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 real 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 12 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 Parallelism and Performance

Not applicable.

9 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.

10 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 10.2). It would usually be more convenient in practice to keep the data file and the options file separate.

10.1 Program Text

```
!   H02CFF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.

    Module h02cffe_mod

!     H02CFF 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                                :: monit, qphx
!     .. Parameters ..
      Real (Kind=nag_wp), Parameter         :: cutoff = -1840000.0_nag_wp
      Integer, Parameter, Public            :: iset = 1, lintvr = 10,          &
                                               mdepth = 2000, nin = 5,         &
                                               ninopt = 7, nout = 6
    Contains
      Subroutine qphx(nstate,ncolh,x,hx)

!       Routine to compute H*x. (In this version of QPHX, the Hessian
!       matrix H is not referenced explicitly.)

!       .. Scalar Arguments ..
        Integer, Intent (In)                :: ncolh, nstate
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out)    :: hx(ncolh)
        Real (Kind=nag_wp), Intent (In)     :: x(ncolh)
!       .. Executable Statements ..
        hx(1) = 2.0_nag_wp*x(1)
        hx(2) = 2.0_nag_wp*x(2)
        hx(3) = 2.0_nag_wp*(x(3)+x(4))
        hx(4) = hx(3)
        hx(5) = 2.0_nag_wp*x(5)
        hx(6) = 2.0_nag_wp*(x(6)+x(7))
        hx(7) = hx(6)

        Return

      End Subroutine qphx
      Subroutine monit(intfnd,nodes,depth,obj,x,bstval,bstsol,bl,bu,n,halt, &
        count)

!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (Inout)  :: bstval
        Real (Kind=nag_wp), Intent (In)     :: obj
        Integer, Intent (Inout)             :: count
        Integer, Intent (In)                :: depth, intfnd, n, nodes
        Logical, Intent (Inout)             :: halt
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In)     :: bl(n), bstsol(n), bu(n), x(n)
!       .. Executable Statements ..
        If (intfnd==0) Then
          bstval = cutoff
        End If

        Return

      End Subroutine monit
    End Module h02cffe_mod
    Program h02cffe

!     H02CFF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: h02cef, h02cff, h02cgf, nag_wp, x04abf, x04acf,   &
```

```
                           x04baf
      Use h02cffe_mod, Only: iset, lintvr, mdepth, monit, nin, ninopt, nout,   &
                           qphx
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Character (*), Parameter             :: fname = 'h02cffe.opt'
!     .. Local Scalars ..
      Real (Kind=nag_wp)                   :: obj
      Integer                              :: i, icol, ifail, inform, iobj,    &
                                              jcol, leniz, lenz, m, miniz,     &
                                              minz, mode, n, ncolh, nname,     &
                                              nnz, ns, outchn, strtgy
      Character (200)                      :: rec
      Character (1)                        :: start
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable      :: a(:), bl(:), bu(:), clamda(:),   &
                                              xs(:), z(:)
      Integer, Allocatable                 :: ha(:), intvar(:), istate(:),     &
                                              iz(:), ka(:)
      Character (8), Allocatable           :: crname(:)
      Character (8)                        :: names(5)
!     .. Executable Statements ..
      Write (rec,99996) 'H02CFF Example Program Results'
      Call x04baf(nout,rec)

!     Skip heading in data file.
      Read (nin,*)

      Read (nin,*) n, m
      Read (nin,*) nnz, iobj, ncolh, start, nname
      Allocate (a(nnz),bl(n+m),bu(n+m),clamda(n+m),xs(n+m),ha(nnz), &
        intvar(lintvr),istate(n+m),ka(n+1),crname(nname))

      Read (nin,*) names(1:5)
      Read (nin,*) crname(1:nname)

!     Read the matrix A from data file. Set up KA.

      jcol = 1
      ka(jcol) = 1

      Do i = 1, nnz

!       Element ( HA( I ), ICOL ) is stored in A( I ).

        Read (nin,*) a(i), ha(i), icol

        If (icol==jcol+1) Then

!         Index in A of the start of the ICOL-th column equals I.

          ka(icol) = i
          jcol = icol
        Else If (icol>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.

          ka((jcol+1):(icol-1)) = i
          ka(icol) = i
          jcol = icol
        End If

      End Do

      ka(n+1) = nnz + 1

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

```
      Read (nin,*) istate(1:n)
      Read (nin,*) xs(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.

      outchn = nout

      Call x04abf(iset,outchn)

!     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 h02cff(ninopt,inform)

      If (inform/=0) Then
        Write (rec,99997) 'H02CFF terminated with INFORM = ', inform
        Call x04baf(nout,rec)
        Go To 100
      End If

      strtgy = 3
      intvar(1:7) = (/2,3,4,5,6,7,-1/)

      Call h02cgf('NoList')

      Call h02cgf('Print Level = 0')

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

      leniz = 1
      lenz = 1
      Allocate (iz(leniz),z(lenz))

      ifail = 1
      Call h02cef(n,m,nnz,iobj,ncolh,qphx,a,ha,ka,bl,bu,start,names,nname, &
        crname,ns,xs,intvar,lintvr,mdepth,istate,miniz,minz,obj,clamda,strtgy, &
        iz,leniz,z,lenz,monit,ifail)

      If (ifail/=14) Then
        Write (rec,99995) ifail
        Call x04baf(nout,rec)
      Else
        Deallocate (iz,z)

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

        ifail = 0
        Call h02cef(n,m,nnz,iobj,ncolh,qphx,a,ha,ka,bl,bu,start,names,nname, &
          crname,ns,xs,intvar,lintvr,mdepth,istate,miniz,minz,obj,clamda, &
          strtgy,iz,leniz,z,lenz,monit,ifail)

!       Print out the best integer solution found

        Write (rec,99999) obj
```

```
        Call x04baf(nout,rec)
        Call x04baf(nout,' Components are')

        Do i = 1, n
          Write (rec,99998) i, xs(i)
          Call x04baf(nout,rec)
        End Do

      End If

100   Continue

99999 Format (1X,'Optimal Integer Value is = ',E20.8)
99998 Format (1X,'X(',I3,') = ',F10.2)
99997 Format (A,I5)
99996 Format (1X,A)
99995 Format (1X,'** Workspace query in H02CEF exited with IFAIL = ',I0)
    End Program h02cffe
```

10.2 Program Data

```
Begin
  Iteration Limit = 125 * (Default = 75)
  Print Level = 1      * (Default = 10)
End
```

```
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                                                :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 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
```

10.3 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
```