# NAG Toolbox for MATLAB

# d02px

## 1 Purpose

d02px computes the solution of a system of ordinary differential equations using interpolation anywhere on an integration step taken by d02pd.

## 2 Syntax

```
[ywant, ypwant, work, wrkint, ifail] = d02px(neq, twant, reqest, nwant,
f, work, wrkint, 'lenint', lenint)
```

## 3 Description

d02px and its associated functions (d02pv, d02pd, d02pw, d02py and d02pz) 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

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

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

d02pd computes the solution at the end of an integration step. Using the information computed on that step d02px computes the solution by interpolation at any point on that step. It cannot be used if **method** $= 3$ was specified in the call to setup function d02pv.

## 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 Parameters

### 5.1 Compulsory Input Parameters

1: **neq – int32 scalar**

*n*, the number of ordinary differential equations in the system to be solved by the integration function.

*Constraint*: **neq** $\geq 1$.

2: **twant – double scalar**

*t*, the value of the independent variable where a solution is desired.

3: **reqest – string**

Determines whether the solution and/or its first derivative are to be computed.

**reqest** = 'S'

Compute the approximate solution only.

**reqest** = 'D'

Compute the approximate first derivative of the solution only.

**reqest** = 'B'

Compute both the approximate solution and its first derivative.

*Constraint*: **reqest** = 'S', 'D' or 'B'.

4: **nwant – int32 scalar**

The number of components of the solution to be computed. The first **nwant** components are evaluated.

*Constraint*: $1 \leq$ **nwant** $\leq n$, where $n$ is specified by **neq** in the prior call to d02pv.

5: **f – string containing name of m-file**

**f** must evaluate the functions $f_i$ (that is the first derivatives $y_i'$) for given values of the arguments $t, y_i$. It must be the same procedure as supplied to d02pd.

Its specification is:

```
[yp] = f(t, y)
```

**Input Parameters**

1: **t – double scalar**

$t$, the current value of the independent variable.

2: **y**$(n)$ **– double array**

The current values of the dependent variables, $y_i$, for $i = 1, 2, \ldots, n$.

**Output Parameters**

1: **yp**$(n)$ **– double array**

The values of $f_i$, for $i = 1, 2, \ldots, n$.

6: **work**$(*)$ **– double array**

**Note**: the dimension of the array **work** must be at least **lenwrk** (see d02pv).

This **must** be the same array as supplied to d02pd and **must** remain unchanged between calls.

7: **wrkint**(**lenint**) **– double array**

Must be the same array as supplied in previous calls, if any, and must remain unchanged between calls to d02px.

## 5.2 Optional Input Parameters

1: **lenint – int32 scalar**

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

*Constraints*:

**lenint** $\geq 1$ if **method** $= 1$ in the prior call to d02pv;
**lenint** $\geq n + 5 \times$ **nwant** if **method** $= 2$ and $n$ is specified by **neq** in the prior call of d02pv.

### 5.3 Input Parameters Omitted from the MATLAB Interface

None.

### 5.4 Output Parameters

1: **ywant**$(*)$ **– double array**

**Note**: the dimension of the array **ywant** must be at least **nwant** if **reqest** = 'S' or 'B', and at least 1 otherwise.

An approximation to the first **nwant** components of the solution at **twant** if **reqest** = 'S' or 'B'. Otherwise **ywant** is not defined.

2: **ypwant**$(*)$ **– double array**

**Note**: the dimension of the array **ypwant** must be at least **nwant** if **reqest** = 'D' or 'B', and at least 1 otherwise.

An approximation to the first **nwant** components of the first derivative at **twant** if **reqest** = 'D' or 'B'. Otherwise **ypwant** is not defined.

3: **work**$(*)$ **– double array**

**Note**: the dimension of the array **work** must be at least **lenwrk** (see d02pv).

Contains information about the integration for use on subsequent calls to d02pd or other associated functions.

4: **wrkint**(**lenint**) **– double array**

The contents are modified.

5: **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, an invalid input value for **nwant** or **lenint** was detected or an invalid call to d02px was made, for example without a previous call to the integration function d02pd, or after an error return from d02pd, or if d02pd was being used with **method** = 3. You cannot continue integrating the problem.

## 7 Accuracy

The computed values will be of a similar accuracy to that computed by d02pd.

## 8 Further Comments

None.

## 9 Example

```
d02pd_f.m
function [yp] = f(t, y)
```

```
  yp = zeros(2, 1);
  yp(1) = y(2);
  yp(2) = -y(1);
```

d02px_f.m

```
function [yp] = f(t, y)
  yp = zeros(2, 1);
  yp(1) = y(2);
  yp(2) = -y(1);
```

```
tstart = 0;
ystart = [0; 1];
tend = 6.283185307179586;
tol = 0.001;
thres = [1e-08; 1e-08];
method = int32(2);
task = 'Complex Task';
errass = false;
lenwrk = int32(64);
neq = int32(2);
twant = 0.3926990816987241;
reqest = 'Both';
nwant = int32(1);
wrkint = zeros(7, 1);
[work, ifail] = ...
       d02pv(tstart, ystart, tend, tol, thres, method, task, errass,
lenwrk);
[tnow, ynow, ypnow, work, ifail] = d02pd('d02pd_f', neq, work);
[ywant, ypwant, workOut, wrkintOut, ifail] = ...
    d02px(neq, twant, reqest, nwant, 'd02px_f', work, wrkint)
```

```
ywant =
    0.3827
ypwant =
    0.9239
workOut =
     array elided
wrkintOut =
    0.7959
   -0.6123
   -0.2181
   -0.0571
    0.0113
    0.0019
   -0.0001
ifail =
           0
```