

# NAG Library Function Document

## nag\_opt\_handle\_set\_linobj (e04rec)

### 1 Purpose

nag\_opt\_handle\_set\_linobj (e04rec) is a part of the NAG optimization modelling suite and defines the linear objective function of the problem.

### 2 Specification

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

void nag_opt_handle_set_linobj (void *handle, Integer nvar,
    const double cvec[], NagError *fail)
```

### 3 Description

After the initialization function nag\_opt\_handle\_init (e04rac) has been called, nag\_opt\_handle\_set\_linobj (e04rec) may be used to define the objective function of the problem as a linear function  $c^T x$  unless the objective function has already been defined by nag\_opt\_handle\_set\_linobj (e04rec), nag\_opt\_handle\_set\_quadobj (e04rfc) or by nag\_opt\_handle\_set\_nlnobj (e04rgc). This will typically be used for linear semidefinite programming problems (SDP)

$$\begin{aligned}
 &\text{minimize}_{x \in \mathbb{R}^n} && c^T x && (a) \\
 &\text{subject to} && \sum_{i=1}^n x_i A_i^k - A_0^k \succeq 0, \quad k = 1, \dots, m_A && (b) \\
 & && l_B \leq Bx \leq u_B && (c) \\
 & && l_x \leq x \leq u_x && (d)
 \end{aligned} \tag{1}$$

or SDP with bilinear matrix inequalities (BMI-SDP) where the objective function has only linear terms. 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: **nvar** – Integer *Input*  
*On entry:*  $n$ , the number of decision variables  $x$  in the problem. It must be unchanged from the value set during the initialization of the handle by nag\_opt\_handle\_init (e04rac).
- 3: **cvec[nvar]** – const double *Input*  
*On entry:* the dense vector  $c$  of the objective function.

- 4:     **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 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

### NE\_ALREADY\_DEFINED

The objective function has already been defined.

### NE\_BAD\_PARAM

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

### 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\_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 2.7.6 in How to Use the NAG Library and its Documentation for further information.

### NE\_NO\_LICENCE

Your licence key may have expired or may not have been installed correctly.  
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

### NE\_PHASE

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

### NE\_REF\_MATCH

On entry, **nvar** =  $\langle value \rangle$ , expected value =  $\langle value \rangle$ .  
Constraint: **nvar** must match the value given during initialization of **handle**.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

`nag_opt_handle_set_linobj` (e04rec) is not threaded in any implementation.

## 9 Further Comments

None.

## **10 Example**

See Section 10 in nag\_opt\_handle\_init (e04rac) for links to all examples in the suite.

---