E04ABF – 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

E04ABF searches for a minimum, in a given finite interval, of a continuous function of a single variable, using function values only. The method (based on quadratic interpolation) is intended for functions which have a continuous first derivative (although it will usually work if the derivative has occasional discontinuities).

2   Specification

```
SUBROUTINE E04ABF(FUNCT, E1, E2, A, B, MAXCAL, X, F, IFAIL)
INTEGER         MAXCAL, IFAIL
real            E1, E2, A, B, X, F
EXTERNAL        FUNCT
```

3   Description

E04ABF is applicable to problems of the form:

$$\text{Minimize}\, F(x) \ \text{subject to}\ \ a \le x \le b.$$

It normally computes a sequence of x values which tend in the limit to a minimum of $F(x)$ subject to the given bounds. It also progressively reduces the interval $[a, b]$ in which the minimum is known to lie. It uses the safeguarded quadratic-interpolation method described in Gill and Murray [1].

The user must supply a subroutine FUNCT to evaluate $F(x)$. The parameters E1 and E2 together specify the accuracy

$$Tol(x) = \text{E1} \times |x| + \text{E2}$$

to which the position of the minimum is required. Note that FUNCT is never called at any point which is closer than $Tol(x)$ to a previous point.

If the original interval $[a, b]$ contains more than one minimum, E04ABF will normally find one of the minima.

4   References

[1]   Gill P E and Murray W (1973) Safeguarded steplength algorithms for optimization using descent methods *NPL Report NAC 37* National Physical Laboratory

5   Parameters

1:   FUNCT — SUBROUTINE, supplied by the user.   *External Procedure*

This routine must be supplied by the user to calculate the value of the function $F(x)$ at any point x in $[a, b]$. It should be tested separately before being used in conjunction with E04ABF.

Its specification is:

```
      SUBROUTINE FUNCT(XC, FC)
      real            XC, FC
```

1:   XC — *real*   *Input*

On entry: the point x at which the value of F is required.

> 2:   FC — **real**   *Output*
>
> *On exit:* FC must be set to the value of the function F at the current point x.

FUNCT must be declared as EXTERNAL in the (sub)program from which E04ABF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2:   E1 — **real**   *Input/Output*

On entry: the relative accuracy to which the position of a minimum is required. (Note that, since E1 is a relative tolerance, the scaling of x is automatically taken into account.)

E1 should be no smaller than 2ϵ, and preferably not much less than $\sqrt{\epsilon}$, where ϵ is the **machine precision**.

On exit: if the user sets E1 to 0.0 (or to any value less than ϵ), E04ABF will reset E1 to the default value $\sqrt{\epsilon}$ before starting the minimization process.

3:   E2 — **real**   *Input/Output*

On entry: the absolute accuracy to which the position of a minimum is required. E2 should be no smaller than 2ϵ.

On exit: if the user sets E2 to 0.0 (or to any value less than ϵ), E04ABF will reset it to the default value $\sqrt{\epsilon}$.

4:   A — **real**   *Input/Output*

On entry: the lower bound a of the interval containing a minimum.

On exit: an improved lower bound on the position of the minimum.

5:   B — **real**   *Input/Output*

On entry: the upper bound b of the interval containing a minimum.

On exit: an improved upper bound on the position of the minimum.

6:   MAXCAL — INTEGER   *Input/Output*

On entry: the maximum number of evaluations of $F(x)$ which the user is prepared to allow.

Constraint: MAXCAL ≥ 3. (Few problems will require more than 30.)

There will be an error exit (see Section 6) after MAXCAL calls of FUNCT.

On exit: the total number of times that FUNCT was actually called.

7:   X — **real**   *Output*

On exit: the estimated position of the minimum.

8:   F — **real**   *Output*

On exit: the function value at the final point given in X.

9:   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 or gives a warning (see Section 6).

For this routine, because the values of output parameters may be useful even if IFAIL $\neq 0$ on exit, users are recommended to set IFAIL to -1 before entry. **It is then essential to test the value of IFAIL on exit**. To suppress the output of an error message when soft failure occurs, set IFAIL to 1.

6 Error Indicators and Warnings

Errors or warnings specified by the routine:

IFAIL = 1

On entry, $(A + E2) \geq B$,

or   MAXCAL < 3,

IFAIL = 2

The number of calls of FUNCT has exceeded MAXCAL. This may have happened simply because MAXCAL was set too small for a particular problem, or may be due to a mistake in the user's routine FUNCT. If no mistake can be found in FUNCT, restart E04ABF (preferably with values of A and B given on exit from the previous call of E04ABF).

7 Accuracy

If $F(x)$ is δ-unimodal (see the Chapter Introduction) for some $\delta < Tol(x)$, where $Tol(x) = E1 \times |x| + E2$, then, on exit, x approximates the minimum of $F(x)$ in the original interval $[a, b]$ with an error less than $3 \times Tol(x)$.

8 Further Comments

Timing depends on the behaviour of $F(x)$, the accuracy demanded and the length of the interval $[a, b]$. Unless $F(x)$ can be evaluated very quickly, the run time will usually be dominated by the time spent in FUNCT.

If $F(x)$ has more than one minimum in the original interval $[a, b]$, E04ABF will determine an approximation x (and improved bounds a and b) for one of the minima.

If E04ABF finds an x such that $F(x - \delta_1) > F(x) < F(x + \delta_2)$ for some $\delta_1, \delta_2 \geq Tol(x)$, the interval $[x - \delta_1, x + \delta_2]$ will be regarded as containing a minimum, even if $F(x)$ is less than $F(x - \delta_1)$ and $F(x + \delta_2)$ only due to rounding errors in the user-supplied routine. Therefore FUNCT should be programmed to calculate $F(x)$ as accurately as possible, so that E04ABF will not be liable to find a spurious minimum.

9 Example

A sketch of the function

$$F(x) = \frac{\sin x}{x}$$

shows that it has a minimum somewhere in the range $[3.5, 5.0]$. The following program shows how E04ABF can be used to obtain a good approximation to the position of a minimum.

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.

```
*     E04ABF Example Program Text.
*     Mark 17 Revised.  NAG Copyright 1995.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             A, B, EPS, F, T, X
      INTEGER          IFAIL, MAXCAL
*     .. External Subroutines ..
      EXTERNAL         E04ABF, FUNCT
```

```
*      .. Executable Statements ..
       WRITE (NOUT,*) 'E04ABF Example Program Results'
*      EPS and T are set to zero so that E04ABF will reset them to
*      their default values
       EPS = 0.0e0
       T = 0.0e0
*      The minimum is known to lie in the range (3.5, 5.0)
       A = 3.5e0
       B = 5.0e0
*      Allow 30 calls of FUNCT
       MAXCAL = 30
       IFAIL = 1
*
       CALL E04ABF(FUNCT,EPS,T,A,B,MAXCAL,X,F,IFAIL)
*
       WRITE (NOUT,*)
       IF (IFAIL.EQ.1) THEN
          WRITE (NOUT,*) 'Parameter outside expected range'
       ELSE
          IF (IFAIL.EQ.2) THEN
             WRITE (NOUT,*)
     +         'Results after MAXCAL function evaluations are'
             WRITE (NOUT,*)
          END IF
          WRITE (NOUT,99999) 'The minimum lies in the interval ', A,
     +      ' to ', B
          WRITE (NOUT,99999) 'Its estimated position is ', X, ','
          WRITE (NOUT,99998) 'where the function value is ', F
          WRITE (NOUT,99997) MAXCAL, 'function evaluations were required'
       END IF
       STOP
*
99999 FORMAT (1X,A,F10.8,A,F10.8)
99998 FORMAT (1X,A,F7.4)
99997 FORMAT (1X,I2,1X,A)
       END
*
       SUBROUTINE FUNCT(XC,FC)
*      Routine to evaluate F(x) at any point in (A, B)
*      .. Scalar Arguments ..
       real            FC, XC
*      .. Intrinsic Functions ..
       INTRINSIC       SIN
*      .. Executable Statements ..
       FC = SIN(XC)/XC
       RETURN
       END
```

9.2   Program Data

None.

9.3 Program Results

```
E04ABF Example Program Results

The minimum lies in the interval 4.49340940 to 4.49340951
Its estimated position is 4.49340945,
where the function value is -0.2172
10 function evaluations were required
```