

NAG 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, COMM, LCOMM, IFAIL)
INTEGER          IOPTS, LCOMM, IFAIL
double precision COMM(LCOMM)
```

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 optional parameter 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 optional parameter 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.

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 optional parameters 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 symbolic names and default values is given in Section 11 in 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: COMM(LCOMM) – *double precision* array *Communication Array*
On exit: COMM **must not** be altered between calls to any of the routines E05JBF, E05JCF, E05JDF, E05JEF, E05JFF, E05JGF, E05JHF, E05JJF, E05JKF and E05JLF.
- 3: LCOMM – INTEGER *Input*
On entry: the dimension of the array COMM as declared in the (sub)program from which E05JCF is called.
Constraint: LCOMM ≥ 100 .
- 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 Section 3.3 in the Essential Introduction 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

On entry, LCOMM < 100,
 or the initialization routine E05JAF has not been called.

IFAIL = 2

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

- (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 are invalid. The correct number of word 'tokens' are present, but a keyword or keyword combination is not valid;
- (d) Begin was found, but end-of-file was found before End was found;
- (e) end-of-file was found before Begin was found.

In cases (c) and (d) all optional parameters that were set from the file before the error was encountered will remain set on exit.

IFAIL = 3

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

IFAIL = 4

Data-type discrepancy: either a real value was associated with an integer optional parameter or a non-numeric value was associated with a numeric optional parameter.

IFAIL = 5

A supplied string associated with a numeric value could not be parsed. Check that all such strings specify valid integer or real values.

IFAIL = 6

A supplied optional-parameter name contained the wrong number of word ‘tokens’, so could not be recognized 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 the global minimum of the ‘peaks’ function in two dimensions

$$F(x, y) = 3(1 - x)^2 \exp(-x^2 - (y + 1)^2) - 10\left(\frac{x}{5} - x^3 - y^5\right) \exp(-x^2 - y^2) - \frac{1}{3} \exp(-(x + 1)^2 - y^2)$$

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

The function F has several local minima and one global minimum in the given box. The global minimum is approximately located at $(0.23, -1.63)$, where the function value is approximately -6.55 .

By specifying an initialization list via LIST, 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 choose $(-1, 0)$ as our initial point (see Section 9.3), and let the initialization list be

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

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

9.1 Program Text

```
*      E05JCF Example Program Text
*      Mark 22 Release. NAG Copyright 2007.
*
*      This program demonstrates the use of routines to set and get
*      values of optional parameters associated with E05JBF
*
*      .. Implicit None Statement ..
      IMPLICIT      NONE
*
*      .. Parameters ..
      INTEGER
      PARAMETER
      (NIN=5,NOUT=6)
      INTEGER
      N, NINMAX, LCOMM
      PARAMETER
      (N=2,NINMAX=192,LCOMM=100)
*
*      .. Local Scalars ..
      DOUBLE PRECISION INFBNB, OBJ
      INTEGER
      I, IBDCHK, IBOUND, IFAIL, IINIT, J, NF, SDLIST,
      +
      STCLIM
```

```

LOGICAL          LMOK, LPLLOT
CHARACTER*3      LCSRCH
*
.. Local Arrays ..
DOUBLE PRECISION BL(N), BU(N), COMM(LCOMM), LIST(N,NINMAX),
+               RUSER(1), X(N)
INTEGER          INITPT(N), IUSER(2), NUMPTS(N)
*
.. External Subroutines ..
EXTERNAL         E05JAF, E05JBF, E05JCF, E05JDF, E05JEF, E05JFF,
+               E05JGF, E05JJF, E05JKF, E05JLF, MONIT, OBJFUN
*
.. External Functions ..
INTEGER          E05JHF
LOGICAL          A00ACF
EXTERNAL         E05JHF, A00ACF
*
.. Executable Statements ..
CONTINUE

*
WRITE (NOUT,*) 'E05JCF Example Program Results'
*
Skip heading in data file
*
READ (NIN,*)

*
Read SDLIST from data file
*
READ (NIN,*) SDLIST

*
LMOK = A00ACF()

*
IF ( .NOT. LMOK) THEN
  WRITE (NOUT,*)
  WRITE (NOUT,*) ' ** A valid licence key was not found'
ELSE IF (SDLIST.GT.NINMAX) THEN
  WRITE (NOUT,*)
  WRITE (NOUT,*) ' ** SDLIST is too large'
ELSE

*
  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 LIST, NUMPTS and INITPT if necessary)
  from data file
*
  READ (NIN,*) IINIT

*
  IF (IINIT.EQ.3) THEN

*
    User is specifying the initialization list
*
    READ (NIN,*) (NUMPTS(I),I=1,N)
    READ (NIN,*) ((LIST(I,J),J=1,NUMPTS(I)),I=1,N)
    READ (NIN,*) (INITPT(I),I=1,N)
  END IF

*
  Read LPLLOT. Its value determines whether MONIT displays

```

```

*      information on the current search box
*
*      READ (NIN,*) LPLOT
*
*      Communicate NOUT and LPLOT through to callbacks
*
*      IUSER(1) = NOUT
*
*      IF (LPLOT) THEN
*          IUSER(2) = 1
*      ELSE
*          IUSER(2) = 0
*      END IF
*
*      Call E05JAF to initialize E05JBF
*
*      IFAIL = -1
*      CALL E05JAF(N,COMM,LCOMM,IFAIL)
*
*      IF (IFAIL.EQ.0) THEN
*
*          Use E05JCF to read some options from the end of the data
*          file
*
*          IFAIL = 0
*          CALL E05JCF(NIN,COMM,LCOMM,IFAIL)
*          WRITE (NOUT,*)
*
*          Use E05JKF to find the value of the integer-valued option
*          'Function Evaluations Limit'
*
*          IFAIL = 0
*          CALL E05JKF('Function Evaluations Limit',NF,COMM,LCOMM,
+              IFAIL)
*          WRITE (NOUT,99999) NF
*
*          Use E05JFF to set the value of the integer-valued option
*          'Static Limit'
*
*          STCLIM = 4*N
*
*          IFAIL = 0
*          CALL E05JFF('Static Limit',STCLIM,COMM,LCOMM,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 it holds its default value
*          (E05JHF returns 0)
*
*          IFAIL = 0
*          IBDCHK = E05JHF('Infinite Bound Size',COMM,LCOMM,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'
*
*          IFAIL = 0
*          CALL E05JLF('Infinite Bound Size',INFBND,COMM,LCOMM,IFAIL)
*          WRITE (NOUT,99996) INFBND
*
*          Use E05JGF to increase the value of the real-valued option
*          'Infinite Bound Size' tenfold
*
*          INFBND = 1.0D1*INFBND
*
*          IFAIL = 0

```

```

      CALL E05JGF('Infinite Bound Size',INFBND,COMM,LCOMM,IFAIL)
*
*      Use E05JDF to set the option 'Local Searches Limit = 40'
*
      IFAIL = 0
      CALL E05JDF('Local Searches Limit = 40',COMM,LCOMM,IFAIL)
*
*      Use E05JEF to set the option 'Local Searches' to 'On'
*
      LCSRCH = 'On'
*
      IFAIL = 0
      CALL E05JEF('Local Searches',LCSRCH,COMM,LCOMM,IFAIL)
*
*      Get that value of 'Local Searches' using E05JJF
*
      IFAIL = 0
      CALL E05JJF('Local Searches',LCSRCH,COMM,LCOMM,IFAIL)
      WRITE (NOUT,99995) LCSRCH
*
*      Solve the problem.
*
      IFAIL = 1
      CALL E05JBF(N,OBJFUN,IBOUND,IINIT,BL,BU,SDLIST,LIST,NUMPTS,
+          INITPT,MONIT,X,OBJ,COMM,LCOMM,IUSER,RUSER,IFAIL)
*
      WRITE (NOUT,*)
*
      IF (IFAIL.GE.0) THEN
          WRITE (NOUT,99994) IFAIL
*
          IF (IFAIL.EQ.0) THEN
              WRITE (NOUT,99993) OBJ
              WRITE (NOUT,99992) (X(I),I=1,N)
              END IF
*
          ELSE
              WRITE (NOUT,99991) IFAIL
              END IF
*
      END IF
*
      END IF
*
      END IF
*
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 "',A3,"' .')
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)
99991 FORMAT (1X,' ** E05JBF returned with IFAIL = ',I5)
      END
      SUBROUTINE OBJFUN(N,X,F,NSTATE,IUSER,RUSER,INFORM)
*
*      Routine to evaluate E05JBF objective function.
*      Mark 22 Release. NAG Copyright 2007.
*
*      .. Scalar Arguments ..
      DOUBLE PRECISION F
      INTEGER          INFORM, N, NSTATE
*
*      .. Array Arguments ..
      DOUBLE PRECISION RUSER(*), X(N)
      INTEGER          IUSER(*)
*
*      .. Local Scalars ..
      DOUBLE PRECISION X1, X2
      INTEGER          NOUT

```

```

*      .. Intrinsic Functions ..
      INTRINSIC      EXP
*      .. 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
*
*            NOUT = IUSER(1)
*            WRITE (NOUT,*)
*            WRITE (NOUT,99999)
*         END IF
*
*         X1 = X(1)
*         X2 = X(2)
*
*         F = 3.0D0*(1.0D0-X1)**2*EXP(-(X1**2)-(X2+1)**2) -
+           1.0D1*(X1/5.0D0-X1**3-X2**5)*EXP(-X1**2-X2**2) -
+           1.0D0/3.0D0*EXP(-(X1+1.0D0)**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,LIST,NUMPTS,INITPT,
+                    NBASKT,XBASKT,BOXL,BOXU,NSTATE,IUSER,RUSER,
+                    INFORM)
*
*      Monitoring routine for E05JBF.
*      Mark 22 Release. NAG Copyright 2007.
*
*      .. Scalar Arguments ..
      INTEGER      INFORM, N, NBASKT, NCALL, NINIT, NSTATE
*      .. Array Arguments ..
      DOUBLE PRECISION BOXL(N), BOXU(N), LIST(N,NINIT), RUSER(*),
+                    XBASKT(N,NBASKT), XBEST(N)
      INTEGER      ICOUNT(6), INITPT(N), IUSER(*), NUMPTS(N)
*      .. Local Scalars ..
      INTEGER      I, IPLOT, J, NOUT
*      .. External Subroutines ..
      EXTERNAL      OUTBOX
*      .. Executable Statements ..
      CONTINUE
*
      INFORM = 0
*
      IF (INFORM.GE.0) THEN
*
*         We are going to allow the iterations to continue.
*         Extract NOUT and PLOT from the integer communication-array
*
*         NOUT = IUSER(1)
*         IPLOT = IUSER(2)
*
*         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,*)
*
        IF ((IPLOT.EQ.1) .AND. (N.EQ.2)) THEN
            WRITE (NOUT,99998)
            WRITE (NOUT,*)
        END IF
*
    END IF
*
    IF ((IPLOT.EQ.1) .AND. (N.EQ.2)) THEN
*
        Display the coordinates of the edges of the current search
        box
*
        CALL OUTBOX(N,BOXL,BOXU,NOUT)
*
    END IF
*
    IF (NSTATE.LE.0) THEN
*
        MONIT is called for the last time
*
        IF ((IPLOT.EQ.1) .AND. (N.EQ.2)) THEN
            WRITE (NOUT,99997)
            WRITE (NOUT,*)
        END IF
*
        WRITE (NOUT,99996) ICOUNT(1)
        WRITE (NOUT,99995) NCALL
        WRITE (NOUT,99994) ICOUNT(2)
        WRITE (NOUT,99993) ICOUNT(3)
        WRITE (NOUT,99992) ICOUNT(4)
        WRITE (NOUT,99991) ICOUNT(5)
        WRITE (NOUT,99990) ICOUNT(6)
        WRITE (NOUT,99989) NBASKT
        WRITE (NOUT,99988)
*
        DO 20 I = 1, N
            WRITE (NOUT,99987) I, (XBASKT(I,J),J=1,NBASKT)
20        CONTINUE
*
        WRITE (NOUT,*)
        WRITE (NOUT,99986)
        WRITE (NOUT,*)
    END IF
*
    END IF
*
    RETURN
*
99999 FORMAT (1X,'*** Begin monitoring information ***')
99998 FORMAT (1X,'<Begin displaying search boxes>')
99997 FORMAT (1X,'<End displaying search boxes>')
99996 FORMAT (1X,'Total sub-boxes =',I5)
99995 FORMAT (1X,'Total function evaluations =',I5)
99994 FORMAT (1X,'Total function evaluations used in local search =',I5)
99993 FORMAT (1X,'Total points used in local search =',I5)
99992 FORMAT (1X,'Total sweeps through levels =',I5)
99991 FORMAT (1X,'Total splits by init. list =',I5)
99990 FORMAT (1X,'Lowest level with nonsplit boxes =',I5)
99989 FORMAT (1X,'Number of candidate minima in the "shopping basket',
+           ' " =',I5)
99988 FORMAT (1X,'Shopping basket:')
99987 FORMAT (1X,'XBASKT(',I3,',',:)=',(6F9.5))
99986 FORMAT (1X,'*** End monitoring information ***')
    END
    SUBROUTINE OUTBOX(N,BOXL,BOXU,NOUT)
*
    Displays edges of box with bounds BOXL and BOXU in format suitable
    for plotting.
*

```

```

*      Mark 22 Release. NAG Copyright 2007.
*
*      .. Scalar Arguments ..
      INTEGER          N, NOUT
*      .. Array Arguments ..
      DOUBLE PRECISION BOXL(N), BOXU(N)
*      .. Executable Statements ..
      CONTINUE

*
      WRITE (NOUT,99999) BOXL(1), BOXL(2)
      WRITE (NOUT,99999) BOXL(1), BOXU(2)
      WRITE (NOUT,99998)
      WRITE (NOUT,99999) BOXL(1), BOXL(2)
      WRITE (NOUT,99999) BOXU(1), BOXL(2)
      WRITE (NOUT,99998)
      WRITE (NOUT,99999) BOXL(1), BOXU(2)
      WRITE (NOUT,99999) BOXU(1), BOXU(2)
      WRITE (NOUT,99998)
      WRITE (NOUT,99999) BOXU(1), BOXL(2)
      WRITE (NOUT,99999) BOXU(1), BOXU(2)
      WRITE (NOUT,99998)

*
      RETURN

*
99999 FORMAT (F20.15,1X,F20.15)
99998 FORMAT (A)
      END

```

9.2 Program Data

E05JCF Example Program Data

3	:	SDLIST
0	:	IBOUND
-3.0 -3.0	:	Lower bounds BL
3.0 3.0	:	Upper bounds BU
3	:	IINIT
3 3 3	:	NUMPTS
-3.0 -1.0 3.0 -3.0 0.0 3.0	:	Matrix LIST
2 2 2	:	INITPT
.FALSE.	:	LPLOT

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 0.11579E+78.
 Option "Local Searches" has the value "ON".

(OBJFUN was just called for the first time)

*** Begin monitoring information ***

```

Total sub-boxes = 180
Total function evaluations = 185
Total function evaluations used in local search = 102
Total points used in local search = 9
Total sweeps through levels = 9
Total splits by init. list = 5
Lowest level with nonsplit boxes = 6

```

```
Number of candidate minima in the "shopping basket" =    2
Shopping basket:
XBASKT( 1,:) =  0.22828 -1.34740
XBASKT( 2,:) = -1.62553  0.20452
```

```
*** End monitoring information ***
```

```
On exit from E05JBF, IFAIL =    0
Final objective value =  -6.55113
Global optimum X =  0.22828 -1.62553
```

