NAG Toolbox for MATLAB

d05by

1 Purpose

d05by computes the fractional quadrature weights associated with the Backward Differentiation Formulae (BDF) of orders 4, 5 and 6. These weights can then be used in the solution of weakly singular equations of Abel type.

2 Syntax

```
[wt, sw, ifail] = d05by(iorder, iq, lenfw)
```

3 Description

d05by computes the weights $W_{n,j}$ and ω_i for a family of quadrature rules related to a BDF method for approximating the integral:

$$\frac{1}{\sqrt{\pi}}\int_0^t \frac{\phi(s)}{\sqrt{t-s}}\,ds \simeq \sqrt{h}\sum_{j=0}^{2p-2} W_{n,j}\phi(jh) + \sqrt{h}\sum_{j=2p-1}^{n} \omega_{n-j}\phi(jh), \qquad 0 \le t \le T, \tag{1}$$

with $t = nh (n \ge 0)$, for some given h. In (1), p is the order of the BDF method used and $W_{n,j}$, ω_i are the fractional starting and the fractional convolution weights respectively. The algorithm for the generation of ω_i is based on Newton's iteration. Fast Fourier transform (FFT) techniques are used for computing these weights and subsequently $W_{n,j}$ (see Baker and Derakhshan 1987 and Henrici 1979 for practical details and Lubich 1986 for theoretical details). Some special functions can be represented as the fractional integrals of simpler functions and fractional quadratures can be employed for their computation (see Lubich 1986). A description of how these weights can be used in the solution of weakly singular equations of Abel type is given in Section 8.

4 References

Baker C T H and Derakhshan M S 1987 Computational approximations to some power series *Approximation Theory* (ed L Collatz, G Meinardus and G Nürnberger) **81** 11–20

Henrici P 1979 Fast Fourier methods in computational complex analysis *SIAM Rev.* **21** 481–529

Lubich Ch 1986 Discretized fractional calculus *SIAM J. Math. Anal.* **17** 704–719

5 Parameters

5.1 Compulsory Input Parameters

1: **iorder – int32 scalar**

p, the order of the BDF method to be used.

Constraint: $4 \le$ **iorder** ≤ 6.

2: **iq – int32 scalar**

Determines the number of weights to be computed. By setting **iq** to a value, $2^{\mathbf{iq}+1}$ fractional convolution weights are computed.

Constraint: **iq** ≥ 0.

3: **lenfw – int32 scalar**

Constraint: **lenfw** $\geq 2^{\mathbf{iq}+2}$.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

ldsw, work, lwk

5.4 Output Parameters

1: **wt**(**lenfw**) – **double array**

The first $2^{\mathbf{iq}+1}$ elements of **wt** contains the fractional convolution weights ω_i, for $i = 0, 1, \ldots, 2^{\mathbf{iq}+1} - 1$. The remainder of the array is used as workspace.

2: **sw**(**ldsw**,$2 \times$ **iorder** $- 1$) – **double array**

sw$(n, j+1)$ contains the fractional starting weights $W_{n-1,j}$, for $n = 1, 2, \ldots,$ $\left(2^{\mathbf{iq}+1} + 2 \times \mathbf{iorder} - 1\right)$; $j = 0, 1, \ldots, 2 \times \mathbf{iorder} - 2$.

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 $= 1$

On entry, **iorder** < 4 or **iorder** > 6,
or **iq** < 0,
or **lenfw** $< 2^{\mathbf{iq}+2}$,
or **ldsw** $< 2^{\mathbf{iq}+1} + 2 \times$ **iorder** $- 1$,
or **lwk** $< 2^{\mathbf{iq}+3}$.

7 Accuracy

None.

8 Further Comments

Fractional quadrature weights can be used for solving weakly singular integral equations of Abel type. In this section, we propose the following algorithm which you may find useful in solving a linear weakly singular integral equation of the form

$$y(t) = f(t) + \frac{1}{\sqrt{\pi}} \int_0^t \frac{K(t,s)y(s)}{\sqrt{t-s}} ds, \qquad 0 \leq t \leq T, \tag{2}$$

using d05by. In (2), $K(t,s)$ and $f(t)$ are given and the solution $y(t)$ is sought on a uniform mesh of size h such that $T = Nh$. Discretization of (2) yields

$$y_n = f(nh) + \sqrt{h} \sum_{j=0}^{2p-2} W_{nj} K(nh, jh) y_j + \sqrt{h} \sum_{j=2p-1}^{n} \omega_{n-j} K(nh, jh) y_j, \tag{3}$$

where $y_n \simeq y(nh)$. We propose the following algorithm for computing y_n from (3) after a call to d05by:

(a) Set $N = 2^{\mathbf{iq}+1} + 2 \times \mathbf{iorder} - 2$ and $h = T/N$.

(b) Equation (3) requires $2 \times \mathbf{iorder} - 2$ starting values, y_j, for $j = 1, 2, \ldots, 2 \times \mathbf{iorder} - 2$, with $y_0 = f(0)$. These starting values can be computed by solving the system

$$y_n = f(nh) + \sqrt{h} \sum_{j=0}^{2\times\mathbf{iorder}-2} \mathbf{sw}(n+1, j+1) K(nh, jh) y_j, \qquad n = 1, 2, \ldots, 2 \times \mathbf{iorder} - 2.$$

(c) Compute the inhomogeneous terms

$$\sigma_n = f(nh) + \sqrt{h} \sum_{j=0}^{2\times\mathbf{iorder}-2} \mathbf{sw}(n+1, j+1) K(nh, jh) y_j, \qquad n = 2 \times \mathbf{iorder} - 1, 2 \times \mathbf{iorder}, \ldots, N.$$

(d) Start the iteration for $n = 2 \times \mathbf{iorder} - 1, 2 \times \mathbf{iorder}, \ldots, N$ to compute y_n from:

$$\left(1 - \sqrt{h}\mathbf{wt}(1) K(nh, nh)\right) y_n = \sigma_n + \sqrt{h} \sum_{j=2\times\mathbf{iorder}-1}^{n-1} \mathbf{wt}(n-j+1) K(nh, jh) y_j.$$

Note that for nonlinear weakly singular equations, the solution of a nonlinear algebraic system is required at step (b) and a single nonlinear equation at step (d).

9 Example

```
iorder = int32(4);
iq = int32(3);
lenfw = int32(32);
[wt, sw, ifail] = d05by(iorder, iq, lenfw)
```

```
wt =
    0.6928
    0.6651
    0.4589
    0.3175
    0.2622
    0.2451
    0.2323
    0.2164
    0.2006
    0.1878
    0.1780
    0.1700
    0.1629
    0.1566
    0.1508
    0.1457
    0.3849
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
         0
sw =
         0         0         0         0         0         0         0
```

```
    0.0565    2.8928   -6.7497   11.6491  -11.1355    5.5374   -1.1223
    0.0371    1.7401   -2.8628    6.5207   -6.4058    3.2249   -0.6583
    0.0300    1.3207   -2.4642    6.3612   -5.4478    2.7025   -0.5481
    0.0258    1.1217   -2.2620    5.3683   -3.7553    2.2132   -0.4549
    0.0230    0.9862   -2.0034    4.5005   -3.2772    2.7262   -0.4320
    0.0208    0.9001   -1.8989    4.2847   -3.5881    2.8201    0.2253
    0.0190    0.8506   -1.9250    4.4164   -4.0181    2.7932    0.1564
    0.0173    0.8177   -1.9697    4.5348   -4.2425    2.7458   -0.0697
    0.0160    0.7886   -1.9781    4.5318   -4.2769    2.6997   -0.2127
    0.0149    0.7603   -1.9548    4.4545   -4.2332    2.6541   -0.2620
    0.0140    0.7338   -1.9198    4.3619   -4.1782    2.6059   -0.2716
    0.0132    0.7097   -1.8842    4.2754   -4.1246    2.5544   -0.2767
    0.0125    0.6880   -1.8497    4.1933   -4.0662    2.5011   -0.2845
    0.0119    0.6681   -1.8153    4.1109   -4.0004    2.4479   -0.2915
    0.0114    0.6497   -1.7805    4.0279   -3.9304    2.3962   -0.2951
    0.0110    0.6327   -1.7461    3.9463   -3.8598    2.3466   -0.2958
    0.0105    0.6168   -1.7126    3.8677   -3.7907    2.2990   -0.2950
    0.0102    0.6020   -1.6804    3.7926   -3.7238    2.2536   -0.2935
    0.0098    0.5882   -1.6495    3.7209   -3.6589    2.2101   -0.2917
    0.0095    0.5752   -1.6199    3.6523   -3.5961    2.1686   -0.2895
    0.0093    0.5631   -1.5916    3.5867   -3.5356    2.1291   -0.2871
    0.0090    0.5517   -1.5644    3.5240   -3.4774    2.0914   -0.2844
ifail =
           0
```