

## NAG Library Function Document

### nag\_quad\_1d\_gen\_vec\_multi\_dimreq (d01rcc)

#### 1 Purpose

The dimension of the arrays that must be passed as actual arguments to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac) are dependent upon a number of factors. `nag_quad_1d_gen_vec_multi_dimreq` (d01rcc) returns the correct size of these arrays enabling `nag_quad_1d_gen_vec_multi_rcomm` (d01rac) to be called successfully.

#### 2 Specification

```
#include <nag.h>
#include <nagd01.h>

void nag_quad_1d_gen_vec_multi_dimreq (Integer ni, Integer *lenxrq,
    Integer *ldfmrq, Integer *sdfmrq, Integer *licmin, Integer *licmax,
    Integer *lcmin, Integer *lcmax, const Integer iopts[],
    const double opts[], NagError *fail)
```

#### 3 Description

`nag_quad_1d_gen_vec_multi_dimreq` (d01rcc) returns the minimum dimension of the arrays **x** (*lenxrq*), **fm** (*ldfmrq* × *sdfmrq*), **icom** (*licmin*) and **com** (*lcmin*) that must be passed to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac) to enable the integration to commence given options currently set for the **ni** integrands. `nag_quad_1d_gen_vec_multi_dimreq` (d01rcc) also returns the upper bounds *licmax* and *lcmax* for the dimension of the arrays **icom** and **com**, that could possibly be required with the chosen options.

All the minimum values *lenxrq*, *ldfmrq*, *sdfmrq*, *licmin* and *lcmin*, and subsequently all the maximum values *licmax* and *lcmax* may be affected if different options are set, and hence `nag_quad_1d_gen_vec_multi_dimreq` (d01rcc) should be called after any options are set, and before the first call to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac).

A segment is here defined as a (possibly maximal) subset of the domain of integration. During subdivision, a segment is bisected into two new segments.

#### 4 References

None.

#### 5 Arguments

- 1: **ni** – Integer *Input*  
*On entry:*  $n_i$ , the number of integrals which will be approximated in the subsequent call to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac).  
*Constraint:* **ni** > 0.
- 2: **lenxrq** – Integer \* *Output*  
*On exit:* *lenxrq*, the minimum dimension of the array **x** that can be used in a subsequent call to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac).

- 3: **ldfmrq** – Integer \* *Output*  
*On exit:* *ldfmrq*, the minimum stride separating row elements of the matrix of values stored in the array **fm** that can be used in a subsequent call to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac).
- 4: **sdfmrq** – Integer \* *Output*  
*On exit:* *sdfmrq*, the minimum number of columns of the matrix of values stored in the array **fm** that can be used in a subsequent call to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac).  
**Note:** the minimum dimension of the array **fm** is  $ldfmrq \times sdfmrq$ .
- 5: **licmin** – Integer \* *Output*  
*On exit:* *licmin*, the minimum dimension of the array **icom** that must be passed to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac) to enable it to calculate a single approximation to all the  $n_i$  integrals over the interval  $[a, b]$  with  $s_{pri}$  initial segments.
- 6: **licmax** – Integer \* *Output*  
*On exit:* *licmax* the dimension of the array **icom** that must be passed to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac) to enable it to exhaust the adaptive process controlled by the currently set options for the  $n_i$  integrals over the interval  $[a, b]$  with  $s_{pri}$  initial segments.
- 7: **lcmmin** – Integer \* *Output*  
*On exit:* *lcmmin*, the minimum dimension of the array **com** that must be passed to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac) to enable it to calculate a single approximation to all the  $n_i$  integrals over the interval  $[a, b]$  with  $s_{pri}$  initial segments.
- 8: **lcmmax** – Integer \* *Output*  
*On exit:* *lcmmax*, the dimension of the array **com** that must be passed to `nag_quad_1d_gen_vec_multi_rcomm` (d01rac) to enable it to exhaust the adaptive process controlled by the currently set options for the  $n_i$  integrals over the interval  $[a, b]$  with  $s_{pri}$  initial segments.
- 9: **iopts** $[dim]$  – const Integer *Communication Array*  
**Note:** the dimension, *dim*, of this array is dictated by the requirements of associated functions that must have been previously called. This array MUST be the same array passed as argument **iopts** in the previous call to `nag_quad_opt_set` (d01zkc).  
*On entry:* the integer option array as returned by `nag_quad_opt_set` (d01zkc).  
*Constraint:* **iopts** must not be changed between calls to `nag_quad_opt_set` (d01zkc), `nag_quad_opt_get` (d01zlc), `nag_quad_1d_gen_vec_multi_dimreq` (d01rcc) and `nag_quad_1d_gen_vec_multi_rcomm` (d01rac).
- 10: **opts** $[dim]$  – const double *Communication Array*  
**Note:** the dimension, *dim*, of this array is dictated by the requirements of associated functions that must have been previously called. This array MUST be the same array passed as argument **opts** in the previous call to `nag_quad_opt_set` (d01zkc).  
*On entry:* the real option array as returned by `nag_quad_opt_set` (d01zkc).  
*Constraint:* **opts** must not be changed between calls to `nag_quad_opt_set` (d01zkc), `nag_quad_opt_get` (d01zlc), `nag_quad_1d_gen_vec_multi_dimreq` (d01rcc) and `nag_quad_1d_gen_vec_multi_rcomm` (d01rac).
- 11: **fail** – NagError \* *Input/Output*  
The NAG error argument (see Section 3.6 in the Essential Introduction).

## 6 Error Indicators and Warnings

### NE\_ALLOC\_FAIL

Dynamic memory allocation failed.  
See Section 3.2.1.2 in the Essential Introduction for further information.

### NE\_BAD\_PARAM

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

### NE\_INT

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

### 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 the Essential Introduction for further information.

### NE\_INVALID\_OPTION

One of the option arrays **iopts** or **opts** has become corrupted. Re-initialize the arrays using `nag_quad_opt_set (d01zkc)`.

### NE\_NO\_LICENCE

Your licence key may have expired or may not have been installed correctly.  
See Section 3.6.5 in the Essential Introduction for further information.

## 7 Accuracy

Not applicable.

## 8 Parallelism and Performance

Not applicable.

## 9 Further Comments

None.

## 10 Example

See Section 10 in `nag_quad_1d_gen_vec_multi_rcomm (d01rac)` for examples of the usage of `nag_quad_1d_gen_vec_multi_dimreq (d01rcc)`.

---