NAG Toolbox for MATLAB

d02ns

1 Purpose

d02ns is a setup function which must be called prior to an integrator in sub-chapter D02M/N, if full matrix linear algebra is required.

2 Syntax

```
[rwork, ifail] = d02ns(neq, ldysav, jceval, nwkjac, rwork)
```

3 Description

d02ns defines the linear algebra to be used as full 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.

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 d02nb or d02ng must be either "temp_tag_xref_d02nbz d02ngz"done**d02nbz d02ngz**missing_entity d02nbz d02ngz 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 d02ns; hence it is permissible for the actual argument to be more descriptive ‘Numerical’, ‘Analytical’ or ‘Default’ on a call to d02ns.

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

4: **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 d02ns is called.

Constraint: **nwkjac** $\geq$ **ldysav** $\times$ (**ldysav** + 1).

5: **rwork**(**50** + **4** $\times$ **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**(**50** + **4** $\times$ **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, **neq** $<$ 1,
or **neq** $>$ **ldysav**,
or **nwkjac** $<$ **ldysav** $\times$ (**ldysav** + 1),
or **jceval** $\neq$ 'N', 'A' or 'D'.

7 Accuracy

Not applicable.

8 Further Comments

d02ns must be called as a setup function before a call to either d02nb or d02ng and may be called as the linear algebra setup function before a call to either d02nm or d02nn.

9 Example

d02nb_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);
```

d02nb_jac.m

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

d02nb_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);
```

```
neq = int32(3);
t = 0;
tout = 10;
y = [1;
     0;
     0];
rwork = zeros(62, 1);
rtol = [0.0001];
atol = [1e-07];
itol = int32(1);
inform = zeros(23, 1, 'int32');
ysave = zeros(3, 6);
wkjac = zeros(12, 1);
itask = int32(4);
itrace = int32(0);
[const, rwork, ifail] = ...
   d02nv(int32(3), int32(6), int32(5), 'Newton', false, zeros(6), ...
    10, 1e-10, 10, 0, int32(200), int32(5), 'Average-L2', rwork);
[rwork,  ifail]  =  d02ns(int32(3),  int32(3),  'Numerical',  int32(12),
rwork);
[tOut, yOut, ydot, rworkOut, informOut, ysaveOut, wkjacOut, ifail] = ...
     d02nb(neq, t, tout, y, rwork, rtol, atol, itol, inform, 'd02nb_fcn',
...
    ysave, 'd02nb_jac', wkjac, 'd02nb_monitr', itask, itrace)
```

```
tOut =
    10
yOut =
    0.8414
    0.0000
    0.1586
ydot =
   -0.0079
   -0.0000
    0.0079
rworkOut =
     array elided
informOut =
          55
         132
          17
```

```
           3
           3
          79
           0
           3
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
           0
ysaveOut =
    0.8414   -0.0041    0.0001   -0.0000    0.0000   -0.0000
    0.0000   -0.0000    0.0000   -0.0000    0.0000   -0.0000
    0.1586    0.0041   -0.0001    0.0000   -0.0000    0.0000
wkjacOut =
    1.0100
         0
   -0.0100
 -380.6996
 -247.4940
  629.1936
   -0.0408
   -0.0040
    2.5831
    1.0000
    3.0000
    3.0000
ifail =
           0
```