NAG Library Routine Document

E04ABF/E04ABA

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/E04ABA 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).

E04ABA is a version of E04ABF that has additional parameters in order to make it safe for use in multithreaded applications (see Section 5).

2 Specification

2.1 Specification for E04ABF

```
SUBROUTINE E04ABF (FUNCT, E1, E2, A, B, MAXCAL, X, F, IFAIL)
INTEGER            MAXCAL, IFAIL
REAL (KIND=nag_wp) E1, E2, A, B, X, F
EXTERNAL           FUNCT
```

2.2 Specification for E04ABA

```
SUBROUTINE E04ABA (FUNCT, E1, E2, A, B, MAXCAL, X, F, IUSER, RUSER,        &
                   IFAIL)
INTEGER            MAXCAL, IUSER(*), IFAIL
REAL (KIND=nag_wp) E1, E2, A, B, X, F, RUSER(*)
EXTERNAL           FUNCT
```

3 Description

E04ABF/E04ABA is applicable to problems of the form:

$$\text{Minimize } F(x) \quad \text{subject to} \quad a \leq x \leq b.$$

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 (1973).

You must supply a 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/E04ABA will normally find one of the minima.

4 References

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*

You must supply this routine 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/E04ABA.

The specification of FUNCT for E04ABF is:

```
SUBROUTINE FUNCT (XC, FC)
REAL (KIND=nag_wp) XC, FC
```

The specification of FUNCT for E04ABA is:

```
SUBROUTINE FUNCT (XC, FC, IUSER, RUSER)
INTEGER             IUSER(*)
REAL (KIND=nag_wp) XC, FC, RUSER(*)
```

1: XC – REAL (KIND=nag_wp) *Input*

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

2: FC – REAL (KIND=nag_wp) *Output*

On exit: must be set to the value of the function F at the current point x.

Note: *the following are additional parameters for specific use with E04ABA. Users of E04ABF therefore need not read the remainder of this description.*

3: IUSER$(*)$ – INTEGER array *User Workspace*

4: RUSER$(*)$ – REAL (KIND=nag_wp) array *User Workspace*

FUNCT is called with the parameters IUSER and RUSER as supplied to E04ABF/E04ABA. You are free to use the arrays IUSER and RUSER to supply information to FUNCT as an alternative to using COMMON global variables.

FUNCT must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which E04ABF/E04ABA is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: E1 – REAL (KIND=nag_wp) *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 you set E1 to 0.0 (or to any value less than ϵ), E1 will be reset to the default value $\sqrt{\epsilon}$ before starting the minimization process.

3: E2 – REAL (KIND=nag_wp) *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 you set E2 to 0.0 (or to any value less than ϵ), E2 will be reset to the default value $\sqrt{\epsilon}$.

4: A – REAL (KIND=nag_wp) *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 (KIND=nag_wp) *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 calls of $F(x)$ to be allowed.

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 (KIND=nag_wp) *Output*

On exit: the estimated position of the minimum.

8: F – REAL (KIND=nag_wp) *Output*

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

9: IFAIL – INTEGER *Input/Output*

Note: *for E04ABA, IFAIL does not occur in this position in the parameter list. See the additional parameters described below.*

On 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.

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.**

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

Note: *the following are additional parameters for specific use with E04ABA. Users of E04ABF therefore need not read the remainder of this description*.

10: IUSER$(*)$ – INTEGER array *User Workspace*

11: RUSER$(*)$ – REAL (KIND=nag_wp) array *User Workspace*

IUSER and RUSER are not used by E04ABF/E04ABA, but are passed directly to FUNCT and may be used to pass information to this routine as an alternative to using COMMON global variables.

12: IFAIL – INTEGER *Input/Output*

Note: see the parameter description for IFAIL above.

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).

Note: E04ABF/E04ABA may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, $(\text{A} + \text{E2}) \geq \text{B}$,
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 FUNCT. If no mistake can be found in FUNCT, restart E04ABF/E04ABA (preferably with the values of A and B given on exit from the previous call of E04ABF/E04ABA).

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

If $F(x)$ is δ-unimodal for some $\delta < Tol(x)$, where $Tol(x) = \text{E1} \times |x| + \text{E2}$, 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 Parallelism and Performance

Not applicable.

9 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/E04ABA will determine an approximation x (and improved bounds a and b) for one of the minima.

If E04ABF/E04ABA 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 subroutine. Therefore FUNCT should be programmed to calculate $F(x)$ as accurately as possible, so that E04ABF/E04ABA will not be liable to find a spurious minimum.

10 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/E04ABA can be used to obtain a good approximation to the position of a minimum.

10.1 Program Text

the following program illustrates the use of E04ABF. An equivalent program illustrating the use of E04ABA is available with the supplied Library and is also available from the NAG web site.

```
!   E04ABF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.
    Module e04abfe_mod

!     E04ABF 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                                :: funct
!     .. Parameters ..
      Integer, Parameter, Public            :: nout = 6
    Contains
      Subroutine funct(xc,fc)
!       Routine to evaluate F(x) at any point in (A, B)

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (Out)     :: fc
        Real (Kind=nag_wp), Intent (In)      :: xc
!       .. Intrinsic Procedures ..
        Intrinsic                            :: sin
!       .. Executable Statements ..
        fc = sin(xc)/xc

        Return

      End Subroutine funct
    End Module e04abfe_mod
    Program e04abfe

!     E04ABF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: e04abf, nag_wp
      Use e04abfe_mod, Only: funct, nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                     :: a, b, e1, e2, f, x
      Integer                                :: ifail, maxcal
!     .. Executable Statements ..
      Write (nout,*) 'E04ABF Example Program Results'

!     E1 and E2 are set to zero so that E04ABF will reset them to
!     their default values

      e1 = 0.0_nag_wp
      e2 = 0.0_nag_wp

!     The minimum is known to lie in the range (3.5, 5.0)
```

```
      a = 3.5_nag_wp
      b = 5.0_nag_wp

!     Allow 30 calls of FUNCT

      maxcal = 30

      ifail = -1
      Call e04abf(funct,e1,e2,a,b,maxcal,x,f,ifail)

      Select Case (ifail)
      Case (0,2)
        Write (nout,*)
        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 Select

99999 Format (1X,A,F11.8,A,F11.8)
99998 Format (1X,A,F7.4)
99997 Format (1X,I2,1X,A)
    End Program e04abfe
```

10.2 Program Data

None.

10.3 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
```