

# NAG Library Routine Document

## D02LYF

**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

D02LYF is a diagnostic routine which may be called after a call of the integrator D02LAF.

### 2 Specification

```
SUBROUTINE D02LYF (NEQ, HNEXT, HUSED, HSTART, NSUCC, NFAIL, NATT, THRES,    &
                  THRESP, RWORK, LRWORK, IFAIL)
INTEGER          NEQ, NSUCC, NFAIL, NATT, LRWORK, IFAIL
REAL (KIND=nag_wp) HNEXT, HUSED, HSTART, THRES(NEQ), THRESP(NEQ),    &
                  RWORK(LRWORK)
```

### 3 Description

D02LYF permits you to extract information about the performance of D02LAF and the setting of some optional parameters. It may be called only after a call of D02LAF.

### 4 References

None.

### 5 Arguments

- |    |                                                                                                                                                                      |               |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| 1: | NEQ – INTEGER                                                                                                                                                        | <i>Input</i>  |
|    | <i>On entry:</i> the number of second-order ordinary differential equations solved by D02LAF. It must be the same as the argument NEQ supplied to D02LAF and D02LXF. |               |
| 2: | HNEXT – REAL (KIND=nag_wp)                                                                                                                                           | <i>Output</i> |
|    | <i>On exit:</i> the next step size which D02LAF, if called, would attempt.                                                                                           |               |
| 3: | HUSED – REAL (KIND=nag_wp)                                                                                                                                           | <i>Output</i> |
|    | <i>On exit:</i> the last successful step size used by D02LAF.                                                                                                        |               |
| 4: | HSTART – REAL (KIND=nag_wp)                                                                                                                                          | <i>Output</i> |
|    | <i>On exit:</i> the initial step size used on the current integration problem by D02LAF.                                                                             |               |
| 5: | NSUCC – INTEGER                                                                                                                                                      | <i>Output</i> |
|    | <i>On exit:</i> the number of steps attempted by D02LAF that have been successful since the start of the current problem.                                            |               |
| 6: | NFAIL – INTEGER                                                                                                                                                      | <i>Output</i> |
|    | <i>On exit:</i> the number of steps attempted by D02LAF that have failed since the start of the current problem.                                                     |               |

- 7: NATT – INTEGER *Output*  
*On exit:* the number of steps attempted before the initial step was successful. Over a large number of problems the cost of an attempted step of this type is approximately half that of a normal attempted step.
- 8: THRES(NEQ) – REAL (KIND=nag\_wp) array *Output*  
*On exit:* the *i*th solution threshold value used in the error control strategy. (See D02LXF.)
- 9: THRESP(NEQ) – REAL (KIND=nag\_wp) array *Output*  
*On exit:* the *i*th derivative threshold value used in the error control strategy. (See D02LXF.)
- 10: RWORK(LRWORK) – REAL (KIND=nag\_wp) array *Communication Array*  
*On entry:* this **must** be the same argument RWORK as supplied to D02LAF. It is used to pass information from D02LAF to D02LYF and therefore the contents of this array **must not** be changed before calling D02LYF.
- 11: LRWORK – INTEGER *Input*  
*On entry:* the dimension of the array RWORK as declared in the (sub)program from which D02LYF is called.  
 This must be the same argument LRWORK as supplied to D02LXF.
- 12: 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 = 1

D02LAF has not been called, or one or both of the arguments NEQ and LRWORK does not match the corresponding argument supplied to D02LXF.

This error exit can be caused if elements of RWORK have been overwritten.

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**

D02LYF is not thread safe and should not be called from a multithreaded user program. Please see Section 3.12.1 in How to Use the NAG Library and its Documentation for more information on thread safety.

D02LYF is not threaded in any implementation.

## **9 Further Comments**

None.

## **10 Example**

See Section 10 in D02LAF.

---