

NAG Library Routine Document

D01GAF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

D01GAF 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 Specification

```
SUBROUTINE D01GAF (X, Y, N, ANS, ER, IFAIL)
  INTEGER          N, IFAIL
  REAL (KIND=nag_wp) X(N), Y(N), ANS, ER
```

3 Description

D01GAF 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 routine 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 Arguments

- | | | |
|----|--|---------------|
| 1: | X(N) – REAL (KIND=nag_wp) array | <i>Input</i> |
| | <i>On entry:</i> the values of the independent variable, i.e., the x_1, x_2, \dots, x_n . | |
| | <i>Constraint:</i> either $X(1) < X(2) < \dots < X(N)$ or $X(1) > X(2) > \dots > X(N)$. | |
| 2: | Y(N) – REAL (KIND=nag_wp) array | <i>Input</i> |
| | <i>On entry:</i> the values of the dependent variable y_i at the points x_i , for $i = 1, 2, \dots, n$. | |
| 3: | N – INTEGER | <i>Input</i> |
| | <i>On entry:</i> n , the number of points. | |
| | <i>Constraint:</i> $N \geq 4$. | |
| 4: | ANS – REAL (KIND=nag_wp) | <i>Output</i> |
| | <i>On exit:</i> the estimated value of the integral. | |

5: ER – REAL (KIND=nag_wp) *Output*

On exit: an estimate of the uncertainty in ANS.

6: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

Indicates that fewer than four points have been supplied to D01GAF.

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.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -399

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL = -999

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

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 D01GAF but on return the absolute value of ER is an approximation to, but not necessarily a bound for, $|I - \text{ANS}|$. If on exit IFAIL > 0, both ANS and ER are returned as zero.

8 Parallelism and Performance

D01GAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

The time taken by D01GAF 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 routine.

10 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.

10.1 Program Text

```

Program d01gafe

!      D01GAF Example Program Text
!
!      Mark 26 Release. NAG Copyright 2016.
!
!      .. Use Statements ..
!      Use nag_library, Only: d01gaf, nag_wp
!      .. Implicit None Statement ..
!      Implicit None
!      .. Parameters ..
!      Integer, Parameter          :: nin = 5, nout = 6
!      .. Local Scalars ..
!      Real (Kind=nag_wp)          :: ans, er
!      Integer                     :: i, ifail, n
!      .. Local Arrays ..
!      Real (Kind=nag_wp), Allocatable :: x(:), y(:)
!      .. Executable Statements ..
!      Write (nout,*) 'D01GAF Example Program Results'

!      Skip heading in data file
!      Read (nin,*)

!      Read (nin,*) n
!      Allocate (x(n),y(n))

!      Read (nin,*)(x(i),y(i),i=1,n)

!      ifail = -1
!      Call d01gaf(x,y,n,ans,er,ifail)

!      Select Case (ifail)
!      Case (0)
!          Write (nout,*)
!          Write (nout,99999) 'Integral = ', ans, '      Estimated error = ', er
!      Case (1)

```

```
      Write (nout,*)
      Write (nout,*) 'Less than 4 points supplied'
Case (2)
      Write (nout,*)
      Write (nout,*) 'Points not in increasing or decreasing order'
Case (3)
      Write (nout,*)
      Write (nout,*) 'Points not all distinct'
End Select

99999 Format (1X,A,F7.4,A,F7.4)
      End Program d01gafe
```

10.2 Program Data

D01GAF Example Program Data

```
21
0.00  4.0000
0.04  3.9936
0.08  3.9746
0.12  3.9432
0.22  3.8153
0.26  3.7467
0.30  3.6697
0.38  3.4943
0.39  3.4719
0.42  3.4002
0.45  3.3264
0.46  3.3014
0.60  2.9412
0.68  2.7352
0.72  2.6344
0.73  2.6094
0.83  2.3684
0.85  2.3222
0.88  2.2543
0.90  2.2099
1.00  2.0000
```

10.3 Program Results

D01GAF Example Program Results

Integral = 3.1414 Estimated error = -0.0001
