NAG Toolbox for MATLAB

d02nt

1 Purpose

d02nt is a setup function which you must call prior to an integrator in sub-chapter D02M/N, if banded matrix linear algebra is required.

2 Syntax

```
[rwork, ifail] = d02nt(neq, ldysav, jceval, ml, mu, nwkjac, njcpvt, rwork)
```

3 Description

d02nt defines the linear algebra to be used as banded matrix linear algebra, permits you to specify the method for calculating the Jacobian and checks the validity of certain input values.

4 References

See the D02M/N Sub-chapter Introduction.

5 Parameters

5.1 Compulsory Input Parameters

1: **neq – int32 scalar**

The number of differential equations.

Constraint: $1 \leq$ **neq** $\leq$ **ldysav**.

2: **ldysav – int32 scalar**

a bound on the maximum number of differential equations to be solved during the integration.

Constraint: **ldysav** $\geq$ **neq**.

3: **jceval – string**

Specifies the technique to be used to compute the Jacobian as follows:

jceval = 'N'

The Jacobian is to be evaluated numerically by the integrator. If this option is used, then the actual argument corresponding to (sub)program **jac** in the call to d02nc or d02nh must be either "temp_tag_xref_d02ncz d02nhz"done**d02ncz d02nhz**missing entity d02ncz d02nhz respectively.

jceval = 'A'

You must supply a (sub)program to evaluate the Jacobian on a call to the integrator.

jceval = 'D'

The default choice is to be made. In this case 'D' is interpreted as 'N'.

Only the first character of the actual parameter **jceval** is passed to d02nt; hence it is permissible for the actual argument to be more descriptive, e.g., ‘Numerical’, ‘Analytical’ or ‘Default’, on a call to d02nt.

Constraint: **jceval** = 'N', 'A' or 'D'.

4: **ml – int32 scalar**

m_L, the number of subdiagonals in the band.

Constraint: $0 \leq \mathbf{ml} \leq \mathbf{neq} - 1$.

5: **mu – int32 scalar**

m_U, the number of superdiagonals in the band.

Constraint: $0 \leq \mathbf{mu} \leq \mathbf{neq} - 1$.

6: **nwkjac – int32 scalar**

The size of the workspace array **wkjac**, which you are supplying to the integrator, as declared in the (sub)program from which d02nt is called.

Constraint: $\mathbf{nwkjac} \geq (2 \times \mathbf{ml} + \mathbf{mu} + 1) \times \mathbf{ldysav}$.

7: **njcpvt – int32 scalar**

The size of the workspace array **jacpvt**, which you are supplying to the integrator, as declared in the (sub)program from which d02nt is called.

Constraint: $\mathbf{njcpvt} \geq \mathbf{ldysav}$.

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 setup function to the integrator and therefore the contents of this array must not be changed before calling the integrator.

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 setup function to the integrator and therefore the contents of this array must not be changed before calling the integrator.

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 $= 1$

On entry, **jceval** $\neq$ 'N', 'A' or 'D',
or **neq** < 1,
or **ml** < 0 or **ml** $>$ **neq** $- 1$,
or **mu** < 0 or **mu** $>$ **neq** $- 1$,
or **neq** $>$ **ldysav**,

or **njcpvt** $<$ **ldysav**,
or **nwkjac** $<$ $(2 \times$ **ml** $+$ **mu** $+ 1) \times$ **ldysav**.

7 Accuracy

Not applicable.

8 Further Comments

d02nt must be called as a setup function before a call to either d02nc or d02nh and may be called as the linear algebra setup function before a call to either d02nm or d02nn.

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, ngu)
  inln=int32(0);
```

```
neq = int32(3);
t = 0;
tout = 5;
y = [1;
     0;
     0];
rwork = zeros(62, 1);
rtol = [0.0001];
atol = [1e-07;
     1e-08;
     1e-07];
itol = int32(2);
inform = zeros(23, 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);
[tOut, yOut, ydot, rworkOut, informOut, ysaveOut, wkjacOut, ...
 jacpvtOut, ifail] = ...
    d02nc(neq, t, tout, y, rwork, rtol, atol, itol, inform, 'd02nc_fcn',
...
    ysave, 'd02nc_jac', wkjac, jacpvt, 'd02nc_monitr', itask, itrace)
```

```
tOut =
     5
yOut =
    0.8915
    0.0000
    0.1085
ydot =
   -0.0124
   -0.0000
    0.0124
rworkOut =
     array elided
informOut =
          57
         248
          12
           4
           4
         246
           0
           3
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
ysaveOut =
  Columns 1 through 7
    0.8864   -0.0092    0.0004   -0.0000    0.0000         0         0
    0.0000   -0.0000    0.0000   -0.0000    0.0000         0         0
    0.1136    0.0092   -0.0004    0.0000   -0.0000         0         0
  Columns 8 through 14
         0         0         0         0    0.0000    0.0000    0.0000
         0         0         0         0    0.0000    0.0000    0.0000
         0         0         0         0   -0.0000   -0.0000   -0.0000
wkjacOut =
    0.9934
 -142.7005
   -0.0387
         0
    0.0027
    0.0385
         0
         0
    0.9767
         0
```

```
         0
         0
   -0.0066
   -0.6194
    0.0043
jacpvtOut =
           1
           2
           3
ifail =
           0
```