NAG Toolbox for MATLAB

d03pz

1 Purpose

d03pz interpolates in the spatial co-ordinate the solution and derivative of a system of partial differential equations (PDEs). The solution must first be computed using one of the finite difference schemes d03pc, d03ph or d03pp, or one of the Keller box schemes d03pe, d03pk or d03pr.

2 Syntax

```
[up, ifail] = d03pz(m, u, x, xp, itype, 'npde', npde, 'npts', npts, 'intpts', intpts)
```

3 Description

d03pz is an interpolation function for evaluating the solution of a system of partial differential equations (PDEs), at a set of user-specified points. The solution of the system of equations (possibly with coupled ordinary differential equations) must be computed using a finite difference scheme or a Keller box scheme on a set of mesh points. d03pz can then be employed to compute the solution at a set of points anywhere in the range of the mesh. It can also evaluate the first spatial derivative of the solution. It uses linear interpolation for approximating the solution.

4 References

None.

5 Parameters

Note: the parameters **x**, **m**, **u**, **npts** and **npde** must be supplied unchanged from the PDE function.

5.1 Compulsory Input Parameters

1: **m – int32 scalar**

The co-ordinate system used. If the call to d03pz follows one of the finite difference functions then **m** must be the same parameter **m** as used in that call. For the Keller box scheme only Cartesian co-ordinate systems are valid and so **m must** be set to zero. No check will be made by d03pz in this case.

m $= 0$

Indicates Cartesian co-ordinates.

m $= 1$

Indicates cylindrical polar co-ordinates.

m $= 2$

Indicates spherical polar co-ordinates.

Constraints:

$0 \leq$ **m** ≤ 2 following a finite difference function;
m $= 0$ following a Keller box scheme function.

2: **u(npde,npts) – double array**

The PDE part of the original solution returned in the parameter **u** by the PDE function.

Constraint: **npde** ≥ 1.

3: **x(npts) – double array**

$\mathbf{x}(i)$, for $i = 1, 2, \ldots, \mathbf{npts}$, must contain the mesh points as used by the PDE function.

4: **xp(intpts) – double array**

$\mathbf{xp}(i)$, for $i = 1, 2, \ldots, \mathbf{intpts}$, must contain the spatial interpolation points.

Constraint: $\mathbf{x}(1) \leq \mathbf{xp}(1) < \mathbf{xp}(2) < \cdots < \mathbf{xp}(\mathbf{intpts}) \leq \mathbf{x}(\mathbf{npts})$.

5: **itype – int32 scalar**

specifies the interpolation to be performed.

itype $= 1$

The solutions at the interpolation points are computed.

itype $= 2$

Both the solutions and their first derivatives at the interpolation points are computed.

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

5.2 Optional Input Parameters

1: **npde – int32 scalar**

the number of PDEs.

Constraint: **npde** ≥ 1.

2: **npts – int32 scalar**

Default: The dimension of the arrays **u**, **x**. (An error is raised if these dimensions are not equal.) the number of mesh points.

Constraint: **npts** ≥ 3.

3: **intpts – int32 scalar**

Default: The dimension of the arrays **xp**, **up**. (An error is raised if these dimensions are not equal.) the number of interpolation points.

Constraint: **intpts** ≥ 1.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **up(npde,intpts,itype) – double array**

If **itype** $= 1$, $\mathbf{up}(i,j,1)$, contains the value of the solution $U_i(x_j, t_{\text{out}})$, at the interpolation points $x_j = \mathbf{xp}(j)$, for $j = 1, 2, \ldots, \mathbf{intpts}$ and $i = 1, 2, \ldots, \mathbf{npde}$.

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

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, **itype** $\neq 1$ or 2,
or **intpts** < 1,
or **npde** < 1,
or **npts** < 3,
or **m** $\neq 0$, 1 or 2,
or the mesh points $\mathbf{x}(i)$, for $i = 1, 2, \ldots,$ **npts**, are not in strictly increasing order.

ifail $= 2$

On entry, the interpolation points $\mathbf{xp}(i)$, for $i = 1, 2, \ldots,$ **intpts**, are not in strictly increasing order.

ifail $= 3$

You are attempting extrapolation, that is, one of the interpolation points $\mathbf{xp}(i)$, for some i, lies outside the interval $[\mathbf{x}(1), \mathbf{x}(\mathbf{npts})]$. Extrapolation is not permitted.

7 Accuracy

See the PDE function documents.

8 Further Comments

None.

9 Example

```
m = int32(1);
u = [0, 0.1760021889329504, 0.3341579089708593, 0.4937980719530722, ...
    0.6511403366936832, 0.8044807535486175, 0.9525489765885864, ...
    1.09423368983943, 1.228514538086573, 1.35444287807973, ...
    1.471137345520289, 1.577784110954285, 1.673622501349812, ...
    1.757611424145824, 1.825804783497759, 1.867752956803207, ...
    1.878359678117952, 1.868385424753035, 1.854505240663775, ...
    1.84852820454846;
     0.9997119383379945, 0.9996043818707437, 0.9995887346960896, ...
    0.9995733176158295, 0.9995516033644839, 0.9995217547212679, ...
    0.9994816709262737, 0.999428326601514, 0.9993571885011239, ...
    0.9992612549235844, 0.9991292197833584, 0.9989381167530054, ...
    0.9984790503092739, 0.9936254861425254, 0.9449226608879052, ...
    0.7493216873033168, 0.4582225956678795, 0.2084439018937778, ...
    0.05247378844653194, 0];
x = [0;
     0.08257934547233232;
     0.1645945902807339;
     0.2454854871407991;
     0.3246994692046835;
     0.4016954246529694;
     0.4759473930370735;
     0.5469481581224268;
     0.6142127126896678;
     0.6772815716257411;
     0.7357239106731316;
```

```
     0.7891405093963936;
     0.8371664782625285;
     0.879473751206489;
     0.9157733266550574;
     0.9458172417006346;
     0.9694002659393304;
     0.9863613034027223;
     0.9965844930066698;
     1];
xp = [0;
     0.4;
     0.6;
     0.8;
     0.9;
     1];
itype = int32(1);
[up, ifail] = d03pz(m, u, x, xp, itype)
```

```
up =
         0    0.8008    1.1988    1.5990    1.7958    1.8485
    0.9997    0.9995    0.9994    0.9988    0.9663   -0.0000
ifail =
           0
```