NAG Toolbox for MATLAB

e04uj

1 Purpose

To supply individual optional parameters to e04ug.

2 Syntax

```
[lwsav, iwsav, rwsav, inform] = e04uj(str, lwsav, iwsav, rwsav)
```

3 Description

e04uj may be used to supply values for optional parameters to e04ug. It is only necessary to call e04uj for those parameters whose values are to be different from their default values. One call to e04uj sets one parameter value.

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

```
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 value. Such numbers may be up to 16 contiguous characters in Fortran'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.

For e04uj, each user-specified option is normally printed as it is defined, on the current advisory message unit (see x04ab), but this printing may be suppressed using the keyword **Nolist**. Thus the statement

```
[lwsav, iwsav, rwsav, inform] = e04uj('Nolist', lwsav, iwsav, rwsav);
```

suppresses printing of this and subsequent options. Printing will automatically be turned on again after a call to e04ug and may be turned on again at any time using the keyword **List**.

For e04uj printing is turned off by default, but may be turned on at any time using the keyword **List**.

Optional parameter settings are preserved following a call to e04ug 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 e04ug.

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 11 of the document for e04ug.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **str – string**

A single valid option string (as described in Section 3 and in Section 11 of the document for e04ug).

2: **lwsav**(**20**) **– logical array**
3: **iwsav**(**550**) **– int32 array**
4: **rwsav**(**550**) **– double array**

The arrays **lwsav**, **iwsav** and **rwsav** **must not** be altered between calls to any of the functions e04wb, e04ug or e04uj.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **lwsav**(**20**) **– logical array**
2: **iwsav**(**550**) **– int32 array**
3: **rwsav**(**550**) **– double array**

The arrays **lwsav**, **iwsav** and **rwsav** **must not** be altered between calls to any of the functions e04wb, e04ug or e04uj.

4: **inform – int32 scalar**

Contains zero if a valid option string has been supplied and a value > 0 otherwise (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

inform $= 5$

The supplied option is invalid. Check that the keywords are neither ambiguous nor misspelt.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

e04ug_confun.m

```
function [mode, f, fjac, user] = ...
   confun(mode, ncnln, njnln, nnzjac, x, fjac, nstate, user)
  f = zeros(ncnln, 1);
```

```
  if (mode == 0 || mode == 2)
    f(1) = 1000*sin(-x(1)-0.25) + 1000*sin(-x(2)-0.25);
    f(2) = 1000*sin(x(1)-0.25) + 1000*sin(x(1)-x(2)-0.25);
    f(3) = 1000*sin(x(2)-x(1)-0.25) + 1000*sin(x(2)-0.25);
  end

  if (mode == 1 || mode == 2)
%   nonlinear jacobian elements for column 1.
    fjac(1) = -1000*cos(-x(1)-0.25);
    fjac(2) = 1000*cos(x(1)-0.25) + 1000*cos(x(1)-x(2)-0.25);
    fjac(3) = -1000*cos(x(2)-x(1)-0.25);
%   nonlinear jacobian elements for column 2.
    fjac(4) = -1000*cos(-x(2)-0.25);
    fjac(5) = -1000*cos(x(1)-x(2)-0.25);
    fjac(6) = 1000*cos(x(2)-x(1)-0.25) + 1000*cos(x(2) -0.25);
  end
```

e04ug_objfun.m

```
function [mode, objf, objgrd, user] = objfun(mode, nonln, x, objgrd,
nstate, user)

  if (mode == 0 || mode == 2)
    objf = 1.0e-6*x(3)^3 + 2.0e-6*x(4)^3/3;
  end

  if (mode == 1 || mode == 2)
    objgrd(1) = 0;
    objgrd(2) = 0;
    objgrd(3) = 3.0e-6*x(3)^2;
    objgrd(4) = 2.0e-6*x(4)^2;
  end
```

```
n = int32(4);
m = int32(6);
ncnln = int32(3);
nonln = int32(4);
njnln = int32(2);
iobj = int32(6);
a = [9.999999999999999e+24;
     9.999999999999999e+24;
     9.999999999999999e+24;
     1;
     -1;
     9.999999999999999e+24;
     9.999999999999999e+24;
     9.999999999999999e+24;
     -1;
     1;
     3;
     -1;
     -1;
     2];
ha = [int32(1);
     int32(2);
     int32(3);
     int32(5);
     int32(4);
     int32(1);
     int32(2);
     int32(3);
     int32(5);
     int32(4);
     int32(6);
     int32(1);
     int32(2);
```

```
     int32(6)];
ka = [int32(1);
     int32(6);
     int32(11);
     int32(13);
     int32(15)];
bl = [-0.55;
     -0.55;
     0;
     0;
     -894.8;
     -894.8;
     -1294.8;
     -0.55;
     -0.55;
     -9.999999999999999e+24];
bu = [0.55;
     0.55;
     1200;
     1200;
     -894.8;
     -894.8;
     -1294.8;
     9.999999999999999e+24;
     9.999999999999999e+24;
     9.999999999999999e+24];
start = 'C';
names = {'Varble 1'; 'Varble 2'; 'Varble 3'; 'Varble 4'; 'NlnCon 1';
'NlnCon 2'; 'NlnCon 3'; 'LinCon 1'; 'LinCon 2'; 'Free Row'};
ns = int32(0);
xs = [0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0];
istate = zeros(10, 1, 'int32');
clamda = [0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0;
     0];
leniz = int32(1000);
lenz = int32(1000);
[cwsav,lwsav,iwsav,rwsav,ifail] = e04wb('e04ug');
[lwsav,iwsav,rwsav,ifail] = e04uj('Major Print level = 1', lwsav, iwsav,
rwsav);
[aOut, nsOut, xsOut, istateOut, clamdaOut, miniz, minz, ninf, sinf, ...
 obj, user, lwsavOut, iwsavOut, rwsavOut, ifail] = ...
     e04ug('e04ug_confun', 'e04ug_objfun', n, m, ncnln, nonln, njnln, ...
           iobj, a, ha, ka, bl, bu, start, names, ns, xs, istate, clamda,
...
          leniz, lenz, lwsav, iwsav, rwsav)
```

```
*** E04UGF

 Parameters
 ----------

 Frequencies.
  Check frequency.........           60          Expand frequency.......
```

```
10000
 Factorization frequency.        50

 QP subproblems.
 Scale tolerance.........  9.00E-01       Minor feasibility tol..  1.05E-
08
 Scale option............         1       Minor optimality tol...  1.05E-
08
 Partial price...........         1       Crash tolerance........  1.00E-
01
  Pivot tolerance.........   2.05E-11           Minor print level......
0
  Crash option............             0          Elastic weight.........
1.00E+02

 The SQP method.
 Minimize................
 Nonlinear objective vars         4       Major optimality tol...  1.05E-
08
  Function precision......   1.72E-13           Unbounded step size....
1.00E+20
 Superbasics limit.......         4       Forward difference int.  4.15E-
07
 Unbounded objective.....  1.00E+15       Central difference int.  5.57E-
05
 Major step limit........  2.00E+00       Derivative linesearch..
  Derivative level........             3          Major iteration limit..
1000
  Linesearch tolerance....   9.00E-01           Verify level...........
0
  Minor iteration limit...          500          Major print level......
1
  Infinite bound size.....   1.00E+20           Iteration limit........
10000

 Hessian approximation.
  Hessian full memory.....                      Hessian updates........
99999999
 Hessian frequency.......  99999999

 Nonlinear constraints.
 Nonlinear constraints...         3       Major feasibility tol..  1.05E-
08
  Nonlinear Jacobian vars.            2          Violation limit........
1.00E+01

 Miscellaneous.
  Variables...............             4          Linear constraints.....
3
  Nonlinear variables.....             4          Linear variables.......
0
 LU factor tolerance.....  5.00E+00       LU singularity tol.....  2.05E-
11
 LU update tolerance.....  5.00E+00       LU density tolerance...  6.00E-
01
  eps (machine precision).   1.11E-16           Monitoring file........
-1
 COLD start..............                Infeasible exit........

 Workspace provided is                 IZ(    1000),  Z(    1000).
 To start solving the problem we need  IZ(     628),  Z(     758).

 confun  sets       6   out of       6   constraint gradients.
 objfun  sets       4   out of       4   objective  gradients.

 Cheap test on confun...

 The Jacobian seems to be OK.

 The largest discrepancy was    4.42E-08  in constraint     3.
```

```
 Cheap test on objfun...

 The objective gradients seem to be OK.
   Gradient   projected   in   two   directions         0.00000000000E+00
0.00000000000E+00
  Difference  approximations                           1.74248004037E-19
4.49092793911E-21

 Variable State      Value         Lower Bound   Upper Bound    Lagr Mult
Residual

 Varble 1     BS    0.118876      -0.55000       0.55000      -1.2529E-07
0.4311
 Varble 2     BS   -0.396234      -0.55000       0.55000       1.9243E-08
0.1538
 Varble 3     BS     679.945            .          1200.0       1.7001E-10
520.1
 Varble 4    SBS     1026.07            .          1200.0      -2.1918E-10
173.9

 Constrnt State      Value         Lower Bound   Upper Bound    Lagr Mult
Residual

 NlnCon 1     EQ    -894.800        -894.80        -894.80        -4.387
3.3644E-09
 NlnCon 2     EQ    -894.800        -894.80        -894.80        -4.106
6.0049E-10
 NlnCon 3     EQ    -1294.80        -1294.8        -1294.8        -5.463
3.3549E-09
 LinCon 1     BS   -0.515110       -0.55000           None            .
3.4890E-02
 LinCon 2     BS    0.515110       -0.55000           None            .
1.065
 Free Row     BS     4091.97            None           None       -1.000
4092.

 Exit E04UGF - Optimal solution found.

 Final objective value =     5126.498
aOut =
   1.0e+03 *
   -0.9327
    1.9565
   -0.7213
    0.0010
   -0.0010
   -0.9893
   -0.9651
    1.5197
   -0.0010
    0.0010
    0.0030
   -0.0010
   -0.0010
    0.0020
nsOut =
           1
xsOut =
   1.0e+03 *
    0.0001
   -0.0004
    0.6799
    1.0261
   -0.8948
   -0.8948
   -1.2948
   -0.0005
    0.0005
    4.0920
```

```
istateOut =
           3
           3
           3
           2
           0
           0
           0
           3
           3
           3
clamdaOut =
   -0.0000
    0.0000
    0.0000
   -0.0000
   -4.3870
   -4.1056
   -5.4633
         0
         0
   -1.0000
miniz =
         628
minz =
         758
ninf =
           0
sinf =
     0
obj =
   5.1265e+03
user =
     0
lwsavOut =
     1
     0
     0
     1
     1
     1
     1
     1
     1
     1
     0
     1
     1
     0
     0
     0
     0
     0
     0
     0
iwsavOut =
     array elided
rwsavOut =
     array elided
ifail =
           0
```