NAG Library Routine Document

D01GBF

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

D01GBF returns an approximation to the integral of a function over a hyper-rectangular region, using a Monte–Carlo method. An approximate relative error estimate is also returned. This routine is suitable for low accuracy work.

2   Specification

```
SUBROUTINE D01GBF (NDIM, A, B, MINCLS, MAXCLS, FUNCTN, EPS, ACC, LENWRK,    &
                   WRKSTR, FINEST, IFAIL)
INTEGER             NDIM, MINCLS, MAXCLS, LENWRK, IFAIL
REAL (KIND=nag_wp)  A(NDIM), B(NDIM), FUNCTN, EPS, ACC, WRKSTR(LENWRK),     &
                    FINEST
EXTERNAL            FUNCTN
```

3   Description

D01GBF uses an adaptive Monte–Carlo method based on the algorithm described in Lautrup (1971). It is implemented for integrals of the form:

$$\int_{a_1}^{b_1} \int_{a_2}^{b_2} \cdots \int_{a_n}^{b_n} f(x_1, x_2, \ldots, x_n) \, dx_n \cdots dx_2 dx_1.$$

Upon entry, unless LENWRK has been set to the minimum value $10 \times$ NDIM, the routine subdivides the integration region into a number of equal volume subregions. Inside each subregion the integral and the variance are estimated by means of pseudorandom sampling. All contributions are added together to produce an estimate for the whole integral and total variance. The variance along each coordinate axis is determined and the routine uses this information to increase the density and change the widths of the sub-intervals along each axis, so as to reduce the total variance. The total number of subregions is then increased by a factor of two and the program recycles for another iteration. The program stops when a desired accuracy has been reached or too many integral evaluations are needed for the next cycle.

4   References

Lautrup B (1971) An adaptive multi-dimensional integration procedure *Proc. 2nd Coll. Advanced Methods in Theoretical Physics, Marseille*

5   Arguments

1:    NDIM – INTEGER   *Input*

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

Constraint: NDIM ≥ 1.

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

On entry: the lower limits of integration, a_i, for $i = 1, 2, \ldots, n$.

3:    B(NDIM) – REAL (KIND=nag_wp) array   *Input*

On entry: the upper limits of integration, b_i, for $i = 1, 2, \ldots, n$.

4:    MINCLS – INTEGER   *Input/Output*

On entry: must be set

either to the minimum number of integrand evaluations to be allowed, in which case MINCLS ≥ 0;

or to a negative value. In this case, the routine assumes that a previous call had been made with the same arguments NDIM, A and B and with either the same integrand (in which case D01GBF continues calculation) or a similar integrand (in which case D01GBF begins the calculation with the subdivision used in the last iteration of the previous call). See also WRKSTR.

On exit: contains the number of integrand evaluations actually used by D01GBF.

5:    MAXCLS – INTEGER   *Input*

On entry: the maximum number of integrand evaluations to be allowed. In the continuation case this is the number of new integrand evaluations to be allowed. These counts do not include zero integrand values.

Constraints:

MAXCLS > MINCLS;
MAXCLS ≥ 4 × (NDIM + 1).

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

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

The specification of FUNCTN is:

```
FUNCTION FUNCTN (NDIM, X)
REAL (KIND=nag_wp) FUNCTN

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.

FUNCTN must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which D01GBF is called. Arguments denoted as *Input* must **not** be changed by this procedure.

7:    EPS – REAL (KIND=nag_wp)   *Input*

On entry: the relative accuracy required.

Constraint: EPS ≥ 0.0.

8:    ACC – REAL (KIND=nag_wp)   *Output*

On exit: the estimated relative accuracy of FINEST.

9:    LENWRK – INTEGER   *Input*

On entry: the dimension of the array WRKSTR as declared in the (sub)program from which D01GBF is called.

For maximum efficiency, LENWRK should be about

$$3 \times \text{NDIM} \times (\text{MAXCLS}/4)^{1/\text{NDIM}} + 7 \times \text{NDIM}.$$

If LENWRK is given the value $10 \times \text{NDIM}$ then the subroutine uses only one iteration of a crude Monte–Carlo method with MAXCLS sample points.

Constraint: $\text{LENWRK} \geq 10 \times \text{NDIM}$.

10:    WRKSTR(LENWRK) – REAL (KIND=nag_wp) array   *Input/Output*

On entry: if $\text{MINCLS} < 0$, WRKSTR must be unchanged from the previous call of D01GBF – except that for a new integrand WRKSTR(LENWRK) must be set to 0.0. See also MINCLS.

On exit: contains information about the current sub-interval structure which could be used in later calls of D01GBF. In particular, WRKSTR(j) gives the number of sub-intervals used along the jth coordinate axis.

11:    FINEST – REAL (KIND=nag_wp)   *Input/Output*

On entry: must be unchanged from a previous call to D01GBF.

On exit: the best estimate obtained for the integral.

12:    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, because for this routine the values of the output arguments may be useful even if $\text{IFAIL} \neq 0$ 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: $\text{IFAIL} = 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6   Error Indicators and Warnings

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

Errors or warnings detected by the routine:

$\text{IFAIL} = 1$

On entry, $\text{NDIM} < 1$,
or $\text{MINCLS} \geq \text{MAXCLS}$,
or $\text{LENWRK} < 10 \times \text{NDIM}$,
or $\text{MAXCLS} < 4 \times (\text{NDIM} + 1)$,
or $\text{EPS} < 0.0$.

$\text{IFAIL} = 2$

MAXCLS was too small for D01GBF to obtain the required relative accuracy EPS. In this case D01GBF returns a value of FINEST with estimated relative error ACC, but ACC will be greater than EPS. This error exit may be taken before MAXCLS nonzero integrand evaluations have actually occurred, if the routine calculates that the current estimates could not be improved before MAXCLS was exceeded.

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

A relative error estimate is output through the argument ACC. The confidence factor is set so that the actual error should be less than ACC 90% of the time. If you want a higher confidence level then a smaller value of EPS should be used.

8 Parallelism and Performance

D01GBF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The running time for D01GBF will usually be dominated by the time used to evaluate the integrand f, so the maximum time that could be used is approximately proportional to MAXCLS.

For some integrands, particularly those that are poorly behaved in a small part of the integration region, D01GBF may terminate with a value of ACC which is significantly smaller than the actual relative error. This should be suspected if the returned value of MINCLS is small relative to the expected difficulty of the integral. Where this occurs, D01GBF should be called again, but with a higher entry value of MINCLS (e.g., twice the returned value) and the results compared with those from the previous call.

9.1 Additional Information

The exact values of FINEST and ACC on return will depend (within statistical limits) on the sequence of random numbers generated within this routine.

10 Example

This example calculates the integral

$$\int_0^1 \int_0^1 \int_0^1 \int_0^1 \frac{4x_1 x_3 \exp(2x_1 x_3)}{(1 + x_2 + x_4)^2} dx_1 dx_2 dx_3 dx_4 = 0.575364.$$

10.1 Program Text

```
!   D01GBF Example Program Text
!   Mark 26 Release. NAG Copyright 2016.

    Module d01gbfe_mod

!       D01GBF 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                          :: functn
!       .. Parameters ..
        Integer, Parameter, Public      :: lenwrk = 500, maxcls = 20000,       &
                                           ndim = 4, nout = 6
    Contains
        Function functn(ndim,x)

!           .. Function Return Value ..
            Real (Kind=nag_wp)          :: functn
!           .. Scalar Arguments ..
            Integer, Intent (In)        :: ndim
!           .. Array Arguments ..
            Real (Kind=nag_wp), Intent (In) :: x(ndim)
!           .. Intrinsic Procedures ..
            Intrinsic                   :: exp
!           .. Executable Statements ..
            functn = 4.0E0_nag_wp*x(1)*x(3)**2*exp(2.0E0_nag_wp*x(1)*x(3))/     &
              (1.0E0_nag_wp+x(2)+x(4))**2

            Return

        End Function functn
    End Module d01gbfe_mod
    Program d01gbfe

!       D01GBF Example Main Program

!       .. Use Statements ..
        Use nag_library, Only: d01gbf, nag_wp
        Use d01gbfe_mod, Only: functn, lenwrk, maxcls, ndim, nout
!       .. Implicit None Statement ..
        Implicit None
!       .. Local Scalars ..
        Real (Kind=nag_wp)              :: acc, eps, finest
        Integer                         :: ifail, mincls
!       .. Local Arrays ..
        Real (Kind=nag_wp), Allocatable :: a(:), b(:), wrkstr(:)
!       .. Executable Statements ..
        Write (nout,*) 'D01GBF Example Program Results'

        Allocate (a(ndim),b(ndim),wrkstr(lenwrk))

        a(1:ndim) = 0.0E0_nag_wp
        b(1:ndim) = 1.0E0_nag_wp
        eps = 0.01E0_nag_wp
        mincls = 1000

        ifail = -1
        Call d01gbf(ndim,a,b,mincls,maxcls,functn,eps,acc,lenwrk,wrkstr,finest, &
          ifail)

        Select Case (ifail)
        Case (0,2)
          Write (nout,*)
          Write (nout,99998) 'Requested accuracy    = ', eps
```

```
      Write (nout,99997) 'Estimated value        = ', finest
      Write (nout,99998) 'Estimated accuracy     = ', acc
      Write (nout,99999) 'Number of evaluations = ', mincls
    End Select

99999 Format (1X,A,I5)
99998 Format (1X,A,E13.2)
99997 Format (1X,A,F13.5)
    End Program d01gbfe
```

10.2 Program Data

None.

10.3 Program Results

```
D01GBF Example Program Results

Requested accuracy    =       0.10E-01
Estimated value       =        0.57572
Estimated accuracy    =       0.92E-02
Number of evaluations =   1728
```