NAG Fortran Library Routine Document

E05JCF

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

E05JCF may be used to supply optional parameters to E05JBF from an external file. The initialization routine E05JAF **must** have been called before calling E05JCF.

2 Specification

```
SUBROUTINE E05JCF (IOPTS, RW, LENRW, IFAIL)
INTEGER             IOPTS, LENRW, IFAIL
double precision    RW(LENRW)
```

3 Description

E05JCF may be used to supply values for optional parameters to E05JBF. E05JCF 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

```
Static Limit = 100
```

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.

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
   Static Limit = 500
End
```

Optional parameter settings are preserved following a call to E05JBF and so the keyword **Defaults** is provided to allow you to reset all the optional parameters to their default values before a subsequent call to E05JBF.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 11 of the document for E05JBF.

4 References

None.

5 Parameters

1: IOPTS – INTEGER *Input*

On entry: the unit number of the option file to be read.

Constraint: IOPTS is a valid unit open for reading

2: RW(LENRW) – ***double precision*** array *Communication Array*

The array RW **must not** be altered between calls to any of the routines E05JBF, E05JCF, E05JDF, E05JEF, E05JFF, E05JGF, E05JHF, E05JKF and E05JLF.

3: LENRW – INTEGER *Input*

On entry: the dimension of the array RW as declared in the (sub)program from which E05JCF is called.

4: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you 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, if you are 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 E05JAF has not been called.

IFAIL $= 2$

Could not read options file on unit IOPTS. This may be due to:

(a) IOPTS is not a valid unit number;

(b) a file is not associated with unit IOPTS, 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.

IFAIL $= 3$

The option file contains an option-name/numerical-value pair where the numerical value is outside the valid range expected by the option name. See Section 11 of the document for E05JBF for allowable values of the optional parameters.

IFAIL $= 4$

Data-type discrepancy: an integer option was attempted to be given a ***double precision*** value.

IFAIL $= 5$

The value to be set could not be parsed. Check that the value specifies a valid INTEGER or ***double precision*** value.

IFAIL $= 6$

The option-name contained the wrong number of word ‘tokens’, so could not be recognised by the routine.

7 Accuracy

Not applicable.

8 Further Comments

E05JDF, E05JEF, E05JFF or E05JGF may also be used to supply optional parameters to E05JBF.

9 Example

This example finds a global minimum of the ‘six-hump camelback’ function in two dimensions

$$F(\mathbf{x}) = \left(4 - 2.1x_1^2 + x_1^4/3\right)x_1^2 + x_1x_2 + \left(-4 + 4x_2^2\right)x_2^2$$

on the box $[-3,3] \times [-2,2]$.

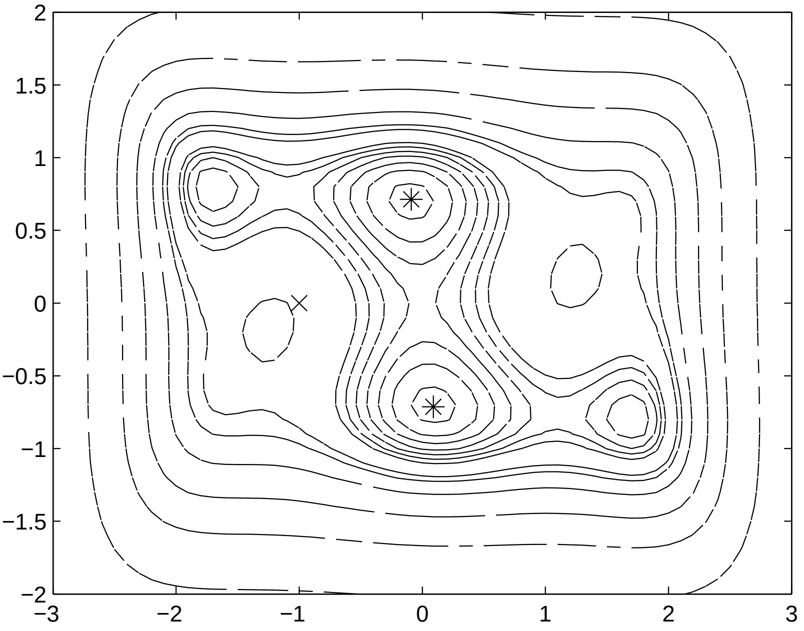

Figure 1
The camelback function ***F***.
The global minima are denoted by $*$, while our start point is labeled with $\times$

The function F has four local minima and two global minima in the given box. The global minima are located at $\pm(0.0898, -0.7126)$, where the function value is -1.0316.

By specifying an initialization list via INLIST, NUMPTS and INITPT we can start E05JBF looking close to one of the local minima and check that it really does move away from that point to one of the global minima.

More precisely, we chose $(-1, 0)$ as our initial point (marked with a cross in Figure 1), and let the initialization list be

$$\begin{pmatrix} -3 & -1 & 3 \\ -2 & 0 & 2 \end{pmatrix}.$$

This example solves the optimization problem using some of the optional parameters described in Section 11 of the document for E05JBF.

9.1 Program Text

```
*     E05JCF Example Program Text
*     Mark 22 Release. NAG Copyright 2007.
      IMPLICIT          NONE
*     .. Parameters ..
      INTEGER           NIN, NOUT
      PARAMETER         (NIN=5,NOUT=6)
      INTEGER           NMAX, LDIN, NINMAX, LENRW
      PARAMETER         (NMAX=2,LDIN=NMAX,NINMAX=192,LENRW=100)
*     .. Local Scalars ..
      DOUBLE PRECISION  INFBND, OBJ
      INTEGER           I, IBDCHK, IBOUND, IFAIL, IINIT, J, N, NF, NINIT,
     +                  STCLIM
      CHARACTER*8       LCSRCH
*     .. Local Arrays ..
      DOUBLE PRECISION  BL(NMAX), BU(NMAX), INLIST(LDIN,NINMAX),
     +                  RUSER(1), RW(LENRW), X(NMAX)
      INTEGER           INITPT(NMAX), IUSER(1), NUMPTS(NMAX)
*     .. External Subroutines ..
      EXTERNAL          E05JAF, E05JBF, E05JCF, E05JDF, E05JEF, E05JFF,
     +                  E05JGF, E05JJF, E05JKF, E05JLF, MONIT, OBJFUN
*     .. External Functions ..
      INTEGER           E05JHF
      EXTERNAL          E05JHF
*     .. Executable Statements ..
      CONTINUE
*
      WRITE (NOUT,*) 'E05JCF Example Program Results'
*
*     This program demonstrates the use of routines to set and get
*     values of optional parameters associated with E05JBF
*
*     Skip heading in data file
*
      READ (NIN,*)
*
*     Read N and NINIT from data file
*
      READ (NIN,*) N, NINIT
*
      IF (N.LE.NMAX .AND. NINIT.LE.NINMAX) THEN
*
*        Read IBOUND, BL, and BU from data file
*
         READ (NIN,*) IBOUND
*
         IF (IBOUND.EQ.0) THEN
*
*           Read in the whole of each bound
*
            READ (NIN,*) (BL(I),I=1,N)
            READ (NIN,*) (BU(I),I=1,N)
*
         ELSE IF (IBOUND.EQ.3) THEN
*
*           Bounds are uniform: read in only the first entry of each
*
            READ (NIN,*) BL(1)
            READ (NIN,*) BU(1)
*
         END IF
*
*        Read in IINIT (and INLIST, NUMPTS and INITPT if necessary)
*        from data file
*
         READ (NIN,*) IINIT
```

```
*
         IF (IINIT.EQ.3) THEN
*
*           User is specifying the initialization list
*
            READ (NIN,*) ((INLIST(I,J),J=1,NINIT),I=1,N)
            READ (NIN,*) (NUMPTS(I),I=1,N)
            READ (NIN,*) (INITPT(I),I=1,N)
*
         END IF
*
*        Call E05JAF to initialize E05JBF
*
         IFAIL = -1
         CALL E05JAF(N,RW,LENRW,IFAIL)
*
*        Use E05JCF to read some options from the end of the data file
*
         CALL E05JCF(NIN,RW,LENRW,IFAIL)
         WRITE (NOUT,*)
*
*        Use E05JKF to find the value of the integer-valued option
*        'Function Evaluations Limit'
*
         CALL E05JKF('Function Evaluations Limit',NF,RW,LENRW,IFAIL)
         WRITE (NOUT,99999) NF
*
*        Use E05JFF to set the value of the integer-valued option
*        'Static Limit'
*
         STCLIM = 4*N
         CALL E05JFF('Static Limit',STCLIM,RW,LENRW,IFAIL)
*
*        Use E05JHF to determine whether the real-valued option
*        'Infinite Bound Size' has been set by us (in which case
*        E05JHF returns 1) or whether is holds its default value
*        (E05JHF returns 0)
*
         IBDCHK = E05JHF('Infinite Bound Size',RW,LENRW,IFAIL)
*
         IF (IBDCHK.EQ.1) THEN
            WRITE (NOUT,99998)
         ELSE IF (IBDCHK.EQ.0) THEN
            WRITE (NOUT,99997)
         END IF
*
*        Use E05JLF to get the value of the real-valued option
*        'Infinite Bound Size'
*
         CALL E05JLF('Infinite Bound Size',INFBND,RW,LENRW,IFAIL)
         WRITE (NOUT,99996) INFBND
*
*        Use E05JGF to increase the value of the real-valued option
*        'Infinite Bound Size' tenfold
*
         INFBND = 1.0D1*INFBND
         CALL E05JGF('Infinite Bound Size',INFBND,RW,LENRW,IFAIL)
*
*        Use E05JDF to set the option 'Local Searches Limit = 40'
*
         CALL E05JDF('Local Searches Limit = 40',RW,LENRW,IFAIL)
*
*        Use E05JEF to set the option 'Local Searches' to 'On'
*
         LCSRCH = 'On'
         CALL E05JEF('Local Searches',LCSRCH,RW,LENRW,IFAIL)
*
*        Get that value of 'Local Searches' using E05JJF
*
         CALL E05JJF('Local Searches',LCSRCH,RW,LENRW,IFAIL)
         WRITE (NOUT,99995) LCSRCH
```

```
*
*        Solve the problem.
*
         IFAIL = -1
         CALL E05JBF(N,OBJFUN,MONIT,IBOUND,IINIT,BL,BU,NINIT,INLIST,
     +               LDIN,NUMPTS,INITPT,X,OBJ,RW,LENRW,IUSER,RUSER,
     +               IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99994) IFAIL
*
         IF (IFAIL.EQ.0) THEN
            WRITE (NOUT,99993) OBJ
            WRITE (NOUT,99992) (X(I),I=1,N)
         END IF
*
      END IF
*
      STOP
*
99999 FORMAT (1X,'Option ''Function Evaluations Limit'' has the value ',
     +       I6,'.')
99998 FORMAT (1X,'Option ''Infinite Bound Size'' has been set by us.')
99997 FORMAT (1X,'Option ''Infinite Bound Size'' holds its default ',
     +       'value.')
99996 FORMAT (1X,'Option ''Infinite Bound Size'' has the value ',E13.5,
     +       '.')
99995 FORMAT (1X,'Option ''Local Searches'' has the value ''',A8,'''.')
99994 FORMAT (1X,'On exit from E05JBF, IFAIL =',I5)
99993 FORMAT (1X,'Final objective value =',F11.5)
99992 FORMAT (1X,'Global optimum X =',2F9.5)
      END

      SUBROUTINE OBJFUN(N,X,F,NSTATE,IUSER,RUSER,INFORM)
*
*     Routine to evaluate objective function.
*     Mark 22 Release. NAG Copyright 2007.
*
*     .. Parameters ..
      DOUBLE PRECISION  FOUR, THREE
      INTEGER           NOUT
      PARAMETER         (FOUR=4.0D0,THREE=3.0D0,NOUT=6)
*     .. Scalar Arguments ..
      DOUBLE PRECISION  F
      INTEGER           INFORM, N, NSTATE
*     .. Array Arguments ..
      DOUBLE PRECISION  RUSER(*), X(N)
      INTEGER           IUSER(*)
*     .. Local Scalars ..
      DOUBLE PRECISION  X1, X2
*     .. Executable Statements ..
      CONTINUE
*
      INFORM = 0
*
      IF (INFORM.GE.0) THEN
*
*        If INFORM >= 0 then we're prepared to evaluate OBJFUN
*        at the current X
*
         IF (NSTATE.EQ.1) THEN
*
*           This is the first call to OBJFUN
*
            WRITE (NOUT,*)
            WRITE (NOUT,99999)
         END IF
*
         X1 = X(1)
         X2 = X(2)
*
```

```
            F = (FOUR-2.1D0*X1**2+X1**4/THREE)*X1**2 + X1*X2 +
     +          (-FOUR+FOUR*X2**2)*X2**2
      END IF
*
      RETURN
*
99999 FORMAT (1X,'(OBJFUN was just called for the first time)')
      END
*
      SUBROUTINE MONIT(N,NCALL,XBEST,ICOUNT,NINIT,INLIST,LDIN,INITPT,
     +                 NBASKT,XBASKT,NSTATE,IUSER,RUSER,INFORM)
*
*     Monitoring routine.
*     Mark 22 Release. NAG Copyright 2007.
*
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Scalar Arguments ..
      INTEGER          INFORM, LDIN, N, NBASKT, NCALL, NINIT, NSTATE
*     .. Array Arguments ..
      DOUBLE PRECISION INLIST(LDIN,NINIT), RUSER(*), XBASKT(N,NBASKT),
     +                 XBEST(N)
      INTEGER          ICOUNT(6), INITPT(N), IUSER(*)
*     .. Local Scalars ..
      INTEGER          I, J
*     .. Executable Statements ..
      CONTINUE
*
      INFORM = 0
*
      IF (INFORM.GE.0) THEN
*
*        We are going to allow the iterations to continue
*
         IF (NSTATE.EQ.0 .OR. NSTATE.EQ.1) THEN
*
*           When NSTATE.EQ.1, MONIT is called for the first time. When
*           NSTATE.EQ.0, MONIT is called for the first AND last time.
*           Display a welcome message
*
            WRITE (NOUT,*)
            WRITE (NOUT,99999)
            WRITE (NOUT,*)
         END IF
*
         IF (NSTATE.LE.0) THEN
*
*           MONIT is called for the last time.
*
            WRITE (NOUT,99998) ICOUNT(1)
            WRITE (NOUT,99997) NCALL
            WRITE (NOUT,99996) ICOUNT(2)
            WRITE (NOUT,99995) ICOUNT(3)
            WRITE (NOUT,99994) ICOUNT(4)
            WRITE (NOUT,99993) ICOUNT(5)
            WRITE (NOUT,99992) ICOUNT(6)
            WRITE (NOUT,99991) NBASKT
            WRITE (NOUT,99990)
*
            DO 20 I = 1, N
               WRITE (NOUT,99989) I, (XBASKT(I,J),J=1,NBASKT)
   20       CONTINUE
*
            WRITE (NOUT,*)
            WRITE (NOUT,99988)
            WRITE (NOUT,*)
         END IF
*
      END IF
*
```

```
      RETURN
*
99999 FORMAT (1X,'*** Begin monitoring information ***')
99998 FORMAT (1X,'Total subboxes =',I5)
99997 FORMAT (1X,'Total function evaluations =',I5)
99996 FORMAT (1X,'Total function evaluations used in local search =',I5)
99995 FORMAT (1X,'Total points used in local search =',I5)
99994 FORMAT (1X,'Total sweeps through levels =',I5)
99993 FORMAT (1X,'Total splits by init. list =',I5)
99992 FORMAT (1X,'Lowest level with nonsplit boxes =',I5)
99991 FORMAT (1X,'Number of candidate minima in the ''shopping basket',
     +       ''' =',I5)
99990 FORMAT (1X,'Shopping basket:')
99989 FORMAT (1X,'XBASKT(',I3,',:) =',(6F9.5))
99988 FORMAT (1X,'*** End monitoring information ***')
      END
```

9.2 Program Data

```
E05JCF Example Program Data
  2   3                                                    : N and NINIT
  0                                                        : IBOUND
  -3.0   -2.0                                              : Lower bounds BL
  3.0   2.0                                                : Upper bounds BU
  3                                                        : IINIT
  -3.0   -1.0   3.0  -2.0   0.0   2.0                      : Matrix INLIST
  3   3   3                                                : NUMPTS
  2   2   2                                                  : Initial-point pointer
INITPT
Begin example options file
* Comment lines like this begin with an asterisk
* Set the maximum number of function evaluations
Function Evaluations Limit = 100000
* Set the local search termination tolerance
Local Searches Tolerance = 1.0D-10
* Set the maximum number of times a given box may be split
Splits Limit = 20
End
```

9.3 Program Results

```
 E05JCF Example Program Results

 Option 'Function Evaluations Limit' has the value 100000.
 Option 'Infinite Bound Size' holds its default value.
 Option 'Infinite Bound Size' has the value   0.10000E+21.
 Option 'Local Searches' has the value 'ON      '.

 (OBJFUN was just called for the first time)

 *** Begin monitoring information ***

 Total subboxes =  136
 Total function evaluations =  166
 Total function evaluations used in local search =   98
 Total points used in local search =    9
 Total sweeps through levels =    9
 Total splits by init. list =    3
 Lowest level with nonsplit boxes =    5
 Number of candidate minima in the 'shopping basket' =    2
 Shopping basket:
 XBASKT( 1,:) =  0.08984 -0.08984
 XBASKT( 2,:) = -0.71266  0.71266

 *** End monitoring information ***


 On exit from E05JBF, IFAIL =    0
 Final objective value =   -1.03163
 Global optimum X =  0.08984 -0.71266
```