NAG Toolbox

nag_ode_ivp_adams_rootdiag (d02qy)

1 Purpose

nag_ode_ivp_adams_rootdiag (d02qy) is a diagnostic function which may be called after a call to the integrator functions nag_ode_ivp_adams_roots (d02qf) or nag_ode_ivp_adams_roots_revcom (d02qg).

2 Syntax

```
[index, itype, events, resids, ifail] = nag_ode_ivp_adams_rootdiag(neqg, rwork, iwork)
[index, itype, events, resids, ifail] = d02qy(neqg, rwork, iwork)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 22: *lrwork* and *liwork* were removed from the interface.

3 Description

nag_ode_ivp_adams_rootdiag (d02qy) should be called only after a call to nag_ode_ivp_adams_roots (d02qf) or nag_ode_ivp_adams_roots_revcom (d02qg) results in the output value **root** = *true*, indicating that a root has been detected. nag_ode_ivp_adams_rootdiag (d02qy) permits you to examine information about the root detected, such as the indices of the event equations for which there is a root, the type of root (odd or even) and the residuals of the event equations.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **neqg** – INTEGER

The number of event functions defined for the integration function. It must be the same argument **neqg** supplied to the setup function nag_ode_ivp_adams_setup (d02qw) and to the integration function (nag_ode_ivp_adams_roots (d02qf) or nag_ode_ivp_adams_roots_revcom (d02qg)).

2: **rwork**(*lrwork*) – REAL (KIND=nag_wp) array

This **must** be the same argument **rwork** as supplied to nag_ode_ivp_adams_roots (d02qf) or nag_ode_ivp_adams_roots_revcom (d02qg). It is used to pass information from the integration function to nag_ode_ivp_adams_rootdiag (d02qy) and therefore the contents of this array **must not** be changed before calling nag_ode_ivp_adams_rootdiag (d02qy).

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

This **must** be the same argument **iwork** as supplied to nag_ode_ivp_adams_roots (d02qf) or nag_ode_ivp_adams_roots_revcom (d02qg). It is used to pass information from the integration function to nag_ode_ivp_adams_rootdiag (d02qy) and therefore the contents of this array **must not** be changed before calling nag_ode_ivp_adams_rootdiag (d02qy).

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **index** – INTEGER

The index k of the event equation $g_k(x, y, y') = 0$ for which the root has been detected.

2: **itype** – INTEGER

Information about the root detected for the event equation defined by **index**. The possible values of **itype** with their interpretations are as follows:

itype $= 1$
A simple root, or lack of distinguishing information available.

itype $= 2$
A root of even multiplicity is believed to have been detected, that is no change in sign of the event function was found.

itype $= 3$
A high-order root of odd multiplicity.

itype $= 4$
A possible root, but due to high multiplicity or a clustering of roots accurate evaluation of the event function was prohibited by round-off error and/or cancellation.

In general, the accuracy of the root is less reliable for values of **itype** > 1.

3: **events**(**neqg**) – INTEGER array

Information about the kth event function on a very small interval containing the root, **t** (see nag_ode_ivp_adams_roots (d02qf) and nag_ode_ivp_adams_roots_revcom (d02qg)), as output from the integration function. All roots lying in this interval are considered indistinguishable numerically and therefore should be regarded as defining a root at **t**. The possible values of **events**(k) with their interpretations are as follows:

events$(k) = 0$
The kth event function did not have a root.

events$(k) = -1$
The kth event function changed sign from positive to negative about a root, in the direction of integration.

events$(k) = 1$
The kth event function changed sign from negative to positive about a root, in the direction of integration.

events$(k) = 2$
A root was identified, but no change in sign was observed.

4: **resids**(**neqg**) – REAL (KIND=nag_wp) array

The value of the kth event function computed at the root, **t** (see nag_ode_ivp_adams_roots (d02qf) and nag_ode_ivp_adams_roots_revcom (d02qg)).

5: **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 $= 1$

An integration function (nag_ode_ivp_adams_roots (d02qf) or nag_ode_ivp_adams_roots_revcom (d02qg)) has not been called, no root was detected or one or more of the arguments *lrwork*, *liwork* and **neqg** does not match the corresponding values supplied to nag_ode_ivp_adams_setup (d02qw). Values for the arguments **index**, **itype**, **events** and **resids** will not have been set.

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

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

None.

9 Example

See Section 10 in nag_ode_ivp_adams_roots (d02qf).

9.1 Program Text

```
function d02qy_example

fprintf('d02qy example results\n\n');

neqf   = nag_int(2);
neqg   = nag_int(2);
vectol = true;
atol   = [1e-06;  1e-06];
rtol   = [0.0001; 0.0001];
onestp = false;
crit   = true;
tcrit  = 10;
hmax   = 0;
maxstp = nag_int(0);
lrwork = 23*(1+neqf) + 14*neqg;
alterg = false;
sophst = true;

lrwork = 23*(1+neqf) + 14*neqg;
rwork  = zeros(lrwork,1);
liwork = 21 + 4*neqg;
iwork  = zeros(liwork, 1, nag_int_name);

% Adams method setup
[statefOut, altergOut, rwork, iwork, ifail] = ...
  d02qw(...
```

```
        'Start', neqf, vectol, atol, rtol, onestp, crit, tcrit, hmax, ...
        maxstp, neqg, alterg, sophst, rwork, iwork);

% Integrate from t = 0 to 10
t    = 0;
tout = 10;
y    = [0; 1];

[t, y, root, rwork, iwork, ifail] = ...
    d02qf(...
          @fcn, t, y, tout, @g, neqg, rwork, iwork);

% Display solution and root statistics.
fprintf('Solution and root functions at t = %7.4f is:\n',t);

[index, itype, events, resids, ifail] = d02qy(neqg, rwork, iwork);

fprintf('\n y :');
fprintf('%10.4f',y);
fprintf('\n g :');
fprintf('%10.4f',resids);
fprintf('\n\nRoot Diagnostics:\n');
fprintf('  Index of event function,  k : %d\n',index);
fprintf('  Type of root                : ');
if itype==1
   fprintf('simple root\n');
elseif itype==2
   fprintf('root of even multiplicity\n');
elseif itype==3
   fprintf('root of odd multiplicity\n');
else
   fprintf('root of high multiplicity or cluster?\n');
end
fprintf('  event function sign change  : ');
if events(index)==-1
   fprintf('positive to negative\n');
elseif events(index)==1
   fprintf('negative to positive\n');
else events(index)==2
   fprintf('no change in sign\n');
end

function f = fcn(neqf, x, y)
  f=zeros(neqf,1);
  f(1)=y(2);
  f(2)=-y(1);

function result = g(neqf, x, y, yp, k)
  if (k == 1)
    result = yp(1)+0.5;
  else
    result = y(1)-sqrt(0.5);
  end
```

9.2 Program Results

```
     d02qy example results

Solution and root functions at t =  0.7854 is:

 y :    0.7071    0.7071
 g :    1.2071    0.0000

Root Diagnostics:
  Index of event function,  k : 2
  Type of root                : simple root
  event function sign change  : negative to positive
```