NAG 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 multidimensional integral (from 1 to 20 dimensions), given the analytic form of the integrand and suitable Gaussian weights and abscissae.

2 Specification

```
FUNCTION D01FBF (NDIM, NPTVEC, LWA, WEIGHT, ABSCIS, FUN, IFAIL)
REAL (KIND=nag_wp) D01FBF
INTEGER            NDIM, NPTVEC(NDIM), LWA, IFAIL
REAL (KIND=nag_wp) WEIGHT(LWA), ABSCIS(LWA), FUN
EXTERNAL           FUN
```

3 Description

D01FBF approximates a multidimensional integral by evaluating the summation

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

given the weights w_{j,i_j} and abscissae x_{j,i_j} for a multidimensional 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 D01BCF or D01TBF 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

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

You must supply a subroutine 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 Arguments

1: NDIM – INTEGER *Input*

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

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 (KIND=nag_wp) array *Input*

On entry: 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 (KIND=nag_wp) array *Input*

On entry: 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 (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 (NDIM, X)
REAL (KIND=nag_wp) FUN
INTEGER            NDIM
REAL (KIND=nag_wp) X(NDIM)
```

1: NDIM – INTEGER *Input*

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

2: X(NDIM) – REAL (KIND=nag_wp) array *Input*

On entry: the coordinates of 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 D01FBF is called. Arguments 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. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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 $= 1$

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

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The accuracy of the computed multidimensional 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 Parallelism and Performance

D01FBF is not threaded in any implementation.

9 Further Comments

The total time taken by D01FBF 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.

10 Example

This example 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: rational Gauss, $a = 1.0$, $b = 2.0$.

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

10.1 Program Text

```
!   D01FBF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module d01fbfe_mod

!     D01FBF 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                           :: fun
!     .. Parameters ..
      Integer, Parameter, Public       :: ndim = 4, nout = 6
    Contains
      Function fun(ndim,x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)             :: fun
!       .. Scalar Arguments ..
        Integer, Intent (In)           :: ndim
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (In) :: x(ndim)
!       .. Intrinsic Procedures ..
        Intrinsic                      :: exp
!       .. Executable Statements ..
        fun = (x(1)*x(2)*x(3))**6/(x(4)+2.0E0_nag_wp)**8*                    &
          exp(-2.0E0_nag_wp*x(2)-0.5E0_nag_wp*x(3)*x(3))

        Return

      End Function fun
    End Module d01fbfe_mod
    Program d01fbfe

!     D01FBF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: d01fbf, d01tbf, nag_wp
      Use d01fbfe_mod, Only: fun, ndim, nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: a, ans, b
      Integer                          :: i, ifail, j, lwa
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: abscis(:), weight(:)
      Integer                          :: nptvec(ndim)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: sum
!     .. Executable Statements ..
      Write (nout,*) 'D01FBF Example Program Results'

      nptvec(1:ndim) = (/4,4,4,4/)
```

```
      lwa = sum(nptvec(1:ndim))
      Allocate (abscis(lwa),weight(lwa))

      j = 1

      Do i = 1, 4

        ifail = 0
        Select Case (i)
        Case (1)
          a = 1.0E0_nag_wp
          b = 2.0E0_nag_wp
          Call d01tbf(0,a,b,nptvec(i),weight(j),abscis(j),ifail)
        Case (2)
          a = 0.0E0_nag_wp
          b = 2.0E0_nag_wp
          Call d01tbf(-3,a,b,nptvec(i),weight(j),abscis(j),ifail)
        Case (3)
          a = 0.0E0_nag_wp
          b = 0.5E0_nag_wp
          Call d01tbf(-4,a,b,nptvec(i),weight(j),abscis(j),ifail)
        Case (4)
          a = 1.0E0_nag_wp
          b = 2.0E0_nag_wp
          Call d01tbf(-5,a,b,nptvec(i),weight(j),abscis(j),ifail)
        End Select

        j = j + nptvec(i)
      End Do

      ifail = 0
      ans = d01fbf(ndim,nptvec,lwa,weight,abscis,fun,ifail)

      Write (nout,*)
      Write (nout,99999) 'Answer = ', ans

99999 Format (1X,A,F10.5)
    End Program d01fbfe
```

10.2 Program Data

None.

10.3 Program Results

```
 D01FBF Example Program Results

 Answer =    0.25065
```