NAG Fortran Library Routine Document

E04NRF

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 E04NQF from an external file. The initialization routine E04NPF **must** have been called prior to calling E04NRF.

2 Specification

```
SUBROUTINE E04NRF (ISPECS, CW, IW, RW, IFAIL)
INTEGER            ISPECS, IW(*), IFAIL
double precision   RW(*)
CHARACTER*8        CW(*)
```

3 Description

E04NRF may be used to supply values for optional parameters to E04NQF. E04NRF 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 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 E04NQF 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 E04NQF.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 11 of the document for E04NQF.

4 References

None.

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: CW(∗) – CHARACTER*8 array *Communication Array*
3: IW(∗) – INTEGER array *Communication Array*
4: RW(∗) – ***double precision*** array *Communication Array*

The arrays CW, IW and RW are defined in the document for E04NPF and **must not** be altered between calls to any of the routines E04NPF, E04NQF, E04NRF, E04NSF, E04NTF, E04NUF, E04NXF and E04NYF.

5: 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 E04NPF 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

E04NSF, E04NTF or E04NUF may also be used to supply optional parameters to E04NQF.

9 Example

To minimize the quadratic function $f(x) = c^T x + \frac{1}{2}x^T H x$, where

$$c = (-200.0, -2000.0, -2000.0, -2000.0, -2000.0, 400.0, 400.0)^T$$

$$H = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 \end{pmatrix}$$

subject to the bounds

$$\begin{aligned} 0 &\leq x_1 \leq 200 \\ 0 &\leq x_2 \leq 2500 \\ 400 &\leq x_3 \leq 800 \\ 100 &\leq x_4 \leq 700 \\ 0 &\leq x_5 \leq 1500 \\ 0 &\leq x_6 \\ 0 &\leq x_7 \end{aligned}$$

and to the linear constraints

$$\begin{aligned} x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 &= 2000 \\ 0.15x_1 + 0.04x_2 + 0.02x_3 + 0.04x_4 + 0.02x_5 + 0.01x_6 + 0.03x_7 &\leq 60 \\ 0.03x_1 + 0.05x_2 + 0.08x_3 + 0.02x_4 + 0.06x_5 + 0.01x_6 &\leq 100 \\ 0.02x_1 + 0.04x_2 + 0.01x_3 + 0.02x_4 + 0.02x_5 &\leq 40 \\ 0.02x_1 + 0.03x_2 + 0.01x_5 &\leq 30 \\ 1500 \leq 0.70x_1 + 0.75x_2 + 0.80x_3 + 0.75x_4 + 0.80x_5 + 0.97x_6 & \\ 250 \leq 0.02x_1 + 0.06x_2 + 0.08x_3 + 0.12x_4 + 0.02x_5 + 0.01x_6 + 0.97x_7 &\leq 300 \end{aligned}$$

The initial point, which is infeasible, is

$$x_0 = (0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0)^T.$$

The optimal solution (to five figures) is

$$x^* = (0.0, 349.40, 648.85, 172.85, 407.52, 271.36, 150.02)^T.$$

One bound constraint and four linear constraints are active at the solution. Note that the Hessian matrix H is positive semi-definite.

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.

```
*     E04NRF Example Program Text
*     Mark 21 Release. NAG Copyright 2004.
      IMPLICIT         NONE
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX, NEMAX
      PARAMETER        (NMAX=100,MMAX=100,NEMAX=100)
      INTEGER          LENCW, LENIW, LENRW
      PARAMETER        (LENCW=600,LENIW=600,LENRW=600)
*     .. Local Scalars ..
      DOUBLE PRECISION BNDINF, FEATOL, OBJ, OBJADD, SINF
      INTEGER          ELMODE, I, ICOL, IFAIL, IOBJ, J, JCOL, LENC, M,
     +                 N, NCOLH, NE, NINF, NNAME, NS
      CHARACTER        START
```

```
      CHARACTER*8      PROB
*     .. Local Arrays ..
      DOUBLE PRECISION ACOL(NEMAX), BL(NMAX+MMAX), BU(NMAX+MMAX), C(1),
     +                 PI(MMAX), RC(NMAX+MMAX), RUSER(1), RW(LENRW),
     +                 X(NMAX+MMAX)
      INTEGER          HELAST(NMAX+MMAX), HS(NMAX+MMAX), INDA(NEMAX),
     +                 IUSER(1), IW(LENIW), LOCA(NMAX+1)
      CHARACTER*8      CUSER(1), CW(LENCW), NAMES(NMAX+MMAX)
*     .. External Subroutines ..
      EXTERNAL         E04NPF, E04NQF, E04NRF, E04NSF, E04NTF, E04NUF,
     +                 E04NXF, E04NYF, QPHX
*     .. Executable Statements ..
      WRITE (NOUT,*) 'E04NRF Example Program Results'
*
*     This program demonstrates the use of routines to set and
*     get values of optional parameters associated with E04NQF.
*
*     Skip heading in data file.
      READ (NIN,*)
      READ (NIN,*) N, M
      IF (N.LE.NMAX .AND. M.LE.MMAX) THEN
*
*        Read NE, IOBJ, NCOLH, START and NNAME from data file.
         READ (NIN,*) NE, IOBJ, NCOLH, START, NNAME
*
*        Read NAMES from data file.
         READ (NIN,*) (NAMES(I),I=1,NNAME)
*
*        Read the matrix ACOL from data file. Set up LOCA.
         JCOL = 1
         LOCA(JCOL) = 1
         DO 40 I = 1, NE
*
*           Element ( INDA( I ), ICOL ) is stored in ACOL( I ).
            READ (NIN,*) ACOL(I), INDA(I), ICOL
*
            IF (ICOL.LT.JCOL) THEN
*              Elements not ordered by increasing column index.
               WRITE (NOUT,99999) 'Element in column', ICOL,
     +           ' found after element in column', JCOL, '. Problem',
     +           ' abandoned.'
               STOP
            ELSE IF (ICOL.EQ.JCOL+1) THEN
*              Index in ACOL of the start of the ICOL-th column equals I.
               LOCA(ICOL) = I
               JCOL = ICOL
            ELSE IF (ICOL.GT.JCOL+1) THEN
*              Index in ACOL of the start of the ICOL-th column equals I,
*              but columns JCOL+1,JCOL+2,...,ICOL-1 are empty. Set the
*              corresponding elements of LOCA to I.
               DO 20 J = JCOL + 1, ICOL - 1
                  LOCA(J) = I
   20          CONTINUE
               LOCA(ICOL) = I
               JCOL = ICOL
            END IF
   40    CONTINUE
*
         LOCA(N+1) = NE + 1
*
         IF (N.GT.ICOL) THEN
*           Columns N,N-1,...,ICOL+1 are empty. Set the corresponding
*           elements of LOCA accordingly.
            DO 60 I = N, ICOL + 1, -1
               LOCA(I) = LOCA(I+1)
   60       CONTINUE
         END IF
*
*        Read BL, BU, HS and X 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,*) (HS(I),I=1,N)
         ELSE IF (START.EQ.'W') THEN
            READ (NIN,*) (HS(I),I=1,N+M)
         END IF
         READ (NIN,*) (X(I),I=1,N)
*
*        We have no explicit objective vector so set LENC = 0; the
*        objective vector is stored in row IOBJ of ACOL.
         LENC = 0
         OBJADD = 0.0D0
         PROB = ' '
*
*        Call E04NPF to initialise E04NQF.
         IFAIL = 0
         CALL E04NPF(CW,LENCW,IW,LENIW,RW,LENRW,IFAIL)
*
*        By default E04NQF does not print monitoring information.
*        Use E04NTF to set the integer-valued option 'Print file'
*        unit number to get information.
         CALL E04NTF('Print file',NOUT,CW,IW,RW,IFAIL)
*
*        Use E04NRF to read some options from the end of the input
*        data file.
         CALL E04NRF(NIN,CW,IW,RW,IFAIL)
         WRITE (NOUT,*)
*
*        Use E04NXF to find the value of integer-valued option
*        'Elastic mode'.
         CALL E04NXF('Elastic mode',ELMODE,CW,IW,RW,IFAIL)
         WRITE (NOUT,99998) ELMODE
*
*        Use E04NUF to set the value of real-valued option
*        'Infinite bound size'.
         BNDINF = 1.0D10
         CALL E04NUF('Infinite bound size',BNDINF,CW,IW,RW,IFAIL)
*
*        Use E04NYF to find the value of real-valued option
*        'Feasibility tolerance'.
         CALL E04NYF('Feasibility tolerance',FEATOL,CW,IW,RW,IFAIL)
         WRITE (NOUT,99997) FEATOL
*
*        Use E04NSF to set the option 'Iterations limit'.
         CALL E04NSF('Iterations limit 50',CW,IW,RW,IFAIL)
*
*        Solve the QP problem.
         IFAIL = -1
         CALL E04NQF(START,QPHX,M,N,NE,NNAME,LENC,NCOLH,IOBJ,OBJADD,
     +               PROB,ACOL,INDA,LOCA,BL,BU,C,NAMES,HELAST,HS,X,PI,
     +               RC,NS,NINF,SINF,OBJ,CW,LENCW,IW,LENIW,RW,LENRW,
     +               CUSER,IUSER,RUSER,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99996) IFAIL
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,99995) OBJ
            WRITE (NOUT,99994) (X(I),I=1,N)
         END IF
*
      END IF
      STOP
*
99999 FORMAT (1X,A,I5,A,I5,A,A)
99998 FORMAT (1X,'Option ''Elastic mode'' has the value ',I3,'.')
99997 FORMAT (1X,'Option ''Feasibility tolerance'' has the value ',1P,
     +       E13.5,'.')
99996 FORMAT (1X,'On exit from E04NQF, IFAIL = ',I5)
99995 FORMAT (1X,'Final objective value = ',1P,E11.3)
99994 FORMAT (1X,'Optimal X = ',7F9.2)
      END
*
```

```
      SUBROUTINE QPHX(NCOLH,X,HX,NSTATE,CUSER,IUSER,RUSER)
*     Routine to compute H*x. (In this version of QPHX, the Hessian
*     matrix H is not referenced explicitly.)
*     .. Parameters ..
      DOUBLE PRECISION TWO
      PARAMETER       (TWO=2.0D+0)
*     .. Scalar Arguments ..
      INTEGER         NCOLH, NSTATE
*     .. Array Arguments ..
      DOUBLE PRECISION HX(NCOLH), RUSER(*), X(NCOLH)
      INTEGER         IUSER(*)
      CHARACTER*8     CUSER(*)
*     .. Executable Statements ..
      HX(1) = TWO*X(1)
      HX(2) = TWO*X(2)
      HX(3) = TWO*(X(3)+X(4))
      HX(4) = HX(3)
      HX(5) = TWO*X(5)
      HX(6) = TWO*(X(6)+X(7))
      HX(7) = HX(6)
      RETURN
      END
```

9.2 Program Data

```
E04NRF Example Program Data
 7  8                      : Values of N and M
48  8  7  'C'  15          : Values of NNZ, IOBJ, NCOLH, START and NNAME

'...X1...'  '...X2...'  '...X3...'  '...X4...'  '...X5...'
'...X6...'  '...X7...'  '..ROW1..'  '..ROW2..'  '..ROW3..'
'..ROW4..'  '..ROW5..'  '..ROW6..'  '..ROW7..'  '..COST..' : End of array NAMES

    0.02   7   1   : Sparse matrix A, ordered by increasing column index;
    0.02   5   1   : each row contains ACOL(i), INDA(i), ICOL (= column index)
    0.03   3   1   : The row indices may be in any order. In this example
    1.00   1   1   : row 8 defines the linear objective term transpose(C)*X.
    0.70   6   1
    0.02   4   1
    0.15   2   1
 -200.00   8   1
    0.06   7   2
    0.75   6   2
    0.03   5   2
    0.04   4   2
    0.05   3   2
    0.04   2   2
    1.00   1   2
-2000.00   8   2
    0.02   2   3
    1.00   1   3
    0.01   4   3
    0.08   3   3
    0.08   7   3
    0.80   6   3
-2000.00   8   3
    1.00   1   4
    0.12   7   4
    0.02   3   4
    0.02   4   4
    0.75   6   4
    0.04   2   4
-2000.00   8   4
    0.01   5   5
    0.80   6   5
    0.02   7   5
    1.00   1   5
    0.02   2   5
    0.06   3   5
    0.02   4   5
```

```
-2000.00   8   5
    1.00   1   6
    0.01   2   6
    0.01   3   6
    0.97   6   6
    0.01   7   6
  400.00   8   6
    0.97   7   7
    0.03   2   7
    1.00   1   7
  400.00   8   7        : End of matrix A

 0.0       0.0       4.0E+02   1.0E+02   0.0       0.0
 0.0       2.0E+03  -1.0E+25  -1.0E+25  -1.0E+25  -1.0E+25
 1.5E+03   2.5E+02  -1.0E+25             : End of lower bounds array BL

 2.0E+02   2.5E+03   8.0E+02   7.0E+02  1.5E+03  1.0E+25
 1.0E+25   2.0E+03   6.0E+01   1.0E+02  4.0E+01  3.0E+01
 1.0E+25   3.0E+02   1.0E+25             : End of upper bounds array BU

 0    0    0    0    0    0    0         : Initial array HS
 0.0  0.0  0.0  0.0  0.0  0.0  0.0       : 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

```
 E04NRF 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

 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
```

```
 LP/QP Parameters
 ----------------
  Minimize..............                        QPsolver  Cholesky......
Cold start.............
  Scale tolerance........       0.900          Feasibility tolerance..   1.00E-04
Iteration limit........        50
  Scale option...........            2          Optimality tolerance...   1.00E-06
Print level............         1
  Crash tolerance........       0.100          Pivot tolerance........   2.04E-11
Partial price..........         1
  Crash option...........            3          Elastic weight.........   1.00E+00
Prtl price section ( A)         7
  Elastic mode...........          1         Elastic objective......           1
Prtl price section (-I)         8

 QP objective
 ------------
  Objective variables....          7         Hessian columns........           7
Superbasics limit......         7
  Nonlin Objective vars..        7        Unbounded step size....  1.00E+10
  Linear Objective vars..        0

 Miscellaneous
 -------------
  LU factor tolerance....      100.00          LU singularity tol.....   2.04E-11
Timing level...........         0
  LU update tolerance....       10.00          LU swap tolerance......   1.03E-04
Debug level............         0
  LU partial  pivoting...                      eps (machine precision)   1.11E-16
System information.....        No

 Nonlinear constraints       0     Linear constraints       8
 Nonlinear variables         7     Linear variables         0
 Jacobian  variables         0     Objective variables      7
 Total constraints           8     Total variables          7

 Itn      1: Feasible constraints

 E04NQF EXIT   0 -- finished successfully
 E04NQF INFO   1 -- optimality conditions satisfied

 Problem name
 No. of iterations                    9   Objective value     -1.8477846771E+06
 No. of Hessian products             16   Linear objective    -2.9886903537E+06
                                          Quadratic objective  1.1409056766E+06
 No. of superbasics                   2   No. of basic nonlinears             4
 No. of degenerate steps              0   Percentage                       0.00
 Max x       (scaled)        3 1.7E+00   Max pi      (scaled)        6 6.6E+06
 Max x                       3 6.5E+02   Max pi                      7 1.5E+04
 Max Prim inf(scaled)        0 0.0E+00   Max Dual inf(scaled)        3 1.5E-09
 Max Primal infeas           0 0.0E+00   Max Dual infeas             9 3.3E-11

 Name                                    Objective Value     -1.8477846771E+06

 Status         Optimal Soln             Iteration      9    Superbasics     2

 Section 1 - Rows

  Number  ...Row.. State  ...Activity...  Slack Activity  ..Lower Limit.  ..Upper
Limit.  .Dual Activity    ..i

        8  ..ROW1..     EQ       2000.00000            .             2000.00000
2000.00000    -12900.76766      1
        9  ..ROW2..     BS            49.23160        -10.76840             None
60.00000         0.00000      2
       10  ..ROW3..     UL           100.00000            .                None
100.00000     -2324.86620      3
       11  ..ROW4..     BS            32.07187         -7.92813             None
40.00000          .           4
       12  ..ROW5..     BS            14.55719        -15.44281             None
```

```
30.00000          .           5
       13   ..ROW6..     LL        1500.00000            .             1500.00000
None       14454.60290      6
       14   ..ROW7..     LL         250.00000            .              250.00000
300.00000     14580.95432      7
        15   ..COST..      BS   -2988690.35370   -2988690.35370               None
None         -1.0          8

 Section 2 - Columns

  Number  .Column. State  ...Activity...  .Obj Gradient.  ..Lower Limit.  ..Upper
Limit.  Reduced Gradnt    m+j

         1   ...X1...      LL             .              -200.00000            .
200.00000      2360.67253      9
         2   ...X2...      BS          349.39923       -1301.20153            .
2500.00000         0.00000     10
         3   ...X3...    SBS          648.85342        -356.59829         400.00000
800.00000          0.00000     11
         4   ...X4...    SBS          172.84743        -356.59829         100.00000
700.00000          0.00000     12
         5   ...X5...      BS          407.52089       -1184.95822            .
1500.00000         0.00000     13
         6   ...X6...      BS          271.35624         1242.75804           .
None           0.00000     14
         7   ...X7...      BS          150.02278         1242.75804           .
None           0.00000     15

 On exit from E04NQF, IFAIL =     0
 Final objective value =  -1.848E+06
 Optimal X =      0.00   349.40   648.85   172.85   407.52   271.36   150.02
```