

NAG Toolbox

nag_quad_1d_data (d01ga)

1 Purpose

nag_quad_1d_data (d01ga) integrates a function which is specified numerically at four or more points, over the whole of its specified range, using third-order finite difference formulae with error estimates, according to a method due to Gill and Miller (1972).

2 Syntax

```
[ans, er, ifail] = nag_quad_1d_data(x, y, 'n', n)
```

```
[ans, er, ifail] = d01ga(x, y, 'n', n)
```

3 Description

nag_quad_1d_data (d01ga) evaluates the definite integral

$$I = \int_{x_1}^{x_n} y(x) dx,$$

where the function y is specified at the n -points x_1, x_2, \dots, x_n , which should be all distinct, and in either ascending or descending order. The integral between successive points is calculated by a four-point finite difference formula centred on the interval concerned, except in the case of the first and last intervals, where four-point forward and backward difference formulae respectively are employed. If n is less than 4, the function fails. An approximation to the truncation error is integrated and added to the result. It is also returned separately to give an estimate of the uncertainty in the result. The method is due to Gill and Miller (1972).

4 References

Gill P E and Miller G F (1972) An algorithm for the integration of unequally spaced data *Comput. J.* **15** 80–83

5 Parameters

5.1 Compulsory Input Parameters

1: $\mathbf{x}(\mathbf{n})$ – REAL (KIND=nag_wp) array

The values of the independent variable, i.e., the x_1, x_2, \dots, x_n .

Constraint: either $\mathbf{x}(1) < \mathbf{x}(2) < \dots < \mathbf{x}(\mathbf{n})$ or $\mathbf{x}(1) > \mathbf{x}(2) > \dots > \mathbf{x}(\mathbf{n})$.

2: $\mathbf{y}(\mathbf{n})$ – REAL (KIND=nag_wp) array

The values of the dependent variable y_i at the points x_i , for $i = 1, 2, \dots, n$.

5.2 Optional Input Parameters

1: \mathbf{n} – INTEGER

Default: the dimension of the arrays \mathbf{x} , \mathbf{y} . (An error is raised if these dimensions are not equal.)
 n , the number of points.

Constraint: $\mathbf{n} \geq 4$.

5.3 Output Parameters

- 1: **ans** – REAL (KIND=nag_wp)
The estimated value of the integral.
- 2: **er** – REAL (KIND=nag_wp)
An estimate of the uncertainty in **ans**.
- 3: **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

Indicates that fewer than four points have been supplied to nag_quad_1d_data (d01ga).

ifail = 2

Values of **x** are neither strictly increasing nor strictly decreasing.

ifail = 3

Two points have the same X-value.

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.

No error is reported arising from the relative magnitudes of **ans** and **er** on return, due to the difficulty when the true answer is zero.

7 Accuracy

No accuracy level is specified by you before calling nag_quad_1d_data (d01ga) but on return the absolute value of **er** is an approximation to, but not necessarily a bound for, $|I - \mathbf{ans}|$. If on exit **ifail** > 0, both **ans** and **er** are returned as zero.

8 Further Comments

The time taken by nag_quad_1d_data (d01ga) depends on the number of points supplied, n .

In their paper, Gill and Miller (1972) do not add the quantity **er** to **ans** before return. However, extensive tests have shown that a dramatic reduction in the error often results from such addition. In other cases, it does not make an improvement, but these tend to be cases of low accuracy in which the modified answer is not significantly inferior to the unmodified one. You have the option of recovering the Gill–Miller answer by subtracting **er** from **ans** on return from the function.

9 Example

This example evaluates the integral

$$\int_0^1 \frac{4}{1+x^2} dx = \pi$$

reading in the function values at 21 unequally spaced points.

9.1 Program Text

```
function d01ga_example
fprintf('d01ga example results\n\n');
x = [0:0.04:1];
y = 4./(1+x.^2);
[ans, err, ifail] = d01ga(x, y);
fprintf('Integral = %9.6f      Estimated error = %9.2e\n',ans,err);
```

9.2 Program Results

```
d01ga example results
Integral =    3.141593      Estimated error = -3.19e-07
```
