NAG Toolbox for MATLAB

d02xj

1 Purpose

d02xj interpolates components of the solution of a system of first-order ordinary differential equations from information provided by the integrators in sub-chapter D02M/N.

2 Syntax

```
[sol, ifail] = d02xj(xsol, m, ysav, neq, x, nqu, hu, h, 'sdysav', sdysav)
```

3 Description

d02xj evaluates the first m components of the solution of a system of ordinary differential equations at any point using natural polynomial interpolation based on information generated by the integrator. This information must be passed unchanged to d02xj. d02xj should not normally be used to extrapolate outside the range of values obtained from the above functions.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **xsol – double scalar**

The point at which the first m components of the solution are to be evaluated. **xsol** should not be an extrapolation point, that is **xsol** should satisfy $(\mathbf{xsol} - \mathbf{x}) \times \mathbf{hu} \leq 0.0$. Extrapolation is permitted but not recommended.

2: **m – int32 scalar**

m, the number of components of the solution whose values at **xsol** are required. The first **m** components are evaluated.

Constraint: $1 \leq \mathbf{m} \leq \mathbf{neq}$.

3: **ysav**(**ldysav**,**sdysav**) **– double array**

ldysav, the first dimension of the array, must be at least 1.

The values provided in the parameter **ysav** on return from the integrator.

4: **neq – int32 scalar**

The value used for the parameter **neq** when calling the integrator.

Constraint: $1 \leq \mathbf{neq} \leq \mathbf{ldysav}$.

5: **x – double scalar**

The latest value at which the solution has been computed, as provided in the parameter **tcur** on return from the optional output d02ny.

6: **nqu – int32 scalar**

The order of the method used up to the latest value at which the solution has been computed, as provided in the parameter **nqu** on return from the optional output d02ny.

Constraint: **nqu** ≥ 1.

7: **hu – double scalar**

The last successful step used, that is the step used in the integration to get to **x**, as provided in the parameter **hu** on return from the optional output d02ny.

8: **h – double scalar**

The next step size to be attempted in the integration, as provided in the parameter **h** on return from the optional output d02ny.

5.2 Optional Input Parameters

1: **sdysav – int32 scalar**

Default: The dimension of the array **ysav**.

the value used for the parameter **sdysav** when calling the integrator.

Constraint: **sdysav** $\geq$ **nqu** $+ 1$.

5.3 Input Parameters Omitted from the MATLAB Interface

ldysav

5.4 Output Parameters

1: **sol**(**m**) **– double array**

The calculated value of the ith component of the solution at **xsol**, for $i = 1, 2, \ldots, m$.

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, **m** < 1,
or **neq** < 1,
or **ldysav** < 1,
or **neq** $>$ **ldysav**,
or **m** $>$ **neq**,
or **nqu** < 1,
or **sdysav** $<$ **nqu** $+ 1$.

ifail $= 2$

On entry, **hu** $= 0.0$ or **h** $= 0.0$. This error can only occur if **h** and **hu** have been changed by you or possibly if the integrator has failed before calling d02xj.

ifail $= 3$

d02xj has been called for extrapolation. Before returning with this error exit, the value of the solution at **xsol** is calculated and placed in **sol**.

7 Accuracy

The solution values returned will be of a similar accuracy to those computed by the integrator.

8 Further Comments

d02xj is that employed for prediction purposes internally by the integrator. It is supplied for purposes of consistency only. You are recommended to employ the C^1 interpolant provided by d02xk wherever possible.

9 Example

```
xsol = 0.02;
m = int32(3);
ysave = [0.9992010426063869, -9.014310377562871e-06, 6.812799519478248e-
10, ...
         4.431544977419294e-13, 0, 3.646603065905307e-06;
       3.637754720562377e-05, -1.43245286845343e-09, -1.690946415679345e-
09, ...
       4.645881910091972e-09, -5.351301001679175e-39, -5.764408311856045e-
06;
      0.0007625798464070455, 9.015742830431324e-06, 1.009666463731707e-09,
...
    -4.64632506458996e-09, -0.2374434471130371, 2.117805245939617e-06];
neq = int32(3);
x = 0.02004966865187021;
nqu = int32(1);
hu = 0.0002271142424177253;
h = 0.0002271142424177253;
[sol, ifail] = d02xj(xsol, m, ysave, neq, x, nqu, hu, h)
```

```
sol =
    0.9992
    0.0000
    0.0008
ifail =
           0
```