1 SUMMARY

This function **supplies integer-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 $0 \le m_i < \beta$, 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:** ID05A, ID05AD. **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

```
NUM = ID05A(INUM)
```

The double precision version

```
NUM = ID05AD(INUM)
```

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

1. ID05A/ID05AD(1) gives β, the base used for the floating-point arithmetic. This value is that returned by the Fortran 90 Numeric Inquiry Function RADIX.
2. ID05A/ID05AD(2) gives n, the number of base β digits used in the mantissa. This value is that returned by the Fortran 90 Numeric Inquiry Function DIGITS.
3. ID05A/ID05AD(3) gives the number of binary digits used to store the exponent of a floating-point number, including its sign.
4. ID05A/ID05AD(4) has the value 0 if a floating-point number is chopped (rounded down) when it is stored after addition and has the value 1 if it correctly rounded to the nearest machine representable number.
5. ID05A/ID05AD(5) has the value 0 if a floating-point number is chopped (rounded down) when it is stored after multiplication and has the value 1 if it correctly rounded to the nearest machine representable number.
6. ID05A/ID05AD(6) is the most negative integer, i, for which $1.0+\beta^i$ and 1.0 are different stored

numbers. On most machines $i = 1 - n$.

7. ID05A/ID05AD(7) is the most negative integer, i, for which $1.0 - \beta^i$ and 1.0 are different stored numbers. On most machines $i = -n$.
8. ID05A/ID05AD(8) is the most negative integer, i, for which β^i is a positive stored number.
9. ID05A/ID05AD(9) is the most negative integer, i, for which β^i is a positive stored number represented to full precision, i.e., so that $m_1 > 0$. For machines without the provision for gradual underflow, ID05A/ID05AD(8) and ID05A/ID05AD(9) will be identical. This value is 1 less than that returned by the Fortran 90 Numeric Inquiry Function MINEXPONENT.
10. ID05A/ID05AD(10) is the largest positive integer, i, for which β^i is an exactly representable finite stored number. This value is 1 less than that returned by the Fortran 90 Numeric Inquiry Function MAXEXPONENT.

INUM is not altered by the function. **Restriction:** $1 \le$ INUM ≤ 10. Note: ID05 does not indicate an error when the value of INUM is out of range; a value of INUM ≤ 0 is treated as 1 and a value of INUM > 10 is treated as 10.

ID05A (ID05AD for the DOUBLE PRECISION version) is an INTEGER 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: $1 \le$ INUM ≤ 10.

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 base, the number of digits used in the exponent and the number of base-digits used in the mantissa on the machine for double precision computations; in this case an IBM 3084Q is used.

```
      INTEGER ID05AD, NUM
      NUM = ID05AD( 1 )
      WRITE( 6, 2000 ) NUM
      NUM = ID05AD( 2 )
      WRITE( 6, 2010 ) NUM
      NUM = ID05AD( 3 )
      WRITE( 6, 2020 ) NUM
      STOP
 2000 FORMAT( ' BASE = ', I8 )
 2010 FORMAT( ' NO. DIGITS IN MANTISSA (DOUBLE PRECISION) = ', I8 )
 2020 FORMAT( ' BITS IN EXPONENT (DOUBLE PRECISION) = ', I8 )
      END
```

This produces the following output

```
 BASE =       16
```

```
  NO. DIGITS IN MANTISSA (DOUBLE PRECISION) =       14
  BITS IN EXPONENT (DOUBLE PRECISION) =        7
```