# D01GCF – NAG Fortran Library Routine Document

**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

D01GCF calculates an approximation to a definite integral in up to 20 dimensions, using the Korobov–Conroy number theoretic method.

## 2 Specification

```
      SUBROUTINE D01GCF(NDIM, FUNCTN, REGION, NPTS, VK, NRAND, ITRANS,
     1                  RES, ERR, IFAIL)
      INTEGER           NDIM, NPTS, NRAND, ITRANS, IFAIL
      real              FUNCTN, VK(NDIM), RES, ERR
      EXTERNAL          FUNCTN, REGION
```

## 3 Description

This routine calculates an approximation to the integral,

$$I = \int_{c_1}^{d_1} dx_1, \ldots, \int_{c_n}^{d_n} dx_n \;\; f(x_1, x_2, \ldots, x_n) \tag{1}$$

using the Korobov–Conroy number theoretic method ([1], [2], [3]). The region of integration defined in (1) is such that generally $c_i$ and $d_i$ may be functions of $x_1, x_2, \ldots, x_{i-1}$, for $i = 2, 3, \ldots, n$, with $c_1$ and $d_1$ constants. The integral is first of all transformed to an integral over the $n$-cube $[0, 1]^n$ by the change of variables

$$x_i = c_i + (d_i - c_i)y_i, \;\; i = 1, 2, \ldots, n.$$

The method then uses as its basis the number theoretic formula for the $n$ -cube, $[0, 1]^n$:

$$\int_0^1 dx_1 \;\ldots \int_0^1 dx_n \; g(x_1, x_2, \ldots, x_n) = \frac{1}{p}\sum_{k=1}^{p} g\left(\left\{k\frac{a_1}{p}\right\}, \ldots, \left\{k\frac{a_n}{p}\right\}\right) - E \tag{2}$$

where $\{x\}$ denotes the fractional part of $x$, $a_1, a_2, \ldots, a_n$ are the so-called optimal coefficients, $E$ is the error and $p$ is a prime integer. (It is strictly only necessary that $p$ be relatively prime to all $a_1, a_2, \ldots, a_n$ and is in fact chosen to be even for some cases in Conroy [3].) The method makes use of properties of the Fourier expansion of $g(x_1, x_2, \ldots, x_n)$ which is assumed to have some degree of periodicity. Depending on the choice of $a_1, a_2, \ldots, a_n$ the contributions from certain groups of Fourier coefficients are eliminated from the error, $E$. Korobov shows that $a_1, a_2, \ldots, a_n$ can be chosen so that the error satisfies

$$E \leq CKp^{-\alpha} \ln^{\alpha\beta} p \tag{3}$$

where $\alpha$ and $C$ are real numbers depending on the convergence rate of the Fourier series, $\beta$ is a constant depending on $n$ and $K$ is a constant depending on $\alpha$ and $n$. There are a number of procedures for calculating these optimal coefficients. Korobov imposes the constraint that

$$\begin{aligned} a_1 &= 1 \\ a_i &= a^{i-1} (\text{mod } p) \end{aligned} \tag{4}$$

and gives a procedure for calculating the parameter, $a$, to satisfy the optimal conditions.

In this routine the periodisation is achieved by the simple transformation

$$x_i = y_i^2(3 - 2y_i), \qquad i = 1, 2, \ldots, n.$$

More sophisticated periodisation procedures are available but in practice the degree of periodisation does not appear to be a critical requirement of the method.

An easily calculable error estimate is not available apart from repetition with an increasing sequence of values of $p$ which can yield erratic results. The difficulties have been studied by Cranley and Patterson [4] who have proposed a Monte Carlo error estimate arising from converting (2) into a stochastic integration rule by the inclusion of a random origin shift which leaves the form of the error (3) unchanged; i.e., in the formula (2), $\left\{k\frac{a_i}{p}\right\}$ is replaced by $\left\{\alpha_i + k\frac{a_i}{p}\right\}$, for $i = 1, 2, \ldots, n$, where each $\alpha_i$, is uniformly distributed over $[0, 1]$. Computing the integral for each of a sequence of random vectors $\alpha$ allows a 'standard error' to be estimated.

This routine provides built-in sets of optimal coefficients, corresponding to six different values of $p$. Alternatively the optimal coefficients may be supplied by the user. Routines D01GYF and D01GZF compute the optimal coefficients for the cases where $p$ is a prime number or $p$ is a product of 2 primes, respectively.

# 4 References

**[1]** Korobov N M (1957) The approximate calculation of multiple integrals using number theoretic methods *Dokl. Acad. Nauk SSSR* **115** 1062–1065

**[2]** Korobov N M (1963) *Number Theoretic Methods in Approximate Analysis* Fizmatgiz, Moscow

**[3]** Conroy H (1967) Molecular Shroedinger equation VIII. A new method for evaluting multi-dimensional integrals *J. Chem. Phys.* **47** 5307–5318

**[4]** Cranley R and Patterson T N L (1976) Randomisation of number theoretic methods for mulitple integration *SIAM J. Numer. Anal.* **13** 904–914

# 5 Parameters

**1:** NDIM — INTEGER *Input*

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

*Constraint:* $1 \le \text{NDIM} \le 20$.

**2:** FUNCTN — ***real*** FUNCTION, supplied by the user. *External Procedure*

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

Its specification is:

```
real FUNCTION FUNCTN(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.

FUNCTN must be declared as EXTERNAL in the (sub)program from which D01GCF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

**3:** REGION — SUBROUTINE, supplied by the user. *External Procedure*

REGION must evaluate the limits of integration in any dimension.

Its specification is:

```
SUBROUTINE REGION(NDIM, X, J, C, D)
INTEGER           NDIM, J
real              X(NDIM), C, D
```

**1:** NDIM — INTEGER *Input*

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

**2:** X(NDIM) — ***real*** array *Input*

*On entry:* $\mathrm{X}(1),\ldots,\mathrm{X}(j-1)$ contain the current values of the first $(j-1)$ variables, which may be used if necessary in calculating $c_j$ and $d_j$.

**3:** J — INTEGER *Input*

*On entry:* the index $j$ for which the limits of the range of integration are required.

**4:** C — ***real*** *Output*

*On exit:* the lower limit $c_j$ of the range of $x_j$.

**5:** D — ***real*** *Output*

*On exit:* the upper limit $d_j$ of the range of $x_j$.

REGION must be declared as EXTERNAL in the (sub)program from which D01GCF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

**4:** NPTS — INTEGER *Input*

*On entry:* the Korobov rule to be used. There are two alternatives depending on the value of NPTS.

(a) $1 \leq \mathrm{NPTS} \leq 6$.

In this case one of six preset rules is chosen using 2129, 5003, 10007, 20011, 40009 or 80021 points depending on the respective value of NPTS being 1, 2, 3, 4, 5 or 6.

(b) $\mathrm{NPTS} > 6$.

NPTS is the number of actual points to be used with corresponding optimal coefficients supplied in the array VK.

*Constraint:* $\mathrm{NPTS} \geq 1$.

**5:** VK(NDIM) — ***real*** array *Input/Output*

*On entry:* if $\mathrm{NPTS} > 6$, VK must contain the $n$ optimal coefficients (which may be calculated using D01GYF or D01GZF); if $\mathrm{NPTS} \leq 6$, VK need not be set.

*On exit:* if $\mathrm{NPTS} > 6$, VK is unchanged; if $\mathrm{NPTS} \leq 6$, VK contains the $n$ optimal coefficients used by the preset rule.

**6:** NRAND — INTEGER *Input*

*On entry:* the number of random samples to be generated in the error estimation (generally a small value, say 3 to 5 is sufficient). The total number of integrand evaluations will be $\mathrm{NRAND} \times \mathrm{NPTS}$.

*Constraint:* $\mathrm{NRAND} \geq 1$.

**7:** ITRANS — INTEGER *Input*

*On entry:* indicates whether the periodising transformation is to be used:

if ITRANS = 0, the transformation is to be used.

if ITRANS ≠ 0, the transformation is to be suppresssed (to cover cases where the integrand may already be periodic or where the user desires to specify a particular transformation in the definition of FUNCTN).

*Suggested value:* ITRANS = 0.

**8:** RES — ***real*** *Output*

*On exit:* an estimate of the value of the integral.

**9:** ERR — ***real*** *Output*

*On exit:* the standard error as computed from NRAND sample values. If NRAND = 1, then ERR contains zero.

**10:** IFAIL — INTEGER *Input/Output*

*On entry:* IFAIL must be set to 0, −1 or 1. For users not familiar with this parameter (described in Chapter P01) the recommended value is 0.

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

# 6 Error Indicators and Warnings

Errors detected by the routine:

IFAIL = 1

On entry, NDIM < 1,
or NDIM > 20.

IFAIL = 2

On entry, NPTS < 1.

IFAIL = 3

On entry, NRAND < 1.

# 7 Accuracy

An estimate of the absolute standard error is given by the value, on exit, of ERR.

# 8 Further Comments

The time taken by the routine will be approximately proportional to NRAND $\times p$, where $p$ is the number of points used.

The exact values of RES and ERR returned by D01GCF will depend (within statistical limits) on the sequence of random numbers generated within the routine by calls to G05CAF. To ensure that the results returned by D01GCF in separate runs are identical, users should call G05CBF immediately before calling D01GCF; to ensure that they are different, call G05CCF.

# 9 Example

This example calculates the integral

$$\int_0^1 \int_0^1 \int_0^1 \int_0^1 \cos(0.5 + 2(x_1 + x_2 + x_3 + x_4) - 4)\, dx_1\, dx_2\, dx_3\, dx_4.$$

## 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.

```
*     D01GCF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NDIM
      PARAMETER        (NDIM=4)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             ERR, RES
      INTEGER          IFAIL, ITRANS, NPTS, NRAND
*     .. Local Arrays ..
      real             VK(NDIM)
*     .. External Functions ..
      real             FUNCT
      EXTERNAL         FUNCT
*     .. External Subroutines ..
      EXTERNAL         D01GCF, REGION
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D01GCF Example Program Results'
      NPTS = 2
      ITRANS = 0
      NRAND = 4
      IFAIL = 0
*
      CALL D01GCF(NDIM,FUNCT,REGION,NPTS,VK,NRAND,ITRANS,RES,ERR,IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,99999) 'Result =', RES, '  Standard error =', ERR
      STOP
*
99999 FORMAT (1X,A,F13.5,A,e10.2)
      END
*
      SUBROUTINE REGION(N,X,J,A,B)
*     .. Scalar Arguments ..
      real              A, B
      INTEGER           J, N
*     .. Array Arguments ..
      real              X(N)
*     .. Executable Statements ..
      A = 0.0e0
      B = 1.0e0
      RETURN
      END
*
      real FUNCTION FUNCT(NDIM,X)
*     .. Scalar Arguments ..
      INTEGER             NDIM
*     .. Array Arguments ..
      real                X(NDIM)
*     .. Local Scalars ..
      real                SUM
      INTEGER             J
```

```
*     .. Intrinsic Functions ..
      INTRINSIC         COS, real
*     .. Executable Statements ..
      SUM = 0.0e0
      DO 20 J = 1, NDIM
         SUM = SUM + X(J)
   20 CONTINUE
      FUNCT = COS(0.5e0+2.0e0*SUM-real(NDIM))
      RETURN
      END
```

## 9.2 Program Data

None.

## 9.3 Program Results

```
D01GCF Example Program Results

Result =      0.43999  Standard error =  0.18E-05
```