NAG Fortran Library Routine Document

D01APF

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

D01APF is an adaptive integrator which calculates an approximation to the integral of a function $g(x)w(x)$ over a finite interval $[a, b]$:

$$I = \int_a^b g(x)w(x)\,dx$$

where the weight function w has end-point singularities of algebraico-logarithmic type.

2 Specification

```
      SUBROUTINE D01APF(G, A, B, ALFA, BETA, KEY, EPSABS, EPSREL, RESULT,
     1                  ABSERR, W, LW, IW, LIW, IFAIL)
      INTEGER           KEY, LW, IW(LIW), LIW, IFAIL
      real              G, A, B, ALFA, BETA, EPSABS, EPSREL, RESULT, ABSERR,
     1                  W(LW)
      EXTERNAL          G
```

3 Description

D01APF is based upon the QUADPACK routine QAWSE (Piessens *et al.* (1983)) and integrates a function of the form $g(x)w(x)$, where the weight function $w(x)$ may have algebraico-logarithmic singularities at the end-points a and/or b. The strategy is a modification of that in D01AKF. We start by bisecting the original interval and applying modified Clenshaw–Curtis integration of orders 12 and 24 to both halves. Clenshaw–Curtis integration is then used on all sub-intervals which have a or b as one of their end-points (Piessens *et al.* (1974)). On the other sub-intervals Gauss–Kronrod (7–15 point) integration is carried out.

A 'global' acceptance criterion (as defined by Malcolm and Simpson (1976)) is used. The local error estimation control 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, Mertens I and Branders M (1974) Integration of functions having end-point singularities *Angew. Inf.* **16** 65–68

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 X.

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 D01APF 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.

Constraint: B $>$ A.

4: ALFA – ***real*** *Input*

On entry: the parameter α in the weight function.

Constraint: ALFA > -1.

5: BETA – ***real*** *Input*

On entry: the parameter β in the weight function.

Constraint: BETA > -1.

6: KEY – INTEGER *Input*

On entry: indicates which weight function is to be used:

if KEY $= 1$, $w(x) = (x-a)^{\alpha}(b-x)^{\beta}$

if KEY $= 2$, $w(x) = (x-a)^{\alpha}(b-x)^{\beta}\ln(x-a)$

if KEY $= 3$, $w(x) = (x-a)^{\alpha}(b-x)^{\beta}\ln(b-x)$

if KEY $= 4$, $w(x) = (x-a)^{\alpha}(b-x)^{\beta}\ln(x-a)\ln(b-x)$

Constraint: KEY $= 1, 2, 3$ or 4.

7: EPSABS – ***real*** *Input*

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

8: EPSREL – ***real*** *Input*

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

9: RESULT – ***real*** *Output*

On exit: the approximation to the integral I.

10: ABSERR – ***real*** *Output*

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

11: W(LW) – ***real*** array *Output*

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

12: LW – INTEGER *Input*

On entry: the dimension of the array W as declared in the (sub)program from which D01APF 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 $\geq$ 8.

13: 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.

14: LIW – INTEGER *Input*

On entry: the dimension of the array IW as declared in the (sub)program from which D01APF is called. The number of sub-intervals into which the interval of integration may be divided cannot exceed LIW.

Suggested value: LIW = LW/4.

Constraint: LIW $\geq$ 2.

15: 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. If the position of a discontinuity or a singularity of algebraico-logarithmic type within the interval can be determined, the interval must be split up at this point and the integrator called on the subranges. If necessary, another integrator, which is designed for handling the difficulty involved, must be used. Alternatively consider relaxing the accuracy requirements specified by EPSABS and EPSREL, or increasing the amount of workspace.

IFAIL = 2

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

IFAIL $= 3$

Extremely bad local integrand behaviour 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, B $\leq$ A,
or ALFA ≤ -1,
or BETA ≤ -1,
or KEY < 1,
or KEY > 4.

IFAIL $= 5$

On entry, LW < 8,
or LIW < 2.

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 D01APF 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, $\int_{a_i}^{b_i} f(x)w(x)\,dx \simeq r_i$ 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:

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

9 Example

To compute:

$$\int_0^1 \ln x \cos(10\pi x)\,dx \quad \text{and} \quad \int_0^1 \frac{\sin(10x)}{\sqrt{x(1-x)}}\,dx.$$

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.

```
*     D01APF 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, NOF
*     .. Local Scalars ..
      real             A, ABSERR, B, EPSABS, EPSREL, RESULT
      INTEGER          IFAIL
*     .. Local Arrays ..
      real             ALFA(2), BETA(2), W(LW)
      INTEGER          INTEGR(2), IW(LIW)
*     .. External Functions ..
      real             FST
      EXTERNAL         FST
*     .. External Subroutines ..
      EXTERNAL         D01APF
*     .. Common blocks ..
      COMMON           /TELNUM/KOUNT, NOF
*     .. Data statements ..
      DATA             ALFA/0.0e0, -0.5e0/
      DATA             BETA/0.0e0, -0.5e0/
      DATA             INTEGR/2, 1/
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D01APF Example Program Results'
      EPSABS = 0.0e0
      EPSREL = 1.0e-04
      A = 0.0e0
      B = 1.0e0
      DO 20 NOF = 1, 2
         KOUNT = 0
         IFAIL = -1
*
         CALL D01APF(FST,A,B,ALFA(NOF),BETA(NOF),INTEGR(NOF),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,*)
         WRITE (NOUT,99998)
     +     'ALFA   - parameter in the weight function = ', ALFA(NOF)
         WRITE (NOUT,99998)
     +     'BETA   - parameter in the weight function = ', BETA(NOF)
         WRITE (NOUT,99997)
     +     'INTEGR - denotes which weight function is to be used = ',
     +     INTEGR(NOF)
         WRITE (NOUT,*)
         IF (IFAIL.NE.0) WRITE (NOUT,99997) 'IFAIL = ', IFAIL
         IF (IFAIL.LE.3) THEN
            WRITE (NOUT,99996)
     +        'RESULT - approximation to the integral = ', RESULT
            WRITE (NOUT,99998)
     +        'ABSERR - estimate of the absolute error = ', ABSERR
            WRITE (NOUT,99997)
     +        'KOUNT  - number of function evaluations = ', KOUNT
            WRITE (NOUT,99997) 'IW(1)  - number of subintervals used = '
     +        , IW(1)
```

```
         END IF
   20 CONTINUE
      STOP
*
99999 FORMAT (1X,A,F10.4)
99998 FORMAT (1X,A,e9.2)
99997 FORMAT (1X,A,I4)
99996 FORMAT (1X,A,F9.5)
      END
*
      real  FUNCTION FST(X)
*     .. Scalar Arguments ..
      real              X
*     .. Scalars in Common ..
      INTEGER           KOUNT, NOF
*     .. Local Scalars ..
      real              A, OMEGA, PI
*     .. External Functions ..
      real              X01AAF
      EXTERNAL          X01AAF
*     .. Intrinsic Functions ..
      INTRINSIC         COS, SIN
*     .. Common blocks ..
      COMMON            /TELNUM/KOUNT, NOF
*     .. Executable Statements ..
      PI = X01AAF(PI)
      KOUNT = KOUNT + 1
      IF (NOF.EQ.1) THEN
         A = 10.0e0*PI
         FST = COS(A*X)
      ELSE
         OMEGA = 10.0e0
         FST = SIN(OMEGA*X)
      END IF
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
 D01APF Example Program Results

 A      - lower limit of integration =     0.0000
 B      - upper limit of integration =     1.0000
 EPSABS - absolute accuracy requested =  0.00E+00
 EPSREL - relative accuracy requested =  0.10E-03

 ALFA   - parameter in the weight function =  0.00E+00
 BETA   - parameter in the weight function =  0.00E+00
 INTEGR - denotes which weight function is to be used =    2

 RESULT - approximation to the integral =  -0.04899
 ABSERR - estimate of the absolute error =  0.11E-06
 KOUNT  - number of function evaluations =  110
 IW(1)  - number of subintervals used =    4

 A      - lower limit of integration =     0.0000
 B      - upper limit of integration =     1.0000
 EPSABS - absolute accuracy requested =  0.00E+00
 EPSREL - relative accuracy requested =  0.10E-03

 ALFA   - parameter in the weight function = -0.50E+00
 BETA   - parameter in the weight function = -0.50E+00
 INTEGR - denotes which weight function is to be used =    1

 RESULT - approximation to the integral =   0.53502
 ABSERR - estimate of the absolute error =  0.19E-11
```

```
 KOUNT  - number of function evaluations =   50
 IW(1)  - number of subintervals used =    2
```
