NAG Fortran Library Routine Document

D01AQF

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

D01AQF calculates an approximation to the Hilbert transform of a function $g(x)$ over $[a, b]$:

$$I = \int_a^b \frac{g(x)}{x - c} \, dx$$

for user-specified values of a, b and c.

2 Specification

```
 SUBROUTINE D01AQF(G, A, B, C, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW,
1                  LIW, IFAIL)
 INTEGER           LW, IW(LIW), LIW, IFAIL
 real              G, A, B, C, EPSABS, EPSREL, RESULT, ABSERR, W(LW)
 EXTERNAL          G
```

3 Description

D01AQF is based upon the QUADPACK routine QAWC (Piessens *et al.* (1983)) and integrates a function of the form $g(x)w(x)$, where the weight function

$$w(x) = \frac{1}{x - c}$$

is that of the Hilbert transform. (If $a < c < b$ the integral has to be interpreted in the sense of a Cauchy principal value.) It is an adaptive routine which employs a 'global' acceptance criterion (as defined by Malcolm and Simpson (1976)). Special care is taken to ensure that c is never the end-point of a sub-interval (Piessens *et al.* (1976)). On each sub-interval (c_1, c_2) modified Clenshaw–Curtis integration of orders 12 and 24 is performed if $c_1 - d \le c \le c_2 + d$ where $d = (c_2 - c_1)/20$. Otherwise the Gauss 7-point and Kronrod 15-point rules are used. The local error estimation is described by Piessens *et al.* (1983).

4 References

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

Piessens R, van Roy-Branders M and Mertens I (1976) The automatic evaluation of Cauchy principal value integrals *Angew. Inf.* **18** 31–35

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

5 Parameters

1: G – ***real*** FUNCTION, supplied by the user. *External Procedure*

G must return the value of the function g at a given point.

Its specification is:

```
real FUNCTION G(X)
real           X
```

1: X – ***real*** *Input*

On entry: the point at which the function g must be evaluated.

G must be declared as EXTERNAL in the (sub)program from which D01AQF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: A – ***real*** *Input*

On entry: the lower limit of integration, a.

3: B – ***real*** *Input*

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

4: C – ***real*** *Input*

On entry: the parameter c in the weight function.

Constraint: C must not equal A or B.

5: EPSABS – ***real*** *Input*

On entry: the absolute accuracy required. If EPSABS is negative, the absolute value is used. See Section 7.

6: EPSREL – ***real*** *Input*

On entry: the relative accuracy required. If EPSREL is negative, the absolute value is used. See Section 7.

7: RESULT – ***real*** *Output*

On exit: the approximation to the integral I.

8: ABSERR – ***real*** *Output*

On exit: an estimate of the modulus of the absolute error, which should be an upper bound for $|I - \text{RESULT}|$.

9: W(LW) – ***real*** array *Output*

On exit: details of the computation, as described in Section 8.

10: LW – INTEGER *Input*

On entry: the dimension of the array W as declared in the (sub)program from which D01AQF is called. The value of LW (together with that of LIW below) imposes a bound on the number of sub-intervals into which the interval of integration may be divided by the routine. 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.

Constraint: LW $\geq$ 4.

11: IW(LIW) – INTEGER array *Output*

On exit: IW(1) contains the actual number of sub-intervals used. The rest of the array is used as workspace.

12: LIW – INTEGER *Input*

On entry: the dimension of the array IW as declared in the (sub)program from which D01AQF is called. The number of sub-intervals into which the interval of integration may be divided cannot exceed LIW.

Suggested value: LIW $= \text{LW}/4$.

Constraint: LIW ≥ 1.

13: 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 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. Another integrator which is designed for handling the type of difficulty involved, must be used. Alternatively consider relaxing the accuracy requirements specified by EPSABS and EPSREL, or increasing the workspace.

IFAIL $= 2$

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

IFAIL $= 3$

Extremely bad local behaviour of $g(x)$ 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 $= 4$

On entry, C $=$ A or C $=$ B.

IFAIL $= 5$

On entry, LW < 4,
or LIW < 1.

7 Accuracy

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

$$|I - \text{RESULT}| \leq tol,$$

where

$$tol = \max\{|\text{EPSABS}|, |\text{EPSREL}| \times |I|\},$$

and EPSABS and EPSREL are user-specified absolute and relative error tolerances. Moreover, it returns the quantity ABSERR which, in normal circumstances satisfies:

$$|I - \text{RESULT}| \leq \text{ABSERR} \leq tol.$$

8 Further Comments

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

If IFAIL $\neq 0$ on exit, then the user may wish to examine the contents of the array W, which contains the end-points of the sub-intervals used by D01AQF 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, $\int_{a_i}^{b_i} g(x)w(x)\,dx \simeq r_i$ and RESULT $= \sum_{i=1}^{n} r_i$. 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:

$$\begin{aligned} a_i &= \text{W}(i), \\ b_i &= \text{W}(n+i), \\ e_i &= \text{W}(2n+i) \text{ and} \\ r_i &= \text{W}(3n+i). \end{aligned}$$

9 Example

To compute the Cauchy principal value of

$$\int_{-1}^{1} \frac{dx}{(x^2 + 0.01^2)(x - \frac{1}{2})}.$$

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.

```
*     D01AQF Example Program Text
*     Mark 14 Revised.  NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          LW, LIW
      PARAMETER        (LW=800,LIW=LW/4)
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Scalars in Common ..
      INTEGER          KOUNT
*     .. Local Scalars ..
      real             A, ABSERR, B, C, EPSABS, EPSREL, RESULT
      INTEGER          IFAIL
*     .. Local Arrays ..
      real             W(LW)
      INTEGER          IW(LIW)
*     .. External Functions ..
      real             FST
      EXTERNAL         FST
*     .. External Subroutines ..
      EXTERNAL         D01AQF
*     .. Common blocks ..
      COMMON           /TELNUM/KOUNT
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D01AQF Example Program Results'
      EPSABS = 0.0e0
      EPSREL = 1.0e-04
```

```
      A = -1.0e0
      B = 1.0e0
      C = 0.5e0
      KOUNT = 0
      IFAIL = -1
*
      CALL D01AQF(FST,A,B,C,EPSABS,EPSREL,RESULT,ABSERR,W,LW,IW,LIW,
     +            IFAIL)
*
      WRITE (NOUT,*)
      WRITE (NOUT,99999) 'A      - lower limit of integration = ', A
      WRITE (NOUT,99999) 'B      - upper limit of integration = ', B
      WRITE (NOUT,99998) 'EPSABS - absolute accuracy requested = ',
     +  EPSABS
      WRITE (NOUT,99998) 'EPSREL - relative accuracy requested = ',
     +  EPSREL
      WRITE (NOUT,99998) 'C      - parameter in the weight function = ',
     +  C
      WRITE (NOUT,*)
      IF (IFAIL.NE.0) WRITE (NOUT,99996) 'IFAIL = ', IFAIL
      IF (IFAIL.LE.3) THEN
         WRITE (NOUT,99997) 'RESULT - approximation to the integral = ',
     +     RESULT
         WRITE (NOUT,99998) 'ABSERR - estimate of the absolute error = '
     +     , ABSERR
         WRITE (NOUT,99996) 'KOUNT  - number of function evaluations = '
     +     , KOUNT
         WRITE (NOUT,99996) 'IW(1)  - number of subintervals used = ',
     +     IW(1)
      END IF
      STOP
*
99999 FORMAT (1X,A,F10.4)
99998 FORMAT (1X,A,e9.2)
99997 FORMAT (1X,A,F9.2)
99996 FORMAT (1X,A,I4)
      END
*
      real FUNCTION FST(X)
*     .. Scalar Arguments ..
      real               X
*     .. Scalars in Common ..
      INTEGER            KOUNT
*     .. Local Scalars ..
      real               AA
*     .. Common blocks ..
      COMMON             /TELNUM/KOUNT
*     .. Executable Statements ..
      KOUNT = KOUNT + 1
      AA = 0.01e0
      FST = 1.0e0/(X**2+AA**2)
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
D01AQF Example Program Results

A      - lower limit of integration =    -1.0000
B      - upper limit of integration =     1.0000
EPSABS - absolute accuracy requested =  0.00E+00
EPSREL - relative accuracy requested =  0.10E-03
C      - parameter in the weight function =  0.50E+00

RESULT - approximation to the integral =   -628.46
ABSERR - estimate of the absolute error =  0.13E-01
KOUNT  - number of function evaluations =  255
IW(1)  - number of subintervals used =    8
```