NAG Library Routine Document

D01ARF

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

D01ARF computes definite and indefinite integrals over a finite range to a specified relative or absolute accuracy, using the method described in Patterson (1968).

2 Specification

```
SUBROUTINE D01ARF (A, B, FUN, RELACC, ABSACC, MAXRUL, IPARM, ACC, ANS,      &
                   N, ALPHA, IFAIL)
INTEGER              MAXRUL, IPARM, N, IFAIL
REAL (KIND=nag_wp) A, B, FUN, RELACC, ABSACC, ACC, ANS, ALPHA(390)
EXTERNAL             FUN
```

3 Description

D01ARF evaluates definite and indefinite integrals of the form:

$$\int_a^b f(t)\, dt$$

using the method described in Patterson (1968).

3.1 Definite Integrals

In this case D01ARF must be called with IPARM $= 0$. By linear transformation the integral is changed to

$$I = \int_{-1}^{+1} F(x)\, dx$$

where

$$F(x) = \frac{b-a}{2} f\left(\frac{b+a+(b-a)x}{2}\right)$$

and is then approximated by an n-point quadrature rule

$$I = \sum_{k=1}^{n} w_k F(x_k)$$

where w_k are the weights and x_k are the abscissae.

The routine uses a family of nine interlacing rules based on the optimal extension of the three-point Gauss rule. These rules use 1, 3, 7, 15, 31, 63, 127, 255 and 511 points and have respective polynomial integrating degrees 1, 5, 11, 23, 47, 95, 191, 383 and 767. Each rule has the property that the next in sequence includes all the points of its predecessor and has the greatest possible increase in integrating degree.

The integration method is based on the successive application of these rules until the absolute value of the difference of two successive results differs by not more than ABSACC, or relatively by not more than RELACC. The result of the last rule used is taken as the value of the integral (ANS), and the absolute difference of the results of the last two rules used is taken as an estimate of the absolute error

(ACC). Due to their interlacing form no integrand evaluations are wasted in passing from one rule to the next.

3.2 Indefinite Integrals

Suppose the value of the integral

$$\int_c^d f(t)\,dt$$

is required for a number of sub-intervals $[c,d]$, all of which lie in an interval $[a,b]$.

In this case D01ARF should first be called with the parameter IPARM $= 1$ and the interval set to $[a,b]$. The routine then calculates the integral over $[a,b]$ **and** the Legendre expansion of the integrand, using the same integrand values. If the routine is subsequently called with IPARM $= 2$ and the interval set to $[c,d]$, the integral over $[c,d]$ is calculated by analytical integration of the Legendre expansion, without further evaluations of the integrand.

For the interval $[-1,1]$ the expansion takes the form

$$F(x) = \sum_{i=0}^{\infty} \alpha_i P_i(x)$$

where $P_i(x)$ is the order i Legendre polynomial. Assuming that the integral over the full range $[-1,1]$ was evaluated to the required accuracy using an n-point rule, then the coefficients

$$\alpha_i = \tfrac{1}{2}(2i-1)\int_{-1}^{+1} P_i(x)F(x)\,dx, \quad i = 0,1,\ldots,m$$

are evaluated by that same rule, up to

$$m = (3n-1)/4.$$

The accuracy for indefinite integration should be of the same order as that obtained for the definite integral over the full range. The indefinite integrals will be exact when $F(x)$ is a polynomial of degree $\leq m$.

4 References

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

5 Parameters

1: A – REAL (KIND=nag_wp) *Input*

On entry: a, the lower limit of integration.

2: B – REAL (KIND=nag_wp) *Input*

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

3: FUN – REAL (KIND=nag_wp) FUNCTION, supplied by the user. *External Procedure*

FUN must return the value of the integrand f at a specified point.

The specification of FUN is:

```
FUNCTION FUN (X)
REAL (KIND=nag_wp) FUN
REAL (KIND=nag_wp) X
```

1: X – REAL (KIND=nag_wp) *Input*

On entry: the point in $[a, b]$ at which the integrand f must be evaluated.

FUN must either be a module subprogram USEd by, or declared as EXTERNAL in, the (sub)program from which D01ARF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

If IPARM $= 2$, FUN is not called.

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

On entry: the relative accuracy required. If convergence according to absolute accuracy is required, RELACC should be set to zero (but see also Section 7). If RELACC < 0.0, its absolute value is used.

If IPARM $= 2$, RELACC is not used.

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

On entry: the absolute accuracy required. If convergence according to relative accuracy is required, ABSACC should be set to zero (but see also Section 7). If ABSACC < 0.0, its absolute value is used.

If IPARM $= 2$, ABSACC is not used.

6: MAXRUL – INTEGER *Input*

On entry: the maximum number of successive rules that may be used.

Constraint: $1 \leq$ MAXRUL ≤ 9. If MAXRUL is outside these limits, the value 9 is assumed.

If IPARM $= 2$, MAXRUL is not used.

7: IPARM – INTEGER *Input*

On entry: indicates the task to be performed by the routine.

IPARM $= 0$
: Only the definite integral over $[a, b]$ is evaluated.

IPARM $= 1$
: As well as the definite integral, the expansion of the integrand in Legendre polynomials over $[a, b]$ is calculated, using the same values of the integrand as used to compute the integral. The expansion coefficients, and some other quantities, are returned in ALPHA for later use in computing indefinite integrals.

IPARM $= 2$
: $f(t)$ is integrated analytically over $[a, b]$ using the previously computed expansion, stored in ALPHA. No further evaluations of the integrand are required. The routine must previously have been called with IPARM $= 1$ and the interval $[a, b]$ must lie within that specified for the previous call. In this case only the arguments A, B, IPARM, ANS, ALPHA and IFAIL are used.

Constraint: IPARM $= 0$, 1 or 2.

8: ACC – REAL (KIND=nag_wp) *Output*

On exit: if IPARM $= 0$ or 1, ACC contains the absolute value of the difference between the last two successive estimates of the integral. This may be used as a measure of the accuracy actually achieved.

If IPARM $= 2$, ACC is not used.

9: ANS – REAL (KIND=nag_wp) *Output*

On exit: the estimated value of the integral.

10: N – INTEGER *Output*

On exit: when IPARM $= 0$ or 1, N contains the number of integrand evaluations used in the calculation of the integral.

If IPARM $= 2$, N is not used.

11: ALPHA(390) – REAL (KIND=nag_wp) array *Input/Output*

On entry: if IPARM $= 2$, ALPHA must contain the coefficients of the Legendre expansions of the integrand, as returned by a previous call of D01ARF with IPARM $= 1$ and a range containing the present range.

If IPARM $= 0$ or 1, ALPHA need not be set on entry.

On exit: if IPARM $= 1$, the first m elements of ALPHA hold the coefficients of the Legendre expansion of the integrand, and the value of m is stored in ALPHA(390). ALPHA must not be changed between a call with IPARM $= 1$ and subsequent calls with IPARM $= 2$.

If IPARM $= 2$, the first m elements of ALPHA are unchanged on exit.

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 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: 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).

Note: D01ARF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL $= 1$

If IPARM $= 0$ or 1, this indicates that all MAXRUL rules have been used and the integral has not converged to the accuracy requested. In this case ANS contains the last approximation to the integral, and ACC contains the difference between the last two approximations. To check this estimate of the integral, D01ARF could be called again to evaluate

$$\int_a^b f(t)\,dt \quad \text{as} \quad \int_a^c f(t)\,dt + \int_c^b f(t)\,dt \quad \text{for some } a < c < b.$$

If IPARM $= 2$, this indicates failure of convergence during the run with IPARM $= 1$ in which the Legendre expansion was created.

IFAIL $= 2$

On entry, IPARM $\neq 0$, 1 or 2

IFAIL $= 3$

The routine is called with IPARM $= 2$ but a previous call with IPARM $= 1$ has been omitted or was invoked with an integration interval of length zero.

IFAIL $= 4$

On entry, with IPARM $= 2$, the interval for indefinite integration is not contained within the interval specified when D01ARF was previously called with IPARM $= 1$.

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

The relative or absolute accuracy required is specified by you in the variables RELACC or ABSACC. D01ARF will terminate whenever either the relative accuracy specified by RELACC or the absolute accuracy specified by ABSACC is reached. One or other of these criteria may be 'forced' by setting the parameter for the other to zero. If both RELACC and ABSACC are specified as zero, then the routine uses the value $10.0 \times$ (***machine precision***) for RELACC.

If on exit IFAIL $= 0$, then it is likely that the result is correct to one or other of these accuracies. If on exit IFAIL $= 1$, then it is likely that neither of the requested accuracies has been reached.

When you have no prior idea of the magnitude of the integral, it is possible that an unreasonable accuracy may be requested, e.g., a relative accuracy for an integral which turns out to be zero, or a small absolute accuracy for an integral which turns out to be very large. Even if failure is reported in such a case, the value of the integral may still be satisfactory. The device of setting the other 'unused' accuracy parameter to a small positive value (e.g., 10^{-9} for an implementation of 11-digit precision) rather than zero, may prevent excessive calculation in such a situation.

To avoid spurious convergence, it is recommended that relative accuracies larger than about 10^{-3} be avoided.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by D01ARF depends on the complexity of the integrand and the accuracy required.

This routine uses the Patterson method over the whole integration interval and should therefore be suitable for well behaved functions. However, for very irregular functions it would be more efficient to submit the differently behaved regions separately for integration.

10 Example

This example evaluates the following integrals

(i) Definite integral only $(\text{IPARM} = 0)$ for

$$\int_0^1 \frac{4}{1+x^2}\,dx \quad \left(\text{ABSACC} = 10^{-5}\right).$$

(ii) Definite integral together with expansion coefficients $(\text{IPARM} = 1)$ for

$$\int_1^2 \sqrt[8]{x}\,dx \quad \left(\text{ABSACC} = 10^{-5}\right).$$

(iii) Indefinite integral using previous expansion $(\text{IPARM} = 2)$ for

$$\int_{1.2}^{1.8} \sqrt[8]{x}\,dx \quad \left(\text{ABSACC} = 10^{-5}\right).$$

10.1 Program Text

```
!   D01ARF Example Program Text
!   Mark 25 Release. NAG Copyright 2014.

    Module d01arfe_mod

!     D01ARF 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                               :: f1, f2
!     .. Parameters ..
      Integer, Parameter, Public           :: maxrul = 0, nout = 6
    Contains
      Function f1(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                   :: f1
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: x
!       .. Executable Statements ..
        f1 = 4.0E0_nag_wp/(1.0E0_nag_wp+x*x)

        Return

      End Function f1
      Function f2(x)

!       .. Function Return Value ..
        Real (Kind=nag_wp)                   :: f2
!       .. Scalar Arguments ..
        Real (Kind=nag_wp), Intent (In)      :: x
!       .. Executable Statements ..
        f2 = x**0.125E0_nag_wp

        Return

      End Function f2
    End Module d01arfe_mod
    Program d01arfe

!     D01ARF Example Main Program

!     .. Use Statements ..
```

```
      Use nag_library, Only: d01arf, nag_wp
      Use d01arfe_mod, Only: f1, f2, maxrul, nout
!     .. Implicit None Statement ..
      Implicit None
!     .. Local Scalars ..
      Real (Kind=nag_wp)                   :: a, absacc, acc, ans, b, relacc
      Integer                              :: ifail, iparm, n
!     .. Local Arrays ..
      Real (Kind=nag_wp)                   :: alpha(390)
!     .. Executable Statements ..
      Write (nout,*) 'D01ARF Example Program Results'

      relacc = 0.0E0_nag_wp
      absacc = 1.0E-5_nag_wp

!     Definite integral of F1(x) - no expansion

      iparm = 0
      a = 0.0E0_nag_wp
      b = 1.0E0_nag_wp

      Write (nout,*)
      Write (nout,*) 'Definite integral of 4/(1+x*x) over (0,1)'

      ifail = -1
      Call d01arf(a,b,f1,relacc,absacc,maxrul,iparm,acc,ans,n,alpha,ifail)

      Select Case (ifail)
      Case (:-1)
        Go To 100
      Case (0,1)
        Write (nout,99999) 'Estimated value of the integral =', ans
        Write (nout,99998) 'Estimated absolute error =', acc
        Write (nout,99997) 'Number of points used =', n
      End Select

!     Definite integral of F2(x) - with expansion

      iparm = 1
      a = 1.0E0_nag_wp
      b = 2.0E0_nag_wp

      Write (nout,*)
      Write (nout,*) 'Definite integral of x**(1/8) over (1,2)'

      ifail = -1
      Call d01arf(a,b,f2,relacc,absacc,maxrul,iparm,acc,ans,n,alpha,ifail)

      Select Case (ifail)
      Case (:-1)
        Go To 100
      Case (0,1)
        Write (nout,99999) 'Estimated value of the integral =', ans
        Write (nout,99998) 'Estimated absolute error =', acc
        Write (nout,99997) 'Number of points used =', n
      End Select

!     Indefinite integral of F2(x)

      iparm = 2
      a = 1.2E0_nag_wp
      b = 1.8E0_nag_wp

      Write (nout,*)
      Write (nout,*) 'Indefinite integral of x**(1/8) over (1.2,1.8)'

      ifail = 0
      Call d01arf(a,b,f2,relacc,absacc,maxrul,iparm,acc,ans,n,alpha,ifail)

      Write (nout,99999) 'Estimated value of the integral =', ans
```

```
100   Continue

99999 Format (1X,A,F9.5)
99998 Format (1X,A,E10.2)
99997 Format (1X,A,I4)
    End Program d01arfe
```

10.2 Program Data

None.

10.3 Program Results

```
 D01ARF Example Program Results

 Definite integral of 4/(1+x*x) over (0,1)
 Estimated value of the integral =  3.14159
 Estimated absolute error =  0.18E-07
 Number of points used =  15

 Definite integral of x**(1/8) over (1,2)
 Estimated value of the integral =  1.04979
 Estimated absolute error =  0.59E-06
 Number of points used =   7

 Indefinite integral of x**(1/8) over (1.2,1.8)
 Estimated value of the integral =  0.63073
```