

# NAG Library Function Document

## nag\_ode\_ivp\_rkts\_diag (d02ptc)

### 1 Purpose

nag\_ode\_ivp\_rkts\_diag (d02ptc) provides details about an integration performed by either nag\_ode\_ivp\_rkts\_range (d02pec), nag\_ode\_ivp\_rkts\_onestep (d02pfc) or nag\_ode\_ivp\_rk\_step\_revcomm (d02pgc).

### 2 Specification

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

void nag_ode_ivp_rkts_diag (Integer *fevals, Integer *stepcost,
    double *waste, Integer *stepsok, double *hnext, Integer iwsav[],
    const double rwsav[], NagError *fail)
```

### 3 Description

nag\_ode\_ivp\_rkts\_diag (d02ptc) and its associated functions (nag\_ode\_ivp\_rkts\_range (d02pec), nag\_ode\_ivp\_rkts\_onestep (d02pfc), nag\_ode\_ivp\_rk\_step\_revcomm (d02pgc), nag\_ode\_ivp\_rk\_interp\_setup (d02phc), nag\_ode\_ivp\_rk\_interp\_eval (d02pjc), nag\_ode\_ivp\_rkts\_setup (d02pqc), nag\_ode\_ivp\_rkts\_reset\_tend (d02prc), nag\_ode\_ivp\_rkts\_interp (d02psc) and nag\_ode\_ivp\_rkts\_errass (d02puc)) solve the initial value problem for a first-order system of ordinary differential equations. The functions, based on Runge–Kutta methods and derived from RKSUITE (see Brankin *et al.* (1991)), integrate

$$y' = f(t, y) \quad \text{given} \quad y(t_0) = y_0$$

where  $y$  is the vector of  $n$  solution components and  $t$  is the independent variable.

After a call to nag\_ode\_ivp\_rkts\_range (d02pec), nag\_ode\_ivp\_rkts\_onestep (d02pfc) or nag\_ode\_ivp\_rk\_step\_revcomm (d02pgc), nag\_ode\_ivp\_rkts\_diag (d02ptc) can be called to obtain information about the cost of the integration and the size of the next step.

### 4 References

Brankin R W, Gladwell I and Shampine L F (1991) RKSUITE: A suite of Runge–Kutta codes for the initial value problems for ODEs *SoftReport 91-S1* Southern Methodist University

### 5 Arguments

- 1: **fevals** – Integer \* *Output*  
*On exit:* the total number of evaluations of  $f$  used in the integration so far; this includes evaluations of  $f$  required for the secondary integration necessary if nag\_ode\_ivp\_rkts\_setup (d02pqc) had previously been called with **errass** = Nag\_ErrorAssess.on.
- 2: **stepcost** – Integer \* *Output*  
*On exit:* the cost in terms of number of evaluations of  $f$  of a typical step with the method being used for the integration. The method is specified by the argument **method** in a prior call to nag\_ode\_ivp\_rkts\_setup (d02pqc).

- 3:     **waste** – double \* *Output*  
*On exit:* the number of attempted steps that failed to meet the local error requirement divided by the total number of steps attempted so far in the integration. A ‘large’ fraction indicates that the integrator is having trouble with the problem being solved. This can happen when the problem is ‘stiff’ and also when the solution has discontinuities in a low-order derivative.
- 4:     **stepsok** – Integer \* *Output*  
*On exit:* the number of accepted steps.
- 5:     **hnext** – double \* *Output*  
*On exit:* the step size the integrator will attempt to use for the next step.
- 6:     **iwsav**[130] – Integer *Communication Array*  
7:     **rwsav**[350] – const double *Communication Array*  
**Note:** the communication **rwsav** used by the other functions in the suite must be used here however, only the first 350 elements will be referenced.  
*On entry:* these must be the same arrays supplied in a previous call to nag\_ode\_ivp\_rkts\_range (d02pec), nag\_ode\_ivp\_rkts\_onestep (d02pfc) or nag\_ode\_ivp\_rk\_step\_revcomm (d02pgc). They must remain unchanged between calls.  
*On exit:* information about the integration for use on subsequent calls to nag\_ode\_ivp\_rkts\_range (d02pec), nag\_ode\_ivp\_rkts\_onestep (d02pfc) or nag\_ode\_ivp\_rk\_step\_revcomm (d02pgc) or other associated functions.
- 8:     **fail** – NagError \* *Input/Output*  
The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

## 6 Error Indicators and Warnings

### NE\_ALLOC\_FAIL

Dynamic memory allocation failed.

See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

### NE\_BAD\_PARAM

On entry, argument *<value>* had an illegal value.

### NE\_INTERNAL\_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

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

See Section 2.7.6 in How to Use the NAG Library and its Documentation for further information.

### NE\_MISSING\_CALL

You cannot call this function before you have called the integrator.

### NE\_NO\_LICENCE

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

See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

**NE\_PREV\_CALL**

On entry, a previous call to the setup function has not been made or the communication arrays have become corrupted, or a catastrophic error has already been detected elsewhere.  
You cannot continue integrating the problem.

**NE\_RK\_INVALID\_CALL**

You have already made one call to this function after the integrator could not achieve specified accuracy.  
You cannot call this function again.

**7 Accuracy**

Not applicable.

**8 Parallelism and Performance**

nag\_ode\_ivp\_rkts\_diag (d02ptc) is not threaded in any implementation.

**9 Further Comments**

When a secondary integration has taken place, that is when global error assessment has been specified using **errass** = Nag\_ErrorAssess\_on in a prior call to nag\_ode\_ivp\_rkts\_setup (d02pqc), then the approximate number of evaluations of  $f$  used in this secondary integration is given by  $2 \times \mathbf{stepsok} \times \mathbf{stepcost}$  for **method** = Nag\_RK\_4\_5 or Nag\_RK\_7\_8 and  $3 \times \mathbf{stepsok} \times \mathbf{stepcost}$  for **method** = Nag\_RK\_2\_3.

**10 Example**

See Section 10 in nag\_ode\_ivp\_rkts\_range (d02pec), nag\_ode\_ivp\_rkts\_onestep (d02pfc), nag\_ode\_ivp\_rkts\_reset\_tend (d02prc), nag\_ode\_ivp\_rkts\_interp (d02psc) and nag\_ode\_ivp\_rkts\_errass (d02puc).

---