NAG Library Routine Document

D01BCF

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

D01BCF 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 Specification

```
SUBROUTINE D01BCF (ITYPE, A, B, C, D, N, WEIGHT, ABSCIS, IFAIL)

INTEGER             ITYPE, N, IFAIL
REAL (KIND=nag_wp) A, B, C, D, WEIGHT(N), ABSCIS(N)
```

3 Description

D01BCF 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) Exponential Gauss

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) Rational Gauss

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 D01BCF may be passed to D01FBF, 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

1: ITYPE – INTEGER *Input*

On entry: indicates the type of quadrature rule.

ITYPE $= 0$
Gauss–Legendre, with normal weights.

ITYPE $= 1$
Gauss–Jacobi, with normal weights.

ITYPE $= -1$
Gauss–Jacobi, with adjusted weights.

ITYPE $= 2$
Exponential Gauss, with normal weights.

ITYPE $= -2$
Exponential Gauss, with adjusted weights.

ITYPE $= 3$
Gauss–Laguerre, with normal weights.

ITYPE $= -3$
Gauss–Laguerre, with adjusted weights.

ITYPE $= 4$
Gauss–Hermite, with normal weights.

ITYPE $= -4$
Gauss–Hermite, with adjusted weights.

ITYPE $= 5$
Rational Gauss, with normal weights.

ITYPE $= -5$
Rational Gauss, with adjusted weights.

Constraint: ITYPE $= 0$, 1, -1, 2, -2, 3, -3, 4, -4, 5 or -5.

2: A – REAL (KIND=nag_wp) *Input*
3: B – REAL (KIND=nag_wp) *Input*
4: C – REAL (KIND=nag_wp) *Input*
5: D – REAL (KIND=nag_wp) *Input*

On entry: 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 $= 1$, -1, 5 or -5. For some rules C and D must not be too large (see Section 6).

6: N – INTEGER *Input*

On entry: 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.

7: WEIGHT(N) – REAL (KIND=nag_wp) array *Output*

On exit: the N weights.

8: ABSCIS(N) – REAL (KIND=nag_wp) array *Output*

On exit: the N abscissae.

9: 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, if you are not familiar with this parameter, the recommended value is 0. **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).

Errors or warnings detected by the routine:

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, $\mathrm{N} < 1$,
or $\mathrm{ITYPE} < -5$,
or $\mathrm{ITYPE} > 5$.

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 $\mathrm{ITYPE} = 0$, $\mathrm{A} \geq \mathrm{B}$;

if $\mathrm{ITYPE} = \pm 1$, $\mathrm{A} \geq \mathrm{B}$ or $\mathrm{C} \leq -1.0$ or $\mathrm{D} \leq -1.0$ or $\mathrm{C} + \mathrm{D} + 2.0 > gmax$;

if $\mathrm{ITYPE} = \pm 2$, $\mathrm{A} \geq \mathrm{B}$ or $\mathrm{C} \leq -1.0$;

if $\mathrm{ITYPE} = \pm 3$, $\mathrm{B} = 0.0$ or $\mathrm{C} \leq -1.0$ or $\mathrm{C} + 1.0 > gmax$;

if $\mathrm{ITYPE} = \pm 4$, $\mathrm{B} \leq 0.0$ or $\mathrm{C} \leq -1.0$ or $(\mathrm{C} + 1.0/2.0) > gmax$;

if $\mathrm{ITYPE} = \pm 5$, $\mathrm{A} + \mathrm{B} = 0.0$ or $\mathrm{C} \leq -1.0$ or $\mathrm{D} \leq \mathrm{C} + 1.0$.

Here *gmax* is the (machine-dependent) largest integer value such that $\Gamma(gmax)$ can be computed without overflow (see the Users' Note for your implementation for S14AAF).

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 X02ALF. 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 $= 6$

Exponential Gauss 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

This example returns the abscissae and (adjusted) weights for the seven-point Gauss–Laguerre formula.

9.1 Program Text

```
    Program d01bcfe

!     D01BCF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: d01bcf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: n = 7, nout = 6
!     .. Local Scalars ..
```

```
      Real (Kind=nag_wp)                :: a, b, c, d
      Integer                           :: ifail, itype, j
!     .. Local Arrays ..
      Real (Kind=nag_wp)                :: abscis(n), weight(n)
!     .. Executable Statements ..
      Write (nout,*) 'D01BCF Example Program Results'

      a = 0.0E0_nag_wp
      b = 1.0E0_nag_wp
      c = 0.0E0_nag_wp
      d = 0.0E0_nag_wp
      itype = -3

      ifail = 0
      Call d01bcf(itype,a,b,c,d,n,weight,abscis,ifail)

      Write (nout,*)
      Write (nout,99999) 'Laguerre formula,', n, ' points'
      Write (nout,*)
      Write (nout,*) '      Abscissae             Weights'
      Write (nout,*)
      Write (nout,99998)(abscis(j),weight(j),j=1,n)

99999 Format (1X,A,I3,A)
99998 Format (1X,E15.5,5X,E15.5)
    End Program d01bcfe
```

9.2 Program Data

None.

9.3 Program Results

```
 D01BCF Example Program Results

 Laguerre formula,  7 points

       Abscissae             Weights

     0.19304E+00         0.49648E+00
     0.10267E+01         0.11776E+01
     0.25679E+01         0.19182E+01
     0.49004E+01         0.27718E+01
     0.81822E+01         0.38412E+01
     0.12734E+02         0.53807E+01
     0.19396E+02         0.84054E+01
```