NAG Toolbox for MATLAB

d01fc

1 Purpose

d01fc attempts to evaluate a multi-dimensional integral (up to 15 dimensions), with constant and finite limits, to a specified relative accuracy, using an adaptive subdivision strategy.

2 Syntax

```
[minpts, acc, finval, ifail] = d01fc(a, b, minpts, maxpts, functn, eps,
'ndim', ndim)
```

3 Description

d01fc returns an estimate of a multi-dimensional integral over a hyper-rectangle (i.e., with constant limits), and also an estimate of the relative error. You set the relative accuracy required, return values for the integrand via a function argument user-supplied real function **functn**, and also set the minimum and maximum acceptable number of calls to **functn** (in **minpts** and **maxpts**).

The function operates by repeated subdivision of the hyper-rectangular region into smaller hyper-rectangles. In each subregion, the integral is estimated using a seventh-degree rule, and an error estimate is obtained by comparison with a fifth-degree rule which uses a subset of the same points. The fourth differences of the integrand along each co-ordinate axis are evaluated, and the subregion is marked for possible future subdivision in half along that co-ordinate axis which has the largest absolute fourth difference.

If the estimated errors, totalled over the subregions, exceed the requested relative error (or if fewer than **minpts** calls to user-supplied real function **functn** have been made), further subdivision is necessary, and is performed on the subregion with the largest estimated error, that subregion being halved along the appropriate co-ordinate axis.

The function will fail if the requested relative error level has not been attained by the time **maxpts** calls to user-supplied real function **functn** have been made; or, if the amount **lenwrk** of working storage is insufficient. A formula for the recommended value of **lenwrk** is given in Section 5. If a smaller value is used, and is exhausted in the course of execution, the function switches to a less efficient mode of operation; only if this mode also breaks down is insufficient storage reported.

d01fc is based on the HALF (sub)program developed by van Dooren and De Ridder 1976. It uses a different basic rule, described in Genz and Malik 1980.

4 References

Genz A C and Malik A A 1980 An Adaptive Algorithm for Numerical Integration over an N-dimensional Rectangular Region *J. Comput. Appl. Math.* **6** 295–302

van Dooren P and De Ridder L 1976 An adaptive algorithm for numerical integration over an N-dimensional cube *J. Comput. Appl. Math.* **2** 207–217

5 Parameters

5.1 Compulsory Input Parameters

1: **a(ndim) – double array**

The lower limits of integration, a_i, for $i = 1, 2, \ldots, n$.

2: **b(ndim) – double array**

The upper limits of integration, b_i, for $i = 1, 2, \ldots, n$.

3: **minpts – int32 scalar**

Must be set to the minimum number of integrand evaluations to be allowed.

4: **maxpts – int32 scalar**

The maximum number of integrand evaluations to be allowed.

Constraints:

maxpts $\geq$ **minpts**;
maxpts $\geq \alpha$, where $\alpha = 2^{\mathbf{ndim}} + 2 \times \mathbf{ndim}^2 + 2 \times \mathbf{ndim} + 1$.

5: **functn – string containing name of m-file**

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

Its specification is:

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

Input Parameters

1: **ndim – int32 scalar**

n, the number of dimensions of the integral.

2: **z(ndim) – double array**

The co-ordinates of the point at which the integrand f must be evaluated.

Output Parameters

1: **result – double scalar**

The result of the function.

6: **eps – double scalar**

The relative error acceptable to you. When the solution is zero or very small relative accuracy may not be achievable but you may still set **eps** to a reasonable value and check for the error exit **ifail** $= 2$.

Constraint: **eps** > 0.0.

5.2 Optional Input Parameters

1: **ndim – int32 scalar**

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

n, the number of dimensions of the integral.

Constraint: $2 \leq$ **ndim** ≤ 15.

5.3 Input Parameters Omitted from the MATLAB Interface

lenwrk, wrkstr

5.4 Output Parameters

1: **minpts – int32 scalar**

Contains the actual number of integrand evaluations used by d01fc.

2: **acc – double scalar**

The estimated relative error in **finval**.

3: **finval – double scalar**

The best estimate obtained for the integral.

4: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Note: d01fc may return useful information for one or more of the following detected errors or warnings.

ifail $= 1$

On entry, **ndim** < 2,
or **ndim** > 15,
or **maxpts** is too small,
or **lenwrk** $< 2 \times$ **ndim** $+ 4$,
or **eps** ≤ 0.0.

ifail $= 2$

maxpts was too small to obtain the required relative accuracy **eps**. On soft failure, **finval** and **acc** contain estimates of the integral and the relative error, but **acc** will be greater than **eps**.

ifail $= 3$

lenwrk was too small. On soft failure, **finval** and **acc** contain estimates of the integral and the relative error, but **acc** will be greater than **eps**.

7 Accuracy

A relative error estimate is output through the parameter **acc**.

8 Further Comments

Execution time will usually be dominated by the time taken to evaluate the user-supplied real function **functn**, and hence the maximum time that could be taken will be proportional to **maxpts**.

9 Example

```
d01fc_functn.m

function result = functn(ndim,z)
  result = 4.0*z(1)*z(3)*z(3)*exp(2.0*z(1)*z(3))/(1.0+z(2)+z(4))^2;
```

```
a = [0;
     0;
     0;
```

```
     0];
b = [1;
     1;
     1;
     1];
minpts = int32(0);
maxpts = int32(8000);
eps = 0.0001;
[minptsOut, acc, finval, ifail] = d01fc(a, b, minpts, maxpts,
'd01fc_functn', eps)
```

```
minptsOut =
        2223
acc =
   9.8932e-05
finval =
    0.5754
ifail =
           0
```