NAG Toolbox

nag_quad_md_simplex (d01pa)

1 Purpose

nag_quad_md_simplex (d01pa) 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 Syntax

```
[vert, minord, finvls, esterr, ifail] = nag_quad_md_simplex(ndim, vert, functn, minord, finvls, 'sdvert', sdvert, 'maxord', maxord)
[vert, minord, finvls, esterr, ifail] = d01pa(ndim, vert, functn, minord, finvls, 'sdvert', sdvert, 'maxord', maxord)
```

3 Description

nag_quad_md_simplex (d01pa) computes a sequence of approximations **finvls**(j), for $j = \mathbf{minord} + 1, \ldots, \mathbf{maxord}$, 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

5.1 Compulsory Input Parameters

1: **ndim** – INTEGER

n, the number of dimensions of the integral.

Constraint: **ndim** ≥ 2.

2: **vert**$(ldvert, \mathbf{sdvert})$ – REAL (KIND=nag_wp) array

$ldvert$, the first dimension of the array, must satisfy the constraint $ldvert \geq \mathbf{ndim} + 1$.

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 nag_quad_md_simplex (d01pa).

3: **functn** – REAL (KIND=nag_wp) FUNCTION, supplied by the user.

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

```
[result] = functn(ndim, x)
```

Input Parameters

1: **ndim** – INTEGER

n, the number of dimensions of the integral.

2: **x**(**ndim**) – REAL (KIND=nag_wp) array

The coordinates of the point at which the integrand f must be evaluated.

Output Parameters

1: **result**

The value of the integrand f at the given point.

4: **minord** – INTEGER

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), $\ldots$, **finvls**(**minord**) have already been computed in a previous call of nag_quad_md_simplex (d01pa).

Constraint: **minord** ≥ 0.

5: **finvls**(**maxord**) – REAL (KIND=nag_wp) array

If **minord** > 0, **finvls**(1), **finvls**(2), $\ldots$, **finvls**(**minord**) must contain approximations to the integral previously computed by nag_quad_md_simplex (d01pa).

5.2 Optional Input Parameters

1: **sdvert** – INTEGER

Default: the second dimension of the array **vert**.

The second dimension of the array **vert**.

Constraint: **sdvert** $\geq 2 \times$ (**ndim** $+ 1$).

2: **maxord** – INTEGER

Default: the dimension of the array **finvls**.

The highest order of approximation to the integral to be computed.

Constraint: **maxord** $>$ **minord**.

5.3 Output Parameters

1: **vert**($ldvert$, **sdvert**) – REAL (KIND=nag_wp) array

These values are unchanged. The rest of the array **vert** is used for workspace and contains information to be used if another call of nag_quad_md_simplex (d01pa) is made with **minord** > 0. In particular **vert**$(n + 1, 2n + 2)$ contains the volume of the simplex.

2: **minord** – INTEGER

minord $=$ **maxord**.

3: **finvls**(**maxord**) – REAL (KIND=nag_wp) array

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 $2j - 1$.

4: **esterr** – REAL (KIND=nag_wp)

An absolute error estimate for **finvls**(**maxord**).

5: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

Constraint: $ldvert \geq$ **ndim** + 1.

Constraint: **maxord** > **minord**.

Constraint: **minord** ≥ 0.

Constraint: **ndim** ≥ 2.

Constraint: **sdvert** ≥ 2 × (**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.

ifail = −399

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

ifail = −999

Dynamic memory allocation failed.

7 Accuracy

An absolute error estimate is output through the argument **esterr**.

8 Further Comments

The running time for nag_quad_md_simplex (d01pa) will usually be dominated by the time used to evaluate the integrand **functn**. The maximum time that could be used by nag_quad_md_simplex (d01pa) will be approximately given by

$$T \times \frac{(\mathbf{maxord} + \mathbf{ndim})!}{(\mathbf{maxord} - 1)!(\mathbf{ndim} + 1)!}$$

where T is the time needed for one call of **functn**.

9 Example

This example demonstrates the use of the function with the integral

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

9.1 Program Text

```
     function d01pa_example

fprintf('d01pa example results\n\n');

ndim = nag_int(3);
vertex = zeros(ndim+1,2*(ndim+1));
vertex(2:ndim+1,1:ndim) = eye(3);
minord = nag_int(0);
finvls = zeros(5,1);

fprintf('Maxord   Estimated      Estimated         Integrand\n');
fprintf('           value         accuracy        evaluations\n');
nevals = 1;
for maxord = nag_int(1:5)
  [vertex, minord, finvls, esterr, ifail] = ...
  d01pa(...
        ndim, vertex, @functn, minord, finvls,'maxord',maxord);

  fprintf('%5d%13.5f%16.3e%15d\n',maxord, finvls(maxord),esterr,nevals);
  nevals = (nevals*(maxord+ndim+1))/maxord;
end

function result = functn(ndim, x)
  u = sum(x);
  result = exp(u)*cos(u);
```

9.2 Program Results

```
     d01pa example results

Maxord   Estimated      Estimated         Integrand
           value         accuracy        evaluations
    1      0.25816       2.582e-01             1
    2      0.25011       8.058e-03             5
    3      0.25000       1.067e-04            15
    4      0.25000       4.098e-07            35
    5      0.25000       1.731e-09            70
```