NAG Library Routine Document

C05AUF

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

C05AUF locates a simple zero of a continuous function from a given starting value. It uses a binary search to locate an interval containing a zero of the function, then Brent's method, which is a combination of nonlinear interpolation, linear extrapolation and bisection, to locate the zero precisely.

2 Specification

```
SUBROUTINE C05AUF (X, H, EPS, ETA, F, A, B, IUSER, RUSER, IFAIL)

INTEGER             IUSER(*), IFAIL
REAL (KIND=nag_wp)  X, H, EPS, ETA, F, A, B, RUSER(*)
EXTERNAL            F
```

3 Description

C05AUF attempts to locate an interval $[a, b]$ containing a simple zero of the function $f(x)$ by a binary search starting from the initial point $x = \mathrm{X}$ and using repeated calls to C05AVF. If this search succeeds, then the zero is determined to a user-specified accuracy by a call to C05AYF. The specifications of routines C05AVF and C05AYF should be consulted for details of the methods used.

The approximation x to the zero α is determined so that at least one of the following criteria is satisfied:

(i) $|x - \alpha| \leq \mathrm{EPS}$,

(ii) $|f(x)| \leq \mathrm{ETA}$.

4 References

Brent R P (1973) *Algorithms for Minimization Without Derivatives* Prentice–Hall

5 Arguments

1: X – REAL (KIND=nag_wp) *Input/Output*

On entry: an initial approximation to the zero.

On exit: if $\mathrm{IFAIL} = 0$ or 4, X is the final approximation to the zero.

If $\mathrm{IFAIL} = 3$, X is likely to be a pole of $f(x)$.

Otherwise, X contains no useful information.

2: H – REAL (KIND=nag_wp) *Input*

On entry: a step length for use in the binary search for an interval containing the zero. The maximum interval searched is $[\mathrm{X} - 256.0 \times \mathrm{H}, \mathrm{X} + 256.0 \times \mathrm{H}]$.

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

3: EPS – REAL (KIND=nag_wp) *Input*

On entry: the termination tolerance on x (see Section 3).

Constraint: $\mathrm{EPS} > 0.0$.

4: ETA – REAL (KIND=nag_wp) *Input*

On entry: a value such that if $|f(x)| \leq$ ETA, x is accepted as the zero. ETA may be specified as 0.0 (see Section 7).

5: F – REAL (KIND=nag_wp) FUNCTION, supplied by the user. *External Procedure*

F must evaluate the function f whose zero is to be determined.

The specification of F is:

```
FUNCTION F (X, IUSER, RUSER)
REAL (KIND=nag_wp) F

INTEGER             IUSER(*)
REAL (KIND=nag_wp) X, RUSER(*)
```

1: X – REAL (KIND=nag_wp) *Input*

On entry: the point at which the function must be evaluated.

2: IUSER($*$) – INTEGER array *User Workspace*
3: RUSER($*$) – REAL (KIND=nag_wp) array *User Workspace*

F is called with the arguments IUSER and RUSER as supplied to C05AUF. You should use the arrays IUSER and RUSER to supply information to F.

F must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub) program from which C05AUF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

6: A – REAL (KIND=nag_wp) *Output*
7: B – REAL (KIND=nag_wp) *Output*

On exit: the lower and upper bounds respectively of the interval resulting from the binary search. If the zero is determined exactly such that $f(x) = 0.0$ or is determined so that $|f(x)| \leq$ ETA at any stage in the calculation, then on exit A $=$ B $= x$.

8: IUSER($*$) – INTEGER array *User Workspace*
9: RUSER($*$) – REAL (KIND=nag_wp) array *User Workspace*

IUSER and RUSER are not used by C05AUF, but are passed directly to F and should be used to pass information to this routine.

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

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, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

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, EPS $= \langle value \rangle$.
Constraint: EPS > 0.0.

On entry, X $= \langle value \rangle$ and H $= \langle value \rangle$.
Constraint: X + H $\neq$ X (to machine accuracy).

IFAIL $= 2$

An interval containing the zero could not be found. Increasing H and calling C05AUF again will increase the range searched for the zero. Decreasing H and calling C05AUF again will refine the mesh used in the search for the zero.

IFAIL $= 3$

Solution may be a pole rather than a zero.

IFAIL $= 4$

The tolerance EPS has been set too small for the problem being solved. However, the value X returned is a good approximation to the zero. EPS $= \langle value \rangle$.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The levels of accuracy depend on the values of EPS and ETA. If full machine accuracy is required, they may be set very small, resulting in an exit with IFAIL $= 4$, although this may involve many more iterations than a lesser accuracy. You are recommended to set ETA $= 0.0$ and to use EPS to control the accuracy, unless you have considerable knowledge of the size of $f(x)$ for values of x near the zero.

8 Parallelism and Performance

C05AUF is not threaded in any implementation.

9 Further Comments

The time taken by C05AUF depends primarily on the time spent evaluating F (see Section 5). The accuracy of the initial approximation X and the value of H will have a somewhat unpredictable effect on the timing.

If it is important to determine an interval of relative length less than $2 \times$ EPS containing the zero, or if F is expensive to evaluate and the number of calls to F is to be restricted, then use of C05AVF followed by C05AZF is recommended. Use of this combination is also recommended when the structure of the problem to be solved does not permit a simple F to be written: the reverse communication facilities of these routines are more flexible than the direct communication of F required by C05AUF.

If the iteration terminates with successful exit and $\mathrm{A} = \mathrm{B} = \mathrm{X}$ there is no guarantee that the value returned in X corresponds to a simple zero and you 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.

10 Example

This example calculates an approximation to the zero of $x - e^{-x}$ using a tolerance of $\mathrm{EPS} = 1.0\mathrm{E}{-5}$ starting from $\mathrm{X} = 1.0$ and using an initial search step $\mathrm{H} = 0.1$.

10.1 Program Text

```
!   C05AUF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module c05aufe_mod

!     C05AUF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                           :: f
!     .. Parameters ..
      Integer, Parameter, Public       :: nout = 6
    Contains
      Function f(x,iuser,ruser)

!       .. Function Return Value ..
        Real (Kind=nag_wp)             :: f
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In) :: x
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Inout) :: ruser(*)
        Integer, Intent (Inout)        :: iuser(*)
!       .. Intrinsic Procedures ..
        Intrinsic                      :: exp
!       .. Executable Statements ..
        f = x - exp(-x)

        Return

      End Function f
    End Module c05aufe_mod
    Program c05aufe

!     C05AUF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: c05auf, nag_wp
      Use c05aufe_mod, Only: f, nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: a, b, eps, eta, h, x
      Integer                          :: ifail
```

```
!     .. Local Arrays ..
      Real (Kind=nag_wp)                 :: ruser(1)
      Integer                            :: iuser(1)
!     .. Executable Statements ..
      Write (nout,*) 'C05AUF Example Program Results'

      x = 1.0E0_nag_wp
      h = 0.1E0_nag_wp
      eps = 1.0E-5_nag_wp
      eta = 0.0E0_nag_wp

      ifail = -1
      Call c05auf(x,h,eps,eta,f,a,b,iuser,ruser,ifail)

      Write (nout,*)

      Select Case (ifail)
      Case (0)
        Write (nout,99999) 'Root is ', x
        Write (nout,99998) 'Interval searched is [', a, ',', b, ']'
      Case (3,4)
        Write (nout,99999) 'Final value = ', x
      End Select

99999 Format (1X,A,F13.5)
99998 Format (1X,A,2(F8.5,A))
    End Program c05aufe
```

10.2 Program Data

None.

10.3 Program Results

```
 C05AUF Example Program Results

 Root is       0.56714
 Interval searched is [ 0.50000, 0.90000]
```