NAG Toolbox for MATLAB

d02qw

1 Purpose

d02qw is a setup function which must be called by you prior to the first call of either of the integration functions d02qf and d02qg, and may be called prior to any subsequent continuation call to these functions.

2 Syntax

```
[statef, alterg, rwork, iwork, ifail] = d02qw(statef, neqf, vectol, atol, rtol, onestp, crit, tcrit, hmax, maxstp, neqg, alterg, sophst, rwork, iwork, 'latol', latol, 'lrtol', lrtol, 'lrwork', lrwork, 'liwork', liwork)
```

3 Description

d02qw permits initialization of the integration method and setting of optional inputs prior to any call of d02qf or d02qg.

It must be called before the first call of either of the functions d02qf or d02qg and it may be called before any continuation call of either of the functions d02qf or d02qg.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **statef – string**

Specifies whether the integration function (d02qf or d02qg) 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'.

2: **neqf – int32 scalar**

the number of ordinary differential equations to be solved by the integration function. **neqf** must remain unchanged on subsequent calls to d02qw with **statef** = 'C' or 'R'.

Constraint: **neqf** ≥ 1.

3: **vectol – logical scalar**

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

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

where w_i is defined as follows:

vectol	w_i
true	**rtol**$(i) \times \lvert y_i \rvert +$ **atol**(i)
false	**rtol**$(1) \times \lvert y_i \rvert +$ **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 d02qw with **statef** = 'C' or 'R'.

4: **atol**(**latol**) – **double array**

The absolute local error tolerance (see **vectol**).

Constraint: **atol**$(i) \geq 0.0$.

5: **rtol**(**lrtol**) – **double array**

The relative local error tolerance (see **vectol**).

Constraints:

rtol$(i) \geq 0.0$;
if **atol**$(i) = 0.0$, **rtol**$(i) \geq 4.0 \times$ ***machine precision***.

6: **onestp** – **logical scalar**

The mode of operation of the integration function. If **onestp** = **true**, the integration function will operate in one-step mode, that is it will return after each successful step. Otherwise the integration function will operate in interval mode, that is it will return at the end of the integration interval.

7: **crit** – **logical scalar**

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.

8: **tcrit** – **double scalar**

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.

9: **hmax** – **double scalar**

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.

10: **maxstp – int32 scalar**

A bound on the number of attempted steps in any one call to the integration function. If **maxstp** ≤ 0 on entry, a value of 1000 is used.

11: **neqg – int32 scalar**

Specifies whether or not root-finding is required in d02qf or d02qg.

neqg ≤ 0

No root-finding is attempted.

neqg > 0

Root-finding is required and **neqg** event functions will be specified for the integration function.

12: **alterg – logical scalar**

Specifies whether or not the event functions have been redefined. **alterg** need not be set if **statef** = 'S'. On subsequent calls to d02qw, 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.

13: **sophst – logical scalar**

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** ≤ 0, **sophst** is not referenced.

14: **rwork**(**lrwork**) **– double array**

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

15: **iwork**(**liwork**) **– int32 array**

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

5.2 Optional Input Parameters

1: **latol – int32 scalar**

Default: The dimension of the array **atol**.

Constraints:

if **vectol** = **true**, **latol** $\geq$ **neqf**;
if **vectol** = **false**, **latol** ≥ 1.

2: **lrtol – int32 scalar**

Default: The dimension of the array **rtol**.

Constraints:

if **vectol** = **true**, **lrtol** $\geq$ **neqf**;
if **vectol** = **false**, **lrtol** ≥ 1.

3: **lrwork – int32 scalar**

Default: The dimension of the array **rwork**.

Constraint: **lrwork** $\geq 21 \times (1 + \mathbf{neqf}) + 2 \times \mathrm{J} + \mathrm{K} \times \mathbf{neqg} + 2$, where

$$\mathrm{J} = \begin{cases} \mathbf{neqf} & \text{if } \mathbf{vectol} = \mathbf{true} \\ 1 & \text{if } \mathbf{vectol} = \mathbf{false} \end{cases}$$

and

$$\mathrm{K} = \begin{cases} 14 & \text{if } \mathbf{sophst} = \mathbf{true} \\ 5 & \text{if } \mathbf{sophst} = \mathbf{false} \end{cases}$$

4: **liwork – int32 scalar**

Default: The dimension of the array **iwork**.

Constraints:

if **sophst** = **true**, **liwork** $\geq 21 + 4 \times$ **neqg**;
if **sophst** = **false**, **liwork** $\geq 21 +$ **neqg**.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **statef – string**

Is set to 'C', except that if an error is detected, **statef** is unchanged.

2: **alterg – logical scalar**

Is set to **false**.

3: **rwork**(**lrwork**) **– double array**

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

4: **iwork**(**liwork**) **– int32 array**

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

5: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

Illegal input detected.

7 Accuracy

Not applicable.

8 Further Comments

Prior to a continuation call of the integration function, you may reset any of the optional parameters by calling d02qw 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 function;
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 function (see d02qf and d02qg), 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.

9 Example

d02qf_fcn.m

```
function f = fcn(neqf, x, y)
  f=zeros(neqf,1);
  f(1)=y(2);
  f(2)=-y(1);
```

d02qf_g.m

```
function result=g(neqf, x, y, yp, k)
  if (k == 1)
    result = yp(1);
  else
    result = y(1);
  end
```

```
t = 0;
y = [0;
     1];
tout = 10;
neqg = int32(2);
rwork = zeros(97,1);
iwork = zeros(29,1,'int32');
[statefOut, altergOut, rwork, iwork, ifail] = ...
      d02qw('S', int32(2), true, [1e-06;1e-06], [0.0001; 0.0001], false,
true, ...
    10, 0, int32(0), int32(2), false, true, rwork, iwork);
[tOut, yOut, root, rworkOut, iworkOut, ifail] = ...
    d02qf('d02qf_fcn', t, y, tout, 'd02qf_g', neqg, rwork, iwork)
```

```
tOut =
     0
yOut =
     0
     1
root =
     1
```

```
rworkOut =
     array elided
iworkOut =
           2
           2
           0
           0
           1
          97
          29
           1
           1
           1
           2
           1
           0
           0
           1
           0
        1000
           0
           0
           0
           0
           2
           0
           0
           0
           0
           0
           0
           0
ifail =
           0
```