C05AVF – NAG Fortran Library Routine Document

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

C05AVF attempts to locate an interval containing a simple zero of a continuous function using a binary search. It uses reverse communication for evaluating the function.

2 Specification

```
SUBROUTINE C05AVF(X, FX, H, BOUNDL, BOUNDU, Y, C, IND, IFAIL)
INTEGER            IND, IFAIL
real               X, FX, H, BOUNDL, BOUNDU, Y, C(11)
```

3 Description

The user must supply an initial point X and a step H. The routine attempts to locate a short interval $[\mathrm{X},\mathrm{Y}] \subset [\mathrm{BOUNDL},\mathrm{BOUNDU}]$ containing a simple zero of $f(x)$.

(On exit we may have X > Y; X is determined as the first point encountered in a binary search where the sign of $f(x)$ differs from the sign of $f(x)$ at the initial input point X.) The routine attempts to locate a zero of $f(x)$ using H, $0.1 \times$ H, $0.01 \times$ H and $0.001 \times$ H in turn as its basic step before quitting with an error exit if unsuccessful.

C05AVF 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 C05AVF again.

4 References

None.

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*

On initial entry: the best available approximation to the zero.

Constraint: X must lie in the closed interval [BOUNDL, BOUNDU] (see below).

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

On final exit: X contains one end of an interval containing the zero, the other end being in Y (below), unless an error has occurred. If IFAIL = 4, X and Y are the end-points of the largest interval searched. If a zero is located exactly, its value is returned in X (and in Y).

2: FX — ***real*** *Input*

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.

3: H — ***real*** *Input/Output*

On initial entry: a basic step size which is used in the binary search for an interval containing a zero. The basic step sizes H, $0.1 \times \mathrm{H}$, $0.01 \times \mathrm{H}$ and $0.001 \times \mathrm{H}$ are used in turn when searching for the zero.

Constraint: either $\mathrm{X} + \mathrm{H}$ or $\mathrm{X} - \mathrm{H}$ must lie inside the closed interval [BOUNDL, BOUNDU] (see below).

H must be sufficiently large that $\mathrm{X} + \mathrm{H} \neq \mathrm{X}$ on the computer.

On final exit: H is undefined.

4: BOUNDL — ***real*** *Input*

5: BOUNDU — ***real*** *Input*

On initial entry: BOUNDL and BOUNDU must contain respectively lower and upper bounds for the interval of search for the zero.

Constraint: BOUNDL < BOUNDU.

6: Y — ***real*** *Input/Output*

On initial entry: Y need not be set.

On final exit: Y contains the closest point found to the final value of X, such that $f(\mathrm{X}) \times f(\mathrm{Y}) \leq 0$. If a value X is found such that $f(\mathrm{X}) = 0$, then Y = X. On final exit with IFAIL = 4, X and Y are the end-points of the largest interval searched.

7: C(11) — ***real*** array *Workspace*

(On final exit with IFAIL = 0 or 4, C(1) contains $f(\mathrm{Y})$.)

8: IND — INTEGER *Input/Output*

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

if IND = 1, FX need not be set;
if IND = −1, FX must contain $f(\mathrm{X})$.

On intermediate exit: IND contains 2 or 3. The calling program must evaluate f at X, storing the result in FX, and re-enter C05AVF with all other parameters unchanged.

On final exit: IND contains 0.

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

9: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

On entry, BOUNDU ≤ BOUNDL,
or $\mathrm{X} \notin$ [BOUNDL, BOUNDU],
or both $\mathrm{X} + \mathrm{H}$ and $\mathrm{X} - \mathrm{H} \notin$ [BOUNDL, BOUNDU].

IFAIL = 2

On initial entry, H is too small to be used to perturb the initial value of X in the search.

IFAIL = 3

The parameter IND is incorrectly set on initial or intermediate entry.

IFAIL = 4

The routine has been unable to determine an interval containing a simple zero starting from the initial value of X and using the step H. A user who has prior knowledge that a simple zero lies in the interval [BOUNDL, BOUNDU], should vary X and H in an attempt to find it. (See also Section 8.)

7 Accuracy

This routine is not intended to be used to obtain accurate approximations to the zero of $f(x)$ but rather to locate an interval containing a zero. This interval can then be used as input to an accurate rootfinder such as C05AZF or C05ADF. The size of the interval determined depends somewhat unpredictably on the choice of X and H. The closer X is to the root and the **smaller** the initial value of H, then, in general, the smaller (more accurate) the interval determined; however, the accuracy of this statement depends to some extent on the behaviour of $f(x)$ near $x = \mathrm{X}$ and on the size of H.

8 Further Comments

For most problems, the time taken on each call to C05AVF will be negligible compared with the time spent evaluating $f(x)$ between calls to C05AVF. However, the initial choices of X and H will clearly affect the number of evaluations of $f(x)$. In general, the closer X is to the root and the **larger** the initial value of H then the less the time taken. (However taking H large can affect the accuracy and reliability of the routine, see below.)

The user is expected to choose BOUNDL and BOUNDU as physically (or mathematically) realistic limits on the interval of search. For example, it may be known, from physical arguments, that no zero of $f(x)$ of interest will lie outside [BOUNDL, BOUNDU]. Alternatively, $f(x)$ may be more expensive to evaluate for some values of X than for others and such expensive evaluations can sometimes be avoided by careful choice of BOUNDL and BOUNDU.

The choice of BOUNDL and BOUNDU affects the search only in that these values provide physical limitations on the search values and that the search is terminated if it seems, from the available information about $f(x)$, that the zero lies outside [BOUNDL, BOUNDU]. In this case (IFAIL = 4 on exit), only one of $f(\mathrm{BOUNDL})$ and $f(\mathrm{BOUNDU})$ may have been evaluated and a zero close to the other end of the interval could be missed. The actual interval searched is returned in the parameters X and Y and the user can call C05AVF again to search the remainder of the original interval.

Though C05AVF is intended primarily for determining an interval containing a zero of $f(x)$, it may be used to shorten a known interval. This could be useful if, for example, a large interval containing the zero is known and it is also known that the root lies close to one end of the interval; by setting X to this end of the interval and H small, a short interval will usually be determined. However, it is worth noting that once any interval containing a zero has been determined, a call to C05AZF will usually be the most efficient way to calculate an interval of specified length containing the zero. To assist in this determination, the information in X, Y, FX and C(1) on successful exit from C05AVF is in the correct form for a call to routine C05AZF with IND = -1.

If the calculation terminates because $f(\mathrm{X}) = 0.0$, then on return Y is set to X. (In fact, Y = X on return only in this case.) In this case, there is no guarantee that the value in X corresponds to a **simple** zero 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 find a sub-interval of $[0.0, 4.0]$ containing a zero of $x^2 - 3x + 2$. The zero nearest to 3.0 is required and so we set X = 3.0 initially.

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.

```
*     C05AVF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             BOUNDL, BOUNDU, FX, H, X, Y
      INTEGER          IFAIL, IND
*     .. Local Arrays ..
      real             C(11)
*     .. External Subroutines ..
      EXTERNAL         C05AVF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C05AVF Example Program Results'
      WRITE (NOUT,*)
      X = 3.0e0
      H = 0.1e0
      BOUNDL = 0.0e0
      BOUNDU = 4.0e0
      IFAIL = 1
      IND = 1
*
   20 CALL C05AVF(X,FX,H,BOUNDL,BOUNDU,Y,C,IND,IFAIL)
*
      IF (IND.NE.0) THEN
         FX = X*X - 3.0e0*X + 2.0e0
         GO TO 20
      ELSE
         IF (IFAIL.GT.0) THEN
            WRITE (NOUT,99997) 'Error exit,  IFAIL =', IFAIL
         ELSE
            WRITE (NOUT,*) 'Interval containing root is (Y,X) where'
            WRITE (NOUT,99999) 'Y =', Y, '   X =', X
            WRITE (NOUT,*) 'Values of f at Y and X are'
            WRITE (NOUT,99998) 'f(Y) =', C(1), '   f(X) =', FX
         END IF
      END IF
      STOP
*
99999 FORMAT (1X,A,F12.4,A,F12.4)
99998 FORMAT (1X,A,F12.2,A,F12.2)
99997 FORMAT (1X,A,I2)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
C05AVF Example Program Results

Interval containing root is (Y,X) where
Y =      2.5000   X =      1.7000
Values of f at Y and X are
f(Y) =        0.75   f(X) =       -0.21
```