NAG Library Routine Document

D02QWF

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

D02QWF is a setup routine which must be called prior to the first call of either of the integrators D02QFF and D02QGF and may be called prior to any continuation call to these routines.

2   Specification

```
SUBROUTINE D02QWF (STATEF, NEQF, VECTOL, ATOL, LATOL, RTOL, LRTOL,        &
                   ONESTP, CRIT, TCRIT, HMAX, MAXSTP, NEQG, ALTERG,       &
                   SOPHST, RWORK, LRWORK, IWORK, LIWORK, IFAIL)

INTEGER           NEQF, LATOL, LRTOL, MAXSTP, NEQG, LRWORK,               &
                  IWORK(LIWORK), LIWORK, IFAIL
REAL (KIND=nag_wp) ATOL(LATOL), RTOL(LRTOL), TCRIT, HMAX,                 &
                  RWORK(LRWORK)
LOGICAL           VECTOL, ONESTP, CRIT, ALTERG, SOPHST
CHARACTER(1)      STATEF
```

3   Description

D02QWF permits initialization of the integration method and setting of optional inputs prior to any call of D02QFF or D02QGF. It must be called before the first call of either of the routines D02QFF or D02QGF and it may be called before any continuation call of either of the routines D02QFF or D02QGF.

4   References

None.

5   Parameters

1:    STATEF – CHARACTER(1)   _Input/Output_

      On entry: specifies whether the integration routine (D02QFF or D02QGF) is to start a new system of ordinary differential equations, restart a system or continue with a system.

      STATEF = 'S'
           Start integration with a new differential system.

      STATEF = 'R'
           Restart integration with the current differential system.

      STATEF = 'C'
           Continue integration with the current differential system.

      Constraint: STATEF = 'S', 'R' or 'C'.

      On exit: is set to 'C', except that if an error is detected, STATEF is unchanged.

2:    NEQF – INTEGER   _Input_

      On entry: the number of ordinary differential equations to be solved by the integration routine. NEQF must remain unchanged on subsequent calls to D02QWF with STATEF = 'C' or 'R'.

      Constraint: NEQF $\geq$ 1.

3:     VECTOL – LOGICAL *Input*

On entry: specifies whether vector or scalar error control is to be employed for the local error test in the integration.

If VECTOL = .TRUE., then vector error control will be used and you must specify values of RTOL(i) and ATOL(i), for $i = 1, 2, \ldots, $ NEQF.

Otherwise scalar error control will be used and you must specify values of just RTOL(1) and ATOL(1).

The error test to be satisfied is of the form

$$\sqrt{\sum_{i=1}^{\text{NEQF}} \left(\frac{e_i}{w_i}\right)^2} \leq 1.0.$$

where w_i is defined as follows:

VECTOL	w_i
.TRUE.	RTOL(i) $\times |y_i|$ + ATOL(i)
.FALSE.	RTOL(1) $\times |y_i|$ + ATOL(1)

and e_i is an estimate of the local error in y_i, computed internally. VECTOL must remain unchanged on subsequent calls to D02QWF with STATEF = 'C' or 'R'.

4:     ATOL(LATOL) – REAL (KIND=nag_wp) array *Input*

On entry: the absolute local error tolerance (see VECTOL).

Constraint: ATOL(i) ≥ 0.0.

5:     LATOL – INTEGER *Input*

On entry: the dimension of the array ATOL as declared in the (sub)program from which D02QWF is called.

Constraints:

        if VECTOL = .TRUE., LATOL $\geq$ NEQF;
        if VECTOL = .FALSE., LATOL ≥ 1.

6:     RTOL(LRTOL) – REAL (KIND=nag_wp) array *Input*

On entry: the relative local error tolerance (see VECTOL).

Constraints:

        RTOL(i) ≥ 0.0;
        if ATOL(i) = 0.0, RTOL(i) $\geq 4.0 \times$ *machine precision*.

7:     LRTOL – INTEGER *Input*

On entry: the dimension of the array RTOL as declared in the (sub)program from which D02QWF is called.

Constraints:

        if VECTOL = .TRUE., LRTOL $\geq$ NEQF;
        if VECTOL = .FALSE., LRTOL ≥ 1.

8:     ONESTP – LOGICAL *Input*

On entry: the mode of operation of the integration routine. If ONESTP = .TRUE., the integration routine will operate in one-step mode, that is it will return after each successful step. Otherwise

the integration routine will operate in interval mode, that is it will return at the end of the integration interval.

9:   CRIT – LOGICAL   *Input*

On entry: specifies whether or not there is a value for the independent variable beyond which integration is not to be attempted. Setting CRIT = .TRUE. indicates that there is such a point, whereas CRIT = .FALSE. indicates that there is no such restriction.

10:   TCRIT – REAL (KIND=nag_wp)   *Input*

On entry: with CRIT = .TRUE., TCRIT must be set to a value of the independent variable beyond which integration is not to be attempted. Otherwise TCRIT is not referenced.

11:   HMAX – REAL (KIND=nag_wp)   *Input*

On entry: if HMAX $\neq$ 0.0, a bound on the absolute step size during the integration is taken to be |HMAX|.

If HMAX = 0.0, no bound is assumed on the step size during the integration.

A bound may be required if there are features of the solution on very short ranges of integration which may be missed. You should try HMAX = 0.0 first.

Note: this parameter only affects the step size if the option CRIT = .TRUE. is being used.

12:   MAXSTP – INTEGER   *Input*

On entry: a bound on the number of attempted steps in any one call to the integration routine. If MAXSTP $\leq$ 0 on entry, a value of 1000 is used.

13:   NEQG – INTEGER   *Input*

On entry: specifies whether or not root-finding is required in D02QFF or D02QGF.

NEQG $\leq$ 0
    No root-finding is attempted.

NEQG $>$ 0
    Root-finding is required and NEQG event functions will be specified for the integration routine.

14:   ALTERG – LOGICAL   *Input/Output*

On entry: specifies whether or not the event functions have been redefined. ALTERG need not be set if STATEF = 'S'. On subsequent calls to D02QWF, if NEQG has been set positive, then ALTERG = .FALSE. specifies that the event functions remain unchanged, whereas ALTERG = .TRUE. specifies that the event functions have changed. Because of the expense in reinitializing the root searching procedure, ALTERG should be set to .TRUE. only if the event functions really have been altered. ALTERG need not be set if the root-finding option is not used.

On exit: is set to .FALSE..

15:   SOPHST – LOGICAL   *Input*

On entry: the type of search technique to be used in the root-finding. If SOPHST = .TRUE. then a sophisticated and reliable but expensive technique will be used, whereas for SOPHST = .FALSE. a simple but less reliable technique will be used. If NEQG $\leq$ 0, SOPHST is not referenced.

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

This **must** be the same parameter RWORK supplied to the integration routine. It is used to pass information to the integration routine and therefore the contents of this array **must not** be changed before calling the integration routine.

17:    LRWORK – INTEGER   *Input*

On entry: the dimension of the array RWORK as declared in the (sub)program from which D02QWF is called.

Constraint: $\text{LRWORK} \geq 21 \times (1 + \text{NEQF}) + 2 \times J + K \times \text{NEQG} + 2$, where

$$J = \begin{cases} \text{NEQF} & \text{if VECTOL} = \text{.TRUE.} \\ 1 & \text{if VECTOL} = \text{.FALSE.} \end{cases}$$

and

$$K = \begin{cases} 14 & \text{if SOPHST} = \text{.TRUE.} \\ 5 & \text{if SOPHST} = \text{.FALSE.} \end{cases}$$

.

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

This **must** be the same parameter IWORK supplied to the integration routine. It is used to pass information to the integration routine and therefore the contents of this array **must not** be changed before calling the integration routine.

19:    LIWORK – INTEGER   *Input*

On entry: the dimension of the array IWORK as declared in the (sub)program from which D02QWF is called.

Constraints:

       if $\text{SOPHST} = \text{.TRUE.}$, $\text{LIWORK} \geq 21 + 4 \times \text{NEQG}$;
       if $\text{SOPHST} = \text{.FALSE.}$, $\text{LIWORK} \geq 21 + \text{NEQG}$.

20:    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: $\text{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6   Error Indicators and Warnings

If on entry $\text{IFAIL} = 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

$\text{IFAIL} = 1$

    Illegal input detected.

$\text{IFAIL} = -99$

    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

Prior to a continuation call of the integration routine, you may reset any of the optional parameters by calling D02QWF with STATEF $=$ 'C'. You may reset:

HMAX	to alter the maximum step size selection;
RTOL, ATOL	to change the error requirements;
MAXSTP	to increase or decrease the number of attempted steps before an error exit is returned;
ONESTP	to change the operation mode of the integration routine;
CRIT, TCRIT	to alter the point beyond which integration must not be attempted; and
NEQG, ALTERG, SOPHST	to alter the number and type of event functions, and also the search method.

If the behaviour of the system of differential equations has altered and you wish to restart the integration method from the value of T output from the integration routine (see D02QFF and D02QGF), then STATEF should be set to STATEF $=$ 'R' and any of the optional parameters may be reset also. If you want to redefine the system of differential equations or start a new integration problem, then STATEF should be set to STATEF $=$ 'S'. Resetting STATEF $=$ 'R' or 'S' on normal continuation calls causes a restart in the integration process, which is very inefficient when not needed.

10 Example

See Section 10 in D02QFF and D02QGF.