NAG Toolbox

nag_quad_md_gauss (d01fb)

1 Purpose

nag_quad_md_gauss (d01fb) computes an estimate of a multidimensional integral (from 1 to 20 dimensions), given the analytic form of the integrand and suitable Gaussian weights and abscissae.

2 Syntax

```
[result, ifail] = nag_quad_md_gauss(nptvec, weight, abscis, fun, 'ndim', ndim,
'lwa', lwa)
```

```
[result, ifail] = d01fb(nptvec, weight, abscis, fun, 'ndim', ndim, 'lwa', lwa)
```

3 Description

nag_quad_md_gauss (d01fb) approximates a multidimensional integral by evaluating the summation

$$\sum_{i_1=1}^{l_1} w_{1,i_1} \sum_{i_2=1}^{l_2} w_{2,i_2} \cdots \sum_{i_n=1}^{l_n} w_{n,i_n} f\left(x_{1,i_1}, x_{2,i_2}, \ldots, x_{n,i_n}\right)$$

given the weights w_{j,i_j} and abscissae x_{j,i_j} for a multidimensional product integration rule (see Davis and Rabinowitz (1975)). The number of dimensions may be anything from 1 to 20.

The weights and abscissae for each dimension must have been placed in successive segments of the arrays **weight** and **abscis**; for example, by calling nag_quad_1d_gauss_wgen (d01bc) or nag_quad_1d_gauss_wres (d01tb) once for each dimension using a quadrature formula and number of abscissae appropriate to the range of each x_j and to the functional dependence of f on x_j.

If normal weights are used, the summation will approximate the integral

$$\int w_1(x_1) \int w_2(x_2) \cdots \int w_n(x_n) f(x_1, x_2, \ldots, x_n) dx_n \cdots dx_2 dx_1$$

where $w_j(x)$ is the weight function associated with the quadrature formula chosen for the jth dimension; while if adjusted weights are used, the summation will approximate the integral

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

You must supply a function to evaluate

$$f(x_1, x_2, \ldots, x_n)$$

at any values of $x_1, x_2, \ldots, x_n$ within the range of integration.

4 References

Davis P J and Rabinowitz P (1975) *Methods of Numerical Integration* Academic Press

5 Parameters

5.1 Compulsory Input Parameters

1: **nptvec**(**ndim**) – INTEGER array

nptvec(j) must specify the number of points in the jth dimension of the summation, for $j = 1, 2, \ldots, n$.

2: **weight**(**lwa**) – REAL (KIND=nag_wp) array

Must contain in succession the weights for the various dimensions, i.e., **weight**(k) contains the ith weight for the jth dimension, with

$$k = \mathbf{nptvec}(1) + \mathbf{nptvec}(2) + \cdots + \mathbf{nptvec}(j-1) + i.$$

3: **abscis**(**lwa**) – REAL (KIND=nag_wp) array

Must contain in succession the abscissae for the various dimensions, i.e., **abscis**(k) contains the ith abscissa for the jth dimension, with

$$k = \mathbf{nptvec}(1) + \mathbf{nptvec}(2) + \cdots + \mathbf{nptvec}(j-1) + i.$$

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

fun must return the value of the integrand f at a specified point.

```
[result] = fun(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 $f(x)$ evaluated at **x**.

5.2 Optional Input Parameters

1: **ndim** – INTEGER

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

n, the number of dimensions of the integral.

Constraint: $1 \leq \mathbf{ndim} \leq 20$.

2: **lwa** – INTEGER

Default: the dimension of the arrays **weight**, **abscis**. (An error is raised if these dimensions are not equal.)

The dimension of the arrays **weight** and **abscis**.

Constraint: $\mathbf{lwa} \geq \mathbf{nptvec}(1) + \mathbf{nptvec}(2) + \cdots + \mathbf{nptvec}(\mathbf{ndim})$.

5.3 Output Parameters

1: **result**

The result of the function.

2: **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 $= 1$

On entry, **ndim** < 1,
or **ndim** > 20,
or **lwa** $<$ **nptvec**$(1) +$ **nptvec**$(2) + \cdots +$ **nptvec**(**ndim**).

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

The accuracy of the computed multidimensional sum depends on the weights and the integrand values at the abscissae. If these numbers vary significantly in size and sign then considerable accuracy could be lost. If these numbers are all positive, then little accuracy will be lost in computing the sum.

8 Further Comments

The total time taken by nag_quad_md_gauss (d01fb) will be proportional to

$$T \times \textbf{nptvec}(1) \times \textbf{nptvec}(2) \times \cdots \times \textbf{nptvec}(\textbf{ndim}),$$

where T is the time taken for one evaluation of **fun**.

9 Example

This example evaluates the integral

$$\int_1^2 \int_0^\infty \int_{-\infty}^\infty \int_1^\infty \frac{(x_1 x_2 x_3)^6}{(x_4+2)^8} e^{-2x_2} e^{-0.5x_3^2} dx_4 dx_3 dx_2 dx_1$$

using adjusted weights. The quadrature formulae chosen are:

x_1: Gauss–Legendre, $a = 1.0$, $b = 2.0$,

x_2: Gauss–Laguerre, $a = 0.0$, $b = 2.0$,

x_3: Gauss–Hermite, $a = 0.0$, $b = 0.5$,

x_4: rational Gauss, $a = 1.0$, $b = 2.0$.

Four points are sufficient in each dimension, as this integral is in fact a product of four one-dimensional integrals, for each of which the chosen four-point formula is exact.

9.1 Program Text

```
function d01fb_example

fprintf('d01fb example results\n\n');

nptvec = [nag_int(4), 4, 4, 4];
a      = [1, 0, 0  , 1];
b      = [2, 2, 0.5, 2];
```

```
key    = [nag_int(0), -3, -4, -5];
j      = 1;
for i = 1:4
  [wgt, absc, ifail] = d01tb(key(i), a(i), b(i), nptvec(i));
  weight(j:j+nptvec(i)-1) = wgt;
  abscis(j:j+nptvec(i)-1) = absc;
  j = j + nptvec(i);
end

[result, ifail] = d01fb( ...
  nptvec, weight, abscis, @fun);

fprintf('Result = %13.5f\n', result);

function result = fun(ndim,x)
  result = ((x(1)*x(2)*x(3))^6/(x(4)+2)^8)*exp(-2*x(2)-x(3)^2/2);
```

9.2 Program Results

```
     d01fb example results

Result =       0.25065
```