NAG 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 arguments in order to make it safe for use in multithreaded applications (see Section 5). The initialization routine E04WBF **must** have been called before calling E04UHA.

2 Specification

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 (KIND=nag_wp) RWSAV(550)
LOGICAL            LWSAV(20)
```

3 Description

E04UHF/E04UHA may be used to supply values for optional parameters to 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 arguments 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 mandatory keyword;

– a phrase that qualifies the keyword;

– a number that specifies an integer or real value. Such numbers may be up to 40 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 before a subsequent call to E04UGF/E04UGA.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 12 in 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 Arguments

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 argument list. See the additional arguments 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 arguments for specific use with E04UHA. Users of E04UHF therefore need not read the remainder of this description.*

3: LWSAV(20) – LOGICAL array *Communication Array*

4: IWSAV(550) – INTEGER array *Communication Array*

5: RWSAV(550) – REAL (KIND=nag_wp) array *Communication Array*

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

6: INFORM – INTEGER *Output*

Note: see the argument description for INFORM above.

6 Error Indicators and Warnings

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 Parallelism and Performance

E04UHF/E04UHA is not threaded in any implementation.

9 Further Comments

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

10 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

$$x_0 = (2, 2, 2, 2, 2)^{\mathrm{T}},$$

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

The optimal solution is

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

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 10.2). It would usually be more convenient in practice to keep the data file and the options file separate.

10.1 Program Text

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 26 Release. NAG Copyright 2016.

    Module e04uhfe_mod

!     E04UHF 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                             :: objfun
!     .. Parameters ..
      Integer, Parameter, Public         :: iset = 1, nin = 5, ninopt = 7,       &
                                            nout = 6
    Contains
      Subroutine objfun(mode,nonln,x,objf,objgrd,nstate,iuser,ruser)
!       Computes the nonlinear part of the objective function and its
!       gradient

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (Out) :: objf
        Integer, Intent (Inout)          :: mode
        Integer, Intent (In)             :: nonln, nstate
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Inout) :: objgrd(nonln), ruser(*)
        Real (Kind=nag_wp), Intent (In) :: x(nonln)
        Integer, Intent (Inout)          :: iuser(*)
!       .. Executable Statements ..
        If (mode==0 .Or. mode==2) Then
          objf = 2.0E+0_nag_wp - x(1)*x(2)*x(3)*x(4)*x(5)/120.0E+0_nag_wp
        End If

        If (mode==1 .Or. mode==2) Then
          objgrd(1) = -x(2)*x(3)*x(4)*x(5)/120.0E+0_nag_wp
          objgrd(2) = -x(1)*x(3)*x(4)*x(5)/120.0E+0_nag_wp
          objgrd(3) = -x(1)*x(2)*x(4)*x(5)/120.0E+0_nag_wp
          objgrd(4) = -x(1)*x(2)*x(3)*x(5)/120.0E+0_nag_wp
          objgrd(5) = -x(1)*x(2)*x(3)*x(4)/120.0E+0_nag_wp
        End If

        Return

      End Subroutine objfun
    End Module e04uhfe_mod
    Program e04uhfe

!     E04UHF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: e04ugf, e04ugm, e04uhf, e04ujf, nag_wp, x04abf,   &
                             x04acf, x04baf
      Use e04uhfe_mod, Only: iset, nin, ninopt, nout, objfun
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Character (*), Parameter           :: fname = 'e04uhfe.opt'
!     .. Local Scalars ..
      Real (Kind=nag_wp)                 :: obj, sinf
      Integer                            :: ifail, inform, iobj, leniz, lenz, m, &
                                            miniz, minz, mode, n, ncnln, ninf,   &
                                            njnln, nname, nnz, nonln, ns, outchn
      Character (80)                     :: rec
```

```
      Character (1)                    :: start
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: a(:), bl(:), bu(:), clamda(:),       &
                                          xs(:), z(:)
      Real (Kind=nag_wp)               :: user(1)
      Integer, Allocatable             :: ha(:), istate(:), iz(:), ka(:)
      Integer                          :: iuser(1)
      Character (8), Allocatable       :: names(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max
!     .. Executable Statements ..
      Write (rec,99998) 'E04UHF Example Program Results'
      Call x04baf(nout,rec)

!     Skip heading in data file.
      Read (nin,*)
      Read (nin,*) n, m
      Read (nin,*) ncnln, nonln, njnln
      Read (nin,*) start, nname
      nnz = 1
      Allocate (ha(nnz),ka(n+1),istate(n+m),a(nnz),bl(n+m),bu(n+m),xs(n+m),    &
        clamda(n+m),names(nname))

      Read (nin,*) names(1:nname)

!     Define the matrix A to contain a dummy 'free' row that consists
!     of a single (zero) element subject to 'infinite' upper and
!     lower bounds. Set up KA.

      iobj = -1

      ka(1) = 1

      a(1) = 0.0E+0_nag_wp
      ha(1) = 1

!     Columns 2,3,...,N of A are empty. Set the corresponding element
!     of KA to 2.

      ka(2:n) = 2
      ka(n+1) = nnz + 1

      Read (nin,*) bl(1:(n+m))
      Read (nin,*) bu(1:(n+m))

      If (start=='C') Then
        Read (nin,*) istate(1:n)
      Else If (start=='W') Then
        Read (nin,*) istate(1:(n+m))
      End If

      Read (nin,*) xs(1:n)

!     Set the unit number for advisory messages to OUTCHN.

      outchn = nout
      Call x04abf(iset,outchn)

!     Set three options using E04UJF.

      Call e04ujf(' Verify Level = -1 ')

      Call e04ujf(' Major Iteration Limit = 25 ')

      Call e04ujf(' Infinite Bound Size = 1.0D+25 ')

!     Open the options file for reading

      mode = 0

      ifail = 0
```

```
      Call x04acf(ninopt,fname,mode,ifail)

!     Read the options file for the remaining options.

      Call e04uhf(ninopt,inform)

      If (inform/=0) Then
        Write (rec,99999) 'E04UJF terminated with INFORM = ', inform
        Call x04baf(nout,rec)
        Go To 100
      End If

!     Solve the problem.
!     First call is a workspace query

      leniz = max(500,n+m)
      lenz = 500
      Allocate (iz(leniz),z(lenz))

      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)

      If (ifail/=0 .And. ifail/=15 .And. ifail/=16) Then
        Write (nout,99999) 'Query call to E04UGF failed with IFAIL =', ifail
        Go To 100
      End If

      Deallocate (iz,z)

!     The length of the workspace required for the basis factors in this
!     problem is longer than the minimum returned by the query

      lenz = 2*minz
      leniz = 2*miniz
      Allocate (iz(leniz),z(lenz))

      ifail = 0
      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)
100   Continue

99999 Format (1X,A,I5)
99998 Format (1X,A)
    End Program e04uhfe
```

10.2 Program Data

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

```
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+26 :End of BL
 1.0  2.0  3.0  4.0  5.0  1.0E+26 :End of BU
 0    0    0    0    0            :End of ISTATE
 2.0  2.0  2.0  2.0  2.0          :End of XS
```

10.3 Program Results

```
 E04UHF Example Program Results


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

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


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

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

 Workspace provided is                   IZ(     500),  Z(     500).
 To start solving the problem we need    IZ(     566),  Z(     670).

 Exit E04UGF - Not enough integer workspace to start solving the problem.

 *** E04UGF

 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.........  2.04E-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.  5.56E-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.....  2.04E-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(    1132),  Z(    1340).
 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
```