NAG Toolbox

nag_ode_ivp_stiff_sparjac_setup (d02nu)

1 Purpose

nag_ode_ivp_stiff_sparjac_setup (d02nu) is a setup function which must be called prior to an integrator in Sub-chapter D02M–N, if sparse matrix linear algebra is required.

2 Syntax

```
[jacpvt, rwork, ifail] = nag_ode_ivp_stiff_sparjac_setup(neq, neqmax, jceval,
nwkjac, ia, ja, njcpvt, sens, u, eta, lblock, rwork, 'nia', nia, 'nja', nja,
'isplit', isplit)
[jacpvt, rwork, ifail] = d02nu(neq, neqmax, jceval, nwkjac, ia, ja, njcpvt,
sens, u, eta, lblock, rwork, 'nia', nia, 'nja', nja, 'isplit', isplit)
```

3 Description

nag_ode_ivp_stiff_sparjac_setup (d02nu) defines the linear algebra to be used as sparse matrix linear algebra, permits you to specify the method for calculating the Jacobian and its structure, 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** – INTEGER

The number of differential equations.

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

2: **neqmax** – INTEGER

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

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

3: **jceval** – CHARACTER(1)

Specifies the technique to be used to compute the Jacobian.

jceval = 'N'
The sparsity structure and the value of the Jacobian are to be determined numerically by the integrator.

jceval = 'S'
The sparsity structure of the Jacobian is supplied in the arrays **ia** and **ja** but its value is to be determined numerically. This is the recommended mode of operation unless it is a simple matter to supply the Jacobian.

jceval = 'A'
The Jacobian will be evaluated by calls to **jac**. The sparsity structure will be estimated by calls to **jac**; that is, no explicit sparsity structure need be supplied in the arrays **ia** and **ja**.

jceval = 'F'
The sparsity structure of the Jacobian is supplied in **ia** and **ja**, and its value will be determined by calls to **jac**. This is the recommended mode of operation if the **jac** is simple to form.

jceval = 'D'
The default choice is to be made. In this case `D' is interpreted as `S'.

If the sparsity structure is supplied in arrays **ia** and **ja**, then any evidence from the numerical or analytical formation of the Jacobian that this structure is not correct, is ignored.

Only the first character of the actual argument **jceval** is passed to nag_ode_ivp_stiff_sparjac_setup (d02nu); hence it is permissible for the actual argument to be more descriptive, e.g., 'Numerical', 'Structural', 'Analytical', 'Full information' or 'Default' in a call to nag_ode_ivp_stiff_sparjac_setup (d02nu).

If the option **jceval** = 'N', 'S' or 'D' is used then the actual argument corresponding to **jac** in the call to nag_ode_ivp_stiff_exp_sparjac (d02nd) or nag_ode_ivp_stiff_imp_sparjac (d02nj) must be either nag_ode_ivp_stiff_exp_sparjac_dummy_jac (d02ndz) or nag_ode_ivp_stiff_imp_sparjac_dummy_jac (d02njz) respectively.

If integration is to be performed by reverse communication (nag_ode_ivp_stiff_exp_revcom (d02nm) or nag_ode_ivp_stiff_imp_revcom (d02nn)) then **jceval** should be set to either `N' or `A'. In this case **ia** and **ja** are not used and their lengths may be set to 1.

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

4: **nwkjac** – INTEGER

Suggested value: **nwkjac** $= 4 \times$ **neqmax** if **jceval** = 'N' or 'A'. If **nwkjac** is less than this estimate, then a message is printed on the current advisory message unit (see nag_file_set_unit_advisory (x04ab)), and execution continues.

The size of the array **wkjac**, which you are supplying to the integrator, as declared in the (sub) program from which nag_ode_ivp_stiff_sparjac_setup (d02nu) is called.

Constraint: if **jceval** = 'S', 'F' or 'D', **nwkjac** $\geq nelement + 2 \times$ **neq**, where $nelement$ is the total number of nonzeros.

5: **ia**(**nia**) – INTEGER array

If **jceval** = 'S', 'F' or 'D', **ia** must contain details of the sparsity pattern to be used for the Jacobian. See **ja**.

ia is not used if **jceval** = 'N' or 'A'.

6: **ja**(**nja**) – INTEGER array

If **jceval** = 'S', 'F' or 'D', **ja** must contain details of the sparsity pattern to be used for the Jacobian. **ja** contains the row indices where nonzero elements occur, reading in column-wise order, and **ia** contains the starting locations in **ja** of the descriptions of columns $1, 2, \ldots,$ **neq** in that order, with **ia**$(1) = 1$. Thus for each column index $j = 1, 2, \ldots,$ **neq**, the values of the row index i in column j where a nonzero element may occur are given by

$$i = \mathbf{ja}(k)$$

where $\mathbf{ia}(j) \leq k < \mathbf{ia}(j+1)$.

Thus the total number of nonzeros, $nelement$, must be **ia**(**neq** $+ 1) - 1$. For example, for the following matrix

$$\begin{pmatrix} x & 0 & x & 0 & 0 \\ 0 & x & x & x & 0 \\ x & x & x & 0 & 0 \\ x & 0 & 0 & x & x \\ 0 & 0 & 0 & x & x \end{pmatrix}$$

where x represents nonzero elements (13 in all) the arrays **ia** and **ja** should be

ia(k)	1	4	6	9	12	14							
ja(k)	1	3	4	2	3	1	2	3	2	4	5	4	5

ja is not used if **jceval** = 'N' or 'A'.

7: **njcpvt** – INTEGER

Suggested value: **njcpvt** $= 20 \times$ **neqmax** if **jceval** = 'N' or 'A'. If **njcpvt** is less than this estimate, then a message is printed on the current advisory message unit (see nag_file_set_unit_advisory (x04ab)), and execution continues.

The length of the array **jacpvt**, which you are supplying to the integrator, as dimensioned in the sub(program) from which nag_ode_ivp_stiff_sparjac_setup (d02nu) is called.

Constraint: if **jceval** = 'S', 'F' or 'D', **njcpvt** $\geq 3 \times nelement + 14 \times$ **neq**, where $nelement$ is the total number of nonzeros.

8: **sens** – REAL (KIND=nag_wp)

A threshold argument used to determine whether or not a matrix element is zero; when **sens** is set to 0.0 on entry, the function will use **sens** $= 100.0 \times$ ***machine precision***. Otherwise the absolute value of **sens** is used.

9: **u** – REAL (KIND=nag_wp)

Should have a value between 0.0 and 0.9999. Otherwise a default value of 0.1 is used. When the sparsity pattern has been evaluated, the first Jacobian computed is decomposed with **u** governing the choice of pivots; subsequent Jacobian decompositions use the same pattern of decomposition until the sparsity pattern is re-evaluated. When searching a row for a pivot, any element is excluded from the search which is less than **u** times the largest of those elements in the row available as pivots. Thus decreasing **u** biases the algorithm towards maintaining sparsity at the expense of numerical stability.

10: **eta** – REAL (KIND=nag_wp)

A relative pivot threshold, below which on subsequent decompositions (as described under **u**), an internal error is provoked.

eta > 1.0
No check on pivot size is made.

eta ≤ 0.0
The default value **eta** $=$ 1.0e−4 is used.

11: **lblock** – LOGICAL

Indicates if preordering is used before decomposition.

If **lblock** $= true$, on entry, the Jacobian matrix is preordered to block lower triangular form before a decomposition is performed (this is the recommended mode). If you know the structure of the Jacobian to be irreducible, that is not permutable to block lower triangular form, then you should set **lblock** $= false$. For example, a Jacobian arising from using the method of lines for parabolic partial differential equations would normally be irreducible. (See the specification of nag_ode_ivp_stiff_sparjac_diag (d02nx) for optional output concerning **lblock**.)

12: **rwork**$(\mathbf{50} + \mathbf{4} \times$ **neqmax**$)$ – REAL (KIND=nag_wp) 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

1: **nia** – INTEGER

Default: the dimension of the array **ia**.

The dimension of the array **ia**.

Constraints:

if **jceval** = 'S', 'F' or 'D', **nia** $\geq$ **neq** + 1;
otherwise **nia** $\geq$ 1.

2: **nja** – INTEGER

Default: the dimension of the array **ja**.

The dimension of the array **ja**.

Constraints:

if **jceval** = 'S', 'F' or 'D', **nja** $\geq$ **ia**(**neq** + 1) − 1;
otherwise **nja** $\geq$ 1.

3: **isplit** – INTEGER

Suggested value: **isplit** = 73, unless you have information from a previous run of a similar problem.

Default: 73

This argument is used for splitting the integer workspace **jacpvt** to effect an efficient decomposition. It must satisfy $1 \leq$ **isplit** ≤ 99. If **isplit** lies outside this range on entry, a default value of 73 is used. An appropriate value for **isplit** for subsequent runs on similar problems is available via the optional output nag_ode_ivp_stiff_sparjac_diag (d02nx).

5.3 Output Parameters

1: **jacpvt**(**njcpvt**) – INTEGER array

Data relating to the Jacobian sparsity structure.

2: **rwork**(**50** + **4** × **neqmax**) – REAL (KIND=nag_wp) array

3: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, an illegal input was detected.

ifail = −99

An unexpected error has been triggered by this routine. Please contact NAG.

ifail = −399

Your licence key may have expired or may not have been installed correctly.

ifail = −999

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

nag_ode_ivp_stiff_sparjac_setup (d02nu) must be called as a setup function before a call to either nag_ode_ivp_stiff_exp_sparjac (d02nd) or nag_ode_ivp_stiff_imp_sparjac (d02nj) and may be called as the linear algebra setup function before a call to nag_ode_ivp_stiff_exp_revcom (d02nm) or nag_ode_ivp_stiff_imp_revcom (d02nn).

9 Example

See Section 10 in nag_ode_ivp_stiff_exp_sparjac (d02nd), nag_ode_ivp_stiff_imp_sparjac (d02nj) and nag_ode_ivp_stiff_imp_revcom (d02nn).

9.1 Program Text

```
function d02nu_example

fprintf('d02nu example results\n\n');

% Initialize variables and arrays for setup routine

n      = nag_int(3);
ord    = nag_int(5);
sdy    = nag_int(ord + 1);
petzld = false;
con    = zeros(6);
tcrit  = 0;
hmin   = 1.0e-10;
hmax   = 10;
h0     = 0;
maxstp = nag_int(200);
mxhnil = nag_int(5);
lrwork = 50 + 4*n;
rwork  = zeros(lrwork);

% Setup integration method using d02nv.
[const, rwork, ifail] = d02nv(...
                              n, sdy, ord, 'Newton', petzld, con, tcrit, ...
                              hmin, hmax, h0, maxstp, mxhnil, 'Average-L2', ...
                              rwork);

% Setup for sparse Jacbian using d02nu
nwkjac = nag_int(100);
njcpvt = nag_int(150);
ia     = nag_int(0);
ja     = nag_int(0);
sens   = 0;
u      = 0.1;
eta    = 1.0e-4;
lblock = true;
jacpvt = zeros(njcpvt, 1);
[jacpvt, rwork, ifail] = d02nu(...
                               n, n, 'Numerical', nwkjac, ia, ja, njcpvt, ...
                               sens, u, eta, lblock, rwork);

% Initialize variables and arrays for integration
t      = 0;
tout   = 10;
y      = [1; 0; 0];
rtol   = [0.0001];
atol   = [1e-07];
itol   = nag_int(1);
inform = zeros(23, 1, nag_int_name);
ysave  = zeros(n, sdy);
```

```
wkjac  = zeros(nwkjac, 1);
itask  = nag_int(1);
itrace = nag_int(0);

% Integrate ODE from t=0 to t=tout, no monitoring, using d02nd
[t, y, ydot, rwork, inform, ysave, wkjac, jacpvt, ifail] = ...
    d02nd(t, tout, y, rwork, rtol, atol, itol, inform, @fcn, ysave, @jac, ...
          wkjac, jacpvt, 'd02nby', itask, itrace);

fprintf('Solution y and derivative y'' at t = %7.4f is:\n',t);
fprintf('\n %10s %10s\n','y','y''');
for i=1:n
  fprintf(' %10.4f %10.4f\n',y(i),ydot(i));
end

function [f, ires] = fcn(neq, t, y, ires)
% Evaluate derivative vector.
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);
```

9.2 Program Results

```
     d02nu example results

Solution y and derivative y' at t = 10.0000 is:

          y         y'
     0.8414    -0.0075
     0.0000    -0.0000
     0.1586     0.0075
```