NAG Fortran Library Routine Document

E04WEF

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 E04WDF from an external file. The initialization routine E04WCF **must** have been called prior to calling E04WEF.

2 Specification

```
SUBROUTINE E04WEF (ISPECS, IW, RW, IFAIL)
INTEGER            ISPECS, IW(*), IFAIL
double precision   RW(*)
```

3 Description

E04WEF may be used to supply values for optional parameters to E04WDF. E04WEF 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 ***double precision*** 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
```

Optional parameter settings are preserved following a call to E04WDF 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 E04WDF.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 11 of the document for E04WDF.

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: ISPECS – INTEGER *Input*

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

Constraint: ISPECS is a valid unit open for reading.

2: IW(∗) – INTEGER array *Communication Array*

3: RW(∗) – ***double precision*** array *Communication Array*

The arrays IW and RW are defined in the document for E04WCF and **must not** be altered between calls to any of the routines E04WCF, E04WDF, E04WEF, E04WFF, E04WGF, E04WHF, E04WKF and E04WLF.

4: 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 (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, for users not familiar with this parameter the recommended value is 0. **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

The initialization routine E04WCF has not been called.

IFAIL = 2

Could not read options file on unit ISPECS. This may be due to:

(a) ISPECS is not a valid unit number;

(b) a file is not associated with unit ISPECS, or if it is, is unavailable for read access;

(c) one or more lines of the options file is invalid. Check that all keywords are neither ambiguous nor misspelt;

(d) **begin** was found, but end-of-file was found before **end** was found;

(e) end-of-file was found before **begin** was found.

7 Accuracy

Not applicable.

8 Further Comments

E04WFF, E04WGF or E04WHF may also be used to supply optional parameters to E04WDF.

9 Example

This is based on Problem 71 in Hock and Schittkowski (1981) and involves the minimization of the nonlinear function

$$F(x) = x_1x_4(x_1 + x_2 + x_3) + x_3$$

subject to the bounds

$$\begin{gathered} 1 \leq x_1 \leq 5 \\ 1 \leq x_2 \leq 5 \\ 1 \leq x_3 \leq 5 \\ 1 \leq x_4 \leq 5 \end{gathered}$$

to the general linear constraint

$$x_1 + x_2 + x_3 + x_4 \leq 20,$$

and to the nonlinear constraints

$$\begin{gathered} x_1^2 + x_2^2 + x_3^2 + x_4^2 \leq 40, \\ x_1x_2x_3x_4 \geq 25. \end{gathered}$$

The initial point, which is infeasible, is

$$x_0 = (1, 5, 5, 1)^T,$$

and $F(x_0) = 16$.

The optimal solution (to five figures) is

$$x^* = (1.0, 4.7430, 3.8211, 1.3794)^T,$$

and $F(x^*) = 17.014$. One bound constraint and both nonlinear constraints are active at the solution.

The document for E04WEF includes an example program to solve the same problem using some of the optional parameters described in Section 11 of the document for E04WDF.

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.

```
*     E04WEF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
      IMPLICIT         NONE
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, NCLMAX, NCNMAX
      PARAMETER        (NMAX=10,NCLMAX=10,NCNMAX=10)
      INTEGER          LDA, LDCJ, LDH
      PARAMETER        (LDA=NCLMAX,LDCJ=NCNMAX,LDH=NMAX)
      INTEGER          LENIW, LENRW
      PARAMETER        (LENIW=600,LENRW=600)
*     .. Local Scalars ..
      DOUBLE PRECISION BNDINF, FEATOL, OBJF
      INTEGER          ELMODE, I, IFAIL, J, MAJITS, N, NCLIN, NCNLN
*     .. Local Arrays ..
      DOUBLE PRECISION A(LDA,NMAX), BL(NMAX+NCLMAX+NCNMAX),
     +                 BU(NMAX+NCLMAX+NCNMAX), CCON(NCNMAX),
     +                 CJAC(LDCJ,NMAX), CLAMDA(NMAX+NCLMAX+NCNMAX),
     +                 GRAD(NMAX), HESS(LDH,NMAX), RUSER(1), RW(LENRW),
     +                 X(NMAX)
```

```
      INTEGER          ISTATE(NMAX+NCLMAX+NCNMAX), IUSER(1), IW(LENIW)
*     .. External Subroutines ..
      EXTERNAL         CONFUN, E04WCF, E04WDF, E04WEF, E04WFF, E04WGF,
     +                 E04WHF, E04WKF, E04WLF, OBJFUN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E04WEF Example Program Results'
*
*     This program demonstrates the use of routines to set and
*     get values of optional parameters associated with E04WDF.
*
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NCLIN, NCNLN
      IF (N.LE.NMAX .AND. NCLIN.LE.NCLMAX .AND. NCNLN.LE.NCNMAX) THEN
*
*        Read A, BL, BU and X from data file
         IF (NCLIN.GT.0) READ (NIN,*) ((A(I,J),J=1,N),I=1,NCLIN)
         READ (NIN,*) (BL(I),I=1,N+NCLIN+NCNLN)
         READ (NIN,*) (BU(I),I=1,N+NCLIN+NCNLN)
         READ (NIN,*) (X(I),I=1,N)
*
*        Call E04WCF to initialise E04WDF.
         IFAIL = -1
         CALL E04WCF(IW,LENIW,RW,LENRW,IFAIL)
*
*        By default E04WDF does not print monitoring information.
*        Use E04WGF to set the integer-valued option 'Print file'
*        unit number to get information.
         CALL E04WGF('Print file',NOUT,IW,RW,IFAIL)
*
*        Use E04WEF to read some options from the end of the input
*        data file.
         CALL E04WEF(NIN,IW,RW,IFAIL)
         WRITE (NOUT,*)
*
*        Use E04WKF to find the value of integer-valued option
*        'Elastic mode'.
         CALL E04WKF('Elastic mode',ELMODE,IW,RW,IFAIL)
         WRITE (NOUT,99999) ELMODE
*
*        Use E04WHF to set the value of real-valued option
*        'Infinite bound size'.
         BNDINF = 1.0D10
         CALL E04WHF('Infinite bound size',BNDINF,IW,RW,IFAIL)
*
*        Use E04WLF to find the value of real-valued option
*        'Feasibility tolerance'.
         CALL E04WLF('Feasibility tolerance',FEATOL,IW,RW,IFAIL)
         WRITE (NOUT,99998) FEATOL
*
*        Use E04WFF to set the option 'Major iterations limit'.
         CALL E04WFF('Major iterations limit 50',IW,RW,IFAIL)
*
*        Solve the problem.
         IFAIL = -1
         CALL E04WDF(N,NCLIN,NCNLN,LDA,LDCJ,LDH,A,BL,BU,CONFUN,OBJFUN,
     +               MAJITS,ISTATE,CCON,CJAC,CLAMDA,OBJF,GRAD,HESS,X,IW,
     +               LENIW,RW,LENRW,IUSER,RUSER,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99997) IFAIL
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,99996) OBJF
            WRITE (NOUT,99995) (X(I),I=1,N)
         END IF
*
      END IF
      STOP
*
99999 FORMAT (1X,'Option ''Elastic mode'' has the value ',I3,'.')
99998 FORMAT (1X,'Option ''Feasibility tolerance'' has the value ',1P,
```

```
     +       E13.5,'.')
99997 FORMAT (1X,'On exit from E04WDF, IFAIL = ',I5)
99996 FORMAT (1X,'Final objective value = ',F11.3)
99995 FORMAT (1X,'Optimal X = ',7F9.2)
      END

      SUBROUTINE OBJFUN(MODE,N,X,OBJF,GRAD,NSTATE,IUSER,RUSER)
*     Routine to evaluate objective function and its 1st derivatives.
*     .. Parameters ..
      DOUBLE PRECISION  ONE, TWO
      PARAMETER         (ONE=1.0D0,TWO=2.0D0)
*     .. Scalar Arguments ..
      DOUBLE PRECISION  OBJF
      INTEGER           MODE, N, NSTATE
*     .. Array Arguments ..
      DOUBLE PRECISION  GRAD(N), RUSER(*), X(N)
      INTEGER           IUSER(*)
*     .. Executable Statements ..
      IF (MODE.EQ.0 .OR. MODE.EQ.2) OBJF = X(1)*X(4)*(X(1)+X(2)+X(3)) +
     +    X(3)
*
      IF (MODE.EQ.1 .OR. MODE.EQ.2) THEN
         GRAD(1) = X(4)*(TWO*X(1)+X(2)+X(3))
         GRAD(2) = X(1)*X(4)
         GRAD(3) = X(1)*X(4) + ONE
         GRAD(4) = X(1)*(X(1)+X(2)+X(3))
      END IF
*
      RETURN
      END
*
      SUBROUTINE CONFUN(MODE,NCNLN,N,LDCJ,NEEDC,X,CCON,CJAC,NSTATE,
     +                  IUSER,RUSER)
*     Routine to evaluate the nonlinear constraints and their 1st
*     derivatives.
*     .. Parameters ..
      DOUBLE PRECISION  ZERO, TWO
      PARAMETER         (ZERO=0.0D0,TWO=2.0D0)
*     .. Scalar Arguments ..
      INTEGER           LDCJ, MODE, N, NCNLN, NSTATE
*     .. Array Arguments ..
      DOUBLE PRECISION  CCON(*), CJAC(LDCJ,*), RUSER(*), X(N)
      INTEGER           IUSER(*), NEEDC(*)
*     .. Local Scalars ..
      INTEGER           I, J
*     .. Executable Statements ..
      IF (NSTATE.EQ.1) THEN
*        First call to CONFUN.  Set all Jacobian elements to zero.
*        Note that this will only work when 'Derivative Level = 3'
*        (the default; see Section 11.2).
         DO 40 J = 1, N
            DO 20 I = 1, NCNLN
               CJAC(I,J) = ZERO
   20       CONTINUE
   40    CONTINUE
      END IF
*
      IF (NEEDC(1).GT.0) THEN
         IF (MODE.EQ.0 .OR. MODE.EQ.2) CCON(1) = X(1)**2 + X(2)**2 +
     +       X(3)**2 + X(4)**2
         IF (MODE.EQ.1 .OR. MODE.EQ.2) THEN
            CJAC(1,1) = TWO*X(1)
            CJAC(1,2) = TWO*X(2)
            CJAC(1,3) = TWO*X(3)
            CJAC(1,4) = TWO*X(4)
         END IF
      END IF
*
      IF (NEEDC(2).GT.0) THEN
         IF (MODE.EQ.0 .OR. MODE.EQ.2) CCON(2) = X(1)*X(2)*X(3)*X(4)
         IF (MODE.EQ.1 .OR. MODE.EQ.2) THEN
```

```
            CJAC(2,1) = X(2)*X(3)*X(4)
            CJAC(2,2) = X(1)*X(3)*X(4)
            CJAC(2,3) = X(1)*X(2)*X(4)
            CJAC(2,4) = X(1)*X(2)*X(3)
         END IF
      END IF
*
      RETURN
      END
```

9.2 Program Data

```
E04WEF Example Program Data
  4   1   2                                                 : N, NCLIN and NCNLN
  1.0   1.0   1.0   1.0                                     : Matrix A
  1.0   1.0   1.0   1.0  -1.0E+25  -1.0E+25  25.0          : Lower bounds BL
  5.0   5.0   5.0   5.0  20.0      40.0      1.0E+25       : Upper bounds BU
  1.0   5.0   5.0   1.0                                     : Initial vector X
Begin example options file
* Comment lines like this begin with an asterisk.
* Switch off output of timing information:
Timing level 0
* Allow elastic variables:
Elastic mode 1
* Set the feasibility tolerance:
Feasibility tolerance 1.0D-4
End
```

9.3 Program Results

```
 E04WEF Example Program Results

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

      Begin example options file
      * Comment lines like this begin with an asterisk.
      * Switch off output of timing information:
      Timing level 0
      * Allow elastic variables:
      Elastic mode 1
      * Set the feasibility tolerance:
      Feasibility tolerance 1.0D-4
      End

 E04WEZ EXIT 100 -- finished successfully
 E04WEZ INFO 101 -- OPTIONS file read

 Option 'Elastic mode' has the value   1.
 Option 'Feasibility tolerance' has the value   1.00000E-04.

 Parameters
 ==========

 Files
 -----
 Solution file.....           0        Old basis file ........           0
(Print file)...........        6
 Insert file.............     0        New basis file ........           0
(Summary file).........        0
 Punch file..............      0       Backup basis file......        0
 Load file...............      0       Dump file..............        0

 Frequencies
 -----------
 Print frequency........         100         Check frequency........          60
Save new basis map.....       100
 Summary frequency......         100         Factorization frequency          50
Expand frequency.......     10000
```

```
 QP subproblems
 --------------
 QPsolver Cholesky......
  Scale tolerance........       0.900          Minor feasibility tol..   1.00E-04
Iteration limit........     10000
  Scale option...........           0          Minor optimality  tol..   1.00E-06
Minor print level......         1
  Crash tolerance........       0.100          Pivot tolerance........   1.11E-15
Partial price..........         1
  Crash option...........           3          Elastic weight.........   1.00E+04
Prtl price section ( A)         4
                                               New superbasics........         99
Prtl price section (-I)         3

 The SQP Method
 --------------
  Minimize...............                                Cold  start.............
Proximal Point method..         1
  Nonlinear objectiv vars           4          Major optimality tol...   2.00E-06
Function precision.....  1.72E-13
  Unbounded step size....   1.00E+10         Superbasics limit......           4
Difference interval....  4.15E-07
  Unbounded objective....   1.00E+15         Hessian dimension......           4
Central difference int.  5.57E-05
   Major  step  limit.......     2.00E+00               Derivative  linesearch..
Derivative option......         3
  Major iterations limit.          50          Linesearch tolerance...    0.90000
Verify level...........         0
  Minor iterations limit.         500          Penalty parameter......   0.00E+00
Major Print Level......         1

 Hessian Approximation
 ---------------------
  Full-Memory Hessian....                       Hessian updates........   99999999
Hessian frequency......  99999999

Hessian flush..........  99999999

 Nonlinear constraints
 ---------------------
  Nonlinear constraints..           2          Major feasibility tol..   1.00E-06
Violation limit........  1.00E+06
 Nonlinear Jacobian vars         4

 Miscellaneous
 -------------
  LU factor tolerance....        1.10          LU singularity tol.....   1.05E-08
Timing level...........         0
  LU update tolerance....        1.10          LU swap tolerance......   1.03E-04
Debug level............         0
  LU partial  pivoting...                       eps (machine precision)   1.11E-16
System information.....        No

 Nonlinear constraints       2     Linear constraints       1
 Nonlinear variables         4     Linear variables         0
 Jacobian  variables         4     Objective variables      4
 Total constraints           3     Total variables          4

 The user has defined       8   out of       8   constraint gradients.
 The user has defined       4   out of       4   objective  gradients.

 Cheap test of user-supplied problem derivatives...

 The constraint gradients seem to be OK.

 -->  The largest discrepancy was    1.84E-07  in constraint     6

 The objective  gradients seem to be OK.

 Gradient projected in one direction   4.99993000077E+00
 Difference approximation              4.99993303560E+00
```

```
   Itns Major Minors     Step    nCon Feasible  Optimal  MeritFunction      L+U
BSwap     nS  condHz Penalty
      2     0       2               1  1.7E+00  2.8E+00  1.6000000E+01        7
2 1.0E+00          _  r
      4     1       2 1.0E+00       2  1.3E-01  3.2E-01  1.7726188E+01        8
1 6.2E+00 8.3E-02 _n r
      5     2       1 1.0E+00       3  3.7E-02  1.7E-01  1.7099571E+01        7
1 2.0E+00 8.3E-02 _s
      6     3       1 1.0E+00       4  2.2E-02  1.1E-02  1.7014005E+01        7
1 1.8E+00 8.3E-02 _
      7     4       1 1.0E+00       5  1.5E-04  6.0E-04  1.7014018E+01        7
1 1.8E+00 9.2E-02 _
      8     5       1 1.0E+00       6 (3.3E-07) 2.3E-05  1.7014017E+01        7
1 1.9E+00 3.6E-01 _
      9     6       1 1.0E+00       7 (4.2E-10)(2.4E-08) 1.7014017E+01        7
1 1.9E+00 3.6E-01 _

 E04WDF EXIT   0 -- finished successfully
 E04WDF INFO   1 -- optimality conditions satisfied

 Problem name                     NLP
 No. of iterations                  9   Objective value      1.7014017287E+01
 No. of major iterations            6   Linear objective     0.0000000000E+00
 Penalty parameter          3.599E-01   Nonlinear objective  1.7014017287E+01
 No. of calls to funobj             8   No. of calls to funcon              8
 No. of superbasics                 1   No. of basic nonlinears             2
 No. of degenerate steps            0   Percentage                       0.00
 Max x                      2 4.7E+00   Max pi                     2 5.5E-01
 Max Primal infeas          0 0.0E+00   Max Dual infeas            3 4.8E-08
 Nonlinear constraint violn    2.7E-09

 Variable           State       Value        Lower bound      Upper bound    Lagr
multiplier       Slack

  variable        1       LL      1.000000           1.000000           5.000000
1.087871            .
 variable     2     FR    4.743000        1.000000        5.000000            .
0.2570
 variable     3     FR    3.821150        1.000000        5.000000            .
1.179
 variable     4     FR    1.379408        1.000000        5.000000            .
0.3794

 Linear constrnt   State       Value        Lower bound      Upper bound    Lagr
multiplier       Slack

 lincon       1     FR    10.94356          None          20.00000            .
9.056

 Nonlin constrnt   State       Value        Lower bound      Upper bound    Lagr
multiplier       Slack

 nlncon        1      UL     40.00000           None           40.00000       -
0.1614686         -0.2700E-08
 nlncon          2       LL      25.00000           25.00000              None
0.5522937         -0.2215E-08

 On exit from E04WDF, IFAIL =     0
 Final objective value =      17.014
 Optimal X =      1.00     4.74     3.82     1.38
```