e_{\min} + j - 1}$, where e_{\min} is the smallest allowable value of the exponent. The numbers FD05A/FD05AD(3) and FD05A/FD05AD(4) will be identical on machines which have no provision for

gradual underflow. This value is that returned by the Numeric Inquiry Function TINY in the ANSI Fortran 90 standard.

5. FD05A/FD05AD(5) is a close approximation to the largest finite positive REAL/DOUBLE PRECISION number which may be stored on the machine. This number is normally $\beta^{e_{\max}+j}(1.0-\beta^{-n})$, where e_{\max} is the largest allowable value of the exponent, and is the value returned by the Numeric Inquiry Function HUGE in the ANSI Fortran 90 standard.

INUM is not altered by the function. **Restriction:** $1 \leq \text{INUM} \leq 5$. Note: FD05 does not indicate an error when the value of INUM is out of range; a value of $\text{INUM} \leq 0$ is treated as 1 and a value of $\text{INUM} > 5$ is treated as 5.

FD05A/FD05AD is a REAL (DOUBLE PRECISION in the D version) function whose value will be set to the required machine constant.

3 GENERAL INFORMATION

Use of common: None.

Other routines called directly: None.

Input/output: None.

Restrictions: $1 \leq \text{INUM} \leq 5$.

4 METHOD

The constants have been set by the original implementor of HSL on your machine. Further detail of the machine representation of floating-point numbers may be found in, for instance, the Dictionary of Computing (Oxford University Press, 1983).

5 EXAMPLE OF USE

As a very simple example, we wish to find the machine precision and the largest finite floating-point number which may be stored on the machine in double precision; in this case an IBM 3084Q is used.

```
DOUBLE PRECISION DNUM, FD05AD
DNUM = FD05AD( 1 )
WRITE( 6, 2000 ) DNUM
DNUM = FD05AD( 5 )
WRITE( 6, 2010 ) DNUM
STOP
2000 FORMAT( ' MACHINE PRECISION(DOUBLE PRECISION) = ', 1P, D12.4 )
2010 FORMAT( ' LARGEST FLOATING-POINT NUMBER (DOUBLE PRECISION) = ',
*          1P, D12.4 )
END
```

This produces the following output

```
MACHINE PRECISION (DOUBLE PRECISION) = 2.2204D-16
LARGEST FLOATING-POINT NUMBER (DOUBLE PRECISION) = 7.2370D+75
```