NAG 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 multidimensional simplex, together with an error estimate for the last approximation.

2 Specification

```
SUBROUTINE D01PAF (NDIM, VERT, LDVERT, SDVERT, FUNCTN, MINORD, MAXORD,     &
                   FINVLS, ESTERR, IFAIL)

INTEGER             NDIM, LDVERT, SDVERT, MINORD, MAXORD, IFAIL
REAL (KIND=nag_wp) VERT(LDVERT,SDVERT), FUNCTN, FINVLS(MAXORD), ESTERR
EXTERNAL            FUNCTN
```

3 Description

D01PAF computes a sequence of approximations FINVLS(j), for $j = \text{MINORD}+1, \ldots, \text{MAXORD}$, to an integral

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

where S is an n-dimensional simplex defined in terms of its $n+1$ vertices. 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 in Grundmann and Moller (1978), and is implemented in an extrapolated form using the theory from de Doncker (1979).

4 References

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

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

5 Parameters

1: NDIM – INTEGER *Input*

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

Constraint: NDIM ≥ 2.

2: VERT(LDVERT, SDVERT) – REAL (KIND=nag_wp) array *Input/Output*

On entry: VERT(i,j) must be set to the jth component of the ith vertex for the simplex integration region, for $i = 1, 2, \ldots, n+1$ and $j = 1, 2, \ldots, n$. If MINORD > 0, VERT must be unchanged since the previous call of D01PAF.

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

3: LDVERT – INTEGER *Input*

On entry: the first dimension of the array VERT as declared in the (sub)program from which D01PAF is called.

Constraint: LDVERT $\geq$ NDIM $+ 1$.

4: SDVERT – INTEGER *Input*

On entry: the second dimension of the array VERT as declared in the (sub)program from which D01PAF is called.

Constraint: SDVERT $\geq 2 \times$ (NDIM $+ 1$).

5: 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 D01PAF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

6: MINORD – INTEGER *Input/Output*

On entry: 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 ≥ 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 (KIND=nag_wp) array *Input/Output*

On entry: if MINORD > 0, FINVLS(1), FINVLS(2), ..., FINVLS(MINORD) must contain approximations to the integral previously computed by D01PAF.

On exit: 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 (KIND=nag_wp) *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. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 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.**

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, LDVERT $= \langle value \rangle$ and NDIM $= \langle value \rangle$.
Constraint: LDVERT $\geq$ NDIM $+ 1$.

On entry, MAXORD $= \langle value \rangle$ and MINORD $= \langle value \rangle$.
Constraint: MAXORD $>$ MINORD.

On entry, MINORD $= \langle value \rangle$.
Constraint: MINORD ≥ 0.

On entry, NDIM $= \langle value \rangle$.
Constraint: NDIM ≥ 2.

On entry, SDVERT $= \langle value \rangle$ and NDIM $= \langle value \rangle$.
Constraint: SDVERT $\geq 2 \times$ (NDIM $+ 1$).

IFAIL $= 2$

The volume of the simplex integration region is too large or too small to be represented on the machine.

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

An absolute error estimate is output through the parameter ESTERR.

8 Parallelism and Performance

D01PAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

D01PAF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

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 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.

10 Example

This example demonstrates the use of the subroutine with the integral

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

10.1 Program Text

```
!   D01PAF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.

    Module d01pafe_mod

!     D01PAF 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            :: mxord = 5, ndim = 3, nout = 6
      Integer, Parameter, Public            :: sdvert = 2*(ndim+1)
      Integer, Parameter, Public            :: ldvert = ndim + 1
    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                             :: cos, exp
!       .. Executable Statements ..
        functn = exp(x(1)+x(2)+x(3))*cos(x(1)+x(2)+x(3))

        Return
```

```
      End Function functn
    End Module d01pafe_mod
    Program d01pafe

!     D01PAF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: d01paf, nag_wp
      Use d01pafe_mod, Only: functn, ldvert, mxord, ndim, nout, sdvert
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                  :: esterr
      Integer                             :: ifail, j, maxord, minord, nevals
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable     :: finvls(:), vert(:,:)
!     .. Executable Statements ..
      Write (nout,*) 'D01PAF Example Program Results'

      Allocate (finvls(mxord),vert(ldvert,sdvert))

      vert(1:ldvert,1:ndim) = 0.0_nag_wp
      Do j = 2, ldvert
        vert(j,j-1) = 1.0_nag_wp
      End Do

      minord = 0
      nevals = 1

      Do maxord = 1, mxord

        ifail = 0
        Call d01paf(ndim,vert,ldvert,sdvert,functn,minord,maxord,finvls, &
          esterr,ifail)

        If (maxord==1) Write (nout,99999)
        Write (nout,99998) maxord, finvls(maxord), esterr, nevals

        nevals = (nevals*(maxord+ndim+1))/maxord
      End Do

99999 Format (/1X,'MAXORD   Estimated      Estimated         Integrand'/1X, &
        '           value         accuracy        evaluations')
99998 Format (1X,I4,F13.5,E16.3,I15)
    End Program d01pafe
```

10.2 Program Data

None.

10.3 Program Results

```
 D01PAF Example Program Results

 MAXORD   Estimated      Estimated         Integrand
            value         accuracy        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
```