NAG Toolbox for MATLAB

d01ba

1 Purpose

d01ba computes an estimate of the definite integral of a function of known analytical form, using a Gaussian quadrature formula with a specified number of abscissae. Formulae are provided for a finite interval (Gauss–Legendre), a semi-infinite interval (Gauss–Laguerre, Gauss–Rational), and an infinite interval (Gauss–Hermite).

2 Syntax

```
[result, ifail] = d01ba(d01xxx, a, b, n, fun)
```

3 Description

3.1 General

d01ba evaluates an estimate of the definite integral of a function $f(x)$, over a finite or infinite range, by n-point Gaussian quadrature (see Davis and Rabinowitz 1975, Fröberg 1970, Ralston 1965 or Stroud and Secrest 1966). The integral is approximated by a summation

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

where the w_i are called the weights, and the x_i the abscissae. A selection of values of n is available. (See Section 5.)

3.2 Both Limits Finite

$$\int_a^b f(x)\,dx.$$

The Gauss–Legendre weights and abscissae are used, and the formula is exact for any function of the form:

$$f(x) = \sum_{i=0}^{2n-1} c_i x^i.$$

The formula is appropriate for functions which can be well approximated by such a polynomial over $[a, b]$. It is inappropriate for functions with algebraic singularities at one or both ends of the interval, such as $(1+x)^{-1/2}$ on $[-1, 1]$.

3.3 One Limit Infinite

$$\int_a^\infty f(x)\,dx \qquad \text{or} \qquad \int_{-\infty}^a f(x)\,dx.$$

Two quadrature formulae are available for these integrals.

(a) The Gauss–Laguerre formula is exact for any function of the form:

$$f(x) = e^{-bx} \sum_{i=0}^{2n-1} c_i x^i.$$

This formula is appropriate for functions decaying exponentially at infinity; the parameter b should be chosen if possible to match the decay rate of the function.

(b) The Gauss–Rational formula is exact for any function of the form:

$$f(x) = \sum_{i=2}^{2n+1} \frac{c_i}{(x+b)^i} = \frac{\sum_{i=0}^{2n-1} c_{2n+1-i}(x+b)^i}{(x+b)^{2n+1}}.$$

This formula is likely to be more accurate for functions having only an inverse power rate of decay for large x. Here the choice of a suitable value of b may be more difficult; unfortunately a poor choice of b can make a large difference to the accuracy of the computed integral.

3.4 Both Limits Infinite

$$\int_{-\infty}^{+\infty} f(x)\,dx.$$

The Gauss–Hermite weights and abscissae are used, and the formula is exact for any function of the form:

$$f(x) = e^{-b(x-a)^2} \sum_{i=0}^{2n-1} c_i x^i.$$

Again, for general functions not of this exact form, the parameter b should be chosen to match if possible the decay rate at $\pm\infty$.

4 References

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

Fröberg C E 1970 *Introduction to Numerical Analysis* Addison–Wesley

Ralston A 1965 *A First Course in Numerical Analysis* pp. 87–90 McGraw–Hill

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

5 Parameters

5.1 Compulsory Input Parameters

1: **d01xxx – string**

String specifying the quadrature formula to be used:

`'d01baz'`, for Gauss–Legendre quadrature on a finite interval;

`'d01bay'`, for Gauss–Rational quadrature on a semi-infinite interval;

`'d01bax'`, for Gauss–Laguerre quadrature on a semi-infinite interval;

`'d01baw'`, for Gauss–Hermite quadrature on an infinite interval.

2: **a – double scalar**

3: **b – double scalar**

The parameters a and b which occur in the integration formulae:

Gauss–Legendre:

a is the lower limit and b is the upper limit of the integral. It is not necessary that $a < b$.

Gauss–Rational:

b must be chosen so as to make the integrand match as closely as possible the exact form given in Section 3.3(b). The range of integration is $[a\infty)$ if $a + b > 0$, and $(-\infty a]$ if $a + b < 0$.

Gauss–Laguerre:

b must be chosen so as to make the integrand match as closely as possible the exact form given in Section 3.3(a). The range of integration is $[a\infty)$ if $b > 0$, and $(-\infty a]$ is $b < 0$.

Gauss–Hermite:

a and b must be chosen so as to make the integrand match as closely as possible the exact form given in Section 3.4.

Constraints:

Gauss–Rational: $\mathbf{a} + \mathbf{b} \neq 0$,;
Gauss–Laguerre: $\mathbf{b} \neq 0$,;
Gauss–Hermite: $\mathbf{b} > 0$..

4: **n – int32 scalar**

n, the number of abscissae to be used.

Constraint: $\mathbf{n} = 1$, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 20, 24, 32, 48 or 64.

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

fun must return the value of the integrand f at a specified point.

Its specification is:

```
[result] = fun(x)
```

Input Parameters

1: **x – double scalar**

The point at which the integrand f must be evaluated.

Output Parameters

1: **result – double scalar**

The result of the function.

Some points to bear in mind when coding **fun** are mentioned in Section 7.

5.2 Optional Input Parameters

None.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **result – double scalar**

The result of the function.

2: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Note: d01ba may return useful information for one or more of the following detected errors or warnings.

$\mathbf{ifail} = 1$

The N-point rule is not among those stored. If the soft fail option is used, the answer is evaluated for the largest valid value of **n** less than the requested value.

$\mathbf{ifail} = 2$

The value of **a** and/or **b** is invalid.

Gauss–Rational: $\mathbf{a} + \mathbf{b} = 0$.

Gauss–Laguerre: $\mathbf{b} = 0$.

Gauss–Hermite: $\mathbf{b} \leq 0$.

If the soft fail option is used, the answer is returned as zero.

7 Accuracy

The accuracy depends on the behaviour of the integrand, and on the number of abscissae used. No tests are carried out in the function to estimate the accuracy of the result. If such an estimate is required, the function may be called more than once, with a different number of abscissae each time, and the answers compared. It is to be expected that for sufficiently smooth functions a larger number of abscissae will give improved accuracy.

Alternatively, the range of integration may be subdivided, the integral estimated separately for each sub-interval, and the sum of these estimates compared with the estimate over the whole range.

The coding of the user-supplied real function **fun** may also have a bearing on the accuracy. For example, if a high-order Gauss–Laguerre formula is used, and the integrand is of the form

$$f(x) = e^{-bx} g(x)$$

it is possible that the exponential term may underflow for some large abscissae. Depending on the machine, this may produce an error, or simply be assumed to be zero. In any case, it would be better to evaluate the expression as:

$$f(x) = \exp(-bx + \ln g(x))$$

Another situation requiring care is exemplified by

$$\int_{-\infty}^{+\infty} e^{-x^2} x^m \, dx = 0, \qquad m \text{ odd.}$$

The integrand here assumes very large values; for example, for $m = 63$, the peak value exceeds 3×10^{33}. Now, if the machine holds floating-point numbers to an accuracy of k significant decimal digits, we could not expect such terms to cancel in the summation leaving an answer of much less than 10^{33-k} (the weights being of order unity); that is instead of zero, we obtain a rather large answer through rounding error. Fortunately, such situations are characterised by great variability in the answers returned by formulae with different values of n. In general, you should be aware of the order of magnitude of the integrand, and should judge the answer in that light.

8 Further Comments

The time taken by d01ba depends on the complexity of the expression for the integrand and on the number of abscissae required.

9 Example

```
d01ba_fun.m

function [result] = d01ba_fun(x)
 result=4.0/(1.0+x^2);
```

```
a = 0;
b = 1;
n = int32(4);
[result, ifail] = d01ba('d01baz', a, b, n, 'd01ba_fun')
```

```
result =
    3.1416
ifail =
           0
```