NAG Fortran Library Routine Document

D01AHF

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

D01AHF computes a definite integral over a finite range to a specified relative accuracy using a method described by Patterson.

2 Specification

```
real FUNCTION D01AHF(A, B, EPSR, NPTS, RELERR, F, NLIMIT, IFAIL)
INTEGER            NPTS, NLIMIT, IFAIL
real               A, B, EPSR, RELERR, F
EXTERNAL           F
```

3 Description

This routine computes a definite integral of the form

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

The method uses as its basis a family of interlacing high precision rules (see Patterson (1968b)) using 1, 3, 7, 15, 31, 63, 127 and 255 nodes. Initially the family is applied in sequence to the integrand. When two successive rules differ relatively by less than the required relative accuracy, the last rule used is taken as the value of the integral and the operation is regarded as successful. If all rules in the family have been applied unsuccessfully, subdivision is invoked. The subdivision strategy is as follows. The interval under scrutiny is divided into two sub-intervals (not always equal). The basic family is then applied to the first sub-interval. If the required accuracy is not obtained, the interval is stored for future examination (see IFAIL $= 2$) and the second sub-interval is examined. Should the basic family again be unsuccessful, then the sub-interval is further subdivided and the whole process repeated. Successful integrations are accumulated as the partial value of the integral. When all possible successful integrations have been completed, those previously unsuccessful sub-intervals placed in store are examined.

A large number of refinements are incorporated to improve the performance. Some of these are:

(a) The rate of convergence of the basic family is monitored and used to make a decision to abort and subdivide before the full sequence has been applied.

(b) The ϵ-algorithm is applied to the basic results in an attempt to increase the convergence rate. See Wynn (1956).

(c) An attempt is made to detect sharp end-point peaks and singularities in each sub-interval and to apply appropriate transformations to smooth the integrand. This consideration is also used to select interval sizes in the subdivision process.

(d) The relative accuracy sought in each sub-interval is adjusted in accordance with its likely contribution to the total integral.

(e) Random transformations of the integrand are applied to improve reliability in some instances.

4 References

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

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

5 Parameters

1: A – ***real*** *Input*

On entry: the lower limit of integration, a.

2: B – ***real*** *Input*

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

3: EPSR – ***real*** *Input*

On entry: the relative accuracy required.

Constraint: EPSR > 0.0.

4: NPTS – INTEGER *Output*

On exit: the number of function evaluations used in the calculation of the integral.

5: RELERR – ***real*** *Output*

On exit: a rough estimate of the relative error achieved.

6: F – ***real*** FUNCTION, supplied by the user. *External Procedure*

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

Its specification is:

```
real FUNCTION F(X)
real            X
```

1: X – ***real*** *Input*

On entry: the point at which the integrand must be evaluated.

F must be declared as EXTERNAL in the (sub)program from which D01AHF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

7: NLIMIT – INTEGER *Input*

On entry: a limit to the number of function evaluations. If NLIMIT ≤ 0, the routine uses a default limit of 10,000.

8: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.

On exit: IFAIL $= 0$ unless the routine detects an error (see Section 6).

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, because for this routine the values of the output parameters may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

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

The integral has not converged to the accuracy requested. It may be worthwhile to try increasing NLIMIT.

IFAIL = 2

Too many unsuccessful levels of subdivision have been invoked.

IFAIL = 3

On entry, EPSR $\leq$ 0.0.

When IFAIL = 1 or 2 a result may be obtained by continuing without further subdivision, but this is likely to be **inaccurate**.

7 Accuracy

The relative accuracy required is specified by the user in the variable EPSR. The routine will terminate whenever the relative accuracy specified by EPSR is judged to have been reached.

If on exit, IFAIL = 0, then it is most likely that the result is correct to the specified accuracy. If, on exit, IFAIL = 1 or IFAIL = 2, then it is likely that the specified accuracy has not been reached.

RELERR is a rough estimate of the relative error achieved. It is a by-product of the computation and is not used to effect the termination of the routine. The outcome of the integration must be judged by the value of IFAIL.

8 Further Comments

The time taken by the routine depends on the complexity of the integrand and the accuracy required.

9 Example

The following program evaluates the integral to a requested relative accuracy of 10^{-5}

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

9.1 Program Text

Note: the listing of the example program presented below uses ***bold italicised*** terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*     D01AHF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      real             A, ANS, B, EPSR, RELERR
      INTEGER          IFAIL, N, NLIMIT
*     .. External Functions ..
      real             D01AHF, FUN
      EXTERNAL         D01AHF, FUN
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D01AHF Example Program Results'
```

```
      A = 0.0e0
      B = 1.0e0
      NLIMIT = 0
      EPSR = 1.0e-5
      IFAIL = 1
      ANS = D01AHF(A,B,EPSR,N,RELERR,FUN,NLIMIT,IFAIL)
      WRITE (NOUT,*)
      IF (IFAIL.NE.0) THEN
         WRITE (NOUT,99997) 'IFAIL = ', IFAIL
         WRITE (NOUT,*)
      END IF
      IF (IFAIL.LE.2) THEN
         WRITE (NOUT,99999) 'Integral = ', ANS
         WRITE (NOUT,*)
         WRITE (NOUT,99998) 'Estimated relative error = ', RELERR
         WRITE (NOUT,*)
         WRITE (NOUT,99997) 'Number of function evaluations = ', N
      END IF
      STOP
*
99999 FORMAT (1X,A,F8.5)
99998 FORMAT (1X,A,e10.2)
99997 FORMAT (1X,A,I4)
      END
*
      real FUNCTION FUN(X)
*     .. Scalar Arguments ..
      real              X
*     .. Executable Statements ..
      FUN = 4.0e0/(1.0e0+X*X)
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 D01AHF Example Program Results

 Integral =  3.14159

 Estimated relative error =   0.58E-08

 Number of function evaluations =   15
```