NAG Fortran Library Routine Document

C05AZF

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

C05AZF locates a simple zero of a continuous function on a given interval by a combination of the methods of linear interpolation, linear extrapolation and bisection. It uses reverse communication for evaluating the function.

2 Specification

```
SUBROUTINE C05AZF(X, Y, FX, TOLX, IR, C, IND, IFAIL)
INTEGER            IR, IND, IFAIL
real               X, Y, FX, TOLX, C(17)
```

3 Description

The user must supply an initial interval $[\mathrm{X},\mathrm{Y}]$ containing a simple zero of the function $f(x)$ (the choice of X and Y must be such that $f(\mathrm{X}) \times f(\mathrm{Y}) \leq 0.0$). The routine combines the methods of bisection, linear interpolation and linear extrapolation (see Dahlquist and Björck (1974)), to find a sequence of sub-intervals of the initial interval such that the final interval $[\mathrm{X},\mathrm{Y}]$ contains the zero and $|\mathrm{X}-\mathrm{Y}|$ is less than some tolerance specified by TOLX and IR (see Section 5). In fact, since the intervals $[\mathrm{X},\mathrm{Y}]$ are determined only so that $f(\mathrm{X}) \times f(\mathrm{Y}) \leq 0$, it is possible that the final interval may contain a discontinuity or a pole of f (violating the requirement that f be continuous). C05AZF checks if the sign change is likely to correspond to a pole of f and gives an error return in this case.

C05AZF returns to the calling program for each evaluation of $f(x)$. On each return the user should set $\mathrm{FX} = f(\mathrm{X})$ and call C05AZF again.

The routine is a modified version of procedure 'zeroin' given by Bus and Dekker (1975).

4 References

Dahlquist G and Björck Å (1974) *Numerical Methods* Prentice-Hall

Bus J C P and Dekker T J (1975) Two efficient algorithms with guaranteed convergence for finding a zero of a function *ACM Trans. Math. Software* **1** 330–345

5 Parameters

Note: this routine uses **reverse communication.** Its use involves an initial entry, intermediate exits and re-entries, and a final exit, as indicated by the **parameter IND**. Between intermediate exits and re-entries, **all parameters other than FX must remain unchanged**.

1: X – ***real*** *Input/Output*
2: Y – ***real*** *Input/Output*

On initial entry: X and Y must define an initial interval containing the zero, such that $f(\mathrm{X}) \times f(\mathrm{Y}) \leq 0$. It is not necessary that $\mathrm{X} < \mathrm{Y}$.

On intermediate exit: X contains the point at which f must be evaluated before re-entry to the routine.

On final exit: X and Y define a smaller interval containing the zero, such that $f(\mathrm{X}) \times f(\mathrm{Y}) \leq 0$, and $|\mathrm{X}-\mathrm{Y}|$ satisfies the accuracy specified by TOLX and IR, unless an error has occurred. If $\mathrm{IFAIL} = 4$, X and Y generally contain very good approximations to a pole; if $\mathrm{IFAIL} = 5$, X and Y generally contain very good approximations to the zero (see Section 6). If a point X is found such that $f(\mathrm{X}) = 0$, then on final exit X = Y (in this case there is no guarantee that X is a simple zero).

3: FX – ***real*** *Input/Output*

On initial entry: if IND = 1, FX need not be set.

If IND = −1, FX must contain $f(\mathrm{X})$ for the initial value of X.

On intermediate re-entry: FX must contain $f(\mathrm{X})$ for the current value of X.

On exit: FX is unchanged, except that after initial entry with IND = −1FX contains the input value of C(1).

4: TOLX – ***real*** *Input*

On initial entry: the accuracy to which the zero is required. The type of error test is specified by IR (below).

Constraint: TOLX > 0.

5: IR – INTEGER *Input*

On initial entry: indicates the type of error test as follows:

if IR = 0, the test is: $|\mathrm{X} - \mathrm{Y}| \leq 2.0 \times \mathrm{TOLX} \times \max(1.0, |\mathrm{Z}|)$;

if IR = 1, the test is: $|\mathrm{X} - \mathrm{Y}| \leq 2.0 \times \mathrm{TOLX}$;

if IR = 2, the test is: $|\mathrm{X} - \mathrm{Y}| \leq 2.0 \times \mathrm{TOLX} \times |\mathrm{Z}|$.

Here Z is the value of x for which $|f(x)|$ is currently known to have the smallest value; Z is calculated internally to C05AZF.

Suggested value: IR = 0.

Constraint: IR = 0, 1 or 2.

6: C(17) – ***real*** array *Input/Output*

On initial entry: if IND = 1, no elements of C need be set.

If IND = −1, C(1) must contain $f(\mathrm{Y})$, other elements of C need not be set.

On final exit: C is undefined.

7: IND – INTEGER *Input/Output*

On initial entry: IND must be set to 1 or −1:

if IND = 1, FX and C(1) need not be set;

if IND = −1, FX and C(1) must contain $f(\mathrm{X})$ and $f(\mathrm{Y})$ respectively.

On intermediate exit: IND contains 2, 3 or 4. The calling program must evaluate f at X, storing the result in FX, and re-enter C05AZF with all other parameters unchanged.

On final exit: IND contains 0.

Constraint: on entry IND = −1, 1, 2, 3 or 4.

8: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

For environments where it might be inappropriate to halt program execution when an error is detected, the value −1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, for users not familiar with this parameter the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, $f(\mathrm{X})$ and $f(\mathrm{Y})$ have the same sign, with $f(\mathrm{X}) \neq 0.0$.

IFAIL $= 2$

On entry, IND $\neq -1$, 1, 2, 3 or 4.

IFAIL $= 3$

On entry, TOLX ≤ 0.0,
or IR $\neq 0$, 1 or 2.

IFAIL $= 4$

An interval $[\mathrm{X}, \mathrm{Y}]$ has been determined satisfying the error tolerance specified by TOLX and IR and such that $f(\mathrm{X}) \times f(\mathrm{Y}) \leq 0$. However, from observation of the values of f during the calculation of $[\mathrm{X}, \mathrm{Y}]$, it seems that the interval $[\mathrm{X}, \mathrm{Y}]$ contains a pole rather than a zero. Note that this error exit is not completely reliable: the error exit may be taken in extreme cases when $[\mathrm{X}, \mathrm{Y}]$ contains a zero, or the error exit may not be taken when $[\mathrm{X}, \mathrm{Y}]$ contains a pole. Both these cases occur most frequently when TOLX is large.

IFAIL $= 5$

The tolerance TOLX is too small for the problem being solved. This indicator is only set when the length of the interval $[\mathrm{X}, \mathrm{Y}]$ containing the zero has been reduced as much as possible without satisfying the accuracy requirement (see Section 3 and Section 5). The values X and Y returned are usually both very good approximations to the zero.

7 Accuracy

The accuracy of the final value X as an approximation of the zero is determined by TOLX and IR as described above. A relative accuracy criterion (IR $= 2$) should not be used when the initial values X and Y are of different orders of magnitude. In this case a change of origin of the independent variable may be appropriate. For example, if the initial interval $[\mathrm{X}, \mathrm{Y}]$ is transformed linearly to the interval $[1, 2]$, then the zero can be determined to a precise number of figures using an absolute (IR $= 1$) or relative (IR $= 2$) error test and the effect of the transformation back to the original interval can also be determined. Except for the accuracy check, such a transformation has no effect on the calculation of the zero.

8 Further Comments

For most problems, the time taken on each call to C05AZF will be negligible compared with the time spent evaluating $f(x)$ between calls to C05AZF.

If the calculation terminates because $f(\mathrm{X}) = 0.0$, then on return Y is set to X. (In fact, $\mathrm{Y} = \mathrm{X}$ on return only in this case and, possibly, when IFAIL $= 5$.) There is no guarantee that the value returned in X corresponds to a **simple** root and the user should check whether it does.

One way to check this is to compute the derivative of f at the point X, preferably analytically, or, if this is not possible, numerically, perhaps by using a central difference estimate.

If $f'(\mathrm{X}) = 0.0$, then X must correspond to a multiple zero of f rather than a simple zero.

9 Example

To calculate a zero of $e^{-x} - x$ with an initial interval $[0, 1]$, TOLX $= 1.0\mathrm{E}{-5}$ and a mixed error test.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     C05AZF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             FX, TOLX, X, Y
      INTEGER          IFAIL, IND, IR
*     .. Local Arrays ..
      real             C(17)
*     .. External Functions ..
      real             F
      EXTERNAL         F
*     .. External Subroutines ..
      EXTERNAL         C05AZF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C05AZF Example Program Results'
      WRITE (NOUT,*)
      WRITE (NOUT,*) ' Iterations'
      WRITE (NOUT,*)
      TOLX = 1.0e-5
      X = 0.0e0
      Y = 1.0e0
      IR = 0
      IFAIL = 1
      IND = 1
*
   20 CALL C05AZF(X,Y,FX,TOLX,IR,C,IND,IFAIL)
*
      IF (IND.NE.0) THEN
         IF (IND.LT.2 .OR. IND.GT.4) THEN
            WRITE (NOUT,99997) 'Failure with IND=', IND, ' at X=', X
         ELSE
            FX = F(X)
            WRITE (NOUT,99999) ' X=', X, '   FX=', FX, '   IND=', IND
            GO TO 20
         END IF
      ELSE
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,*) ' Solution'
            WRITE (NOUT,*)
            WRITE (NOUT,99998) ' X=', X, '   Y=', Y
         ELSE
            WRITE (NOUT,99997) 'IFAIL = ', IFAIL
            IF (IFAIL.EQ.4 .OR. IFAIL.EQ.5) WRITE (NOUT,99998) 'X =', X,
     +          '  Y =', Y
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,F8.5,A,e12.4,A,I2)
99998 FORMAT (1X,A,F8.5,A,F8.5)
99997 FORMAT (1X,A,I2,A,F10.4)
      END
*
      real FUNCTION F(X)
*     .. Scalar Arguments ..
      real            X
*     .. Intrinsic Functions ..
```

```
      INTRINSIC       EXP
*     .. Executable Statements ..
      F = EXP(-X) - X
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 C05AZF Example Program Results

  Iterations

  X= 0.00000   FX=  0.1000E+01   IND= 2
  X= 1.00000   FX= -0.6321E+00   IND= 3
  X= 0.61270   FX= -0.7081E-01   IND= 4
  X= 0.56384   FX=  0.5182E-02   IND= 4
  X= 0.56717   FX= -0.4242E-04   IND= 4
  X= 0.56714   FX= -0.2538E-07   IND= 4
  X= 0.56714   FX=  0.7810E-05   IND= 4

  Solution

  X= 0.56714   Y= 0.56714
```