NAG Fortran Library Routine Document

D01FBF

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

D01FBF computes an estimate of a multi-dimensional integral (from 1 to 20 dimensions), given the analytic form of the integrand and suitable Gaussian weights and abscissae.

2 Specification

```
real FUNCTION D01FBF(NDIM, NPTVEC, LWA, WEIGHT, ABSCIS, FUN, IFAIL)
INTEGER              NDIM, NPTVEC(NDIM), LWA, IFAIL
real                 WEIGHT(LWA), ABSCIS(LWA), FUN
EXTERNAL             FUN
```

3 Description

This routine approximates a multi-dimensional integral by evaluating the summation

$$\sum_{i_1=1}^{l_1} w_{1,i_1} \sum_{i_2=1}^{l_2} w_{2,i_2} \cdots \sum_{i_n=1}^{l_n} w_{n,i_n} f(x_{1,i_1}, x_{2,i_2}, \ldots, x_{n,i_n})$$

given the weights w_{j,i_j} and abscissae x_{j,i_j} for a multi-dimensional product integration rule (see Davis and Rabinowitz (1975)). The number of dimensions may be anything from 1 to 20.

The weights and abscissae for each dimension must have been placed in successive segments of the arrays WEIGHT and ABSCIS; for example, by calling D01BBF or D01BCF once for each dimension using a quadrature formula and number of abscissae appropriate to the range of each x_j and to the functional dependence of f on x_j.

If normal weights are used, the summation will approximate the integral

$$\int w_1(x_1) \int w_2(x_2) \cdots \int w_n(x_n) f(x_1, x_2, \ldots, x_n)\, dx_n \cdots dx_2\, dx_1$$

where $w_j(x)$ is the weight function associated with the quadrature formula chosen for the jth dimension; while if adjusted weights are used, the summation will approximate the integral

$$\int \int \cdots \int f(x_1, x_2, \ldots, x_n)\, dx_n \cdots dx_2\, dx_1.$$

The user must supply a routine to evaluate

$$f(x_1, x_2, \ldots, x_n)$$

at any values of $x_1, x_2, \ldots, x_n$ within the range of integration.

4 References

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

5 Parameters

1: NDIM – INTEGER *Input*

On entry: the number of dimensions of the integral, n.

Constraint: $1 \leq \text{NDIM} \leq 20$.

2: NPTVEC(NDIM) – INTEGER array *Input*

On entry: NPTVEC(j) must specify the number of points in the jth dimension of the summation, for $j = 1, 2, \ldots, n$.

3: LWA – INTEGER *Input*

On entry: the dimension of the arrays WEIGHT and ABSCIS as declared in the (sub)program from which D01FBF is called.

Constraint: $\text{LWA} \geq \text{NPTVEC}(1) + \text{NPTVEC}(2) + \cdots + \text{NPTVEC}(\text{NDIM})$.

4: WEIGHT(LWA) – ***real*** array *Input*

On entry: WEIGHT must contain in succession the weights for the various dimensions, i.e., WEIGHT(k) contains the ith weight for the jth dimension, with

$$k = \text{NPTVEC}(1) + \text{NPTVEC}(2) + \cdots + \text{NPTVEC}(j-1) + i.$$

5: ABSCIS(LWA) – ***real*** array *Input*

On entry: ABSCIS must contain in succession the abscissae for the various dimensions, i.e., ABSCIS(k) contains the ith abscissa for the jth dimension, with

$$k = \text{NPTVEC}(1) + \text{NPTVEC}(2) + \cdots + \text{NPTVEC}(j-1) + i.$$

6: FUN – ***real*** FUNCTION, supplied by the user. *External Procedure*

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

Its specification is:

```
real FUNCTION FUN(NDIM, X)
INTEGER          NDIM
real             X(NDIM)
```

1: NDIM – INTEGER *Input*

On entry: the number of dimensions of the integral, n.

2: X(NDIM) – ***real*** array *Input*

On entry: the co-ordinates of the point at which the integrand must be evaluated.

FUN must be declared as EXTERNAL in the (sub)program from which D01FBF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

7: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

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, for users 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.**

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 $= 1$

On entry, NDIM < 1,
or NDIM > 20,
or LWA $<$ NPTVEC(1) $+$ NPTVEC(2) $+ \cdots +$ NPTVEC(NDIM).

7 Accuracy

The accuracy of the computed multi-dimensional sum depends on the weights and the integrand values at the abscissae. If these numbers vary significantly in size and sign then considerable accuracy could be lost. If these numbers are all positive, then little accuracy will be lost in computing the sum.

8 Further Comments

The total time taken by the routine will be proportional to

$$T \times \text{NPTVEC}(1) \times \text{NPTVEC}(2) \times \cdots \times \text{NPTVEC}(\text{NDIM}),$$

where T is the time taken for one evaluation of FUN.

9 Example

This example program evaluates the integral

$$\int_1^2 \int_0^\infty \int_{-\infty}^\infty \int_1^\infty \frac{(x_1 x_2 x_3)^6}{(x_4+2)^8} e^{-2x_2} e^{-0.5x_3^2} \, dx_4 \, dx_3 \, dx_2 \, dx_1$$

using adjusted weights. The quadrature formulae chosen are:

x_1 : Gauss–Legendre, $a = 1.0$, $b = 2.0$,

x_2 : Gauss–Laguerre, $a = 0.0$, $b = 2.0$,

x_3 : Gauss–Hermite, $a = 0.0$, $b = 0.5$,

x_4 : Gauss–Rational, $a = 1.0$, $b = 2.0$.

Four points are sufficient in each dimension, as this integral in is in fact a product of four one-dimensional integrals, for each of which the chosen four-point formula is exact.

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     D01FBF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NDIM, LWAMAX
      PARAMETER        (NDIM=4,LWAMAX=16)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             A, ANS, B
      INTEGER          I, IFAIL, ITYPE, IW, LWA
*     .. Local Arrays ..
      real             ABSCIS(LWAMAX), WEIGHT(LWAMAX)
      INTEGER          NPTVEC(NDIM)
*     .. External Functions ..
```

```
      real              D01FBF, FUN
      EXTERNAL          D01FBF, FUN
*     .. External Subroutines ..
      EXTERNAL          D01BAW, D01BAX, D01BAY, D01BAZ, D01BBF
*     .. Data statements ..
      DATA              NPTVEC/4, 4, 4, 4/
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D01FBF Example Program Results'
      LWA = 0
      DO 20 I = 1, NDIM
         LWA = LWA + NPTVEC(I)
   20 CONTINUE
      IF (LWAMAX.GE.LWA) THEN
         ITYPE = 1
         IW = 1
         A = 1.0e0
         B = 2.0e0
         IFAIL = 0
*
         CALL D01BBF(D01BAZ,A,B,ITYPE,NPTVEC(1),WEIGHT(IW),ABSCIS(IW),
     +               IFAIL)
*
         IW = IW + NPTVEC(1)
         A = 0.0e0
         B = 2.0e0
*
         CALL D01BBF(D01BAX,A,B,ITYPE,NPTVEC(2),WEIGHT(IW),ABSCIS(IW),
     +               IFAIL)
*
         IW = IW + NPTVEC(2)
         A = 0.0e0
         B = 0.5e0
*
         CALL D01BBF(D01BAW,A,B,ITYPE,NPTVEC(3),WEIGHT(IW),ABSCIS(IW),
     +               IFAIL)
*
         IW = IW + NPTVEC(3)
         A = 1.0e0
         B = 2.0e0
*
         CALL D01BBF(D01BAY,A,B,ITYPE,NPTVEC(4),WEIGHT(IW),ABSCIS(IW),
     +               IFAIL)
*
         IFAIL = 0
*
         ANS = D01FBF(NDIM,NPTVEC,LWA,WEIGHT,ABSCIS,FUN,IFAIL)
*
         WRITE (NOUT,*)
         WRITE (NOUT,99999) 'Answer = ', ANS
      END IF
      STOP
*
99999 FORMAT (1X,A,F10.5)
      END
*
      real FUNCTION FUN(NDIM,X)
*     .. Scalar Arguments ..
      INTEGER           NDIM
*     .. Array Arguments ..
      real              X(NDIM)
*     .. Intrinsic Functions ..
      INTRINSIC         EXP
*     .. Executable Statements ..
      FUN = (X(1)*X(2)*X(3))**6/(X(4)+2.0e0)**8*EXP(-2.0e0*X(2)
     +      -0.5e0*X(3)*X(3))
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 D01FBF Example Program Results

 Answer =    0.25065
```