NAG Library Routine Document

C05AXF

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

C05AXF attempts to locate a zero of a continuous function using a continuation method based on a secant iteration. It uses reverse communication for evaluating the function.

2 Specification

```
SUBROUTINE C05AXF(X, FX, TOL, IR, SCAL, C, IND, IFAIL)
INTEGER            IR, IND, IFAIL
double precision   X, FX, TOL, SCAL, C(26)
```

3 Description

C05AXF uses a modified version of an algorithm given in Swift and Lindfield (1978) to compute a zero α of a continuous function $f(x)$. The algorithm used is based on a continuation method in which a sequence of problems

$$f(x) - \theta_r f(x_0), \qquad r = 0, 1, \ldots, m$$

are solved, where $1 = \theta_0 > \theta_1 > \cdots > \theta_m = 0$ (the value of m is determined as the algorithm proceeds) and where x_0 is your initial estimate for the zero of $f(x)$. For each θ_r the current problem is solved by a robust secant iteration using the solution from earlier problems to compute an initial estimate.

You must supply an error tolerance TOL. TOL is used directly to control the accuracy of solution of the final problem ($\theta_m = 0$) in the continuation method, and $\sqrt{\mathrm{TOL}}$ is used to control the accuracy in the intermediate problems ($\theta_1, \theta_2, \ldots, \theta_{m-1}$).

4 References

Swift A and Lindfield G R (1978) Comparison of a continuation method for the numerical solution of a single nonlinear equation *Comput. J.* **21** 359–362

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 – ***double precision*** *Input/Output*

On initial entry: an initial approximation to the zero.

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

On final exit: the final approximation to the zero.

2: FX – ***double precision*** *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: must contain $f(\mathrm{X})$ for the current value of X.

3: TOL – ***double precision*** *Input*

On initial entry: a value that controls the accuracy to which the zero is determined. TOL is used in determining the convergence of the secant iteration used at each stage of the continuation process. It is used directly when solving the last problem ($\theta_m = 0$ in Section 3), and $\sqrt{\mathrm{TOL}}$ is used for the problem defined by θ_r, $r < m$. Convergence to the accuracy specified by TOL is not guaranteed, and so you are recommended to find the zero using at least two values for TOL to check the accuracy obtained.

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

4: IR – INTEGER *Input*

On initial entry: indicates the type of error test required, as follows. Solving the problem defined by θ_r, $1 \leq r \leq m$, involves computing a sequence of secant iterates $x_r^0, x_r^1, \ldots$. This sequence will be considered to have converged only if:

for $\mathrm{IR} = 0$,

$$\left|x_r^{(i+1)} - x_r^{(i)}\right| \leq \mathit{eps} \times \max\left(1.0, \left|x_r^{(i)}\right|\right),$$

for $\mathrm{IR} = 1$,

$$\left|x_r^{(i+1)} - x_r^{(i)}\right| \leq \mathit{eps},$$

for $\mathrm{IR} = 2$,

$$\left|x_r^{(i+1)} - x_r^{(i)}\right| \leq \mathit{eps} \times \left|x_r^{(i)}\right|,$$

for some $i > 1$; here *eps* is either TOL or $\sqrt{\mathrm{TOL}}$ as discussed above. Note that there are other subsidiary conditions (not given here) which must also be satisfied before the secant iteration is considered to have converged.

Constraint: $\mathrm{IR} = 0$, 1 or 2.

5: SCAL – ***double precision*** *Input*

On initial entry: a factor for use in determining a significant approximation to the derivative of $f(x)$ at $x = x_0$, the initial value. A number of difference approximations to $f'(x_0)$ are calculated using

$$f'(x_0) \sim (f(x_0 + h) - f(x_0))/h$$

where $|h| < |\mathrm{SCAL}|$ and h has the same sign as SCAL. A significance (cancellation) check is made on each difference approximation and the approximation is rejected if insignificant.

Suggested value: the square root of the ***machine precision***.

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

6: $\mathrm{C}(26)$ – ***double precision*** array *Communication Array*

($\mathrm{C}(5)$ contains the current θ_r, this value may be useful in the event of an error exit.)

7: IND – INTEGER *Input/Output*

On initial entry: must be set to 1 or -1.

$\mathrm{IND} = 1$
FX need not be set.

$\mathrm{IND} = -1$
FX must contain $f(\mathrm{X})$.

On intermediate exit: contains 2, 3 or 4. The calling program must evaluate f at X, storing the result in FX, and re-enter C05AXF with all other parameters unchanged.

On final exit: contains 0.

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

8: IFAIL – INTEGER *Input/Output*

On initial entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

On final 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, because for this routine the values of the output parameters may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **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, TOL ≤ 0.0,
or IR $\neq 0$, 1 or 2.

IFAIL $= 2$

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

IFAIL $= 3$

SCAL is too small, or significant derivatives of f cannot be computed (this can happen when f is almost constant and nonzero, for any value of SCAL).

IFAIL $= 4$

The current problem in the continuation sequence cannot be solved, see C(5) for the value of θ_r. The most likely explanation is that the current problem has no solution, either because the original problem had no solution or because the continuation path passes through a set of insoluble problems. This latter reason for failure should occur rarely, and not at all if the initial approximation to the zero is sufficiently close. Other possible explanations are that TOL is too small and hence the accuracy requirement is too stringent, or that TOL is too large and the initial approximation too poor, leading to successively worse intermediate solutions.

IFAIL $= 5$

Continuation away from the initial point is not possible. This error exit will usually occur if the problem has not been properly posed or the error requirement is extremely stringent.

IFAIL $= 6$

The final problem (with $\theta_m = 0$) cannot be solved. It is likely that too much accuracy has been requested, or that the zero is at $\alpha = 0$ and IR $= 2$.

7 Accuracy

The accuracy of the approximation to the zero depends on TOL and IR. In general decreasing TOL will give more accurate results. Care must be exercised when using the relative error criterion (IR $= 2$).

If the zero is at X = 0, or if the initial value of X and the zero bracket the point X = 0, it is likely that an error exit with IFAIL = 4, 5 or 6 will occur.

It is possible to request too much or too little accuracy. Since it is not possible to achieve more than machine accuracy, a value of TOL $\ll$ ***machine precision*** should not be input and may lead to an error exit with IFAIL = 4, 5 or 6. For the reasons discussed under IFAIL = 4 in Section 6, TOL should not be taken too large, say no larger than TOL = 1.0D−3.

8 Further Comments

For most problems, the time taken on each call to C05AXF will be negligible compared with the time spent evaluating $f(x)$ between calls to C05AXF. However, the initial value of X and the choice of TOL will clearly affect the timing. The closer that X is to the root, the less evaluations of f required. The effect of the choice of TOL will not be large, in general, unless TOL is very small, in which case the timing will increase.

9 Example

This example calculates a zero of $x - e^{-x}$ with initial approximation $x_0 = 1.0$, and TOL = 1.0D−3 and 1.0D−4.

9.1 Program Text

```
*     C05AXF Example Program Text
*     Mark 14 Revised. NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      DOUBLE PRECISION FX, SCAL, TOL, X
      INTEGER          I, IFAIL, IND, IR
*     .. Local Arrays ..
      DOUBLE PRECISION C(26)
*     .. External Functions ..
      DOUBLE PRECISION X02AJF
      EXTERNAL         X02AJF
*     .. External Subroutines ..
      EXTERNAL         C05AXF
*     .. Intrinsic Functions ..
      INTRINSIC        EXP, SQRT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'C05AXF Example Program Results'
      SCAL = SQRT(X02AJF())
      IR = 0
      DO 40 I = 3, 4
         TOL = 10.0D0**(-I)
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'TOL = ', TOL
         WRITE (NOUT,*)
         X = 1.0D0
         IFAIL = 1
         IND = 1
*
   20    CALL C05AXF(X,FX,TOL,IR,SCAL,C,IND,IFAIL)
*
         IF (IND.NE.0) THEN
            FX = X - EXP(-X)
            GO TO 20
         ELSE
            IF (IFAIL.LT.0) THEN
               WRITE (NOUT,*)
               WRITE (NOUT,99998) ' ** C05AXF returned with IFAIL = ',
     +           IFAIL
               GO TO 60
            ELSE IF (IFAIL.GT.0) THEN
               WRITE (NOUT,99998) 'Error exit, IFAIL =', IFAIL
```

```
               IF (IFAIL.EQ.4 .OR. IFAIL.EQ.6) THEN
                  WRITE (NOUT,99997) 'Final value = ', X, ' THETA = ',
     +              C(5), ' LAMBDA = ', C(7)
               END IF
            ELSE
               WRITE (NOUT,99997) 'Root is ', X
            END IF
         END IF
   40 CONTINUE
   60 CONTINUE
*
99999 FORMAT (1X,A,E10.4)
99998 FORMAT (1X,A,I5)
99997 FORMAT (1X,A,F14.5,A,F10.2,A,F10.2)
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 C05AXF Example Program Results

 TOL = 0.1000E-02

 Root is         0.56715

 TOL = 0.1000E-03

 Root is         0.56715
```