H02BZF – NAG Fortran Library Routine Document

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.

Warning. The specification of the parameter LIWORK changed at Mark 16: the minimum dimension of the array IWORK has been increased by N + 3.

1 Purpose

H02BZF extracts more information associated with the solution of an integer programming problem computed by H02BBF.

2 Specification

```
 SUBROUTINE H02BZF(N, M, BL, BU, CLAMDA, ISTATE, IWORK, LIWORK,
1                  RWORK, LRWORK, IFAIL)
 INTEGER           N, M, ISTATE(N+M), IWORK(LIWORK), LIWORK,
1                  LRWORK, IFAIL
 real              BL(N+M), BU(N+M), CLAMDA(N+M), RWORK(LRWORK)
```

3 Description

H02BZF extracts the following information associated with the solution of an integer programming problem computed by H02BBF. The upper and lower bounds used for the solution, the Lagrange multipliers (costs), and the status of the variables at the solution.

In the branch and bound method employed by H02BBF, the arrays BL and BU are used to impose restrictions on the values of the integer variables in each sub-problem. That is, if the variable x_j is restricted to take value v_j in a particular sub-problem, then BL(j) = BU(j) = v_j is set in the sub-problem. Thus, on exit from this routine, some of the elements of BL and BU which correspond to integer variables may contain these imposed values, rather than those originally supplied to H02BBF.

4 References

None.

5 Parameters

1: N — INTEGER *Input*

On entry: this **must** be the same parameter N as supplied to H02BBF.

Constraint: N > 0.

2: M — INTEGER *Input*

On entry: this **must** be the same parameter M as supplied to H02BBF.

Constraint: M ≥ 0.

3: BL(N+M) — ***real*** array *Output*

On exit: if H02BBF exits with IFAIL = 0, 7 or 9, the values in the array BL contain the lower bounds imposed on the integer solution for all the constraints. The first N elements contain the lower bounds on the variables, and the next M elements contain the lower bounds for the general linear constraints (if any).

4: BU(N+M) — ***real*** array *Output*

On exit: if H02BBF exits with IFAIL = 0, 7 or 9, the values in the array BU contain the upper bounds imposed on the integer solution for all the constraints. The first N elements contain the upper bounds on the variables, and the next M elements contain the upper bounds for the general linear constraints (if any).

5: CLAMDA(N+M) — ***real*** array *Output*

On exit: if H02BBF exits with IFAIL = 0, 7 or 9, the values in the array CLAMDA contain the values of the Lagrange multipliers for each constraint with respect to the current working set. The first N elements contain the multipliers (reduced costs) for the bound constraints on the variables, and the next M elements contain the multipliers (shadow costs) for the general linear constraints (if any).

6: ISTATE(N+M) — INTEGER array *Output*

On exit: if H02BBF exits with IFAIL = 0, 7 or 9, the values in the array ISTATE indicate the status of the constraints in the working set at an integer solution. Otherwise, ISTATE indicates the composition of the working set at the final iterate. The significance of each possible value of ISTATE(j) is as follows.

ISTATE(j)	**Meaning**
-2	The constraint violates its lower bound by more than TOLFES (the feasibility tolerance, see H02BBF).
-1	The constraint violates its upper bound by more than TOLFES.
0	The constraint is satisfied to within TOLFES, but is not in the working set.
1	This inequality constraint is included in the working set at its lower bound.
2	This inequality constraint is included in the working set at its upper bound.
3	This constraint is included in the working set as an equality. This value of ISTATE can occur only when BL(j) = BU(j).
4	This corresponds to an integer solution being declared with x_j being temporarily fixed at its current value. This value of ISTATE can occur only when IFAIL = 0, 7 or 9 on exit from H02BBF.

7: IWORK(LIWORK) — INTEGER array *Workspace*

This **must** be the same parameter IWORK as supplied to H02BBF. It is used to pass information from H02BBF to H02BZF and therefore the contents of this array **must not** be changed before calling H02BZF.

8: LIWORK — INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which H02BZF is called.

9: RWORK(LRWORK) — ***real*** array *Workspace*

This **must** be the same parameter RWORK as supplied to H02BBF. It is used to pass information from H02BBF to H02BZF and therefore the contents of this array **must not** be changed before calling H02BZF.

10: LRWORK — INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which H02BZF is called.

11: IFAIL — INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

On exit: IFAIL = 0 unless the routine detects an error (see Section 6).

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 detected by the routine:

IFAIL = 1

On entry, N ≤ 0,
or M < 0.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

One of the applications of integer programming is to the so-called diet problem. Given the nutritional content of a selection of foods, the cost of each food, the amount available of each food and the consumer's minimum daily energy requirements, the problem is to find the cheapest combination. This gives rise to the following problem:

minimize

$$c^T x$$

subject to

$$Ax \geq b, 0 \leq x \leq u,$$

where

$$c = (3\ 24\ 13\ 9\ 20\ 19)^T, x = (x_1, x_2, x_3, x_4, x_5, x_6)^T$$

is integer,

$$A = \begin{pmatrix} 110 & 205 & 160 & 160 & 420 & 260 \\ 4 & 32 & 13 & 8 & 4 & 14 \\ 2 & 12 & 54 & 285 & 22 & 80 \end{pmatrix}, \quad b = \begin{pmatrix} 2000 \\ 55 \\ 800 \end{pmatrix}$$

and $u = (4\ 3\ 2\ 8\ 2\ 2)^T$.

The rows of A correspond to energy, protein and calcium and the columns of A correspond to oatmeal, chicken, eggs, milk, pie and bacon respectively.

The following program solves the above problem to obtain the optimal integer solution and then examines the effect of increasing the energy required to 2200 units.

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.

```
*     H02BZF Example Program Text
*     Mark 16 Revised. NAG Copyright 1993.
*     .. Parameters ..
      INTEGER          NIN, NOUT
      PARAMETER        (NIN=5,NOUT=6)
      INTEGER          NMAX, MMAX
      PARAMETER        (NMAX=10,MMAX=10)
      INTEGER          LDA
      PARAMETER        (LDA=MMAX)
      INTEGER          LIWORK, LRWORK
      PARAMETER        (LIWORK=1000,LRWORK=1000)
*     .. Local Scalars ..
      real             BIGBND, INIVAL, OBJMIP, TOLFES, TOLIV
      INTEGER          I, IFAIL, INTFST, ITMAX, J, M, MAXDPT, MAXNOD,
     +                 MSGLVL, N
*     .. Local Arrays ..
      real             A(LDA,NMAX), BL(NMAX+MMAX), BU(NMAX+MMAX),
     +                 CLAMDA(NMAX+MMAX), CVEC(NMAX), RWORK(LRWORK),
     +                 X(NMAX)
      INTEGER          INTVAR(NMAX), ISTATE(NMAX+MMAX), IWORK(LIWORK)
      CHARACTER*8      NAMES(NMAX+MMAX)
*     .. External Subroutines ..
      EXTERNAL         H02BBF, H02BZF, OUTSOL
*     .. Executable Statements ..
      WRITE (NOUT,*) 'H02BZF Example Program Results'
*     Skip heading in data file
      READ (NIN,*)
      READ (NIN,*) N, M
      IF (N.LE.NMAX .AND. M.LE.MMAX) THEN
*
*        Read ITMAX, MSGLVL, MAXNOD, INTFST, MAXDPT, TOLFES, TOLIV,
*        CVEC, A, BIGBND, BL, BU, INTVAR and X from data file
*
         READ (NIN,*) ITMAX, MSGLVL
         READ (NIN,*) MAXNOD
         READ (NIN,*) INTFST, MAXDPT
         READ (NIN,*) TOLFES, TOLIV
         READ (NIN,*) (CVEC(J),J=1,N)
         READ (NIN,*) (NAMES(J),(A(I,J),I=1,M),J=1,N)
         READ (NIN,*) BIGBND
         READ (NIN,*) (BL(I),I=1,N)
         READ (NIN,*) (NAMES(N+I),BL(N+I),I=1,M)
         READ (NIN,*) (BU(I),I=1,N+M)
         READ (NIN,*) (INTVAR(I),I=1,N)
         READ (NIN,*) (X(I),I=1,N)
*
*        Solve the IP problem using H02BBF
*
         IFAIL = -1
*
         CALL H02BBF(ITMAX,MSGLVL,N,M,A,LDA,BL,BU,INTVAR,CVEC,MAXNOD,
     +               INTFST,MAXDPT,TOLIV,TOLFES,BIGBND,X,OBJMIP,IWORK,
     +               LIWORK,RWORK,LRWORK,IFAIL)
*
         IF (IFAIL.EQ.0 .OR. IFAIL.EQ.7 .OR. IFAIL.EQ.9) THEN
            WRITE (NOUT,99999) 'IP objective value = ', OBJMIP
*
*           Get information about the solution
*
            IFAIL = 0
*
            CALL H02BZF(N,M,BL,BU,CLAMDA,ISTATE,IWORK,LIWORK,RWORK,
     +                  LRWORK,IFAIL)
*
*           Print the solution
```

```
*
            CALL OUTSOL(N,M,A,LDA,BL,BU,X,ISTATE,CLAMDA,BIGBND,NAMES,
     +                  NOUT)
*
*           Increase the energy requirements and solve the modified IP
*           problem using the current IP solution as the starting point
*
            INIVAL = BL(N+1)
            READ (NIN,*) BL(N+1)
            WRITE (NOUT,99998) ’Increase the energy requirements from’,
     +        INIVAL, ’to’, BL(N+1)
*
            IFAIL = -1
*
            CALL H02BBF(ITMAX,MSGLVL,N,M,A,LDA,BL,BU,INTVAR,CVEC,MAXNOD,
     +                  INTFST,MAXDPT,TOLIV,TOLFES,BIGBND,X,OBJMIP,
     +                  IWORK,LIWORK,RWORK,LRWORK,IFAIL)
*
            IF (IFAIL.EQ.0 .OR. IFAIL.EQ.7 .OR. IFAIL.EQ.9) THEN
               WRITE (NOUT,99999) ’IP objective value = ’, OBJMIP
*
*              Get information about the solution
*
               IFAIL = 0
*
               CALL H02BZF(N,M,BL,BU,CLAMDA,ISTATE,IWORK,LIWORK,RWORK,
     +                     LRWORK,IFAIL)
*
*              Print the solution
*
               CALL OUTSOL(N,M,A,LDA,BL,BU,X,ISTATE,CLAMDA,BIGBND,NAMES,
     +                     NOUT)
*
            ELSE
               WRITE (NOUT,99997) ’ H02BBF terminated with IFAIL = ’,
     +           IFAIL
            END IF
         ELSE
            WRITE (NOUT,99997) ’ H02BBF terminated with IFAIL = ’, IFAIL
         END IF
      END IF
      STOP
*
99999 FORMAT (//1X,A,1P,G16.4)
99998 FORMAT (//1X,A,2X,1P,G10.4,2X,A,2X,1P,G10.4)
99997 FORMAT (1X,A,I3)
      END
      SUBROUTINE OUTSOL(N,M,A,LDA,BL,BU,X,ISTATE,CLAMDA,BIGBND,NAMES,
     +                  NOUT)
*     .. Scalar Arguments ..
      real              BIGBND
      INTEGER           LDA, M, N, NOUT
*     .. Array Arguments ..
      real              A(LDA,*), BL(N+M), BU(N+M), CLAMDA(N+M), X(N)
      INTEGER           ISTATE(N+M)
      CHARACTER*8       NAMES(N+M)
*     .. Local Scalars ..
      real              B1, B2, RES, RES2, V, WLAM
      INTEGER           IS, J, K
      CHARACTER*80      REC
*     .. Local Arrays ..
      CHARACTER*2       LSTATE(-2:4)
*     .. External Functions ..
      real              sdot
      EXTERNAL          sdot
*     .. Intrinsic Functions ..
      INTRINSIC         ABS
*     .. Data statements ..
      DATA              LSTATE(-2)/'--'/, LSTATE(-1)/'++'/,
     +                  LSTATE(0)/'FR'/, LSTATE(1)/'LL'/,
     +                  LSTATE(2)/'UL'/, LSTATE(3)/'EQ'/,
     +                  LSTATE(4)/'TF'/
*     .. Executable Statements ..
*
      WRITE (NOUT,99999)
      DO 20 J = 1, N + M
         B1 = BL(J)
         B2 = BU(J)
         WLAM = CLAMDA(J)
         IS = ISTATE(J)
         IF (J.LE.N) THEN
*           The variables  x.
            K = J
            V = X(J)
         ELSE
*           The linear constraints  A*x.
            IF (J.EQ.N+1) WRITE (NOUT,99998)
            K = J - N
            V = sdot(N,A(K,1),LDA,X,1)
         END IF
*
*        Print a line for the j-th variable or constraint.
*
         RES = V - B1
         RES2 = B2 - V
         IF (ABS(RES).GT.ABS(RES2)) RES = RES2
         WRITE (REC,99997) NAMES(J), LSTATE(IS), V, B1, B2, WLAM, RES
         IF (B1.LE.-BIGBND) REC(29:42) = '     None     '
         IF (B2.GE.BIGBND) REC(43:56) = '     None     '
         WRITE (NOUT,'(A)') REC
   20 CONTINUE
      RETURN
*
99999 FORMAT (//1X,'Varbl',3X,'State',5X,'Value',5X,'Lower Bound',3X,
     +      'Upper Bound',4X,'Lagr Mult',3X,'Residual',/)
99998 FORMAT (//1X,'L Con',3X,'State',5X,'Value',5X,'Lower Bound',3X,
     +      'Upper Bound',4X,'Lagr Mult',3X,'Residual',/)
99997 FORMAT (1X,A8,2X,A2,1X,1P,3G14.4,1P,G12.4,1P,G12.4)
      END
```

9.2 Program Data

```
H02BZF Example Program Data
 6  3                                           :Values of N and M
 0  0                                           :Values of ITMAX and MSGLVL
 0                                              :Value of MAXNOD
```

```
0  9                                         :Values of INTFST and MAXDPT
0.0  0.0                                     :Values of TOLFES and TOLIV
3.0  24.0  13.0  9.0  20.0  19.0             :End of CVEC
'Oatmeal'    110.0    4.0      2.0
'Chicken'    205.0   32.0     12.0
'Eggs'       160.0   13.0     54.0
'Milk'       160.0    8.0    285.0
'Pie'        420.0    4.0     22.0
'Bacon'      260.0   14.0     80.0                    :End of matrix A
1.0E+20                                               :Value of BIGBND
0.0  0.0  0.0  0.0  0.0  0.0
'Energy'  2000.0  'Protein'  55.0  'Calcium'  800.0     :End of BL
4.0  3.0  2.0  8.0  2.0  2.0  1.0E+20  1.0E+20  1.0E+20 :End of BU
1    1    1    1    1    1                            :End of INTVAR
0.0  0.0  0.0  0.0  0.0  0.0                          :End of X
2200.0                                      :Change 'Energy' in RHS
```

9.3 Program Results

```
H02BZF Example Program Results


IP objective value =        97.00


Varbl   State     Value     Lower Bound   Upper Bound    Lagr Mult   Residual

Oatmeal   EQ      4.000         4.000         4.000        3.000      0.0000E+00
Chicken   LL     0.0000E+00    0.0000E+00     3.000        24.00      0.0000E+00
Eggs      LL     0.0000E+00    0.0000E+00     2.000        13.00      0.0000E+00
Milk      LL      5.000         5.000         8.000        9.000      0.0000E+00
Pie       EQ      2.000         2.000         2.000        20.00      0.0000E+00
Bacon     LL     0.0000E+00    0.0000E+00     2.000        19.00      0.0000E+00


L Con   State     Value     Lower Bound   Upper Bound    Lagr Mult   Residual

Energy    FR      2080.         2000.         None       0.0000E+00   80.00
Protein   FR      64.00         55.00         None       0.0000E+00   9.000
Calcium   FR      1477.         800.0         None       0.0000E+00   677.0


Increase the energy requirements from   2000.      to   2200.


IP objective value =        106.0


Varbl   State     Value     Lower Bound   Upper Bound    Lagr Mult   Residual

Oatmeal   EQ      4.000         4.000         4.000        3.000      0.0000E+00
Chicken   LL     0.0000E+00    0.0000E+00     3.000        24.00      0.0000E+00
Eggs      LL     0.0000E+00    0.0000E+00     2.000        13.00      0.0000E+00
Milk      LL      6.000         6.000         8.000        9.000      0.0000E+00
Pie       EQ      2.000         2.000         2.000        20.00      0.0000E+00
Bacon     LL     0.0000E+00    0.0000E+00     2.000        19.00      0.0000E+00
```

```
 L Con    State      Value      Lower Bound   Upper Bound    Lagr Mult   Residual

 Energy    FR       2240.         2200.          None        0.0000E+00   40.00
 Protein   FR       72.00         55.00          None        0.0000E+00   17.00
 Calcium   FR       1762.         800.0          None        0.0000E+00   962.0
```