

NAG Library Routine Document

H02CFF

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 H02CEF from an external file.

2 Specification

```
SUBROUTINE H02CFF(IOPTNS, INFORM)
INTEGER          IOPTNS, INFORM
```

3 Description

H02CFF may be used to supply values for optional parameters to H02CEF. H02CFF 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 equal 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 *double precision* value. Such numbers may be up to 16 contiguous characters in Fortran 77'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 = 1
End
```

Normally each line of the file is 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 = 1
End
```

Printing will automatically be turned on again after a call to H02CEF and may be turned on again at any time using the keyword List.

Optional parameter settings are preserved following a call to H02CEF, 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 H02CEF.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 11 in H02CEF.

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*
- On exit:* contains zero if the options file has been successfully read and a value > 0 otherwise, as indicated below.
- 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.

6 Error Indicators and Warnings

If a line is not recognized as a valid option, then a warning message is output on the current advisory message unit (see X04ABF).

7 Accuracy

Not applicable.

8 Further Comments

H02CGF may also be used to supply optional parameters to H02CEF. Note that if E04NKF/E04NKA is used in the same program as H02CEF, then in general H02CFF will also affect the options used by E04NKF/E04NKA.

9 Example

This example solves the same problem as the example for H02CEF, but in addition illustrates the use of H02CFF and H02CGF to set optional parameters for H02CEF.

In this example the options file read by H02CFF 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

```
*      H02CFF Example Program Text
*      Mark 19 Release. NAG Copyright 1999.
*      .. Parameters ..
INTEGER          NIN, NOUT
PARAMETER       (NIN=5,NOUT=6)
INTEGER          NMAX, MMAX, NNZMAX, LENIZ, LENZ, LINTVR, MM
PARAMETER       (NMAX=100,MMAX=100,NNZMAX=100,LENIZ=100000,
```

```

+          LENZ=100000,LINTVR=10,MM=2000)
*  .. Local Scalars ..
DOUBLE PRECISION OBJ
INTEGER          I, ICOL, IFAIL, INFORM, IOBJ, J, JCOL, M, MINIZ,
+          MINZ, N, NCOLH, NNAME, NNZ, NS, OUTCHN, STRTGY
CHARACTER        START
*  .. Local Arrays ..
DOUBLE PRECISION A(NNZMAX), BL(NMAX+MMAX), BU(NMAX+MMAX),
+          CLAMDA(NMAX+MMAX), XS(NMAX+MMAX), Z(LENZ)
INTEGER          HA(NNZMAX), INTVAR(LINTVR), ISTATE(NMAX+MMAX),
+          IZ(LENIZ), KA(NMAX+1)
CHARACTER*8      CRNAME(NMAX+MMAX), NAMES(5)
*  .. External Subroutines ..
EXTERNAL          H02CEF, H02CFF, H02CGF, MONIT, QPHX, X04ABF
*  .. Executable Statements ..
WRITE (NOUT,*) 'H02CFF Example Program Results'
OUTCHN = NOUT
*  Skip heading in data file.
READ (NIN,*)
READ (NIN,*) N, M
IF (N.LE.NMAX .AND. M.LE.MMAX) THEN
*
*      Read NNZ, IOBJ, NCOLH, START and NNAME from data file.
*
*      READ (NIN,*) NNZ, IOBJ, NCOLH, START, NNAME
*
*      Read NAMES and CRNAME from data file.
*
*      READ (NIN,*) (NAMES(I),I=1,5)
*      READ (NIN,*) (CRNAME(I),I=1,NNAME)
*
*      Read the matrix A from data file. Set up KA.
*
*      JCOL = 1
*      KA(JCOL) = 1
*      DO 40 I = 1, NNZ
*
*          Element ( HA( I ), ICOL ) is stored in A( I ).
*
*          READ (NIN,*) A(I), HA(I), ICOL
*
*          IF (ICOL.EQ.JCOL+1) THEN
*
*              Index in A of the start of the ICOL-th column equals I.
*
*              KA(ICOL) = I
*              JCOL = ICOL
*          ELSE IF (ICOL.GT.JCOL+1) THEN
*
*              Index in A 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 KA to I.
*
*              DO 20 J = JCOL + 1, ICOL - 1
*                  KA(J) = I
*              CONTINUE
*              KA(ICOL) = I
*              JCOL = ICOL
*          END IF
*      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)
*      READ (NIN,*) (ISTATE(I),I=1,N)
*      READ (NIN,*) (XS(I),I=1,N)
*
*      Set three options using H02CGF.
*

```

```

      CALL H02CGF(' Check Frequency = 10 ')
*
      CALL H02CGF(' Feasibility Tolerance = 0.00001 ')
*
      CALL H02CGF(' Infinite Bound Size = 1.0D+25 ')
*
      Set the unit number for advisory messages to OUTCHN.
*
      CALL X04ABF(1,OUTCHN)
*
      Read the options file for the remaining options.
*
      CALL H02CFF(NIN,INFORM)
*
      IF (INFORM.NE.0) THEN
        WRITE (NOUT,99999) 'H02CFF terminated with INFORM = ',
+          INFORM
        GO TO 60
      END IF
*
      STRTGY = 3
      INTVAR(1) = 2
      INTVAR(2) = 3
      INTVAR(3) = 4
      INTVAR(4) = 5
      INTVAR(5) = 6
      INTVAR(6) = 7
      INTVAR(7) = -1
*
      CALL H02CGF('NoList')
      CALL H02CGF('Print Level = 0')
*
      Solve the QP problem.
*
      IFAIL = 0
*
      CALL H02CEF(N,M,NNZ,IOBJ,NCOLH,QPHX,A,HA,KA,BL,BU,START,NAMES,
+        NNAME,CRNAME,NS,XS,INTVAR,LINTVR,MM,ISTATE,MINIZ,
+        MINZ,OBJ,CLAMDA,STRTGY,IZ,LENIZ,Z,LENZ,MONIT,IFAIL)
*
      Print out the best integer solution found
*
      WRITE (NOUT,99999) OBJ, (I,XS(I),I=1,N)
    END IF
  60 CONTINUE
*
99999 FORMAT (' Optimal Integer Value is = ',E20.8,/' Components are ',
+  /(' X(',I3,') = ',F10.2))
END
*
      SUBROUTINE QPHX(NSTATE,NCOLH,X,HX)
*
      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), X(NCOLH)
*
      .. 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)
*

```

```

      END
*
      SUBROUTINE MONIT(INTFND,NODES,DEPTH,OBJ,X,BSTVAL,BSTSOL,BL,BU,N,
+          HALT,COUNT)
*
      .. Parameters ..
      DOUBLE PRECISION CUTOFF
      PARAMETER      (CUTOFF=-1840000.0D+0)
*
      .. Scalar Arguments ..
      DOUBLE PRECISION BSTVAL, OBJ
      INTEGER          COUNT, DEPTH, INTFND, N, NODES
      LOGICAL          HALT
*
      .. Array Arguments ..
      DOUBLE PRECISION BL(N), BSTSOL(N), BU(N), X(N)
*
      .. Executable Statements ..
      IF (INTFND.EQ.0) BSTVAL = CUTOFF
*
      END

```

9.2 Program Data

H02CFF Example Program Data

```

7  8      :Values of N and M
48 8  7  'C' 15 :Values of NNZ, IOBJ, NCOLH, START and NNAME
'      '      '      '      '      '      ' :End of NAMES
'...X1...' '...X2...' '...X3...' '...X4...' '...X5...'
'...X6...' '...X7...' '..ROW1..' '..ROW2..' '..ROW3..'
'..ROW4..' '..ROW5..' '..ROW6..' '..ROW7..' '..COST..' :End of CRNAME
0.02  7  1
0.02  5  1
0.03  3  1
1.00  1  1
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
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 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 BU
0         0         0         0         0         0         0         0 :End of ISTATE
0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 :End of XS
Begin
  Iteration Limit = 125 * (Default = 75)
  Print Level = 1      * (Default = 10)
End

```

9.3 Program Results

H02CFF Example Program Results

Calls to H02CGF

```

Check Frequency = 10
Feasibility Tolerance = 0.00001
Infinite Bound Size = 1.0D+25

```

OPTIONS file

```

Begin
  Iteration Limit = 125 * (Default = 75)
  Print Level = 1      * (Default = 10)
End
Optimal Integer Value is =      -0.18475180E+07
Components are
X( 1) =      0.00
X( 2) =     355.00
X( 3) =     645.00
X( 4) =     164.00
X( 5) =     410.00
X( 6) =     275.00
X( 7) =     151.00

```
