nag_ode_ivp_adams_setup (d02qwc)

1. Purpose

nag_ode_ivp_adams_setup (d02qwc) is a setup function which must be called by the user prior to the first call of the integration function nag_ode_ivp_adams_roots (d02qfc) and may be called prior to any subsequent continuation call of the integrator.

2. Specification

```
#include <nag.h>
#include <nagd02.h>

void nag_ode_ivp_adams_setup(Nag_Start *state, Integer neqf,
            Boolean vectol, double atol[], double rtol[],
            Boolean one_step, Boolean crit, double tcrit,
            double hmax, Integer max_step, Integer neqg, Boolean *alter_g,
            Boolean sophist, Nag_ODE_Adams *opt, NagError *fail)
```

3. Description

This function permits initialisation of the integration method and setting of integration inputs prior to any call of nag_ode_ivp_adams_roots (d02qfc).

It must be called before the first call of the function nag_ode_ivp_adams_roots (d02qfc) and it may be called before any continuation call of nag_ode_ivp_adams_roots (d02qfc).

4. Parameters

state

Input: specifies whether the integration routine nag_ode_ivp_adams_roots (d02qfc) is to start a new system of ordinary differential equations, restart a system or continue with a system. **state** is interpreted as follows:
state = **Nag_NewStart** start integration with a new differential system;
state = **Nag_ReStart** restart integration with the current differential system;
state = **Nag_Continue** continue integration with the current differential system.
Constraint: **state** = **Nag_NewStart**, **Nag_ReStart** or **Nag_Continue**
Output: **state** is set to **Nag_Continue**, except that if an error is detected, **state** is unchanged.

neqf

Input: the number of ordinary differential equations to be solved by the integration routine. **neqf** must remain unchanged on subsequent calls to nag_ode_ivp_adams_setup with **state** = **Nag_Continue** or **Nag_ReStart**.
Constraint: **neqf** ≥ 1.

vectol

Input: specifies whether vector or scalar error control is to be employed for the local error test in the integration.
If **vectol** = **TRUE**, then vector error control will be used and the user must specify values of **rtol**[i] and **atol**[i], for $i = 0, 1, \ldots,$**neqf**-1.
Otherwise scalar error control will be used and the user must specify values of just **rtol**[0] and **atol**[0].

The error test to be satisfied is of the form

$$\sqrt{\sum_{i=1}^{\mathbf{neqf}} \left(\frac{e_i}{w_i}\right)^2} \leq 1.0,$$

where w_i is defined as follows:

vectol	w_i
TRUE	**rtol**$[i-1] \times |y_i| +$ **atol**$[i-1]$
FALSE	**rtol**$[0] \times |y_i| +$ **atol**$[0]$

and e_i is an estimate of the local error in y_i, computed internally. **vectol** must remain unchanged on subsequent calls to nag_ode_ivp_adams_setup with **state** = **Nag_Continue** or **Nag_ReStart**.

atol[neqf]
Input: the absolute local error tolerance (see **vectol**).
Constraint: **atol**$[i] \geq 0.0$.

rtol[neqf]
Input: the relative local error tolerance (see **vectol**).
Constraints: **rtol**$[i] \geq 0.0$,
rtol$[i] \geq 4.0\times$***machine precision*** if **atol**$[i] = 0.0$.

one_step
Input: the mode of operation of the integration routine. If **one_step** = **TRUE**, the integration routine will operate in one-step mode, that is it will return after each successful step. Otherwise the integration routine will operate in interval mode, that is it will return at the end of the integration interval.

crit
Input: specifies whether or not there is a value for the independent variable beyond which integration is not to be attempted. Setting **crit** = **TRUE** indicates that there is such a point, whereas **crit** = **FALSE** indicates that there is no such restriction.

tcrit
Input: with **crit** = **TRUE**, **tcrit** must be set to a value of the independent variable beyond which integration is not to be attempted. Otherwise **tcrit** is not referenced.

hmax
Input: if **hmax** $\neq$ 0.0 then a bound on the absolute step size during the integration is taken to be |**hmax**|. If **hmax** = 0.0 on entry, then no bound is assumed on the step size during the integration.

A bound may be required if there are features of the solution on very short ranges of integration which may be missed. The user should try **hmax** = 0.0 first.

Note: this parameter only affects the step size if the option **crit** = **TRUE** is being used.

max_step
Input: a bound on the number of attempted steps in any one call to the integration routine. If **max_step** ≤ 0 on entry, a value of 1000 is used.

neqg
Input: specifies whether or not root-finding is required in nag_ode_ivp_adams_roots (d02qfc). If **neqg** ≤ 0 then no root-finding is attempted. If **neqg** > 0 then root-finding is required and **neqg** event functions will be specified for the integration routine.

alter_g
Input: specifies whether or not the event functions have been redefined. **alter_g** need not be set if **state** = **Nag_NewStart**. On subsequent calls to nag_ode_ivp_adams_setup, if **neqg** has been set positive, then **alter_g** = **FALSE** specifies that the event functions remain unchanged, whereas **alter_g** = **TRUE** specifies that the event functions have changed. Because of the expense in reinitialising the root searching procedure, **alter_g** should be set to **TRUE** only if the event functions really have been altered. **alter_g** need not be set if the root-finding option is not used.
Output: **alter_g** is set to **FALSE**, except that if an error is detected, **alter_g** is unchanged.

sophist
Input: the type of search technique to be used in the root-finding. If **sophist** = **TRUE** then a sophisticated and reliable but expensive technique will be used, whereas for **sophist** = **FALSE** a simple but less reliable technique will be used. If **neqg** ≤ 0 then **sophist** is not referenced.

opt

Output: the structure of type Nag_ODE_Adams will have been initialised to appropriate values for entry to the integration routine nag_ode_ivp_adams_roots (d02qfc). **opt** must be passed unchanged to the integration routine.

Memory will have been allocated by nag_ode_ivp_adams_setup to several pointers within **opt**, this memory is used by the integration routine nag_ode_ivp_adams_roots (d02qfc) and the interpolation routine nag_ode_ivp_adams_interp (d02qzc). The library function nag_ode_ivp_adams_free (d02qyc) is provided so that this memory can be freed by the user when all calls to nag_ode_ivp_adams_roots (d02qfc) and nag_ode_ivp_adams_interp (d02qzc) have been completed. A call to nag_ode_ivp_adams_free (d02qyc) may also be made prior to reentering nag_ode_ivp_adams_setup with **state** = **Nag_Newstart**.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

NE_STATE

state not equal to **Nag_NewStart** on first call.

NE_BAD_PARAM

On entry parameter **state** had an illegal value.

NE_ALLOC_FAIL

Memory allocation failed.

NE_INT_ARG_LT

On entry, **neqf** must not be less than 1: **neqf** = ⟨*value*⟩.

NE_NEQF_CHANGED

state = ⟨*string*⟩ but **neqf** has been changed. **neqf** was ⟨*value*⟩ but is now ⟨*value*⟩.

NE_VECTOL_CHANGED

state = ⟨*string*⟩ but **vectol** has been changed. **vectol** was ⟨*string*⟩ but is now ⟨*string*⟩.

NE_NEQG_CHANGED

alter_g = **FALSE** but **neqg** has been changed. **neqg** was ⟨*value*⟩ but is now ⟨*value*⟩.

NE_REAL_ARG_LT

On entry, **atol**[⟨*value*⟩] must not be less than 0.0: **atol**[⟨*value*⟩] = ⟨*value*⟩.
On entry, **rtol**[⟨*value*⟩] must not be less than 0.0: **rtol**[⟨*value*⟩] = ⟨*value*⟩.

NE_REAL_LT_COND

When **atol**[⟨*value*⟩] = 0.0, **rtol**[⟨*value*⟩] must not be less than $4 \times \varepsilon$. **rtol**[⟨*value*⟩] = ⟨*value*⟩, $4 \times \varepsilon$ = ⟨*value*⟩.

NE_BOOL_NOT_SET

The Boolean argument **crit** has not been set to **TRUE** or **FALSE**.

6. Further Comments

Prior to a continuation call of the integration routine, the user may reset some of the parameters by calling nag_ode_ivp_adams_setup with **state** = **Nag_Continue**. The user may reset:

(a) **hmax**	- to alter the maximum step size selection;
(b) **rtol**, **atol**	- to change the error requirements;
(c) **max_step**	- to increase or decrease the number of attempted steps before an error exit is returned;
(d) **one_step**	- to change the operation mode of the integration routine;
(e) **crit**, **tcrit**	- to alter the point beyond which integration must not be attempted; and
(f) **neqg**, **alter_g**, **sophist**	- to alter the number and type of event functions, and also the search method.

If the behaviour of the system of differential equations has altered and the user wishes to restart the integration method from the value of **t** output from the integration routine, then **state** should be set to **Nag_ReStart** and some of the integration parameters may be reset also. If the user wants to redefine the system of differential equations or start a new integration problem, then **state** should be set to **Nag_NewStart**. Resetting **state** to **Nag_ReStart** or **Nag_NewStart** on normal continuation calls causes a restart in the integration process, which is very inefficient when not needed.

6.1. Accuracy

Not applicable.

6.2. References

None.

7. See Also

nag_ode_ivp_adams_roots (d02qfc)
nag_ode_ivp_adams_free (d02qyc)

8. Example

See example program for nag_ode_ivp_adams_roots (d02qfc).