NAG Toolbox

nag_opt_uncon_conjgrd_option_string (e04dk)

1 Purpose

To supply individual optional parameters to nag_opt_uncon_conjgrd_comp (e04dg).

2 Syntax

```
[lwsav, iwsav, rwsav, inform] = nag_opt_uncon_conjgrd_option_string(str, lwsav, iwsav, rwsav)
[lwsav, iwsav, rwsav, inform] = e04dk(str, lwsav, iwsav, rwsav)
```

3 Description

nag_opt_uncon_conjgrd_option_string (e04dk) may be used to supply values for optional parameters to nag_opt_uncon_conjgrd_comp (e04dg). It is only necessary to call nag_opt_uncon_conjgrd_option_string (e04dk) for those arguments whose values are to be different from their default values. One call to nag_opt_uncon_conjgrd_option_string (e04dk) sets one argument 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 nag_opt_uncon_conjgrd_option_string (e04dk), each user-specified option is normally printed as it is defined, on the current advisory message unit (see nag_file_set_unit_advisory (x04ab)), but this printing may be suppressed using the keyword **Nolist**. Thus the statement

```
[lwsav, iwsav, rwsav, inform] = e04dk('Nolist', lwsav, iwsav, rwsav);
```

suppresses printing of this and subsequent options. Printing will automatically be turned on again after a call to nag_opt_uncon_conjgrd_comp (e04dg) and may be turned on again at any time using the keyword **List**.

For nag_opt_uncon_conjgrd_option_string (e04dk) 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 nag_opt_uncon_conjgrd_comp (e04dg) 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 nag_opt_uncon_conjgrd_comp (e04dg).

A complete list of optional parameters, their abbreviations, synonyms and default values is given in Section 12 in nag_opt_uncon_conjgrd_comp (e04dg).

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **str** – CHARACTER(*)

A single valid option string (as described in Section 3 and in Section 12 in nag_opt_uncon_conjgrd_comp (e04dg)).

2: **lwsav**(**120**) – LOGICAL array
3: **iwsav**(**610**) – INTEGER array
4: **rwsav**(**475**) – REAL (KIND=nag_wp) array

The arrays **lwsav**, **iwsav** and **rwsav** **must not** be altered between calls to any of the functions nag_opt_uncon_conjgrd_option_string (e04dk), nag_opt_uncon_conjgrd_comp (e04dg) or nag_opt_init (e04wb).

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **lwsav**(**120**) – LOGICAL array
2: **iwsav**(**610**) – INTEGER array
3: **rwsav**(**475**) – REAL (KIND=nag_wp) array

4: **inform** – INTEGER

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

6 Error Indicators and Warnings

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

9.1 Program Text

```
     function e04dk_example

fprintf('e04dk example results\n\n');

% Initialize minimizer and set some options
[cwsav,lwsav,iwsav,rwsav,ifail] = e04wb('e04dg');
[lwsav, iwsav, rwsav, inform]   = e04dk(...
                                  'Iteration Limit = 25', lwsav, iwsav, rwsav);
[lwsav, iwsav, rwsav, inform]   = e04dk(...
                                       'Print Level = 0', lwsav, iwsav, rwsav);
[lwsav, iwsav, rwsav, inform]   = e04dk(...
                                       'Nolist', lwsav, iwsav, rwsav);
[lwsav, iwsav, rwsav, inform]   = e04dk(...
                                      'Verify Level = 1', lwsav, iwsav, rwsav);
[lwsav, iwsav, rwsav, inform]   = e04dk(...
                                'Maximum Step length = 100', lwsav, iwsav, rwsav);

% Initial guess.
x = [-1;  1];

% Minimize
[iter, objf, objgrd, x, user, lwsav, iwsav, rwsav, ifail] = ...
   e04dg(...
         @objfun, x, lwsav, iwsav, rwsav);

fprintf('Number of iterations        = %10d\n',iter);
fprintf('Value of objective function = %10.3e\n',objf);
fprintf('Value of df/dx              = %10.3e\n',objgrd(1));
fprintf('Value of df/dy              = %10.3e\n',objgrd(2));
fprintf('Location of minimum         = (%7.3f,%7.3f)\n',x);

function [mode, objf, objgrd, user] = objfun(mode, n, x, nstate, user)
   a     = x(1);
   b     = x(2);
   expa  = exp(a);
   objf  = expa*((2*a+b)^2 + (b+1)^2);
   if (mode == 2)
      objgrd(1) = 4*expa*(2*a+b) + objf;
      objgrd(2) = 2*expa*(2*a+2*b+1.0);
   else
      objgrd = zeros(2,1);
   end
```

9.2 Program Results

```
     e04dk example results

Number of iterations        =         10
Value of objective function =  5.293e-14
Value of df/dx              =  9.125e-07
Value of df/dy              =  8.316e-07
Location of minimum         = (  0.500, -1.000)
```