

NAG Toolbox for MATLAB

d02qz

1 Purpose

d02qz interpolates components of the solution of a non-stiff system of first-order differential equations from information provided by the integrator functions d02qf or d02qg.

2 Syntax

```
[ywant, ypwant, ifail] = d02qz(neqf, twant, nwant, rwork, iwork,
'lrwork', lrwork, 'liwork', liwork)
```

3 Description

d02qz evaluates the first **nwant** components of the solution of a non-stiff system of first-order ordinary differential equations at any point using the method of Watts and Shampine 1986 and information generated by d02qf or d02qg. d02qz should not normally be used to extrapolate outside the current range of the values produced by the integration function.

4 References

Watts H A and Shampine L F 1986 Smoother interpolants for Adams codes *SIAM J. Sci. Statist. Comput.* **7** 334–345

5 Parameters

5.1 Compulsory Input Parameters

1: **neqf** – int32 scalar

The number of first-order ordinary differential equations being solved by the integration function. It must contain the same value as the parameter **neqf** in a prior call to the setup function d02qw.

2: **twant** – double scalar

The point at which components of the solution and derivative are to be evaluated. **twant** should not normally be an extrapolation point, that is **twant** should satisfy

$$told \leq \mathbf{twant} \leq T,$$

or if integration is proceeding in the negative direction

$$told \geq \mathbf{twant} \geq T,$$

where *told* is the previous integration point and is, to within rounding, **tcurr** – **hlast** (see d02qx). Extrapolation is permitted but not recommended and **ifail** = 2 is returned whenever extrapolation is attempted.

3: **nwant** – int32 scalar

the number of components of the solution and derivative whose values at **twant** are required. The first **nwant** components are evaluated.

Constraint: $1 \leq \mathbf{nwant} \leq \mathbf{neqf}$.

4: **rwork(lrwork) – double array**

This **must** be the same parameter **rwork** as supplied to d02qw and to d02qf or d02qg. It is used to pass information from these functions to d02qz. Therefore its contents **must not** be changed prior to a call to d02qz.

5: **iwork(liwork) – int32 array**

This **must** be the same parameter **iwork** as supplied to d02qw and to d02qf or d02qg. It is used to pass information from these functions to d02qz. Therefore its contents **must not** be changed prior to a call to d02qz.

5.2 Optional Input Parameters1: **lrwork – int32 scalar**

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

This must be the same parameter **lrwork** as supplied to d02qw.

2: **liwork – int32 scalar**

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

This must be the same parameter **liwork** as supplied to d02qw.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters1: **ywant(nwant) – double array**

The calculated value of the i th component of the solution at **twant**, for $i = 1, 2, \dots, \mathbf{nwant}$.

2: **ypwant(nwant) – double array**

The calculated value of the i th component of the derivative at **twant**, for $i = 1, 2, \dots, \mathbf{nwant}$.

3: **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

An integration function (d02qf or d02qg) has not been called, no integration steps have been taken since the last call to d02qw with **statef** = 'S', one or more of the parameters **lrwork**, **liwork** and **neqf** does not match the same parameter supplied to d02qw, or **nwant** does not satisfy $1 \leq \mathbf{nwant} \leq \mathbf{neqf}$.

ifail = 2

d02qz has been called for extrapolation. The values of the solution and its derivative at **twant** have been calculated and placed in **ywant** and **ypwant** before returning with this warning (see Section 7).

These error exits may be caused by overwriting elements of **rwork** and **iwork**.

7 Accuracy

The error in interpolation is of a similar order to the error arising from the integration. The same order of accuracy can be expected when extrapolating using d02qz. However, the actual error in extrapolation will, in general, be much larger than for interpolation.

8 Further Comments

When interpolation for only a few components is required then it is more efficient to order the components of interest so that they are numbered first.

9 Example

d02qf_fcn.m

```
function f = fcn(neqf, x, y)
    f=zeros(neqf,1);
    f(1)=y(2);
    f(2)=-y(1);
```

d02qf_g.m

```
function result=g(neqf, x, y, yp, k)
    if (k == 1)
        result = yp(1);
    else
        result = y(1);
    end
```

```
neqf = int32(2);
twant = 0.1963495408493621;
nwant = int32(2);
rwork = zeros(69,1);
iwork = zeros(21,1,'int32');

t = 0;
tinc = 0.0625*pi;
tout = 8*tinc;
y = [0;1];

[statefOut, altergOut, rwork, iwork, ifail] = ...
    d02qw('S', int32(2), true, [1e-08;1e-08], [0.0001; 0.0001], true,
    true, ...
    8*0.0625*pi, 2, int32(500), int32(0), false, true, rwork, iwork);

jj = 1;
fprintf('\n      T          Y(1)      Y(2)\n');
fprintf(' %6.4f   %7.4f  %7.4f\n', t, y(1), y(2));
output(1,1:3) = [t, y(1), y(2)];

while (t < tout)
    [t, y, root, rwork, iwork, ifail] = ...
        d02qf('d02qf_fcn', t, y, tout, 'd02qf_g', int32(0), rwork, iwork);
    while (twant <= t)
        [ywant, ypwant, ifail] = d02qz(neqf, twant, nwant, rwork, iwork);
        fprintf(' %6.4f   %7.4f  %7.4f\n', twant, ywant(1), ywant(2));
        jj = jj+1;
        output(jj,1:3) = [twant, ywant(1), ywant(2)];
        twant = jj*tinc;
    end
end
```

T	Y(1)	Y(2)
0.0000	0.0000	1.0000
0.1963	0.1951	0.9808
0.3927	0.3827	0.9239
0.5890	0.5556	0.8315
0.7854	0.7071	0.7071
0.9817	0.8315	0.5556
1.1781	0.9239	0.3827
1.3744	0.9808	0.1951
1.5708	1.0000	-0.0000