NAG Library Routine Document

D01ATF

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

D01ATF is a general purpose integrator which calculates an approximation to the integral of a function $f(x)$ over a finite interval $[a, b]$:

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

2 Specification

```
SUBROUTINE D01ATF (F, A, B, EPSABS, EPSREL, RESULT, ABSERR, W, LW, IW,     &
                   LIW, IFAIL)

INTEGER            LW, IW(LIW), LIW, IFAIL
REAL (KIND=nag_wp) A, B, EPSABS, EPSREL, RESULT, ABSERR, W(LW)
EXTERNAL           F
```

3 Description

D01ATF is based on the QUADPACK routine QAGS (see Piessens *et al.* (1983)). It is an adaptive routine, using the Gauss 10-point and Kronrod 21-point rules. The algorithm, described in de Doncker (1978), incorporates a global acceptance criterion (as defined by Malcolm and Simpson (1976)) together with the ϵ-algorithm (see Wynn (1956)) to perform extrapolation. The local error estimation is described in Piessens *et al.* (1983).

The routine is suitable as a general purpose integrator, and can be used when the integrand has singularities, especially when these are of algebraic or logarithmic type.

D01ATF requires a subroutine to evaluate the integrand at an array of different points and is therefore amenable to parallel execution. Otherwise the algorithm is identical to that used by D01AJF.

4 References

de Doncker E (1978) An adaptive extrapolation algorithm for automatic integration *ACM SIGNUM Newsl.* **13(2)** 12–18

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

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

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

5 Parameters

1: F – SUBROUTINE, supplied by the user. *External Procedure*

F must return the values of the integrand f at a set of points.

The specification of F is:

```
SUBROUTINE F (X, FV, N)
```

```
INTEGER            N
REAL (KIND=nag_wp) X(N), FV(N)
```

1: X(N) – REAL (KIND=nag_wp) array *Input*

On entry: the points at which the integrand f must be evaluated.

2: FV(N) – REAL (KIND=nag_wp) array *Output*

On exit: FV(j) must contain the value of f at the point X(j), for $j = 1, 2, \ldots, \mathrm{N}$.

3: N – INTEGER *Input*

On entry: the number of points at which the integrand is to be evaluated. The actual value of N is always 21.

F must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which D01ATF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

2: A – REAL (KIND=nag_wp) *Input*

On entry: a, the lower limit of integration.

3: B – REAL (KIND=nag_wp) *Input*

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

4: EPSABS – REAL (KIND=nag_wp) *Input*

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

5: EPSREL – REAL (KIND=nag_wp) *Input*

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

6: RESULT – REAL (KIND=nag_wp) *Output*

On exit: the approximation to the integral I.

7: ABSERR – REAL (KIND=nag_wp) *Output*

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

8: W(LW) – REAL (KIND=nag_wp) array *Output*

On exit: details of the computation see Section 9 for more information.

9: LW – INTEGER *Input*

On entry: the dimension of the array W as declared in the (sub)program from which D01ATF is called. The value of LW (together with that of LIW) 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 $\mathrm{LW}/4$. The more difficult the integrand, the larger LW should be.

Suggested value: $\mathrm{LW} = 800$ to 2000 is adequate for most problems.

Constraint: $\mathrm{LW} \geq 4$.

10: 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.

11: LIW – INTEGER *Input*

On entry: the dimension of the array IW as declared in the (sub)program from which D01ATF is called. The number of sub-intervals into which the interval of integration may be divided cannot exceed LIW.

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

Constraint: $\mathrm{LIW} \geq 1$.

12: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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, because for this routine the values of the output parameters may be useful even if $\mathrm{IFAIL} \neq 0$ 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.**

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

6 Error Indicators and Warnings

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

Note: D01ATF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

$\mathrm{IFAIL} = 1$

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 local difficulty within the interval can be determined (e.g., a singularity of the integrand or its derivative, a peak, a discontinuity, etc.) you will probably gain from splitting up the interval at this point and calling the integrator on the subranges. If necessary, 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 amount of workspace.

$\mathrm{IFAIL} = 2$

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

$\mathrm{IFAIL} = 3$

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 $\mathrm{IFAIL} = 1$.

$\mathrm{IFAIL} = 4$

The requested tolerance cannot be achieved because the extrapolation does not increase the accuracy satisfactorily; the returned result is the best which can be obtained. The same advice applies as in the case of $\mathrm{IFAIL} = 1$.

IFAIL $= 5$

The integral is probably divergent, or slowly convergent. Please note that divergence can occur with any nonzero value of IFAIL.

IFAIL $= 6$

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

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

D01ATF cannot guarantee, but in practice usually achieves, the following accuracy:

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

where

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

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

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

8 Parallelism and Performance

Not applicable.

9 Further Comments

If IFAIL $\neq 0$ on exit, then you may wish to examine the contents of the array W, which contains the end points of the sub-intervals used by D01ATF along with the integral contributions and error estimates over the 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, $\int_{a_i}^{b_i} f(x)\, dx \simeq r_i$ and $\mathrm{RESULT} = \sum_{i=1}^{n} r_i$, unless D01ATF terminates while testing for divergence of the integral (see Section 3.4.3 of Piessens *et al.* (1983)). In this case, RESULT (and ABSERR) are taken to be the values returned from the extrapolation process. 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:

$a_i = \mathrm{W}(i)$,

$b_i = \mathrm{W}(n + i)$,

$e_i = \mathrm{W}(2n + i)$ and

$r_i = \mathrm{W}(3n + i)$.

10 Example

This example computes

$$\int_0^{2\pi} \frac{x \sin(30x)}{\sqrt{1 - (x/2\pi)^2}} dx.$$

10.1 Program Text

```
!   D01ATF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.

    Module d01atfe_mod

!     D01ATF Example Program Module:
!            Parameters and User-defined Routines

!     .. Use Statements ..
      Use nag_library, Only: nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Accessibility Statements ..
      Private
      Public                               :: f
!     .. Parameters ..
      Integer, Parameter, Public           :: lw = 800, nout = 6
      Integer, Parameter, Public           :: liw = lw/4
!     .. Local Scalars ..
      Real (Kind=nag_wp), Public, Save     :: pi
    Contains
      Subroutine f(x,fv,n)

!       .. Scalar Arguments ..
        Integer, Intent (In)                 :: n
!       .. Array Arguments ..
        Real (Kind=nag_wp), Intent (Out)     :: fv(n)
        Real (Kind=nag_wp), Intent (In)      :: x(n)
!       .. Intrinsic Procedures ..
        Intrinsic                            :: sin, sqrt
!       .. Executable Statements ..
        fv(1:n) = x(1:n)*sin(30.0E0_nag_wp*x(1:n))/sqrt(1.0E0_nag_wp-x(1:n)**2 &
          /(4.0E0_nag_wp*pi**2))

        Return

      End Subroutine f
    End Module d01atfe_mod
    Program d01atfe

!     D01ATF Example Main Program

!     .. Use Statements ..
      Use nag_library, Only: d01atf, nag_wp, x01aaf
      Use d01atfe_mod, Only: f, liw, lw, nout, pi
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                   :: a, abserr, b, epsabs, epsrel,    &
                                              result
      Integer                              :: ifail
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable      :: w(:)
      Integer, Allocatable                 :: iw(:)
!     .. Executable Statements ..
```

```
      Write (nout,*) 'D01ATF Example Program Results'

      Allocate (w(lw),iw(liw))

      pi = x01aaf(pi)
      epsabs = 0.0_nag_wp
      epsrel = 1.0E-04_nag_wp
      a = 0.0_nag_wp
      b = 2.0_nag_wp*pi

      ifail = -1
      Call d01atf(f,a,b,epsabs,epsrel,result,abserr,w,lw,iw,liw,ifail)

      If (ifail>=0) Then
        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
      End If
      If (ifail>=0 .And. ifail<=5) Then
        Write (nout,*)
        Write (nout,99997) 'RESULT', 'approximation to the integral', result
        Write (nout,99998) 'ABSERR', 'estimate of the absolute error', abserr
        Write (nout,99996) 'IW(1) ', 'number of subintervals used', iw(1)
      End If

99999 Format (1X,A6,' - ',A30,' = ',F10.4)
99998 Format (1X,A6,' - ',A30,' = ',E9.2)
99997 Format (1X,A6,' - ',A30,' = ',F9.5)
99996 Format (1X,A6,' - ',A30,' = ',I4)
    End Program d01atfe
```

10.2 Program Data

None.

10.3 Program Results

```
 D01ATF Example Program Results

 A      -     lower limit of integration =     0.0000
 B      -     upper limit of integration =     6.2832
 EPSABS -    absolute accuracy requested =  0.00E+00
 EPSREL -    relative accuracy requested =  0.10E-03

 RESULT -  approximation to the integral =  -2.54326
 ABSERR - estimate of the absolute error =  0.13E-04
 IW(1)  -    number of subintervals used =   19
```