NAG Library Routine Document

D01RCF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of **bold italicised** terms and other implementation-dependent details.

1   Purpose

The dimension of the arrays that must be passed as actual arguments to D01RAF are dependent upon a number of factors. D01RCF returns the correct size of these arrays enabling D01RAF to be called successfully.

2   Specification

```
SUBROUTINE D01RCF (NI, LENXRQ, LDFMRQ, SDFMRQ, LICMIN, LICMAX, LCMIN,      &
                   LCMAX, IOPTS, OPTS, IFAIL)

INTEGER           NI, LENXRQ, LDFMRQ, SDFMRQ, LICMIN, LICMAX, LCMIN,      &
                   LCMAX, IOPTS(*), IFAIL
REAL (KIND=nag_wp) OPTS(*)
```

3   Description

D01RCF returns the minimum dimension of the arrays X ($lenxrq$), FM ($ldfmrq \times sdfmrq$), ICOM ($licmin$) and COM ($lcmin$) that must be passed to D01RAF to enable the integration to commence given options currently set for the NI integrands. D01RCF 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 D01RCF should be called after any options are set, and before the first call to D01RAF.

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 D01RAF.

    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 D01RAF.

3:    LDFMRQ – INTEGER   *Output*

    On exit: $ldfmrq$, the minimum leading dimension of the array FM that can be used in a subsequent call to D01RAF.

4: SDFMRQ – INTEGER *Output*

On exit: $sdfmrq$, the minimum second dimension of the array FM that can be used in a subsequent call to D01RAF.

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 D01RAF 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 D01RAF 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: LCMIN – INTEGER *Output*

On exit: $lcmin$, the minimum dimension of the array COM that must be passed to D01RAF 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: LCMAX – INTEGER *Output*

On exit: $lcmax$, the dimension of the array COM that must be passed to D01RAF 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(∗) – INTEGER array *Communication Array*

Note: the dimension 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 D01ZKF.

On entry: the integer option array as returned by D01ZKF.

Constraint: IOPTS must not be changed between calls to D01ZKF, D01ZLF, D01RCF and D01RAF.

10: OPTS(∗) – REAL (KIND=nag_wp) array *Communication Array*

Note: the dimension 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 D01ZKF.

On entry: the real option array as returned by D01ZKF.

Constraint: OPTS must not be changed between calls to D01ZKF, D01ZLF, D01RCF and D01RAF.

11: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, −1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value −1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value −1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 21

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

IFAIL = 1001

One of the option arrays IOPTS or OPTS has become corrupted. Re-initialize the arrays using D01ZKF.

IFAIL = −99

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = −399

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = −999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Not applicable.

8 Parallelism and Performance

D01RCF is not threaded in any implementation.

9 Further Comments

None.

10 Example

See Section 10 in D01RAF for examples of the usage of D01RCF.