NAG Library Function Document

nag_opt_handle_set_nlnconstr (e04rkc)

1 Purpose

nag_opt_handle_set_nlnconstr (e04rkc) is a part of the NAG optimization modelling suite and defines the number of nonlinear constraints of the problem as well as the sparsity structure of their first derivatives.

2 Specification

```
#include <nag.h>
#include <nage04.h>
```

```
void nag_opt_handle_set_nlnconstr (void *handle, Integer ncnln,
     const double bl[], const double bu[], Integer nnzgd,
     const Integer irowgd[], const Integer icolgd[], NagError *fail)
```

3 Description

After the initialization function nag_opt_handle_init (e04rac) has been called, nag_opt_handle_set_nlnconstr (e04rkc) may be used to define the nonlinear constraints $l_g \leq g(x) \leq u_g$ of the problem unless the nonlinear constraints have already been defined. This will typically be used for nonlinear programming problems (NLP) of the kind:

$$\begin{array}{lll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & f(x) & \text{(a)} \\ \text{subject to} & l_g \leq g(x) \leq u_g & \text{(b)} \\ & l_B \leq Bx \leq u_B & \text{(c)} \\ & l_x \leq x \leq u_x & \text{(d)} \end{array} \qquad (1)$$

where n is the number of the decision variables x, m_g is the number of the nonlinear constraints (in (1)(b)) and $g(x)$, l_g and u_g are m_g-dimensional vectors. Linear constraints ((1)(c)), which require no separate gradient information, can be introduced by nag_opt_handle_set_linconstr (e04rjc) and Box constraints ((1)(d)) can be introduced by nag_opt_handle_set_simplebounds (e04rhc).

Note that upper and lower bounds are specified for all the constraints. This form allows full generality in specifying various types of constraint. In particular, the jth constraint may be defined as an equality by setting $l_j = u_j$. If certain bounds are not present, the associated elements l_j or u_j may be set to special values that are treated as $-\infty$ or $+\infty$. See the description of the optional parameter **Infinite Bound Size** of the solver nag_opt_handle_solve_ipopt (e04stc). Its value is denoted as $bigbnd$ further in this text. Note that the bounds are interpreted based on its value at the time of calling this function and any later alterations to **Infinite Bound Size** will not affect these constraints.

Since each nonlinear constraint is most likely to involve a small subset of the decision variables, the partial derivatives of the constraint functions with respect to those variables are best expressed as a sparse Jacobian matrix of m_g rows and n columns. The row and column positions of all the nonzero derivatives must be registered with the handle through nag_opt_handle_set_nlnconstr (e04rkc).

The values of the nonlinear constraint functions and their nonzero gradients at particular points in the decision variable space will be communicated to the NLP solver by user-supplied functions (e.g., **confun** and **congrd** for nag_opt_handle_solve_ipopt (e04stc)).

See nag_opt_handle_init (e04rac) for more details.

4 References

None.

5 Arguments

1: **handle** – void * *Input*

On entry: the handle to the problem. It needs to be initialized by nag_opt_handle_init (e04rac) and **must not** be changed.

2: **ncnln** – Integer *Input*

On entry: m_g, the number of nonlinear constraints (number of rows of the Jacobian matrix).

If **ncnln** $= 0$, no nonlinear constraints will be defined and **bl**, **bu**, **nnzgd**, **irowgd** and **icolgd** will not be referenced and may be **NULL**.

Constraint: **ncnln** ≥ 0.

3: **bl**[**ncnln**] – const double *Input*
4: **bu**[**ncnln**] – const double *Input*

On entry: **bl** and **bu** define lower and upper bounds of the nonlinear constraints, l_g and u_g, respectively. To define the jth constraint as equality, set **bl**$[j-1] =$ **bu**$[j-1] = \beta$, where $|\beta| < bigbnd$. To specify a nonexistent lower bound (i.e., $l_j = -\infty$), set **bl**$[j-1] \leq -bigbnd$; to specify a nonexistent upper bound, set **bu**$[j-1] \geq bigbnd$.

Constraints:

bl$[j-1] \leq$ **bu**$[j-1]$, for $j = 1, 2, \ldots,$ **ncnln**;
bl$[j-1] < bigbnd$, for $j = 1, 2, \ldots,$ **ncnln**;
bu$[j-1] > -bigbnd$, for $j = 1, 2, \ldots,$ **ncnln**.

5: **nnzgd** – Integer *Input*

On entry: **nnzgd** gives the number of nonzeros in the Jacobian matrix.

Constraint: if **ncnln** > 0, **nnzgd** > 0.

6: **irowgd**[**nnzgd**] – const Integer *Input*
7: **icolgd**[**nnzgd**] – const Integer *Input*

On entry: arrays **irowgd** and **icolgd** store the sparsity structure (pattern) of the Jacobian matrix as **nnzgd** nonzeros in coordinate storage (CS) format (see Section 2.1.1 in the f11 Chapter Introduction). The matrix has dimensions **ncnln** $\times n$. **irowgd** specifies one-based row indices and **icolgd** specifies one-based column indices. No particular order of elements is expected, but elements should not repeat and the same order should be used when the Jacobian is evaluated for the solver, e.g., the value of $\frac{\partial g_i}{\partial x_j}$ where $i =$ **irowgd**$[l-1]$ and $j =$ **icolgd**$[l-1]$ should be stored in **gdx**$[l-1]$, for $l = 1, 2, \ldots,$ **nnzgd**.

Constraints:

$1 \leq$ **irowgd**$[l-1] \leq$ **ncnln**, for $l = 1, 2, \ldots,$ **nnzgd**;
$1 \leq$ **icolgd**$[l-1] \leq n$, for $l = 1, 2, \ldots,$ **nnzgd**.

8: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_ALREADY_DEFINED

A set of nonlinear constraints has already been defined.

NE_BAD_PARAM

On entry, argument $\langle value\rangle$ had an illegal value.

NE_BOUND

On entry, $j = \langle value\rangle$, $\mathbf{bl}[j-1] = \langle value\rangle$, $bigbnd = \langle value\rangle$.
Constraint: $\mathbf{bl}[j-1] < bigbnd$.

On entry, $j = \langle value\rangle$, $\mathbf{bl}[j-1] = \langle value\rangle$ and $\mathbf{bu}[j-1] = \langle value\rangle$.
Constraint: $\mathbf{bl}[j-1] \le \mathbf{bu}[j-1]$.

On entry, $j = \langle value\rangle$, $\mathbf{bu}[j-1] = \langle value\rangle$, $bigbnd = \langle value\rangle$.
Constraint: $\mathbf{bu}[j-1] > -bigbnd$.

NE_HANDLE

The supplied **handle** does not define a valid handle to the data structure for the NAG optimization modelling suite. It has not been initialized by nag_opt_handle_init (e04rac) or it has been corrupted.

NE_INT

On entry, **ncnln** $= \langle value\rangle$.
Constraint: **ncnln** ≥ 0.

On entry, **nnzgd** $= \langle value\rangle$.
Constraint: **nnzgd** > 0.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in How to Use the NAG Library and its Documentation for further information.

NE_INVALID_CS

On entry, $i = \langle value\rangle$, $\mathbf{icolgd}[i-1] = \langle value\rangle$ and $n = \langle value\rangle$.
Constraint: $1 \le \mathbf{icolgd}[i-1] \le n$.

On entry, $i = \langle value\rangle$, $\mathbf{irowgd}[i-1] = \langle value\rangle$ and **ncnln** $= \langle value\rangle$.
Constraint: $1 \le \mathbf{irowgd}[i-1] \le$ **ncnln**.

On entry, more than one element of structural Jacobian matrix has row index $\langle value\rangle$ and column index $\langle value\rangle$.
Constraint: each element of structural Jacobian matrix must have a unique row and column index.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in How to Use the NAG Library and its Documentation for further information.

NE_PHASE

The Hessian of the nonlinear objective has already been defined, nonlinear constraints cannot be added.

The problem cannot be modified in this phase any more, the solver has already been called.

7 Accuracy

Not applicable.

8 Parallelism and Performance

nag_opt_handle_set_nlnconstr (e04rkc) is not threaded in any implementation.

9 Further Comments

9.1 Additional Licensor

Parts of the code for e04rkc are distributed according to terms imposed by another licensor. Please refer to the list of Library licensors available on the NAG Website for further details.

10 Example

See Section 10 in nag_opt_handle_solve_ipopt (e04stc).
