NAG Toolbox for MATLAB

d02nz

1 Purpose

d02nz is a setup function which must be called, if optional inputs need resetting, prior to a continuation call to any of the integrators in sub-chapter D02M/N.

2 Syntax

```
[rwork, ifail] = d02nz(ldysav, tcrit, h, hmin, hmax, maxstp, mxhnil, rwork)
```

3 Description

d02nz is provided to permit you to reset many of the parameters which control the integration 'on the fly', that is in conjunction with the interrupt facility permitted through the parameter **itask** of the integrator (e.g., see d02nb). In addition to a number of parameters which you can set initially through one of the integrator setup functions, the step size to be attempted on the next step may be changed.

4 References

See the D02M/N Sub-chapter Introduction.

5 Parameters

5.1 Compulsory Input Parameters

1: **ldysav – int32 scalar**

the value used for the parameter **ldysav** when calling the integrator.

Constraint: **ldysav** ≥ 1.

2: **tcrit – double scalar**

A point beyond which integration must not be attempted. The use of **tcrit** is described under the parameter **itask** in the specification for the integrator (e.g., see d02nb). A value, 0.0 say, must be specified even if **itask** subsequently specifies that **tcrit** will not be used.

3: **h – double scalar**

The next step size to be attempted. Set **h** $= 0.0$ if the current value of **h** is not to be changed.

4: **hmin – double scalar**

The minimum absolute step size to be allowed. Set **hmin** $= 0.0$ if this option is not required. Set **hmin** < 0.0 if the current value of **hmin** is not to be changed.

5: **hmax – double scalar**

The maximum absolute step size to be allowed. Set **hmax** $= 0.0$ if this option is not required. Set **hmax** < 0.0 if the current value of **hmax** is not to be changed.

6: **maxstp – int32 scalar**

The maximum number of steps to be attempted during one call to the integrator after which it will return with **ifail** $= 2$ (see d02nc). Set **maxstp** $= 0$ if this option is not required. Set **maxstp** < 0 if the current value of **maxstp** is not to be changed.

7: **mxhnil – int32 scalar**

The maximum number of warnings printed (if **itrace** ≥ 0, e.g., see d02nb) per problem when $t + h = t$ on a step ($h =$ current step size). If **mxhnil** ≤ 0, a default value of 10 is assumed.

8: **rwork**$(\mathbf{50} + \mathbf{4} \times \mathbf{ldysav})$ **– double array**

This must be the same workspace array as the array **rwork** supplied to the integrator. It is used to pass information from the integrator to d02nz and therefore its contents must not be changed before calling d02nz.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **rwork**$(\mathbf{50} + \mathbf{4} \times \mathbf{ldysav})$ **– double array**

This must be the same workspace array as the array **rwork** supplied to the integrator. It is used to pass information from the integrator to d02nz and therefore its contents must not be changed before calling d02nz.

2: **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 $= 1$

ldysav < 1.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

d02nc_fcn.m

```
function [f, ires] = fcn(neq, t, y, ires)
      f = zeros(3,1);
      f(1) = -0.04d0*y(1) + 1.0d4*y(2)*y(3);
      f(2) = 0.04d0*y(1) - 1.0d4*y(2)*y(3) - 3.0d7*y(2)*y(2);
      f(3) = 3.0d7*y(2)*y(2);
```

d02nc_jac.m

```
function p = jac(neq, t, y, h, d, ml, mu, pIn)
  p = zeros(ml+mu+1, neq);
  hxd = h*d;
  p(1,1) = 1.0d0 - hxd*(-0.04d0);
  p(2,1) = -hxd*(1.0d4*y(3));
  p(3,1) = -hxd*(1.0d4*y(2));
  p(1,2) = -hxd*(0.04d0);
  p(2,2) = 1.0d0 - hxd*(-1.0d4*y(3)-6.0d7*y(2));
  p(3,2) = -hxd*(-1.0d4*y(2));
  p(1,3) = -hxd*(6.0d7*y(2));
  p(2,3) = 1.0d0 - hxd*(0.0d0);
```

d02nc_monitr.m

```
function [hnext, y, imon, inln, hmin, hmax] = ...
    monitr(neq, neqmax, t, hlast, hnext, y, ydot, ysave, r, acor, imon,
hmin, hmax, nqu)
  inln=int32(0);
```

```
neqmax = int32(3);
tcrit = 0;
h = 0.7;
hmin = 1e-10;
hmax = 10;
maxstp = int32(200);
mxhnil = int32(5);
rwork = zeros(62,1);
neq = int32(3);
t = 0;
tout = 5;
y = [1; 0; 0];
rtol = [0.0001];
atol = [1e-07; 1e-08; 1e-07];
itol = int32(2);
inform = zeros(32,1,'int32');
ysave = zeros(3, 14);
wkjac = zeros(15, 1);
jacpvt = zeros(3, 1, 'int32');
itask = int32(1);
itrace = int32(0);
[const, rwork, ifail] = ...
    d02nw(int32(3), int32(14), int32(11), zeros(6), 0, 1e-10, 10, 0, ...
    int32(200), int32(5), 'Average-L2', rwork);
[rwork, ifail] = ...
         d02nt(int32(3),  int32(3),  'Analytical',  int32(1),  int32(2),
int32(15), ...
    int32(3), rwork);
[t, y, ydot, rwork, inform, ysave, wkjac, jacpvt, ifail] = ...
    d02nc(neq, t, tout, y, rwork, rtol, atol, itol, inform, 'd02nc_fcn',
...
    ysave, 'd02nc_jac', wkjac, jacpvt, 'd02nc_monitr', itask, itrace);
[rwork, ifail] = d02nz(neqmax, tcrit, h, hmin, hmax, maxstp, mxhnil,
rwork)
```

```
rwork =
     array elided
ifail =
           0
```