NAG Library Routine Document

D01BAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

D01BAF 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, rational Gauss), and an infinite interval (Gauss–Hermite).

2 Specification

```
FUNCTION D01BAF (D01XXX, A, B, N, FUN, IFAIL)
REAL (KIND=nag_wp) D01BAF
INTEGER            N, IFAIL
REAL (KIND=nag_wp) A, B, FUN
EXTERNAL           D01XXX, FUN
```

3 Description

3.1 General

D01BAF 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

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

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 rational Gauss 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

1: D01XXX – SUBROUTINE, supplied by the NAG Library. *External Procedure*

The name of the routine indicates the quadrature formula:

D01BAZ, for Gauss–Legendre quadrature on a finite interval;

D01BAY, for rational Gauss 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.

The name used must be declared as EXTERNAL in the subroutine from which D01BAF is called.

2: A – REAL (KIND=nag_wp) *Input*
3: B – REAL (KIND=nag_wp) *Input*

On entry: 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 $a < b$.

Rational Gauss:
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:

Rational Gauss: A + B $\neq$ 0.0;
Gauss–Laguerre: B $\neq$ 0.0;
Gauss–Hermite: B $>$ 0.

4: N – INTEGER *Input*

On entry: n, the number of abscissae to be used.

Constraint: N = 1, 2, 3, 4, 5, 6, 8, 10, 12, 14, 16, 20, 24, 32, 48 or 64.

5: FUN – REAL (KIND=nag_wp) FUNCTION, supplied by the user. *External Procedure*

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

The specification of FUN is:

```
FUNCTION FUN (X)
REAL (KIND=nag_wp) FUN
REAL (KIND=nag_wp) X
```

1: X – REAL (KIND=nag_wp) *Input*

On entry: the point at which the integrand f must be evaluated.

FUN must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which D01BAF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

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

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, because for this routine the values of the output parameters may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Note: D01BAF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

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.

IFAIL $= 2$

The value of A and/or B is invalid.

Rational Gauss: A + B $= 0.0$.

Gauss–Laguerre: B $= 0.0$.

Gauss–Hermite: B ≤ 0.0.

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

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

The accuracy depends on the behaviour of the integrand, and on the number of abscissae used. No tests are carried out in D01BAF to estimate the accuracy of the result. If such an estimate is required, the routine 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 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

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

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 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by D01BAF depends on the complexity of the expression for the integrand and on the number of abscissae required.

10 Example

This example evaluates the integrals

$$\int_0^1 \frac{4}{1+x^2} \, dx = \pi$$

by Gauss–Legendre quadrature;

$$\int_2^\infty \frac{1}{x^2 \ln x} \, dx = 0.378671$$

by rational Gauss quadrature with $b = 0$;

$$\int_2^\infty \frac{e^{-x}}{x} \, dx = 0.048901$$

by Gauss–Laguerre quadrature with $b = 1$; and

$$\int_{-\infty}^{+\infty} e^{-3x^2-4x-1} \, dx = \int_{-\infty}^{+\infty} e^{-3(x+1)^2} e^{2x+2} dx = 1.428167$$

by Gauss–Hermite quadrature with $a = -1$ and $b = 3$.

The formulae with $n = 4, 8, 16$ are used in each case.

10.1 Program Text

```
!   D01BAF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.

    Module d01bafe_mod

!     D01BAF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
```

```
      Public                                :: fun1, fun2, fun3, fun4
    Contains
      Function fun1(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                  :: fun1
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)     :: x
!       .. Executable Statements ..
        fun1 = 4.0E0_nag_wp/(1.0E0_nag_wp+x*x)

        Return

      End Function fun1
      Function fun2(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                  :: fun2
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)     :: x
!       .. Intrinsic Procedures ..
        Intrinsic                           :: log
!       .. Executable Statements ..
        fun2 = 1.0E0_nag_wp/(x*x*log(x))

        Return

      End Function fun2
      Function fun3(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                  :: fun3
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)     :: x
!       .. Intrinsic Procedures ..
        Intrinsic                           :: exp
!       .. Executable Statements ..
        fun3 = exp(-x)/x

        Return

      End Function fun3
      Function fun4(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                  :: fun4
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)     :: x
!       .. Intrinsic Procedures ..
        Intrinsic                           :: exp
!       .. Executable Statements ..
        fun4 = exp(-3.0E0_nag_wp*x*x-4.0E0_nag_wp*x-1.0E0_nag_wp)

        Return

      End Function fun4
    End Module d01bafe_mod
    Program d01bafe

!     D01BAF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: d01baf, d01baw, d01bax, d01bay, d01baz, nag_wp
      Use d01bafe_mod, Only: fun1, fun2, fun3, fun4
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter                    :: nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)                    :: a, ans, b
      Integer                               :: i, icase, ifail, nstor
```

```
!     .. Executable Statements ..
      Write (nout,*) 'D01BAF Example Program Results'

cases: Do icase = 1, 4
        Write (nout,*)
        Select Case (icase)
        Case (1)
          Write (nout,*) 'Gauss-Legendre example'
          a = 0.0_nag_wp
          b = 1.0_nag_wp
        Case (2)
          Write (nout,*) 'Gauss-Rational example'
          a = 2.0_nag_wp
          b = 0.0_nag_wp
        Case (3)
          Write (nout,*) 'Gauss-Laguerre example'
          a = 2.0_nag_wp
          b = 1.0_nag_wp
        Case (4)
          Write (nout,*) 'Gauss-Hermite  example'
          a = -1.0_nag_wp
          b = 3.0_nag_wp
        End Select

        Do i = 1, 3
          nstor = 2**(i+1)

          ifail = -1
          Select Case (icase)
          Case (1)
            ans = d01baf(d01baz,a,b,nstor,fun1,ifail)
          Case (2)
            ans = d01baf(d01bay,a,b,nstor,fun2,ifail)
          Case (3)
            ans = d01baf(d01bax,a,b,nstor,fun3,ifail)
          Case (4)
            ans = d01baf(d01baw,a,b,nstor,fun4,ifail)
          End Select

          If (ifail<0) Exit cases
          If (ifail==0 .Or. ifail==1) Write (nout,99999) nstor, ans

        End Do
        Write (nout,*)

      End Do cases

99999 Format (1X,I5,' Points     Answer = ',F10.5)
    End Program d01bafe
```

10.2 Program Data

None.

10.3 Program Results

```
 D01BAF Example Program Results

 Gauss-Legendre example
     4 Points     Answer =    3.14161
     8 Points     Answer =    3.14159
    16 Points     Answer =    3.14159


 Gauss-Rational example
     4 Points     Answer =    0.37910
     8 Points     Answer =    0.37876
    16 Points     Answer =    0.37869
```

```
 Gauss-Laguerre example
     4 Points     Answer =    0.04887
     8 Points     Answer =    0.04890
    16 Points     Answer =    0.04890


 Gauss-Hermite  example
     4 Points     Answer =    1.42803
     8 Points     Answer =    1.42817
    16 Points     Answer =    1.42817
```