NAG Toolbox for MATLAB

d01bc

1 Purpose

d01bc returns the weights (normal or adjusted) and abscissae for a Gaussian integration rule with a specified number of abscissae. Six different types of Gauss rule are allowed.

2 Syntax

```
[weight, abscis, ifail] = d01bc(itype, a, b, c, d, n)
```

3 Description

d01bc returns the weights w_i and abscissae x_i for use in the summation

$$S = \sum_{i=1}^{n} w_i f(x_i)$$

which approximates a definite integral (see Davis and Rabinowitz 1975 or Stroud and Secrest 1966). The following types are provided:

(a) Gauss–Legendre

$$S \simeq \int_a^b f(x)\,dx, \qquad \text{exact for } f(x) = P_{2n-1}(x).$$

Constraint: $b > a$.

(b) Gauss–Jacobi

normal weights:

$$S \simeq \int_a^b (b-x)^c (x-a)^d f(x)\,dx, \qquad \text{exact for } f(x) = P_{2n-1}(x),$$

adjusted weights:

$$S \simeq \int_a^b f(x)\,dx, \qquad \text{exact for } f(x) = (b-x)^c (x-a)^d P_{2n-1}(x).$$

Constraint: $c > -1$, $d > -1$, $b > a$.

(c) Gauss–Exponential

normal weights:

$$S \simeq \int_a^b \left| x - \frac{a+b}{2} \right|^c f(x)\,dx, \qquad \text{exact for } f(x) = P_{2n-1}(x),$$

adjusted weights:

$$S \simeq \int_a^b f(x)\,dx, \qquad \text{exact for } f(x) = \left| x - \frac{a+b}{2} \right|^c P_{2n-1}(x).$$

Constraint: $c > -1$, $b > a$.

(d) Gauss–Laguerre

normal weights:

$$\begin{aligned} S &\simeq \int_a^{\infty} |x-a|^c e^{-bx} f(x)\,dx \qquad (b>0), \\ &\simeq \int_{-\infty}^{a} |x-a|^c e^{-bx} f(x)\,dx \qquad (b<0), \qquad \text{exact for } f(x) = P_{2n-1}(x), \end{aligned}$$

adjusted weights:

$$\begin{aligned} S &\simeq \int_a^{\infty} f(x)\,dx \qquad (b>0), \\ &\simeq \int_{-\infty}^{a} f(x)\,dx \qquad (b<0), \qquad \text{exact for } f(x) = |x-a|^c e^{-bx} P_{2n-1}(x). \end{aligned}$$

Constraint: $c > -1$, $b \neq 0$.

(e) Gauss–Hermite

normal weights:

$$S \simeq \int_{-\infty}^{+\infty} |x-a|^c e^{-b(x-a)^2} f(x)\,dx, \qquad \text{exact for } f(x) = P_{2n-1}(x),$$

adjusted weights:

$$S \simeq \int_{-\infty}^{+\infty} f(x)\,dx, \qquad \text{exact for } f(x) = |x-a|^c e^{-b(x-a)^2} P_{2n-1}(x).$$

Constraint: $c > -1$, $b > 0$.

(f) Gauss–Rational

normal weights:

$$\begin{aligned} S &\simeq \int_a^{\infty} \tfrac{|x-a|^c}{|x+b|^d} f(x)\,dx \qquad (a+b>0), \\ &\simeq \int_{-\infty}^{a} \tfrac{|x-a|^c}{|x+b|^d} f(x)\,dx \qquad (a+b<0), \qquad \text{exact for } f(x) = P_{2n-1}\left(\tfrac{1}{x+b}\right), \end{aligned}$$

adjusted weights:

$$\begin{aligned} S &\simeq \int_a^{\infty} f(x)\,dx \qquad (a+b>0), \\ &\simeq \int_{-\infty}^{a} f(x)\,dx \qquad (a+b<0), \qquad \text{exact for } f(x) = \tfrac{|x-a|^c}{|x+b|^d} P_{2n-1}\left(\tfrac{1}{x+b}\right). \end{aligned}$$

Constraint: $c > -1$, $d > c+1$, $a+b \neq 0$.

In the above formulae, $P_{2n-1}(x)$ stands for any polynomial of degree $2n-1$ or less in x.

The method used to calculate the abscissae involves finding the eigenvalues of the appropriate tridiagonal matrix (see Golub and Welsch 1969). The weights are then determined by the formula

$$w_i = \left\{ \sum_{j=0}^{n-1} P_j^*(x_i)^2 \right\}^{-1}$$

where $P_j^*(x)$ is the jth orthogonal polynomial with respect to the weight function over the appropriate interval.

The weights and abscissae produced by d01bc may be passed to d01fb, which will evaluate the summations in one or more dimensions.

4 References

Davis P J and Rabinowitz P 1975 *Methods of Numerical Integration* Academic Press

Golub G H and Welsch J H 1969 Calculation of Gauss quadrature rules *Math. Comput.* **23** 221–230

Stroud A H and Secrest D 1966 *Gaussian Quadrature Formulas* Prentice–Hall

5 Parameters

5.1 Compulsory Input Parameters

1: **itype – int32 scalar**

Indicates the type of quadrature rule.

itype $= 0$

Gauss–Legendre.

itype $= 1$

Gauss–Jacobi.

itype $= 2$

Gauss–Exponential.

itype $= 3$

Gauss–Laguerre.

itype $= 4$

Gauss–Hermite.

itype $= 5$

Gauss–Rational.

The above values give the normal weights; the adjusted weights are obtained if the value of **itype** above is negated.

Constraint: $-5 \leq$ **itype** ≤ 5.

2: **a – double scalar**

3: **b – double scalar**

4: **c – double scalar**

5: **d – double scalar**

The parameters a, b, c and d which occur in the quadrature formulae. **c** is not used if **itype** $= 0$; **d** is not used unless **itype** $= \pm 1$ or ± 5. For some rules **c** and **d** must not be too large (See Section 6.)

Constraints:

if **itype** $= 0$, **a** $<$ **b**;
if **itype** $= \pm 1$, **a** $<$ **b** and **c** > -1 and **d** > -1;
if **itype** $= \pm 2$, **a** $<$ **b** and **c** > -1;
if **itype** $= \pm 3$, **b** $\neq 0$ and **c** > -1;
if **itype** $= \pm 4$, **b** > 0 and **c** > -1;
if **itype** $= \pm 5$, **a** $+$ **b** $\neq 0$ and **c** > -1 and **d** $>$ **c** $+ 1$.

6: **n – int32 scalar**

n, the number of weights and abscissae to be returned. If **itype** $= -2$ or -4 and **c** $\neq 0.0$, an odd value of **n** may raise problems (see **ifail** $= 6$).

Constraint: **n** > 0.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **weight(n) – double array**

The **n** weights.

2: **abscis(n) – double array**

The **n** abscissae.

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

The algorithm for computing eigenvalues of a tridiagonal matrix has failed to obtain convergence. If the soft fail option is used, the values of the weights and abscissae on return are indeterminate.

ifail = 2

On entry, **n** < 1,
or **itype** < -5,
or **itype** > 5.

If the soft fail option is used, weights and abscissae are returned as zero.

ifail = 3

a, **b**, **c** or **d** is not in the allowed range:

if **itype** $= 0$, $\mathbf{a} \geq \mathbf{b}$;

if **itype** $= \pm 1$, $\mathbf{a} \geq \mathbf{b}$ or $\mathbf{c} \leq -1.0$ or $\mathbf{d} \leq -1.0$ or $\mathbf{c} + \mathbf{d} + 2.0 > gmax$;

if **itype** $= \pm 2$, $\mathbf{a} \geq \mathbf{b}$ or $\mathbf{c} \leq -1.0$;

if **itype** $= \pm 3$, $\mathbf{b} = 0.0$ or $\mathbf{c} \leq -1.0$ or $\mathbf{c} + 1.0 > gmax$;

if **itype** $= \pm 4$, $\mathbf{b} \leq 0.0$ or $\mathbf{c} \leq -1.0$ or $(\mathbf{c} + 1.0/2.0) > gmax$;

if **itype** $= \pm 5$, $\mathbf{a} + \mathbf{b} = 0.0$ or $\mathbf{c} \leq -1.0$ or $\mathbf{d} \leq \mathbf{c} + 1.0$.

Here $gmax$ is the (machine-dependent) largest integer value such that $\Gamma(gmax)$ can be computed without overflow.

If the soft fail option is used, weights and abscissae are returned as zero.

ifail = 4

One or more of the weights are larger than $rmax$, the largest floating-point number on this machine. $rmax$ is given by the function x02al. If the soft fail option is used, the overflowing weights are returned as $rmax$. Possible solutions are to use a smaller value of **n**; or, if using adjusted weights, to change to normal weights.

ifail = 5

One or more of the weights are too small to be distinguished from zero on this machine. If the soft fail option is used, the underflowing weights are returned as zero, which may be a usable approximation. Possible solutions are to use a smaller value of **n**; or, if using normal weights, to change to adjusted weights.

ifail $= 6$

Gauss–Exponential or Gauss–Hermite adjusted weights with **n** odd and **c** $\neq 0.0$. Theoretically, in these cases:

for **c** > 0.0, the central adjusted weight is infinite, and the exact function $f(x)$ is zero at the central abscissa.

for **c** < 0.0, the central adjusted weight is zero, and the exact function $f(x)$ is infinite at the central abscissa.

In either case, the contribution of the central abscissa to the summation is indeterminate.

In practice, the central weight may not have overflowed or underflowed, if there is sufficient rounding error in the value of the central abscissa.

If the soft fail option is used, the weights and abscissa returned may be usable; you must be particularly careful not to 'round' the central abscissa to its true value without simultaneously 'rounding' the central weight to zero or ∞ as appropriate, or the summation will suffer. It would be preferable to use normal weights, if possible.

Note: remember that, when switching from normal weights to adjusted weights or vice versa, redefinition of $f(x)$ is involved.

7 Accuracy

The accuracy depends mainly on n, with increasing loss of accuracy for larger values of n. Typically, one or two decimal digits may be lost from machine accuracy with $n \simeq 20$, and three or four decimal digits may be lost for $n \simeq 100$.

8 Further Comments

The major portion of the time is taken up during the calculation of the eigenvalues of the appropriate tridiagonal matrix, where the time is roughly proportional to n^3.

9 Example

```
itype = int32(-3);
a = 0;
b = 1;
c = 0;
d = 0;
n = int32(7);
[weight, abscis, ifail] = d01bc(itype, a, b, c, d, n)
```

```
weight =
    0.4965
    1.1776
    1.9182
    2.7718
    3.8412
    5.3807
    8.4054
abscis =
    0.1930
    1.0267
    2.5679
    4.9004
    8.1822
   12.7342
```

```
   19.3957
ifail =
           0
```