NAG Fortran Library Routine Document

E04UHF/E04UHA

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

To supply optional parameters to E04UGF/E04UGA from an external file. More precisely, E04UHF must be used to supply optional parameters to E04UGF and E04UHA must be used to supply optional parameters to E04UGA.

E04UHA is a version of E04UHF that has additional parameters in order to make it safe for use in multithreaded applications (see Section 5 below). The initialisation routine E04WBF **must** have been called prior to calling E04UHA.

2 Specifications

2.1 Specification for E04UHF

```
SUBROUTINE E04UHF(IOPTNS, INFORM)
INTEGER           IOPTNS, INFORM
```

2.2 Specification for E04UHA

```
SUBROUTINE E04UHA(IOPTNS, LWSAV, IWSAV, RWSAV, INFORM)
INTEGER           IOPTNS, IWSAV(550), INFORM
real              RWSAV(550)
LOGICAL           LWSAV(20)
```

3 Description

E04UHF/E04UHA may be used to supply values for optional parameters to the corresponding routines E04UGF/E04UGA. E04UHF/E04UHA reads an external file and each line of the file defines a single optional parameter. It is only necessary to supply values for those parameters whose values are to be different from their default values.

Each optional parameter is defined by a single character string, of up to 72 characters, consisting of one or more items. The items associated with a given option must be separated by spaces, or equals signs [=]. Alphabetic characters may be upper or lower case. The string

```
Print level = 1
```

is an example of a string used to set an optional parameter. For each option the string contains one or more of the following items:

(a) A mandatory keyword.

(b) A phrase that qualifies the keyword.

(c) A number that specifies an INTEGER or ***real*** value. Such numbers may be up to 16 contiguous characters in Fortran's I, F, E or D formats, terminated by a space if this is not the last item on the line.

Blank strings and comments are ignored. A comment begins with an asterisk (*) and all subsequent characters in the string are regarded as part of the comment.

The file containing the options must start with **begin** and must finish with **end**. An example of a valid options file is:

```
Begin * Example options file
   Print level = 5
End
```

For E04UHF each line of the file is normally printed as it is read, on the current advisory message unit (see X04ABF), but printing may be suppressed using the keyword **nolist**. To suppress printing of **begin**, **nolist** must be the first option supplied as in the file:

```
Begin
   Nolist
   Print level = 5
End
```

Printing will automatically be turned on again after a call to E04UGF or E04UHF and may be turned on again at any time using the keyword **list**.

For E04UHA printing is turned off by default, but may be turned on at any time using the keyword **list**.

Optional parameter settings are preserved following a call to E04UGF/E04UGA and so the keyword **defaults** is provided to allow you to reset all the optional parameters to their default values prior to a subsequent call to E04UGF/E04UGA.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 11 of the document for E04UGF/E04UGA.

4 References

Hock W and Schittkowski K (1981) *Test Examples for Nonlinear Programming Codes. Lecture Notes in Economics and Mathematical Systems* **187** Springer-Verlag

5 Parameters

1: IOPTNS – INTEGER *Input*

On entry: the unit number of the options file to be read.

Constraint: $0 \leq \text{IOPTNS} \leq 99$.

2: INFORM – INTEGER *Output*

Note: *for E04UHA, INFORM does not occur in this position in the parameter list. See the additional parameters described below.*

On exit: contains zero if the options file has been successfully read and a value > 0 otherwise (see Section 6).

Note: *the following are additional parameters for specific use with E04UHA. Users of E04UHF therefore need not read the remainder of this section.*

2: LWSAV(20) – LOGICAL array *Workspace*

3: IWSAV(550) – INTEGER array *Workspace*

4: RWSAV(550) – ***real*** array *Workspace*

The arrays LWSAV, IWSAV and RWSAV **must not** be altered between calls to any of the routines E04WBF, E04UGA, E04UHA or E04UJA.

5: INFORM – INTEGER *Output*

On exit: contains zero if the options file has been successfully read and a value > 0 otherwise (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the routine:

INFORM = 1

IOPTNS is not in the range $[0, 99]$.

INFORM = 2

begin was found, but end-of-file was found before **end** was found.

INFORM = 3

end-of-file was found before **begin** was found.

INFORM = 4

Not used.

INFORM = 5

One or more lines of the options file is invalid. Check that all keywords are neither ambiguous nor misspelt.

7 Accuracy

Not applicable.

8 Further Comments

E04UJF/E04UJA may also be used to supply optional parameters to the corresponding routines E04UGF/E04UGA.

9 Example

This is Problem 45 from Hock and Schittkowski (1981) and involves the minimization of the nonlinear function

$$f(x) = 2 - \tfrac{1}{120} \times x_1 x_2 x_3 x_4 x_5$$

subject to the bounds

$$\begin{aligned} 0 \leq x_1 \leq 1, \\ 0 \leq x_2 \leq 2, \\ 0 \leq x_3 \leq 3, \\ 0 \leq x_4 \leq 4, \\ 0 \leq x_5 \leq 5. \end{aligned}$$

The initial point, which is infeasible, is

$$x_0 = (2,\ 2,\ 2,\ 2,\ 2)^T,$$

and $f(x_0) = 1.7333$ (to five figures).

The optimal solution is

$$x^* = (1,\ 2,\ 3,\ 4,\ 5)^T,$$

and $f(x^*) = 1$. All the bounds are active at the solution.

In this example the options file read by E04UHF/E04UHA is appended to the data file for the program (see Section 9.2). It would usually be more convenient in practice to keep the data file and the options file separate.

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.

Note: *the following program illustrates the use of E04UHF. An equivalent program illustrating the use of E04UHA is available with the supplied Library and is also available from the NAG web site.*

```
*     E04UHF Example Program Text.
*     Mark 20 Revised. NAG Copyright 2001.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          MMAX, NMAX, LENIZ, LENZ
      PARAMETER        (MMAX=100,NMAX=100,LENIZ=1000,LENZ=1000)
*     .. Local Scalars ..
      real             OBJ, SINF
      INTEGER          I, IFAIL, INFORM, IOBJ, J, M, MINIZ, MINZ, N,
     +                 NCNLN, NINF, NJNLN, NNAME, NNZ, NONLN, NS
      CHARACTER        START
*     .. Local Arrays ..
      real             A(1), BL(NMAX+MMAX), BU(NMAX+MMAX),
     +                 CLAMDA(NMAX+MMAX), USER(1), XS(NMAX+MMAX),
     +                 Z(LENZ)
      INTEGER          HA(1), ISTATE(NMAX+MMAX), IUSER(1), IZ(LENIZ),
     +                 KA(NMAX+1)
      CHARACTER*8      NAMES(NMAX+MMAX)
*     .. External Subroutines ..
      EXTERNAL         E04UGF, E04UGM, E04UHF, E04UJF, OBJFUN, X04ABF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E04UHF Example Program Results'
*     Skip heading in data file.
      READ (NIN,*)
      READ (NIN,*) N, M
      IF (N.LE.NMAX .AND. M.LE.MMAX) THEN
*
*        Read NCNLN, NONLN and NJNLN from data file.
*
         READ (NIN,*) NCNLN, NONLN, NJNLN
*
*        Read START, NNAME and NAMES from data file.
*
         READ (NIN,*) START, NNAME
         IF (NNAME.EQ.N+M) READ (NIN,*) (NAMES(I),I=1,N+M)
*
*        Define the matrix A to contain a dummy 'free' row that consists
*        of a single (zero) element subject to 'infinite' upper and lowe
*        bounds. Set up KA.
*
         NNZ = 1
         IOBJ = -1
*
         KA(1) = 1
*
         A(1) = 0.0e+0
         HA(1) = 1
*
*        Columns 2,3,...,N of A are empty. Set the corresponding element
*        of KA to 2.
*
         DO 20 J = 2, N
            KA(J) = 2
   20    CONTINUE
*
         KA(N+1) = NNZ + 1
*
*        Read BL, BU, ISTATE and XS from data file.
*
         READ (NIN,*) (BL(I),I=1,N+M)
         READ (NIN,*) (BU(I),I=1,N+M)
```

```
         IF (START.EQ.'C') THEN
            READ (NIN,*) (ISTATE(I),I=1,N)
         ELSE IF (START.EQ.'W') THEN
            READ (NIN,*) (ISTATE(I),I=1,N+M)
         END IF
         READ (NIN,*) (XS(I),I=1,N)
*
*        Set three options using E04UJF.
*
         CALL E04UJF(' Verify Level = -1 ')
*
         CALL E04UJF(' Major Iteration Limit = 25 ')
*
         CALL E04UJF(' Infinite Bound Size = 1.0D+25 ')
*
*        Set the unit number for advisory messages to NOUT.
*
         CALL X04ABF(1,NOUT)
*
*        Read the options file for the remaining options.
*
         CALL E04UHF(NIN,INFORM)
*
         IF (INFORM.NE.0) THEN
            WRITE (NOUT,99999) 'E04UHF terminated with INFORM = ',
     +        INFORM
            STOP
         END IF
*
*        Solve the problem.
*
         IFAIL = -1
*
         CALL E04UGF(E04UGM,OBJFUN,N,M,NCNLN,NONLN,NJNLN,IOBJ,NNZ,A,HA,
     +               KA,BL,BU,START,NNAME,NAMES,NS,XS,ISTATE,CLAMDA,
     +               MINIZ,MINZ,NINF,SINF,OBJ,IZ,LENIZ,Z,LENZ,IUSER,
     +               USER,IFAIL)
*
      END IF
*
      STOP
*
99999 FORMAT (1X,A,I3)
      END
*
      SUBROUTINE OBJFUN(MODE,NONLN,X,OBJF,OBJGRD,NSTATE,IUSER,USER)
*     Computes the nonlinear part of the objective function and its
*     gradient
*     .. Scalar Arguments ..
      real               OBJF
      INTEGER            MODE, NONLN, NSTATE
*     .. Array Arguments ..
      real               OBJGRD(NONLN), USER(*), X(NONLN)
      INTEGER            IUSER(*)
*     .. Executable Statements ..
*
      IF (MODE.EQ.0 .OR. MODE.EQ.2) OBJF = 2.0e+0 - X(1)*X(2)*X(3)*X(4)
     +    *X(5)/120.0e+0
*
      IF (MODE.EQ.1 .OR. MODE.EQ.2) THEN
         OBJGRD(1) = -X(2)*X(3)*X(4)*X(5)/120.0e+0
         OBJGRD(2) = -X(1)*X(3)*X(4)*X(5)/120.0e+0
         OBJGRD(3) = -X(1)*X(2)*X(4)*X(5)/120.0e+0
         OBJGRD(4) = -X(1)*X(2)*X(3)*X(5)/120.0e+0
         OBJGRD(5) = -X(1)*X(2)*X(3)*X(4)/120.0e+0
      END IF
*
      END
```

9.2 Program Data

```
E04UHF Example Program Data
  5   1             :Values of N and M
  0   5   0         :Values of NCNLN, NONLN and NJNLN
 'C'  6             :Values of START and NNAME
'Varble 1' 'Varble 2' 'Varble 3' 'Varble 4' 'Varble 5' 'DummyRow' :End of NAMES
 0.0  0.0  0.0  0.0  0.0 -1.0E+25 :End of BL
 1.0  2.0  3.0  4.0  5.0  1.0E+25 :End of BU
 0    0    0    0    0           :End of ISTATE
 2.0  2.0  2.0  2.0  2.0         :End of XS
Begin * Example options file for E04UHF
   Check Frequency = 25   * (Default = 60 )
   Crash Tolerance = 0.05 * (Default = 0.1)
End
```

9.3 Program Results

```
 E04UHF Example Program Results


 Calls to E04UJF
 ---------------

       Verify Level = -1
       Major Iteration Limit = 25
       Infinite Bound Size = 1.0E+25


 OPTIONS file
 ------------

      Begin * Example options file for E04UHF
         Check Frequency = 25   * (Default = 60 )
         Crash Tolerance = 0.05 * (Default = 0.1)
      End

 *** E04UGF
 *** Start of NAG Library implementation details ***

 Implementation title: Generalised Base Version
            Precision: FORTRAN double precision
         Product Code: FLBAS20D
                 Mark: 20A

 *** End of NAG Library implementation details ***

 Parameters
 ----------

 Frequencies.
 Check frequency.........        25       Expand frequency.......     10000
 Factorization frequency.       100

 QP subproblems.
 Scale tolerance.........  9.00E-01       Minor feasibility tol..  1.05E-08
 Scale option............         2       Minor optimality tol...  1.05E-08
 Partial price...........        10       Crash tolerance........  5.00E-02
 Pivot tolerance.........  2.05E-11       Minor print level......         0
 Crash option............         3       Elastic weight.........  1.00E+00

 The SQP method.
 Minimize................
 Nonlinear objective vars         5       Major optimality tol...  1.05E-08
 Function precision......  1.72E-13       Unbounded step size....  1.00E+20
 Superbasics limit.......         5       Forward difference int.  4.15E-07
 Unbounded objective.....  1.00E+15       Central difference int.  5.57E-05
 Major step limit........  2.00E+00       Derivative linesearch..
 Derivative level........         3       Major iteration limit..        25
 Linesearch tolerance....  9.00E-01       Verify level...........        -1
```

```
Minor iteration limit...       500       Major print level......        10
Infinite bound size.....  1.00E+25       Iteration limit........     10000

Hessian approximation.
Hessian full memory.....                 Hessian updates........  99999999
Hessian frequency.......  99999999

Nonlinear constraints.
Nonlinear constraints...         0       Nonlinear Jacobian vars         0

Miscellaneous.
Variables...............         5       Linear constraints.....         1
Nonlinear variables.....         5       Linear variables.......         0
LU factor tolerance.....  1.00E+02       LU singularity tol.....  2.05E-11
LU update tolerance.....  1.00E+01       LU density tolerance...  6.00E-01
eps (machine precision).  1.11E-16       Monitoring file........        -1
COLD start..............                 Infeasible exit........

Workspace provided is                 IZ(    1000),  Z(    1000).
To start solving the problem we need  IZ(     566),  Z(     670).

Itn      0 -- Partial price reduced from              10 to                1.

Itn      0 -- Feasible linear rows.

Itn      0 -- Norm(x-x0) minimized. Sum of infeasibilities = 0.00E+00.

objfun  sets       5   out of       5   objective  gradients.


 Maj  Mnr    Step      Objective Optimal Cond Hz PD
   0    3 0.0E+00   1.866667E+00 3.3E-02 1.0E+00 TF  R
   1    2 1.5E+01   1.550000E+00 7.5E-02 1.0E+00 TF n
   2    2 6.7E+00   1.200000E+00 1.0E-01 1.0E+00 TF n
   3    1 5.0E+00   1.000000E+00 0.0E+00 1.0E+00 TT n

Exit from NP problem after     3 major iterations,
                               8 minor iterations.

Variable State     Value       Lower Bound  Upper Bound   Lagr Mult  Residual

Varble 1    UL    1.00000           .         1.0000      -1.000         .
Varble 2    UL    2.00000           .         2.0000     -0.5000         .
Varble 3    UL    3.00000           .         3.0000     -0.3333         .
Varble 4    UL    4.00000           .         4.0000     -0.2500         .
Varble 5    UL    5.00000           .         5.0000     -0.2000         .

Constrnt State     Value       Lower Bound  Upper Bound   Lagr Mult  Residual

DummyRow    BS    0.00000          None          None     -1.000         .

Exit E04UGF - Optimal solution found.

Final objective value =     1.000000
```