NAG Fortran Library Routine Document

E04NGF/E04NGA

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 E04NFF/E04NFA from an external file. More precisely, E04NGF must be used to supply optional parameters to E04NFF and E04NGA must be used to supply optional parameters to E04NFA.

E04NGA is a version of E04NGF 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 E04NGA.

2 Specifications

2.1 Specification for E04NGF

```
SUBROUTINE E04NGF(IOPTNS, INFORM)
INTEGER           IOPTNS, INFORM
```

2.2 Specification for E04NGA

```
SUBROUTINE E04NGA(IOPTNS, LWSAV, IWSAV, RWSAV, INFORM)
INTEGER           IOPTNS, IWSAV(610), INFORM
real              RWSAV(475)
LOGICAL           LWSAV(120)
```

3 Description

E04NGF/E04NGA may be used to supply values for optional parameters to the corresponding routines E04NFF/E04NFA. E04NGF/E04NGA 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 E04NGF 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 E04NFF or E04NGF and may be turned on again at any time using the keyword **list**.

For E04NGA 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 E04NFF/E04NFA 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 E04NFF/E04NFA.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 11 of the document for E04NFF/E04NFA.

4 References

None.

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 E04NGA, 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 E04NGA. Users of E04NGF therefore need not read the remainder of this section.*

2: LWSAV(120) – LOGICAL array *Workspace*

3: IWSAV(610) – INTEGER array *Workspace*

4: RWSAV(475) – ***real*** array *Workspace*

The arrays LWSAV, IWSAV and RWSAV **must not** be altered between calls to any of the routines E04WBF, E04NFA, E04NGA or E04NHA.

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

E04NHF/E04NHA may also be used to supply optional parameters to the corresponding routines E04NFF/E04NFA.

9 Example

This example solves the same problem as the example for E04NFF/E04NFA, but in addition illustrates the use of E04NGF/E04NGA and E04NHF/E04NHA to set optional parameters for E04NFF/E04NFA.

In this example the options file read by E04NGF/E04NGA 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 E04NGF. An equivalent program illustrating the use of E04NGA is available with the supplied Library and is also available from the NAG web site.*

```
*     E04NGF Example Program Text
*     Mark 18 Revised.  NAG Copyright 1997.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, NCMAX
      PARAMETER        (NMAX=10,NCMAX=10)
      INTEGER          LDA
      PARAMETER        (LDA=NCMAX)
      INTEGER          LIWORK, LWORK
      PARAMETER        (LIWORK=1000,LWORK=10000)
*     .. Local Scalars ..
      real             OBJ
```

```
      INTEGER          I, IFAIL, INFORM, ITER, J, LDH, N, NCLIN
      CHARACTER        UPLO
*     .. Local Arrays ..
      real             A(LDA,NMAX), AX(NCMAX), BL(NMAX+NCMAX),
     +                 BU(NMAX+NCMAX), CLAMDA(NMAX+NCMAX), CVEC(NMAX),
     +                 H(NMAX*(NMAX+1)/2), WORK(LWORK), X(NMAX)
      INTEGER          ISTATE(NMAX+NCMAX), IWORK(LIWORK)
*     .. External Subroutines ..
      EXTERNAL         E04NFF, E04NGF, E04NHF, QPHESS, X04ABF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E04NGF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, NCLIN
      IF (N.LE.NMAX .AND. NCLIN.LE.NCMAX) THEN
*
*        Read CVEC, A, BL, BU, X, UPLO and H from data file
*
         READ (NIN,*) (CVEC(I),I=1,N)
         READ (NIN,*) ((A(I,J),J=1,N),I=1,NCLIN)
         READ (NIN,*) (BL(I),I=1,N+NCLIN)
         READ (NIN,*) (BU(I),I=1,N+NCLIN)
         READ (NIN,*) (X(I),I=1,N)
         READ (NIN,*) UPLO
         IF (UPLO.EQ.'U') THEN
*           Read the upper triangle of H
            READ (NIN,*) ((H(J+(2*N-I)*(I-1)/2),J=I,N),I=1,N)
         ELSE IF (UPLO.EQ.'L') THEN
*           Read the lower triangle of H
            READ (NIN,*) ((H(I+(2*N-J)*(J-1)/2),J=1,I),I=1,N)
         END IF
         LDH = N*(N+1)/2
*
*        Set four options using E04NHF
*
         CALL E04NHF(' Print Level = 1 ')
*
         CALL E04NHF(' Check Frequency = 10 ')
*
         CALL E04NHF(' Crash Tolerance = 0.05 ')
*
         CALL E04NHF(' 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 E04NGF(NIN,INFORM)
*
         IF (INFORM.NE.0) THEN
            WRITE (NOUT,99999) 'E04NGF terminated with INFORM = ',
     +        INFORM
            STOP
         END IF
*
*        Solve the problem
*
         IFAIL = -1
*
         CALL E04NFF(N,NCLIN,A,LDA,BL,BU,CVEC,H,LDH,QPHESS,ISTATE,X,
     +               ITER,OBJ,AX,CLAMDA,IWORK,LIWORK,WORK,LWORK,IFAIL)
*
      END IF
      STOP
*
99999 FORMAT (1X,A,I3)
      END
*
      SUBROUTINE QPHESS(N,JTHCOL,HESS,LDHESS,X,HX)
```

```
*     In this version of QPHESS, the lower triangle of the matrix H is
*     stored in packed form (by columns) in the one-dimensional array
*     HESS. More precisely, the lower triangle of H must be stored with
*     element H(i,j) in HESS(i+(2*N-j)*(j-1)/2) for i .ge. j.
*     Note that storing the lower triangle of H in packed form (by
*     columns) is equivalent to storing the upper triangle of H in
*     packed form (by rows).
*     Note also that LDHESS is used to define the length of HESS, and
*     must therefore be at least N*(N+1)/2.
*     .. Scalar Arguments ..
      INTEGER           JTHCOL, LDHESS, N
*     .. Array Arguments ..
      real              HESS(LDHESS), HX(N), X(N)
*     .. Local Scalars ..
      real              S
      INTEGER           I, INC, J, L, LP1
*     .. Executable Statements ..
      IF (JTHCOL.NE.0) THEN
*        Special case -- extract one column of  H.
         L = JTHCOL
         INC = N
         DO 20 I = 1, JTHCOL
            HX(I) = HESS(L)
            INC = INC - 1
            L = L + INC
   20    CONTINUE
         L = L - INC + 1
         IF (JTHCOL.LT.N) THEN
            LP1 = L
            DO 40 I = JTHCOL + 1, N
               HX(I) = HESS(LP1)
               LP1 = LP1 + 1
   40       CONTINUE
         END IF
      ELSE
*        Normal case.
         L = 0
         DO 80 I = 1, N
            S = 0.0e0
            DO 60 J = I, N
               L = L + 1
               S = S + HESS(L)*X(J)
   60       CONTINUE
            HX(I) = S
   80    CONTINUE
         L = 0
         DO 120 J = 1, N - 1
            L = L + 1
            DO 100 I = J + 1, N
               L = L + 1
               HX(I) = HX(I) + HESS(L)*X(J)
  100       CONTINUE
  120    CONTINUE
      END IF
      RETURN
      END
```

9.2 Program Data

```
E04NGF Example Program Data
  7  7                                                :Values of N and NCLIN
 -0.02  -0.20  -0.20  -0.20  -0.20   0.04   0.04   :End of CVEC
  1.00   1.00   1.00   1.00   1.00   1.00   1.00
  0.15   0.04   0.02   0.04   0.02   0.01   0.03
  0.03   0.05   0.08   0.02   0.06   0.01   0.00
  0.02   0.04   0.01   0.02   0.02   0.00   0.00
  0.02   0.03   0.00   0.00   0.01   0.00   0.00
  0.70   0.75   0.80   0.75   0.80   0.97   0.00
  0.02   0.06   0.08   0.12   0.02   0.01   0.97   :End of matrix A
 -0.01  -0.10     -0.01     -0.04     -0.10     -0.01     -0.01
```

```
 -0.13  -1.0e+25  -1.0e+25  -1.0e+25  -1.0e+25  -9.92e-02 -3.0e-03 :End of BL
  0.01   0.15      0.03      0.02      0.05      1.0e+25   1.0e+25
 -0.13  -4.9e-03  -6.4e-03  -3.7e-03  -1.2e-03   1.0e+25   2.0e-03 :End of BU
 -0.01  -0.03   0.00  -0.01  -0.10   0.02   0.01   :End of X
  'L'                                              :End of UPLO
  2.00
  0.00   2.00
  0.00   0.00   2.00
  0.00   0.00   2.00   2.00
  0.00   0.00   0.00   0.00   2.00
  0.00   0.00   0.00   0.00   0.00  -2.00
  0.00   0.00   0.00   0.00   0.00  -2.00  -2.00   :End of matrix H
Begin   Example options file for  E04NGF
   Feasibility Phase Iteration Limit  =   5    *  (Default = 70)
   Optimality  Phase Iteration Limit  =  10    *  (Default = 70)
End
```

9.3 Program Results

```
 E04NGF Example Program Results


 Calls to E04NHF
 ---------------

       Print Level = 1
       Check Frequency = 10
       Crash Tolerance = 0.05
       Infinite Bound Size = 1.0E+25


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

      Begin   Example options file for  E04NGF
         Feasibility Phase Iteration Limit  =   5    *  (Default = 70)
         Optimality  Phase Iteration Limit  =  10    *  (Default = 70)
      End

 *** E04NFF
 *** 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
 ----------

 Problem type...........       QP2

 Linear constraints.....         7       Feasibility tolerance..  1.05E-08
 Variables..............         7       Optimality tolerance...  1.72E-13
 Hessian rows...........         7       Rank tolerance.........  1.11E-14

 Infinite bound size....  1.00E+25       COLD start.............
 Infinite step size.....  1.00E+25       EPS (machine precision)  1.11E-16

 Check frequency........        10       Expand frequency.......         5
 Minimum sum of infeas..        NO       Crash tolerance........  5.00E-02

 Max degrees of freedom.         7       Print level............         1
 Feasibility phase itns.         5       Monitoring file........        -1
 Optimality  phase itns.        10

 Workspace provided is     IWORK(    1000),  WORK(   10000).
 To solve problem we need  IWORK(      17),  WORK(     189).
```

```
 Varbl State     Value       Lower Bound   Upper Bound    Lagr Mult   Slack

 V   1    LL  -1.000000E-02 -1.000000E-02  1.000000E-02  0.4700         .
 V   2    FR  -6.986465E-02 -0.100000      0.150000            .      3.0135E-02
 V   3    FR   1.825915E-02 -1.000000E-02  3.000000E-02        .      1.1741E-02
 V   4    FR  -2.426081E-02 -4.000000E-02  2.000000E-02        .      1.5739E-02
 V   5    FR  -6.200564E-02 -0.100000      5.000000E-02        .      3.7994E-02
 V   6    FR   1.380544E-02 -1.000000E-02      None            .      2.3805E-02
 V   7    FR   4.066496E-03 -1.000000E-02      None            .      1.4066E-02


 L Con State     Value       Lower Bound   Upper Bound    Lagr Mult   Slack

 L   1    EQ  -0.130000     -0.130000     -0.130000      -1.908      2.7756E-17
 L   2    FR  -5.879898E-03      None     -4.900000E-03        .      9.7990E-04
 L   3    UL  -6.400000E-03      None     -6.400000E-03 -0.3144      8.6736E-19
 L   4    FR  -4.537323E-03      None     -3.700000E-03        .      8.3732E-04
 L   5    FR  -2.915996E-03      None     -1.200000E-03        .      1.7160E-03
 L   6    LL  -9.920000E-02 -9.920000E-02      None        1.955     -1.3878E-17
 L   7    LL  -3.000000E-03 -3.000000E-03  2.000000E-03    1.972     -4.3368E-19

 Exit E04NFF - Optimal QP solution.

 Final QP objective value =   0.3703165E-01

 Exit from QP problem after     8 iterations.
```