NAG Library Function Document

nag_ode_ivp_rk_interp_eval (d02pjc)

1 Purpose

nag_ode_ivp_rk_interp_eval (d02pjc) evaluates the interpolant calculated by nag_ode_ivp_rk_interp_set up (d02phc), following an integration step performed by nag_ode_ivp_rk_step_revcomm (d02pgc) to solve an initial value problem.

2 Specification

```
#include <nag.h>
#include <nagd02.h>
void nag_ode_ivp_rk_interp_eval (Nag_Boolean icheck, Integer n,
     Integer nwant, double t, Integer ideriv, double sol[], double wcomm[],
     Integer lwcomm, Integer iwsav[], double rwsav[], NagError *fail)
```

3 Description

When integrating using the reverse communication Runge–Kutta integrator nag_ode_ivp_rk_step_rev comm (d02pgc), the solution or its derivatives can be obtained inexpensively between steps by interpolation. nag_ode_ivp_rk_interp_setup (d02phc) is called after a step by nag_ode_ivp_rk_ste p_revcomm (d02pgc) from a previous value of t $(= t_{k-1})$ to its current value, $t = t_k$ (i.e., a kth successful time-step has been taken). nag_ode_ivp_rk_interp_eval (d02pjc) can then be called to evaluate interpolated approximations of the function or its derivatives at any value of t in the interval (t_{k-1}, t_k).

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: **icheck** – Nag_Boolean *Input*

On entry: indicates whether consistency checks on input arguments should be performed

icheck = Nag_FALSE
Don't perform checks on input arguments.

icheck = Nag_TRUE
Perform consistency checks on input arguments.

It is recommended to use **icheck** = Nag_TRUE on the first call following a call to nag_ode_ivp_rk_interp_setup (d02phc) and to set **icheck** = Nag_FALSE on subsequent calls within the last step to avoid the overhead of argument checking.

2: **n** – Integer *Input*

On entry: n, the dimension of the system of ODEs being integrated.

Constraint: this must be the same value as supplied in a previous call to nag_ode_ivp_rkts_setup (d02pqc).

3: **nwant** – Integer *Input*

On entry: only the first **nwant** system components to be computed. This should be the same value as passed to nag_ode_ivp_rk_interp_setup (d02phc) when computing the interpolant.

Constraint: **nwant** = **nwant** passed to nag_ode_ivp_rk_interp_setup (d02phc).

4: **t** – double *Input*

On entry: t, the value of the independent variable where a solution is desired. Although any value of t can be supplied, accurate solutions can only be obtained for values in the range of the last time-step taken by nag_ode_ivp_rk_step_revcomm (d02pgc).

5: **ideriv** – Integer *Input*

On entry:

ideriv = 0
: Compute approximations to the first **nwant** components of the solution $y(t)$.

ideriv = 1
: Compute approximations to the first **nwant** components of the first derivatives of the solution $y'(t)$.

Constraint: **ideriv** = 0 or 1.

6: **sol**[**nwant**] – double *Output*

On exit:

ideriv = 0
: The first **nwant** components of the solution $y(t)$.

ideriv = 1
: The first **nwant** components of the first derivatives of the solution $y'(t)$.

7: **wcomm**[**lwcomm**] – double *Communication Array*

On entry: this must be the same array supplied in a previous call to nag_ode_ivp_rk_interp_setup (d02phc). It must remain unchanged between calls.

8: **lwcomm** – Integer *Input*

On entry: length of **wcomm**. This should be the same value as supplied in a previous call to nag_ode_ivp_rk_interp_setup (d02phc).

If in a previous call to nag_ode_ivp_rkts_setup (d02pqc):

method = Nag_RK_2_3, **lwcomm** must be at least 1.

method = Nag_RK_4_5, **lwcomm** must be at least $\mathbf{n} + \max(\mathbf{n}, 5 \times \mathbf{nwant})$.

method = Nag_RK_7_8, **lwcomm** $\geq 8 \times$ **nwant**.

9: **iwsav**[**130**] – Integer *Communication Array*

10: **rwsav**[$\mathbf{32} \times \mathbf{n} + \mathbf{350}$] – double *Communication Array*

On entry: these must be the same arrays supplied in a previous call nag_ode_ivp_rk_step_rev comm (d02pgc). They must remain unchanged between calls.

On exit: information about the integration for use on subsequent calls to nag_ode_ivp_rk_ step_revcomm (d02pgc), nag_ode_ivp_rk_interp_setup (d02phc) or other associated functions.

11: **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_INT

On entry, **ideriv** = ⟨*value*⟩.
Constraint: **ideriv** = 0 or 1.

On entry, **lwcomm** = ⟨*value*⟩.
Constraint: for **method** = Nag_RK_2_3, **lwcomm** ≥ 1.

NE_INT_2

On entry, **lwcomm** = ⟨*value*⟩ and **nwant** = ⟨*value*⟩.
Constraint: for **method** = Nag_RK_7_8, **lwcomm** $\geq 8 \times$ **nwant**.

NE_INT_3

On entry, **lwcomm** = ⟨*value*⟩, **n** = ⟨*value*⟩ and **nwant** = ⟨*value*⟩.
Constraint: for **method** = Nag_RK_4_5, **lwcomm** $\geq$ **n** + max(**n**, $5 \times$ **nwant**).

NE_INT_CHANGED

On entry, **n** = ⟨*value*⟩, but the value passed to the setup routine was **n** = ⟨*value*⟩.

On entry, **nwant** = ⟨*value*⟩, but on interpolation setup **nwant** = ⟨*value*⟩.
Constraint: **nwant** must be unchanged from setup.

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

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.

You cannot call this function before you have called the interpolation setup.

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_INI

The previous call to the interpolation setup function returned an error.

7 Accuracy

The computed values will be of a similar accuracy to that computed by nag_ode_ivp_rk_step_revcomm (d02pgc).

8 Parallelism and Performance

nag_ode_ivp_rk_interp_eval (d02pjc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

See Section 10 in nag_ode_ivp_rk_step_revcomm (d02pgc).