

# NAG Fortran Library Routine Document

## D01PAF

**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

D01PAF returns a sequence of approximations to the integral of a function over a multi-dimensional simplex, together with an error estimate for the last approximation.

### 2 Specification

```

SUBROUTINE D01PAF (NDIM, VERTEX, IV1, IV2, FUNCTN, MINORD, MAXORD,
1                FINVLS, ESTERR, IFAIL)
  INTEGER          NDIM, IV1, IV2, MINORD, MAXORD, IFAIL
  real           VERTEX(IV1,IV2), FUNCTN, FINVLS(MAXORD), ESTERR
  EXTERNAL        FUNCTN

```

### 3 Description

The subroutine computes a sequence of approximations  $\text{FINVLS}(j)$ , for  $j = \text{MINORD} + 1, \text{MINORD} + 2, \dots, \text{MAXORD}$ , to an integral

$$\int_S f(x_1, x_2, \dots, x_n) dx_1 dx_2 \cdots dx_n$$

where  $S$  is an  $n$ -dimensional simplex defined in terms of its  $n + 1$  vertices.  $\text{FINVLS}(j)$  is an approximation which will be exact (except for rounding errors) whenever the integrand is a polynomial of total degree  $2j - 1$  or less.

The type of method used has been described by Grundmann and Moller (1978), and is implemented in an extrapolated form using the theory from de Doncker (1979).

### 4 References

Grundmann A and Moller H M (1978) Invariant integration formulas for the  $n$ -simplex by combinatorial methods *SIAM J. Numer. Anal.* **15** 282–290

de Doncker E (1979) New Euler–Maclaurin Expansions and their application to quadrature over the  $s$ -dimensional simplex *Math. Comput.* **33** 1003–1018

### 5 Parameters

1: NDIM – INTEGER *Input*

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

*Constraint:*  $\text{NDIM} \geq 2$ .

2: VERTEX(IV1,IV2) – **real** array *Input/Output*

*On entry:*  $\text{VERTEX}(i, j)$  must be set to the  $j$ th component of the  $i$ th vertex for the simplex integration region, for  $i = 1, 2, \dots, n + 1$ ;  $j = 1, 2, \dots, n$ . If  $\text{MINORD} > 0$ , VERTEX must be unchanged since the previous call of D01PAF.

*On exit:* these values are unchanged. The rest of the array VERTEX is used for workspace and contains information to be used if another call of D01PAF is made with  $\text{MINORD} > 0$ . In particular  $\text{VERTEX}(n + 1, 2n + 2)$  contains the volume of the simplex.

- 3: IV1 – INTEGER *Input*  
*On entry:* the first dimension of the array VERTEX as declared in the (sub)program from which D01PAF is called.  
*Constraint:*  $IV1 \geq NDIM + 1$ .
- 4: IV2 – INTEGER *Input*  
*On entry:* the second dimension of the array VERTEX as declared in the (sub)program from which D01PAF is called.  
*Constraint:*  $IV2 \geq 2 \times (NDIM + 1)$ .
- 5: 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:

<pre> <b>real</b> FUNCTION FUNCTN(NDIM, X)   INTEGER          NDIM   <b>real</b>           X(NDIM) </pre>	
<p>1: NDIM – INTEGER <span style="float: right;"><i>Input</i></span>  <i>On entry:</i> the number of dimensions of the integral, <math>n</math>.</p>	
<p>2: X(NDIM) – <i>real</i> array <span style="float: right;"><i>Input</i></span>  <i>On entry:</i> the co-ordinates of the point at which the integrand must be evaluated.</p>	

FUNCTN must be declared as EXTERNAL in the (sub)program from which D01PAF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

- 6: MINORD – INTEGER *Input/Output*  
*On entry:* MINORD must specify the highest order of the approximations currently available in the array FINVLS. MINORD = 0 indicates an initial call; MINORD > 0 indicates that FINVLS(1), FINVLS(2), ..., FINVLS(MINORD) have already been computed in a previous call of D01PAF.  
*Constraint:* MINORD  $\geq 0$ .  
*On exit:* MINORD = MAXORD.
- 7: MAXORD – INTEGER *Input*  
*On entry:* the highest order of approximation to the integral to be computed.  
*Constraint:* MAXORD > MINORD.
- 8: FINVLS(MAXORD) – *real* array *Input/Output*  
*On entry:* FINVLS(1), FINVLS(2), ..., FINVLS(MINORD) must contain approximations to the integral previously computed by D01PAF.  
*On exit:* FINVLS contains these values unchanged, and the newly computed values FINVLS(MINORD + 1), FINVLS(MINORD + 2), ..., FINVLS(MAXORD). FINVLS( $j$ ) is an approximation to the integral of polynomial degree  $2j - 1$ .
- 9: ESTERR – *real* *Output*  
*On exit:* an absolute error estimate for FINVLS(MAXORD).

10: 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 < 2,  
or IV1M < NDIM + 1,  
or IV2 < 2 × (NDIM + 1),  
or MINORD < 0,  
or MAXORD ≤ MINORD.

IFAIL = 2

The volume of the simplex integration region (computed as a determinant by F03AAF) is too large or too small to be representable in the machine.

## 7 Accuracy

An absolute error estimate is output through the parameter ESTERR.

## 8 Further Comments

The running time for D01PAF will usually be dominated by the time used to evaluate the integrand FUNCTN. The maximum time that could be used by D01PAF will be approximately given by

$$T \times \frac{(\text{MAXORD} + \text{NDIM})!}{(\text{MAXORD} - 1)! (\text{NDIM} + 1)!}$$

where  $T$  is the time needed for one call of FUNCTN.

## 9 Example

A program demonstrating the use of the subroutine with the integral

$$\int_0^1 \int_0^{1-x} \int_0^{1-x-y} \exp(x + y + z) \cos(x + y + z) dz dy dx = \frac{1}{4}$$

is given below.

## 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.

```

*      D01PAF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
INTEGER          NDIM, IV1, IV2, MXORD
PARAMETER       (NDIM=3,IV1=NDIM+1,IV2=2*(NDIM+1),MXORD=5)
INTEGER          NOUT
PARAMETER       (NOUT=6)
*      .. Local Scalars ..
real           ESTERR
INTEGER          IFAIL, J, K, MAXORD, MINORD, NEVALS
*      .. Local Arrays ..
real           FINVLS(MXORD), VERTEX(IV1,IV2)
*      .. External Functions ..
real           FUNCTN
EXTERNAL        FUNCTN
*      .. External Subroutines ..
EXTERNAL        D01PAF
*      .. Executable Statements ..
WRITE (NOUT,*) 'D01PAF Example Program Results'
DO 40 J = 1, IV1
  DO 20 K = 1, NDIM
    VERTEX(J,K) = 0.0e0
20  CONTINUE
    IF (J.GT.1) VERTEX(J,J-1) = 1.0e0
40  CONTINUE
    MINORD = 0
    NEVALS = 1
    WRITE (NOUT,*)
    WRITE (NOUT,*)
+   'MAXORD  Estimated          Estimated          Integrand'
    WRITE (NOUT,*)
+   '          value          accuracy          evaluations'
    DO 60 MAXORD = 1, MXORD
      IFAIL = 0
*
      CALL D01PAF(NDIM,VERTEX,IV1,IV2,FUNCTN,MINORD,MAXORD,FINVLS,
+             ESTERR,IFAIL)
*
      WRITE (NOUT,99999) MAXORD, FINVLS(MAXORD), ESTERR, NEVALS
      NEVALS = (NEVALS*(MAXORD+NDIM+1))/MAXORD
60  CONTINUE
    STOP
*
99999 FORMAT (1X,I4,F13.5,e16.3,I15)
END
*
real FUNCTION FUNCTN(NDIM,X)
*      .. Scalar Arguments ..
INTEGER          NDIM
*      .. Array Arguments ..
real           X(NDIM)
*      .. Intrinsic Functions ..
INTRINSIC        COS, EXP
*      .. Executable Statements ..
FUNCTN = EXP(X(1)+X(2)+X(3))*COS(X(1)+X(2)+X(3))
RETURN
END

```

## 9.2 Program Data

None.

### 9.3 Program Results

D01PAF Example Program Results

MAXORD	Estimated value	Estimated accuracy	Integrand evaluations
1	0.25816	0.258E+00	1
2	0.25011	0.806E-02	5
3	0.25000	0.107E-03	15
4	0.25000	0.410E-06	35
5	0.25000	0.173E-08	70

---