NAG Toolbox for MATLAB

d01am

1 Purpose

d01am calculates an approximation to the integral of a function $f(x)$ over an infinite or semi-infinite interval $[a,b]$:

$$I = \int_a^b f(x)\,dx.$$

2 Syntax

```
[result, abserr, w, iw, ifail] = d01am(f, bound, inf, epsabs, epsrel,
'lw', lw, 'liw', liw)
```

3 Description

d01am is based on the QUADPACK routine QAGI (see Piessens *et al.* 1983). The entire infinite integration range is first transformed to $[0,1]$ using one of the identities:

$$\int_{-\infty}^a f(x)\,dx = \int_0^1 f\left(a - \frac{1-t}{t}\right)\frac{1}{t^2}\,dt$$

$$\int_a^{\infty} f(x)\,dx = \int_0^1 f\left(a + \frac{1-t}{t}\right)\frac{1}{t^2}\,dt$$

$$\int_{-\infty}^{\infty} f(x)\,dx = \int_0^{\infty}(f(x) + f(-x))\,dx = \int_0^1 \left[f\left(\frac{1-t}{t}\right) + f\left(\frac{-1+t}{t}\right)\right]\frac{1}{t^2}\,dt$$

where a represents a finite integration limit. An adaptive procedure, based on the Gauss 7-point and Kronrod 15-point rules, is then employed on the transformed integral. The algorithm, described in de Doncker 1978, incorporates a global acceptance criterion (as defined by Malcolm and Simpson 1976) together with the ϵ-algorithm (see Wynn 1956) to perform extrapolation. The local error estimation is described in Piessens *et al.* 1983.

4 References

de Doncker E 1978 An adaptive extrapolation algorithm for automatic integration *ACM SIGNUM Newsl.* **13 (2)** 12–18

Malcolm M A and Simpson R B 1976 Local versus global strategies for adaptive quadrature *ACM Trans. Math. Software* **1** 129–146

Piessens R, de Doncker–Kapenga E, Überhuber C and Kahaner D 1983 *QUADPACK, A Subroutine Package for Automatic Integration* Springer–Verlag

Wynn P 1956 On a device for computing the $e_m(S_n)$ transformation *Math. Tables Aids Comput.* **10** 91–96

5 Parameters

5.1 Compulsory Input Parameters

1: **f – string containing name of m-file**

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

Its specification is:

```
[result] = f(x)
```

Input Parameters

1: **x – double scalar**

The point at which the integrand f must be evaluated.

Output Parameters

1: **result – double scalar**

The result of the function.

2: **bound – double scalar**

The finite limit of the integration range (if present). **bound** is not used if the interval is doubly infinite.

3: **inf – int32 scalar**

Indicates the kind of integration range.

inf $= 1$

The range is $[\textbf{bound}, +\infty)$.

inf $= -1$

The range is $(-\infty, \textbf{bound}]$.

inf $= 2$

The range is $(-\infty, +\infty)$.

Constraint: **inf** $= -1$, 1 or 2.

4: **epsabs – double scalar**

The absolute accuracy required. If **epsabs** is negative, the absolute value is used. See Section 7.

5: **epsrel – double scalar**

The relative accuracy required. If **epsrel** is negative, the absolute value is used. See Section 7.

5.2 Optional Input Parameters

1: **lw – int32 scalar**

Default: The dimension of the array **w**.

The value of **lw** (together with that of **liw**) imposes a bound on the number of sub-intervals into which the interval of integration may be divided by the function. The number of sub-intervals cannot exceed **lw**/4. The more difficult the integrand, the larger **lw** should be.

Suggested value: **lw** $= 800$ to 2000 is adequate for most problems.

Default: 800

Constraint: **lw** ≥ 4.

2: **liw – int32 scalar**

Default: The dimension of the array **iw**.

The number of sub-intervals into which the interval of integration may be divided cannot exceed **liw**.

Suggested value: **liw** $= $ **lw**$/4$.

Default: **lw**$/4$

Constraint: **liw** ≥ 1.

5.3 Input Parameters Omitted from the MATLAB Interface

None.

5.4 Output Parameters

1: **result – double scalar**

The approximation to the integral I.

2: **abserr – double scalar**

An estimate of the modulus of the absolute error, which should be an upper bound for $|I -$ **result**$|$.

3: **w(lw) – double array**

Details of the computation, as described in Section 8.

4: **iw(liw) – int32 array**

iw(1) contains the actual number of sub-intervals used. The rest of the array is used as workspace.

5: **ifail – int32 scalar**

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

6 Error Indicators and Warnings

Note: d01am may return useful information for one or more of the following detected errors or warnings.

ifail $= 1$

The maximum number of subdivisions allowed with the given workspace has been reached without the accuracy requirements being achieved. Look at the integrand in order to determine the integration difficulties. If the position of a local difficulty within the interval can be determined (e.g., a singularity of the integrand or its derivative, a peak, a discontinuity, etc.) you will probably gain from splitting up the interval at this point and calling d01am on the infinite subrange and an appropriate integrator on the finite subrange. Alternatively, consider relaxing the accuracy requirements specified by **epsabs** and **epsrel**, or increasing the amount of workspace.

ifail $= 2$

Round-off error prevents the requested tolerance from being achieved. Consider requesting less accuracy.

ifail $= 3$

Extremely bad local integrand behaviour causes a very strong subdivision around one (or more) points of the interval. The same advice applies as in the case of **ifail** $= 1$.

ifail $= 4$

The requested tolerance cannot be achieved because the extrapolation does not increase the accuracy satisfactorily; the returned result is the best which can be obtained. The same advice applies as in the case of **ifail** $= 1$.

ifail $= 5$

The integral is probably divergent, or slowly convergent. Please note that divergence can occur with any nonzero value of **ifail**.

ifail $= 6$

On entry, **lw** < 4,
or **liw** < 1,
or **inf** $\neq -1$, 1 or 2.

7 Accuracy

d01am cannot guarantee, but in practice usually achieves, the following accuracy:

$$|I - \textbf{result}| \leq tol,$$

where

$$tol = \max\{|\textbf{epsabs}|, |\textbf{epsrel}| \times |I|\},$$

and **epsabs** and **epsrel** are user-specified absolute and relative error tolerances. Moreover, it returns the quantity **abserr** which, in normal circumstances, satisfies

$$|I - \textbf{result}| \leq \textbf{abserr} \leq tol.$$

8 Further Comments

The time taken by d01am depends on the integrand and the accuracy required.

If **ifail** $\neq 0$ on exit, then you may wish to examine the contents of the array **w**, which contains the end points of the sub-intervals used by d01am along with the integral contributions and error estimates over these sub-intervals.

Specifically, for $i = 1, 2, \ldots, n$, let r_i denote the approximation to the value of the integral over the sub-interval $[a_i, b_i]$ in the partition of $[a, b]$ and e_i be the corresponding absolute error estimate. Then, $\int_{a_i}^{b_i} f(x)\,dx \simeq r_i$ and **result** $= \sum_{i=1}^{n} r_i$, unless d01am terminates while testing for divergence of the integral (see Section 3.4.3 of Piessens *et al.* 1983). In this case, **result** (and **abserr**) are taken to be the values returned from the extrapolation process. The value of n is returned in **iw**(1), and the values a_i, b_i, e_i and r_i are stored consecutively in the array **w**, that is:

$a_i = \mathbf{w}(i)$,

$b_i = \mathbf{w}(n + i)$,

$e_i = \mathbf{w}(2n + i)$ and

$r_i = \mathbf{w}(3n + i)$.

Note: this information applies to the integral transformed to $(0, 1)$ as described in Section 3, not to the original integral.

9 Example

```
d01am_f.m

function [result] = d01amf_f(x)
  result=1/(x+1)/sqrt(x);
```

```
bound = 0;
inf = int32(1);
```

```
epsabs = 0;
epsrel = 0.0001;
[result, abserr, w, iw, ifail] = d01am('d01am_f', bound, inf, epsabs,
epsrel)
```

```
result =
    3.1416
abserr =
   2.6515e-05
w =
     array elided
iw =
     array elided
ifail =
           0
```