NAG Toolbox for MATLAB

d02qx

1 Purpose

d02qx is a diagnostic function which may be called after a call to either of the integration functions d02qf and d02qg.

2 Syntax

```
[yp, tcurr, hlast, hnext, odlast, odnext, nsucc, nfail, tolfac, badcmp,
ifail] = d02qx(neqf, rwork, iwork, 'lrwork', lrwork, 'liwork', liwork)
```

3 Description

d02qx permits you to extract information about the performance of d02qf or d02qg. It may only be called after a call to d02qf or d02qg.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **neqf – int32 scalar**

the number of first-order ordinary differential equations solved by the integration function. It must be the same parameter **neqf** supplied to the setup function d02qw and the integration functions d02qf or d02qg.

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

This **must** be the same parameter **rwork** as supplied to d02qf or d02qg. It is used to pass information from the integration function to d02qx and therefore the contents of this array **must not** be changed before calling d02qx.

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

This **must** be the same parameter **iwork** as supplied to d02qf or d02qg. It is used to pass information from the integration function to d02qx and therefore the contents of this array **must not** be changed before calling d02qx.

5.2 Optional Input Parameters

1: **lrwork – int32 scalar**

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

This must be the same parameter **lrwork** as supplied to d02qw.

2: **liwork – int32 scalar**

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

This must be the same parameter **liwork** as supplied to d02qw.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **yp(neqf) – double array**

The approximate derivative of the solution component y_i, as supplied in y_i on output from the integration function 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 function is **tout** and **tcurr** $\neq$ **tout**, in which case they are obtained by interpolation.

2: **tcurr – double scalar**

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 function) in the direction of integration, but may be further.

3: **hlast – double scalar**

The last successful step size used by the integrator.

4: **hnext – double scalar**

The next step size which the integration function would attempt.

5: **odlast – int32 scalar**

The order of the method last used (successfully) by the integration function.

6: **odnext – int32 scalar**

The order of the method which the integration function would attempt on the next step.

7: **nsucc – int32 scalar**

The number of steps attempted by the integration function that have been successful since the start of the current problem.

8: **nfail – int32 scalar**

The number of steps attempted by the integration function that have failed since the start of the current problem.

9: **tolfac – double scalar**

A tolerance scale factor, **tolfac** ≥ 1.0, returned when the integration function exits with **ifail** $= 3$. If **rtol** and **atol** are uniformly scaled up by a factor of **tolfac** and d02qw is called, the next call to the integration function is deemed likely to succeed.

10: **badcmp – int32 scalar**

If the integration function returned with **ifail** $= 4$, then **badcmp** specifies the index of the component which forced the error exit. Otherwise **badcmp** is 0.

11: **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

An integration function (d02qf or d02qg) has not been called or one or more of the parameters **lrwork**, **liwork** and **neqf** does not match the corresponding parameter supplied to d02qw.

This error exit may be caused by overwriting elements of **rwork**.

7 Accuracy

Not applicable.

8 Further Comments

You should call d02qy for information about any roots detected by d02qf or d02qg.

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');
neqf = int32(2);
[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);
[t, y, root, rwork, iwork, ifail] = ...
    d02qf('d02qf_fcn', t, y, tout, 'd02qf_g', neqg, rwork, iwork);
[yp, tcurr, hlast, hnext, odlast, odnext, nsucc, nfail, tolfac, badcmp,
ifail] = ...
    d02qx(neqf, rwork, iwork)
```

```
yp =
     1
     0
tcurr =
     0
hlast =
     0
```

```
hnext =
     0
odlast =
           0
odnext =
           0
nsucc =
           0
nfail =
           0
tolfac =
     1
badcmp =
           0
ifail =
           0
```