NAG Library Routine Document

D02QYF

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

D02QYF is a diagnostic routine which may be called after a call to the integrator routines D02QFF or D02QGF.

2 Specification

```
SUBROUTINE D02QYF (NEQG, INDEX, ITYPE, EVENTS, RESIDS, RWORK, LRWORK,      &
                   IWORK, LIWORK, IFAIL)
INTEGER            NEQG, INDEX, ITYPE, EVENTS(NEQG), LRWORK,               &
                   IWORK(LIWORK), LIWORK, IFAIL
REAL (KIND=nag_wp) RESIDS(NEQG), RWORK(LRWORK)
```

3 Description

D02QYF should be called only after a call to D02QFF or D02QGF results in the output value ROOT = .TRUE., indicating that a root has been detected. D02QYF permits you to examine information about the root detected, such as the indices of the event equations for which there is a root, the type of root (odd or even) and the residuals of the event equations.

4 References

None.

5 Parameters

1: NEQG – INTEGER *Input*

On entry: the number of event functions defined for the integration routine. It must be the same parameter NEQG supplied to the setup routine D02QWF and to the integration routine (D02QFF or D02QGF).

2: INDEX – INTEGER *Output*

On exit: the index k of the event equation $g_k(x, y, y') = 0$ for which the root has been detected.

3: ITYPE – INTEGER *Output*

On exit: information about the root detected for the event equation defined by INDEX. The possible values of ITYPE with their interpretations are as follows:

ITYPE = 1
: A simple root, or lack of distinguishing information available.

ITYPE = 2
: A root of even multiplicity is believed to have been detected, that is no change in sign of the event function was found.

ITYPE = 3
: A high-order root of odd multiplicity.

ITYPE $= 4$
A possible root, but due to high multiplicity or a clustering of roots accurate evaluation of the event function was prohibited by round-off error and/or cancellation.

In general, the accuracy of the root is less reliable for values of ITYPE > 1.

4: EVENTS(NEQG) – INTEGER array *Output*

On exit: information about the kth event function on a very small interval containing the root, T (see D02QFF and D02QGF), as output from the integration routine. All roots lying in this interval are considered indistinguishable numerically and therefore should be regarded as defining a root at T. The possible values of EVENTS(k) with their interpretations are as follows:

EVENTS(k) $= 0$
The kth event function did not have a root.

EVENTS(k) $= -1$
The kth event function changed sign from positive to negative about a root, in the direction of integration.

EVENTS(k) $= 1$
The kth event function changed sign from negative to positive about a root, in the direction of integration.

EVENTS(k) $= 2$
A root was identified, but no change in sign was observed.

5: RESIDS(NEQG) – REAL (KIND=nag_wp) array *Output*

On exit: the value of the kth event function computed at the root, T (see D02QFF and D02QGF).

6: RWORK(LRWORK) – REAL (KIND=nag_wp) array *Communication Array*

On entry: this **must** be the same parameter RWORK as supplied to D02QFF or D02QGF. It is used to pass information from the integration routine to D02QYF and therefore the contents of this array **must not** be changed before calling D02QYF.

7: LRWORK – INTEGER *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which D02QYF is called.

This must be the same parameter LRWORK as supplied to D02QWF.

8: IWORK(LIWORK) – INTEGER array *Communication Array*

On entry: this **must** be the same parameter IWORK as supplied to D02QFF or D02QGF. It is used to pass information from the integration routine to D02QYF and therefore the contents of this array **must not** be changed before calling D02QYF.

9: LIWORK – INTEGER *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which D02QYF is called.

This must be the same parameter LIWORK as supplied to D02QWF.

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 parameter, 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 $= 1$

An integration routine (D02QFF or D02QGF) has not been called, no root was detected or one or more of the parameters LRWORK, LIWORK and NEQG does not match the corresponding values supplied to D02QWF. Values for the arguments INDEX, ITYPE, EVENTS and RESIDS will not have been set.

This error exit may be caused by overwriting elements of IWORK.

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 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 D02QFF.