NAG Library Routine Document

D02QXF

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

D02QXF is a diagnostic routine which may be called after a call to either of the integration routines D02QFF and D02QGF.

2   Specification

```
SUBROUTINE D02QXF (NEQF, YP, TCURR, HLAST, HNEXT, ODLAST, ODNEXT, NSUCC,    &
                   NFAIL, TOLFAC, BADCMP, RWORK, LRWORK, IWORK, LIWORK,     &
                   IFAIL)
INTEGER            NEQF, ODLAST, ODNEXT, NSUCC, NFAIL, BADCMP, LRWORK,      &
                   IWORK(LIWORK), LIWORK, IFAIL
REAL (KIND=nag_wp) YP(NEQF), TCURR, HLAST, HNEXT, TOLFAC,                   &
                   RWORK(LRWORK)
```

3   Description

D02QXF permits you to extract information about the performance of D02QFF or D02QGF. It may only be called after a call to D02QFF or D02QGF.

4   References

None.

5   Parameters

1:    NEQF – INTEGER   *Input*

    On entry: the number of first-order ordinary differential equations solved by the integration routine. It must be the same parameter NEQF supplied to the setup routine D02QWF and the integration routines D02QFF or D02QGF.

2:    YP(NEQF) – REAL (KIND=nag_wp) array   *Output*

    On exit: the approximate derivative of the solution component y_i, as supplied in y_i on output from the integration routine at the output value of T. These values are obtained by the evaluation of $y' = f(x, y)$ except when the output value of the parameter T in the call to the integration routine is TOUT and TCURR $\neq$ TOUT, in which case they are obtained by interpolation.

3:    TCURR – REAL (KIND=nag_wp)   *Output*

    On exit: the value of the independent variable which the integrator has actually reached. TCURR will always be at least as far as the output value of the argument T (from the integration routine) in the direction of integration, but may be further.

4:    HLAST – REAL (KIND=nag_wp)   *Output*

    On exit: the last successful step size used by the integrator.

5:    HNEXT – REAL (KIND=nag_wp)   *Output*

    On exit: the next step size which the integration routine would attempt.

6:     ODLAST – INTEGER   *Output*

    On exit: the order of the method last used (successfully) by the integration routine.

7:     ODNEXT – INTEGER   *Output*

    On exit: the order of the method which the integration routine would attempt on the next step.

8:     NSUCC – INTEGER   *Output*

    On exit: the number of steps attempted by the integration routine that have been successful since the start of the current problem.

9:     NFAIL – INTEGER   *Output*

    On exit: the number of steps attempted by the integration routine that have failed since the start of the current problem.

10:    TOLFAC – REAL (KIND=nag_wp)   *Output*

    On exit: a tolerance scale factor, $TOLFAC \geq 1.0$, returned when the integration routine exits with $IFAIL = 3$. If RTOL and ATOL are uniformly scaled up by a factor of TOLFAC and D02QWF is called, the next call to the integration routine is deemed likely to succeed.

11:    BADCMP – INTEGER   *Output*

    On exit: if the integration routine returned with $IFAIL = 4$, then BADCMP specifies the index of the component which forced the error exit. Otherwise BADCMP is 0.

12:    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 D02QXF and therefore the contents of this array **must not** be changed before calling D02QXF.

13:    LRWORK – INTEGER   *Input*

    On entry: the dimension of the array RWORK as declared in the (sub)program from which D02QXF is called.

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

14:    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 D02QXF and therefore the contents of this array **must not** be changed before calling D02QXF.

15:    LIWORK – INTEGER   *Input*

    On entry: the dimension of the array IWORK as declared in the (sub)program from which D02QXF is called.

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

16:    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 or one or more of the parameters LRWORK, LIWORK and NEQF does not match the corresponding parameter supplied to D02QWF.

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

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

You should call D02QYF for information about any roots detected by D02QFF or D02QGF.

10   Example

See Section 10 in D02QFF.