NAG Library Routine Document

C05AWF

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

C05AWF attempts to locate a zero of a continuous function using a continuation method based on a secant iteration.

2 Specification

```
SUBROUTINE C05AWF (X, EPS, ETA, F, NFMAX, IUSER, RUSER, IFAIL)

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

3 Description

C05AWF attempts to obtain an approximation to a simple zero α of the function $f(x)$ given an initial approximation x to α. The zero is found by a call to C05AXF whose specification should be consulted for details of the method used.

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

(i) $|x - \alpha| \sim \text{EPS}$,

(ii) $|f(x)| < \text{ETA}$.

4 References

None.

5 Arguments

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

On entry: an initial approximation to the zero.

On exit: the final approximation to the zero, unless IFAIL = 1, 2 or 5, in which case it contains no useful information.

2: EPS – REAL (KIND=nag_wp)   *Input*

On entry: an absolute tolerance to control the accuracy to which the zero is determined. In general, the smaller the value of EPS the more accurate X will be as an approximation to α. Indeed, for very small positive values of EPS, it is likely that the final approximation will satisfy $|X - \alpha| < \text{EPS}$. You are advised to call the routine with more than one value for EPS to check the accuracy obtained.

Constraint: EPS > 0.0.

3: ETA – REAL (KIND=nag_wp)   *Input*

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

4:     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 C05AWF. 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 C05AWF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

5:     NFMAX – INTEGER   *Input*

On entry: the maximum permitted number of calls to F from C05AWF. If F is inexpensive to evaluate, NFMAX should be given a large value (say > 1000).

Constraint: $\text{NFMAX} > 0$.

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

IUSER and RUSER are not used by C05AWF, but are passed directly to F and should be used to pass information to this routine.

8:     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: $\text{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6   Error Indicators and Warnings

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

Errors or warnings detected by the routine:

$\text{IFAIL} = 1$

On entry, $\text{EPS} = \langle value \rangle$.
Constraint: $\text{EPS} > 0.0$.

On entry, NFMAX = ⟨*value*⟩.
Constraint: NFMAX > 0.

IFAIL = 2

Internal scale factor invalid for this problem. Consider using C05AXF instead and setting SCAL.

IFAIL = 3

Either F has no zero near X or too much accuracy has been requested. Check the coding of F or increase EPS.

IFAIL = 4

More than NFMAX calls have been made to F.

NFMAX may be too small for the problem (because X is too far away from the zero), or F has no zero near X, or too much accuracy has been requested in calculating the zero. Increase NFMAX, check the coding of F or increase EPS.

IFAIL = 5

A serious error occurred in an internal call to an auxiliary routine.

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 = 3 or 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

C05AWF is not threaded in any implementation.

9   Further Comments

The time taken by C05AWF depends primarily on the time spent evaluating the function f (see Section 5) and on how close the initial value of X is to the zero.

If a more flexible way of specifying the function f is required or if you wish to have closer control of the calculation, then the reverse communication routine C05AXF is recommended instead of C05AWF.

10 Example

This example calculates the zero of $f(x) = e^{-x} - x$ from a starting value X = 1.0. Two calculations are made with EPS = 1.0E−3 and 1.0E−4 for comparison purposes, with ETA = 0.0 in both cases.

10.1 Program Text

```
!   C05AWF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module c05awfe_mod

!      C05AWF 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 = exp(-x) - x

          Return

       End Function f
    End Module c05awfe_mod
    Program c05awfe

!      C05AWF Example Main Program

!      .. Use Statements ..
       Use nag_library, Only: c05awf, nag_wp
       Use c05awfe_mod, Only: f, nout
!      .. Implicit None Statement ..
       Implicit None
!      .. Local Scalars ..
       Real (Kind=nag_wp)                :: eps, eta, x
       Integer                           :: ifail, k, nfmax
!      .. Local Arrays ..
       Real (Kind=nag_wp)                :: ruser(1)
       Integer                           :: iuser(1)
!      .. Executable Statements ..
       Write (nout,*) 'C05AWF Example Program Results'

       Write (nout,*)

!      Repeat with tolerance eps set to varying powers of 10:

loop: Do k = 3, 4
          eps = 10.0E0_nag_wp**(-k)
          x = 1.0E0_nag_wp
          eta = 0.0E0_nag_wp
```

```
      nfmax = 200

      ifail = -1
      Call c05awf(x,eps,eta,f,nfmax,iuser,ruser,ifail)

      Select Case (ifail)
      Case (0)
        Write (nout,99999) 'With EPS = ', eps, '    root = ', x
      Case (:-1)
        Exit loop
      Case (3,4)
        Write (nout,99999) 'With EPS = ', eps, ' final value = ', x
      End Select

    End Do loop

99999 Format (1X,A,E10.2,A,F14.5)
    End Program c05awfe
```

10.2 Program Data

None.

10.3 Program Results

```
 C05AWF Example Program Results

 With EPS =   0.10E-02   root =        0.56715
 With EPS =   0.10E-03   root =        0.56715
```