NAG Library Routine Document

X02AJF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

X02AJF returns ϵ, the value ***machine precision***.

2 Specification

```
FUNCTION X02AJF ()
REAL (KIND=nag_wp) X02AJF
```

3 Description

X02AJF returns ***machine precision***, computed as $\epsilon = \frac{1}{2} \times b^{1-p}$, where b is the arithmetic base (see X02BHF) and p is the number of significant base-b digits (see X02BJF).

It is important to note that the definition of ϵ here differs from that in ISO Fortran 95 (1997).

4 References

ISO Fortran 95 (1997) ISO Fortran 95 programming language (ISO/IEC 1539–1:1997)

5 Arguments

None.

6 Error Indicators and Warnings

None.

7 Accuracy

None.

8 Parallelism and Performance

X02AJF is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example prints the values of all the functions in Chapter X02. The results will vary from one implementation of the Library to another.

10.1 Program Text

```
    Program x02ajfe

!     X02AJF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: nag_wp, x02ahf, x02ajf, x02akf, x02alf, x02amf,   &
                             x02anf, x02bbf, x02bef, x02bhf, x02bjf, x02bkf,   &
                             x02blf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                :: nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)                :: largest_arg, largest_pos, machpr,    &
                                           safe_complex, safe_real,             &
                                           smallest_pos
      Integer                           :: largest_pos_int, model_b,            &
                                           model_emax, model_emin, model_p,     &
                                           prec
!     .. Executable Statements ..
      Write (nout,*) 'X02AJF Example Program Results'
      Write (nout,*)
      Write (nout,*) '(results are machine-dependent)'
      Write (nout,*)
      Write (nout,*) 'The basic parameters of the model'
      Write (nout,*)
      model_b = x02bhf()
      Write (nout,99999) ' X02BHF = ', model_b, '  (the model parameter B)'
      model_p = x02bjf()
      Write (nout,99999) ' X02BJF = ', model_p, '  (the model parameter P)'
      model_emin = x02bkf()
      Write (nout,99999) ' X02BKF = ', model_emin,                              &
        '  (the model parameter EMIN)'
      model_emax = x02blf()
      Write (nout,99999) ' X02BLF = ', model_emax,                              &
        '  (the model parameter EMAX)'
      Write (nout,*)
      Write (nout,*) 'Derived parameters of floating-point arithmetic'
      Write (nout,*)
      machpr = x02ajf()
      Write (nout,99998) ' X02AJF = ', machpr, '  (the machine precision)'
      smallest_pos = x02akf()
      Write (nout,99998) ' X02AKF = ', smallest_pos,                            &
        '  (the smallest positive model number)'
      largest_pos = x02alf()
      Write (nout,99998) ' X02ALF = ', largest_pos,                             &
        '  (the largest positive model number)'
      safe_real = x02amf()
      Write (nout,99998) ' X02AMF = ', safe_real,                               &
        '  (the real safe range parameter)'
      safe_complex = x02anf()
      Write (nout,99998) ' X02ANF = ', safe_complex,                            &
        '  (the complex safe range parameter)'
      Write (nout,*)
      Write (nout,*)                                                            &
        'Parameters of other aspects of the computing environment'
      Write (nout,*)
      largest_arg = x02ahf(0.0E0_nag_wp)
      Write (nout,99996) ' X02AHF = ', largest_arg,                             &
        '  (largest argument for SIN and COS)'
      largest_pos_int = x02bbf(0.0E0_nag_wp)
      Write (nout,99997) ' X02BBF = ', largest_pos_int,                         &
        '  (largest positive integer)'
      prec = x02bef(0.0E0_nag_wp)
      Write (nout,99997) ' X02BEF = ', prec, '  (precision in decimal digits)'
```

```
99999 Format (1X,A,I7,1X,A)
99998 Format (1X,A,1P,E26.18e3,1X,A)
99997 Format (1X,A,I20,1X,A)
99996 Format (1X,A,1P,E20.8e3,1X,A)
    End Program x02ajfe
```

10.2 Program Data

None.

10.3 Program Results

```
 X02AJF Example Program Results

 (results are machine-dependent)

 The basic parameters of the model

  X02BHF =       2   (the model parameter B)
  X02BJF =      53   (the model parameter P)
  X02BKF =   -1021   (the model parameter EMIN)
  X02BLF =    1024   (the model parameter EMAX)

 Derived parameters of floating-point arithmetic

  X02AJF =  1.11022302462515654OE-016   (the machine precision)
  X02AKF =  2.22507385850720138E-308   (the smallest positive model number)
  X02ALF =  1.79769313486231570E+308   (the largest positive model number)
  X02AMF =  2.22507385850720187E-308   (the real safe range parameter)
  X02ANF =  2.22507385850720187E-308   (the complex safe range parameter)

 Parameters of other aspects of the computing environment

  X02AHF =      1.42724769E+045   (largest argument for SIN and COS)
  X02BBF =           2147483647   (largest positive integer)
  X02BEF =                   15   (precision in decimal digits)
```