NAG Library Function Document

nag_pde_interp_1d_coll (d03pyc)

1 Purpose

nag_pde_interp_1d_coll (d03pyc) may be used in conjunction with either nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc). It computes the solution and its first derivative at user-specified points in the spatial coordinate.

2 Specification

```
#include <nag.h>
#include <nagd03.h>
void nag_pde_interp_1d_coll (Integer npde, const double u[], Integer nbkpts,
     const double xbkpts[], Integer npoly, Integer npts, const double xp[],
     Integer intpts, Integer itype, double up[], double rsave[],
     Integer lrsave, NagError *fail)
```

3 Description

nag_pde_interp_1d_coll (d03pyc) is an interpolation function for evaluating the solution of a system of partial differential equations (PDEs), or the PDE components of a system of PDEs with coupled ordinary differential equations (ODEs), at a set of user-specified points. The solution of a system of equations can be computed using nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc) on a set of mesh points; nag_pde_interp_1d_coll (d03pyc) can then be employed to compute the solution at a set of points other than those originally used in nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc). It can also evaluate the first derivative of the solution. Polynomial interpolation is used between each of the break-points **xbkpts**$[i-1]$, for $i = 1, 2, \ldots,$ **nbkpts**. When the derivative is needed (**itype** $= 2$), the array **xp**[**intpts** $- 1$] must not contain any of the break-points, as the method, and consequently the interpolation scheme, assumes that only the solution is continuous at these points.

4 References

None.

5 Arguments

Note: the arguments **u**, **npts**, **npde**, **xbkpts**, **nbkpts**, **rsave** and **lrsave** must be supplied unchanged from either nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc).

1: **npde** – Integer *Input*

On entry: the number of PDEs.

Constraint: **npde** ≥ 1.

2: **u**[**npde** $\times$ **npts**] – const double *Input*

On entry: the PDE part of the original solution returned in the argument **u** by the function nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc).

3: **nbkpts** – Integer *Input*

On entry: the number of break-points.

Constraint: **nbkpts** ≥ 2.

4: **xbkpts**[**nbkpts**] – const double *Input*

On entry: **xbkpts**$[i-1]$, for $i = 1, 2, \ldots,$ **nbkpts**, must contain the break-points as used by nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc).

Constraint: **xbkpts**$[0] <$ **xbkpts**$[1] < \cdots <$ **xbkpts**[**nbkpts** $- 1]$.

5: **npoly** – Integer *Input*

On entry: the degree of the Chebyshev polynomial used for approximation as used by nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc).

Constraint: $1 \leq$ **npoly** ≤ 49.

6: **npts** – Integer *Input*

On entry: the number of mesh points as used by nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc).

Constraint: **npts** $= ($**nbkpts** $- 1) \times$ **npoly** $+ 1$.

7: **xp**[**intpts**] – const double *Input*

On entry: **xp**$[i-1]$, for $i = 1, 2, \ldots,$ **intpts**, must contain the spatial interpolation points.

Constraints:

xbkpts$[0] \leq$ **xp**$[0] <$ **xp**$[1] < \cdots <$ **xp**[**intpts** $- 1] \leq$ **xbkpts**[**nbkpts** $- 1$];
if **itype** $= 2$, **xp**$[i-1] \neq$ **xbkpts**$[j-1]$, for $i = 1, 2, \ldots,$ **intpts** and
$j = 2, 3, \ldots,$ **nbkpts** $- 1$.

8: **intpts** – Integer *Input*

On entry: the number of interpolation points.

Constraint: **intpts** ≥ 1.

9: **itype** – Integer *Input*

On entry: specifies the interpolation to be performed.

itype $= 1$
The solution at the interpolation points are computed.

itype $= 2$
Both the solution and the first derivative at the interpolation points are computed.

Constraint: **itype** $= 1$ or 2.

10: **up**$[dim]$ – double *Output*

Note: the dimension, *dim*, of the array **up** must be at least **npde** $\times$ **intpts** $\times$ **itype**.

The element **UP**(i, j, k) is stored in the array element **up**$[(k-1) \times$ **npde** $\times$ **intpts** $+ (j-1) \times$ **npde** $+ i - 1]$.

On exit: if **itype** $= 1$, **UP**$(i, j, 1)$, contains the value of the solution $U_i(x_j, t_{\text{out}})$, at the interpolation points $x_j =$ **xp**$[j-1]$, for $j = 1, 2, \ldots,$ **intpts** and $i = 1, 2, \ldots,$ **npde**.

If **itype** $= 2$, **UP**$(i, j, 1)$ contains $U_i(x_j, t_{\text{out}})$ and **UP**$(i, j, 2)$ contains $\dfrac{\partial U_i}{\partial x}$ at these points.

11: **rsave**[**lrsave**] – double *Communication Array*

The array **rsave** contains information required by nag_pde_interp_1d_coll (d03pyc) as returned by nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc). The contents of **rsave** must not be changed from the call to nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc). Some elements of this array are overwritten on exit.

12: **lrsave** – Integer *Input*

On entry: the size of the workspace **rsave**, as in nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc).

13: **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_EXTRAPOLATION

Extrapolation is not allowed.

NE_INCOMPAT_PARAM

On entry, **itype** $= 2$ and at least one interpolation point coincides with a break-point, i.e., interpolation point no ⟨*value*⟩ with value ⟨*value*⟩ is close to break-point ⟨*value*⟩ with value ⟨*value*⟩.

NE_INT

On entry, **intpts** ≤ 0: **intpts** $=$ ⟨*value*⟩.

On entry, **itype** $=$ ⟨*value*⟩.
Constraint: **itype** $= 1$ or 2.

On entry, **nbkpts** $=$ ⟨*value*⟩.
Constraint: **nbkpts** ≥ 2.

On entry, **npde** $=$ ⟨*value*⟩.
Constraint: **npde** > 0.

On entry, **npoly** $=$ ⟨*value*⟩.
Constraint: **npoly** > 0.

NE_INT_3

On entry, **npts** $=$ ⟨*value*⟩, **nbkpts** $=$ ⟨*value*⟩ and **npoly** $=$ ⟨*value*⟩.
Constraint: **npts** $= ($**nbkpts** $- 1) \times$ **npoly** $+ 1$.

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_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_NOT_STRICTLY_INCREASING

On entry, break-points **xbkpts** badly ordered: $I = \langle value \rangle$, **xbkpts**$[I-1] = \langle value \rangle$, $J = \langle value \rangle$ and **xbkpts**$[J-1] = \langle value \rangle$.

On entry, interpolation points **xp** badly ordered: $I = \langle value \rangle$, **xp**$[I-1] = \langle value \rangle$, $J = \langle value \rangle$ and **xp**$[J-1] = \langle value \rangle$.

7 Accuracy

See the documents for nag_pde_parab_1d_coll (d03pdc) or nag_pde_parab_1d_coll_ode (d03pjc).

8 Parallelism and Performance

nag_pde_interp_1d_coll (d03pyc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

See Section 10 in nag_pde_parab_1d_coll (d03pdc).