NAG Toolbox

nag_quad_1d_indef (d01ar)

1 Purpose

nag_quad_1d_indef (d01ar) computes definite and indefinite integrals over a finite range to a specified relative or absolute accuracy, using the method described in Patterson (1968).

2 Syntax

```
[acc, ans, n, alpha, ifail] = nag_quad_1d_indef(a, b, fun, relacc, absacc,
maxrul, iparm, alpha)
```

```
[acc, ans, n, alpha, ifail] = d01ar(a, b, fun, relacc, absacc, maxrul, iparm,
alpha)
```

3 Description

nag_quad_1d_indef (d01ar) evaluates definite and indefinite integrals of the form:

$$\int_a^b f(t)\,dt$$

using the method described in Patterson (1968).

3.1 Definite Integrals

In this case nag_quad_1d_indef (d01ar) must be called with **iparm** $= 0$. By linear transformation the integral is changed to

$$I = \int_{-1}^{+1} F(x)\,dx$$

where

$$F(x) = \frac{b-a}{2} f\left(\frac{b+a+(b-a)x}{2}\right)$$

and is then approximated by an n-point quadrature rule

$$I = \sum_{k=1}^{n} w_k F(x_k)$$

where w_k are the weights and x_k are the abscissae.

The function uses a family of nine interlacing rules based on the optimal extension of the three-point Gauss rule. These rules use 1, 3, 7, 15, 31, 63, 127, 255 and 511 points and have respective polynomial integrating degrees 1, 5, 11, 23, 47, 95, 191, 383 and 767. Each rule has the property that the next in sequence includes all the points of its predecessor and has the greatest possible increase in integrating degree.

The integration method is based on the successive application of these rules until the absolute value of the difference of two successive results differs by not more than **absacc**, or relatively by not more than **relacc**. The result of the last rule used is taken as the value of the integral (**ans**), and the absolute difference of the results of the last two rules used is taken as an estimate of the absolute error (**acc**). Due to their interlacing form no integrand evaluations are wasted in passing from one rule to the next.

3.2 Indefinite Integrals

Suppose the value of the integral

$$\int_c^d f(t)\,dt$$

is required for a number of sub-intervals $[c,d]$, all of which lie in an interval $[a,b]$.

In this case nag_quad_1d_indef (d01ar) should first be called with the argument **iparm** $= 1$ and the interval set to $[a,b]$. The function then calculates the integral over $[a,b]$ **and** the Legendre expansion of the integrand, using the same integrand values. If the function is subsequently called with **iparm** $= 2$ and the interval set to $[c,d]$, the integral over $[c,d]$ is calculated by analytical integration of the Legendre expansion, without further evaluations of the integrand.

For the interval $[-1,1]$ the expansion takes the form

$$F(x) = \sum_{i=0}^{\infty} \alpha_i P_i(x)$$

where $P_i(x)$ is the order i Legendre polynomial. Assuming that the integral over the full range $[-1,1]$ was evaluated to the required accuracy using an n-point rule, then the coefficients

$$\alpha_i = \tfrac{1}{2}(2i-1)\int_{-1}^{+1} P_i(x)F(x)\,dx, \quad i = 0,1,\ldots,m$$

are evaluated by that same rule, up to

$$m = (3n-1)/4.$$

The accuracy for indefinite integration should be of the same order as that obtained for the definite integral over the full range. The indefinite integrals will be exact when $F(x)$ is a polynomial of degree $\leq m$.

4 References

Patterson T N L (1968) The Optimum addition of points to quadrature formulae *Math. Comput.* **22** 847–856

5 Parameters

5.1 Compulsory Input Parameters

1: **a** – REAL (KIND=nag_wp)

a, the lower limit of integration.

2: **b** – REAL (KIND=nag_wp)

b, the upper limit of integration. It is not necessary that $a < b$.

3: **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(x)
```

Input Parameters

1: **x** – REAL (KIND=nag_wp)

The point in $[a,b]$ at which the integrand f must be evaluated.

Output Parameters

1: **result**

The value of $f(x)$ evaluated at **x**.

If **iparm** $= 2$, **fun** is not called.

4: **relacc** – REAL (KIND=nag_wp)

The relative accuracy required. If convergence according to absolute accuracy is required, **relacc** should be set to zero (but see also Section 7). If **relacc** < 0.0, its absolute value is used.

If **iparm** $= 2$, **relacc** is not used.

5: **absacc** – REAL (KIND=nag_wp)

The absolute accuracy required. If convergence according to relative accuracy is required, **absacc** should be set to zero (but see also Section 7). If **absacc** < 0.0, its absolute value is used.

If **iparm** $= 2$, **absacc** is not used.

6: **maxrul** – INTEGER

The maximum number of successive rules that may be used.

Constraint: $1 \leq$ **maxrul** ≤ 9. If **maxrul** is outside these limits, the value 9 is assumed.

If **iparm** $= 2$, **maxrul** is not used.

7: **iparm** – INTEGER

Indicates the task to be performed by the function.

iparm $= 0$
: Only the definite integral over $[a, b]$ is evaluated.

iparm $= 1$
: As well as the definite integral, the expansion of the integrand in Legendre polynomials over $[a, b]$ is calculated, using the same values of the integrand as used to compute the integral. The expansion coefficients, and some other quantities, are returned in **alpha** for later use in computing indefinite integrals.

iparm $= 2$
: $f(t)$ is integrated analytically over $[a, b]$ using the previously computed expansion, stored in **alpha**. No further evaluations of the integrand are required. The function must previously have been called with **iparm** $= 1$ and the interval $[a, b]$ must lie within that specified for the previous call. In this case only the arguments **a**, **b**, **iparm**, **ans**, **alpha** and **ifail** are used.

Constraint: **iparm** $= 0$, 1 or 2.

8: **alpha**(**390**) – REAL (KIND=nag_wp) array

If **iparm** $= 2$, **alpha** must contain the coefficients of the Legendre expansions of the integrand, as returned by a previous call of nag_quad_1d_indef (d01ar) with **iparm** $= 1$ and a range containing the present range.

If **iparm** $= 0$ or 1, **alpha** need not be set on entry.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: **acc** – REAL (KIND=nag_wp)

If **iparm** $= 0$ or 1, **acc** contains the absolute value of the difference between the last two successive estimates of the integral. This may be used as a measure of the accuracy actually achieved.

If **iparm** $= 2$, **acc** is not used.

2: **ans** – REAL (KIND=nag_wp)

The estimated value of the integral.

3: **n** – INTEGER

When **iparm** $= 0$ or 1, **n** contains the number of integrand evaluations used in the calculation of the integral.

If **iparm** $= 2$, **n** is not used.

4: **alpha**(**390**) – REAL (KIND=nag_wp) array

If **iparm** $= 1$, the first m elements of **alpha** hold the coefficients of the Legendre expansion of the integrand, and the value of m is stored in **alpha**(390). **alpha** must not be changed between a call with **iparm** $= 1$ and subsequent calls with **iparm** $= 2$.

If **iparm** $= 2$, the first m elements of **alpha** are unchanged on exit.

5: **ifail** – INTEGER

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

6 Error Indicators and Warnings

Note: nag_quad_1d_indef (d01ar) may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the function:

ifail $= 1$ (*warning*)

If **iparm** $= 0$ or 1, this indicates that all **maxrul** rules have been used and the integral has not converged to the accuracy requested. In this case **ans** contains the last approximation to the integral, and **acc** contains the difference between the last two approximations. To check this estimate of the integral, nag_quad_1d_indef (d01ar) could be called again to evaluate

$$\int_a^b f(t)\,dt \quad \text{as} \quad \int_a^c f(t)\,dt + \int_c^b f(t)\,dt \quad \text{for some } a < c < b.$$

If **iparm** $= 2$, this indicates failure of convergence during the run with **iparm** $= 1$ in which the Legendre expansion was created.

ifail $= 2$

On entry, **iparm** $\neq 0$, 1 or 2

ifail $= 3$

The function is called with **iparm** $= 2$ but a previous call with **iparm** $= 1$ has been omitted or was invoked with an integration interval of length zero.

ifail $= 4$

On entry, with **iparm** $= 2$, the interval for indefinite integration is not contained within the interval specified when nag_quad_1d_indef (d01ar) was previously called with **iparm** $= 1$.

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 relative or absolute accuracy required is specified by you in the variables **relacc** or **absacc**. nag_quad_1d_indef (d01ar) will terminate whenever either the relative accuracy specified by **relacc** or the absolute accuracy specified by **absacc** is reached. One or other of these criteria may be 'forced' by setting the argument for the other to zero. If both **relacc** and **absacc** are specified as zero, then the function uses the value $10.0 \times$ (***machine precision***) for **relacc**.

If on exit **ifail** $= 0$, then it is likely that the result is correct to one or other of these accuracies. If on exit **ifail** $= 1$, then it is likely that neither of the requested accuracies has been reached.

When you have no prior idea of the magnitude of the integral, it is possible that an unreasonable accuracy may be requested, e.g., a relative accuracy for an integral which turns out to be zero, or a small absolute accuracy for an integral which turns out to be very large. Even if failure is reported in such a case, the value of the integral may still be satisfactory. The device of setting the other 'unused' accuracy argument to a small positive value (e.g., 10^{-9} for an implementation of 11-digit precision) rather than zero, may prevent excessive calculation in such a situation.

To avoid spurious convergence, it is recommended that relative accuracies larger than about 10^{-3} be avoided.

8 Further Comments

The time taken by nag_quad_1d_indef (d01ar) depends on the complexity of the integrand and the accuracy required.

This function uses the Patterson method over the whole integration interval and should therefore be suitable for well behaved functions. However, for very irregular functions it would be more efficient to submit the differently behaved regions separately for integration.

9 Example

This example evaluates the following integrals

(i) Definite integral only (**iparm** $= 0$) for

$$\int_0^1 \frac{4}{1+x^2}\,dx \quad \left(\textbf{absacc} = 10^{-5}\right).$$

(ii) Definite integral together with expansion coefficients (**iparm** $= 1$) for

$$\int_1^2 \sqrt[8]{x}\,dx \quad \left(\textbf{absacc} = 10^{-5}\right).$$

(iii) Indefinite integral using previous expansion (**iparm** $= 2$) for

$$\int_{1.2}^{1.8} \sqrt[8]{x}\, dx \quad \left(\mathbf{absacc} = 10^{-5}\right).$$

9.1 Program Text

```
     function d01ar_example

fprintf('d01ar example results\n\n');

absacc = 1e-05;
relacc = 0;
maxrul = nag_int(0);
alpha  = zeros(390,1);

% Integral 1
iparm = nag_int(0);
a = 0;
b = 1;

[acc1, ans1, n1, alpha, ifail] = ...
    d01ar(...
           a, b, @f1, relacc, absacc, maxrul, iparm, alpha);

fprintf('Definite integral of 4/(1+x*x) over (0,1)\n');
fprintf('Estimated value of the integral = %9.5f\n', ans1);
fprintf('Estimated absolute error        = %10.1e\n', acc1);
fprintf('Number of points used           = %4d\n\n', n1);

% Integral 2
iparm = nag_int(1);
a = 1;
b = 2;

[acc2, ans2, n2, alpha, ifail] = ...
    d01ar(...
           a, b, @f2, relacc, absacc, maxrul, iparm, alpha);

fprintf('Definite integral of x^(1/8) over (1,2)\n');
fprintf('Estimated value of the integral = %9.5f\n', ans2);
fprintf('Estimated absolute error        = %10.1e\n', acc2);
fprintf('Number of points used           = %4d\n\n', n2);

% Integral 3
iparm = nag_int(2);
a = 1.2;
b = 1.8;

[acc3, ans3, n3, alpha, ifail] = ...
    d01ar(...
           a, b, @f2, relacc, absacc, maxrul, iparm, alpha);

fprintf('Indefinite integral of x^(1/8) over (1.2,1.8)\n');
fprintf('Estimated value of the integral = %9.5f\n', ans3);

function f = f1(x)
  f = 4/(1+x^2);

function f = f2(x)
  f = x^(1/8);
```

9.2 Program Results

```
     d01ar example results

Definite integral of 4/(1+x*x) over (0,1)
Estimated value of the integral =   3.14159
Estimated absolute error        =    1.8e-08
Number of points used           =   15

Definite integral of xˆ(1/8) over (1,2)
Estimated value of the integral =   1.04979
Estimated absolute error        =    5.9e-07
Number of points used           =    7

Indefinite integral of xˆ(1/8) over (1.2,1.8)
Estimated value of the integral =   0.63073
```