1 SUMMARY

This subroutine **allows an implementor of the Harwell Subroutine Library on a new machine range to generate the machine constant functions** ID05 **and** FD05 used by many of the subroutines in the library. It is not anticipated that this subroutine be used for any other purpose. Specific machine constants will normally be obtained directly from ID05 and FD05. Details of the constants generated are available in the specification sheets for these functions.

Warning: The implementor is strongly advised to check the output from this subroutine to see that the machine constants generated are correct.

ATTRIBUTES — Version: 1.0.0. **Types:** ZE02AM. **Calls:** None. **Original date:** April 1988. **Origin:** N.I.M. Gould and S. Marlow, Harwell.

2 HOW TO USE THE PACKAGE

2.1 Argument list

```
CALL ZE02AM(MP, NP)
```

MP is an INTEGER variable whose value specifies the Fortran output unit to which the generated subroutines ID05A, ID05AD, FD05A and FD05AD will be sent. The format of the output obtained is illustrated in §5. The value of MP is not altered by the subroutine. **Restriction:** MP must be a valid unit number.

NP is an INTEGER variable whose value specifies a Fortran input/output unit which is used to test the values of the machine constants generated. This value is not altered by the subroutine. **Restriction:** NP must be a valid unit number.

2.2 Common

The subroutine uses the common block ZE02EM.

```
COMMON / ZE02EM / IUSEXP
```

IUSEXP is an INTEGER variable. On some machines, certain values of the exponent field are reserved for special purposes. We assume that our machine reserves IUSEXP values of the exponent for such purposes. IUSEXP is provided, with a default value of 0, in the BLOCK DATA subprogram ZE02BM. If a floating-point overflow occurs when ZE02AM is used, IUSEXP is too small and should be incremented by one (and the program rerun) until the overflows cease. If IEEE standard arithmetic is detected, IUSEXP is known to have the value 2 and will automatically be reset to this value.

2.3 Warning

This subroutine will generate four floating-point underflows as it calculates the required machine constants. The implementor should therefore ensure that execution of the program is not terminated by underflows, if necessary by issuing appropriate instructions to the Fortran compiler used.

A floating-point overflow may also occur. If this happens, the parameter IUSEXP should be increased and the program rerun (see §2.2).

3 GENERAL INFORMATION

Use of common: See §2.2

Other routines called directly: calls the private subroutines ZE02CM, ZE02DM, ZE02M, ZE02MD, ZE02P, ZE02PD, ZE02T, ZE02TD and contains the BLOCK DATA subprogram ZE02BM.

Input/output: Under the control of the parameters MP and NP.

Restrictions: MP and NP must be valid unit numbers.

4 METHOD

This subroutine is a partial translation of a C program written by W. J. Cody. Modifications have been made to account for machines which allow gradual underflow of floating-point numbers.

5 EXAMPLE OF USE

As a very simple example, we use the subroutine to generate the subprograms ID05A, ID05AD, FD05A and FD05AD appropriate for the IBM 3084Q. The output is sent to Fortran unit 6 and unit 51 is used for the intermediate checking.

```
      INTEGER MP, NP
      EXTERNAL ZE02BM
      MP = 6
      NP = 51
      CALL ZE02AM( MP, NP )
      STOP
      END
```

This produces the following output

```
      INTEGER FUNCTION ID05A( INUM )
      INTEGER INUM, IC( 10 )
C
C  INTEGER CONSTANTS (SINGLE PRECISION ARITHMETIC).
C
C  OBTAINED FROM H.S.L. SUBROUTINE ZE02AM.
C  NICK GOULD AND SID MARLOW, HARWELL, APRIL 1988.
C
C  IC(1) THE BASE (RADIX) OF THE FLOATING-POINT ARITHMETIC.
C  IC(2) THE NUMBER OF BASE IC(1) DIGITS IN THE SIGNIFICAND.
C  IC(3) THE NUMBER OF BITS USED FOR THE EXPONENT
C  IC(4) = 0 FLOATING-POINT ADDITION CHOPS, = 1 IT ROUNDS.
C  IC(5) THE NUMBER OF GUARD DIGITS FOR MULTIPLICATION.
C  IC(6) LARGEST -VE INTEGER:1.0 + REAL(IC(1))**IC(6) > 1.0.
C  IC(7) LARGEST -VE INTEGER:1.0 - REAL(IC(1))**IC(7) < 1.0.
C  IC(8) LARGEST -VE INTEGER: REAL(IC(1))**IC(8) > 0.0.
C  IC(9) LARGEST -VE INTEGER: REAL(IC(1))**IC(9) IS NORMAL.
C  IC(10) LARGEST +VE INTEGER: REAL(IC(1))**IC(10) FINITE.
C
      DATA IC( 1 )   /           16 /
      DATA IC( 2 )   /            6 /
      DATA IC( 3 )   /            7 /
      DATA IC( 4 )   /            0 /
      DATA IC( 5 )   /            1 /
      DATA IC( 6 )   /           -5 /
      DATA IC( 7 )   /           -6 /
      DATA IC( 8 )   /          -65 /
      DATA IC( 9 )   /          -65 /
      DATA IC( 10 ) /            62 /
      IF ( INUM .LE. 0 .OR. INUM .GE. 11 ) THEN
         PRINT 2000, INUM
         STOP
      ELSE
```

```
         ID05A = IC( INUM )
      ENDIF
      RETURN
 2000 FORMAT( ' INUM =', I3, ' OUT OF RANGE IN ID05A.',
     *        ' EXECUTION TERMINATED.' )
      END
      INTEGER FUNCTION ID05AD( INUM )
      INTEGER INUM, IC( 10 )
C
C  INTEGER CONSTANTS (DOUBLE PRECISION ARITHMETIC).
C
C  OBTAINED FROM H.S.L. SUBROUTINE ZE02AM.
C  NICK GOULD AND SID MARLOW, HARWELL, APRIL 1988.
C
C  IC(1) THE BASE (RADIX) OF THE FLOATING-POINT ARITHMETIC.
C  IC(2) THE NUMBER OF BASE IC(1) DIGITS IN THE SIGNIFICAND.
C  IC(3) THE NUMBER OF BITS USED FOR THE EXPONENT
C  IC(4) = 0 FLOATING-POINT ADDITION CHOPS, = 1 IT ROUNDS.
C  IC(5) THE NUMBER OF GUARD DIGITS FOR MULTIPLICATION.
C  IC(6) LARGEST -VE INTEGER:1.0 + DBLE(IC(1))**IC(6) > 1.0.
C  IC(7) LARGEST -VE INTEGER:1.0 - DBLE(IC(1))**IC(7) < 1.0.
C  IC(8) LARGEST -VE INTEGER: DBLE(IC(1))**IC(8) > 0.0.
C  IC(9) LARGEST -VE INTEGER: REAL(IC(1))**IC(9) IS NORMAL.
C  IC(10) LARGEST +VE INTEGER: REAL(IC(1))**IC(10) FINITE.
C
      DATA IC( 1 )   /           16 /
      DATA IC( 2 )   /           14 /
      DATA IC( 3 )   /            7 /
      DATA IC( 4 )   /            0 /
      DATA IC( 5 )   /            1 /
      DATA IC( 6 )   /          -13 /
      DATA IC( 7 )   /          -14 /
      DATA IC( 8 )   /          -65 /
      DATA IC( 9 )   /          -65 /
      DATA IC( 10 ) /            62 /
      IF ( INUM .LE. 0 .OR. INUM .GE. 11 ) THEN
         PRINT 2000, INUM
         STOP
      ELSE
         ID05AD = IC( INUM )
      ENDIF
      RETURN
 2000 FORMAT( ' INUM =', I3, ' OUT OF RANGE IN ID05AD.',
     *        ' EXECUTION TERMINATED.' )
      END
      REAL FUNCTION FD05A( INUM )
      INTEGER INUM
      REAL RC( 5 )
C
C  REAL CONSTANTS (SINGLE PRECISION ARITHMETIC).
C
C  OBTAINED FROM H.S.L. SUBROUTINE ZE02AM.
C  NICK GOULD AND SID MARLOW, HARWELL, APRIL 1988.
C
C  RC(1) THE SMALLEST POSITIVE NUMBER: 1.0 + RC(1) > 1.0.
C  RC(2) THE SMALLEST POSITIVE NUMBER: 1.0 - RC(2) < 1.0.
C  RC(3) THE SMALLEST NONZERO +VE REAL NUMBER.
C  RC(4) THE SMALLEST FULL PRECISION +VE REAL NUMBER.
C  RC(5) THE LARGEST FINITE +VE REAL NUMBER.
C
      DATA RC( 1 ) /     0.953674317E-06 /
```

```
      DATA RC( 2 ) /     0.596046449E-07 /
      DATA RC( 3 ) /     0.539760536E-78 /
      DATA RC( 4 ) /     0.539760536E-78 /
      DATA RC( 5 ) /     0.723700514E+76 /
      IF ( INUM .LE. 0 .OR. INUM .GE. 6 ) THEN
         PRINT 2000, INUM
         STOP
      ELSE
         FD05A = RC( INUM )
      ENDIF
      RETURN
 2000 FORMAT( ' INUM =', I3, ' OUT OF RANGE IN FD05A.',
     *        ' EXECUTION TERMINATED.' )
      END
      DOUBLE PRECISION FUNCTION FD05AD( INUM )
      INTEGER INUM
      DOUBLE PRECISION DC( 5 )
C
C  REAL CONSTANTS (DOUBLE PRECISION ARITHMETIC).
C
C  OBTAINED FROM H.S.L. SUBROUTINE ZE02AM.
C  NICK GOULD AND SID MARLOW, HARWELL, APRIL 1988.
C
C  DC(1) THE SMALLEST POSITIVE NUMBER: 1.0 + DC(1) > 1.0.
C  DC(2) THE SMALLEST POSITIVE NUMBER: 1.0 - DC(2) < 1.0.
C  DC(3) THE SMALLEST NONZERO +VE REAL NUMBER.
C  DC(4) THE SMALLEST FULL PRECISION +VE REAL NUMBER.
C  DC(5) THE LARGEST FINITE +VE REAL NUMBER.
C
      DATA DC( 1 ) /     0.22204460492503130D-15 /
      DATA DC( 2 ) /     0.13877787807814457D-16 /
      DATA DC( 3 ) /     0.53976053469340279D-78 /
      DATA DC( 4 ) /     0.53976053469340279D-78 /
      DATA DC( 5 ) /     0.72370055773322621D+76 /
      IF ( INUM .LE. 0 .OR. INUM .GE. 6 ) THEN
         PRINT 2000, INUM
         STOP
      ELSE
         FD05AD = DC( INUM )
      ENDIF
      RETURN
 2000 FORMAT( ' INUM =', I3, ' OUT OF RANGE IN FD05AD.',
     *        ' EXECUTION TERMINATED.' )
      END
```