NAG Fortran Library Routine Document

D01DAF

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

D01DAF attempts to evaluate a double integral to a specified absolute accuracy by repeated applications of the method described by Patterson.

2 Specification

```
SUBROUTINE D01DAF (YA, YB, PHI1, PHI2, F, ABSACC, ANS, NPTS, IFAIL)

INTEGER             NPTS, IFAIL
double precision    YA, YB, ABSACC, ANS
EXTERNAL            PHI1, PHI2, F
```

3 Description

D01DAF attempts to evaluate a definite integral of the form

$$I = \int_a^b \int_{\phi_1(y)}^{\phi_2(y)} f(x,y)\, dx\, dy$$

where a and b are constants and $\phi_1(y)$ and $\phi_2(y)$ are functions of the variable y.

The integral is evaluated by expressing it as

$$I = \int_a^b F(y)\, dy, \qquad \text{where} \qquad F(y) = \int_{\phi_1(y)}^{\phi_2(y)} f(x,y)\, dx.$$

Both the outer integral I and the inner integrals $F(y)$ are evaluated by the method, described by Patterson (1968a) and Patterson (1969), of the optimum addition of points to Gauss quadrature formulae.

This method uses a family of interlacing common point formulae. Beginning with the 3-point Gauss rule, formulae using 7, 15, 31, 63, 127 and finally 255 points are derived. Each new formula contains all the pivots of the earlier formulae so that no function evaluations are wasted. Each integral is evaluated by applying these formulae successively until two results are obtained which differ by less than the specified absolute accuracy.

4 References

Patterson T N L (1968a) On some Gauss and Lobatto based integration formulae *Math. Comput.* **22** 877–881

Patterson T N L (1969) The optimum addition of points to quadrature formulae, errata *Math. Comput.* **23** 892

5 Parameters

1: YA – ***double precision*** *Input*

On entry: a, the lower limit of the integral.

2: YB – ***double precision*** *Input*

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

3: PHI1 – ***double precision*** FUNCTION, supplied by the user *External Procedure*

PHI1 must return the lower limit of the inner integral for a given value of y.

Its specification is:

```
double precision FUNCTION PHI1 (Y)
double precision               Y
```

1: Y – ***double precision*** *Input*

On entry: the value of y for which the lower limit must be evaluated.

PHI1 must be declared as EXTERNAL in the (sub)program from which D01DAF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

4: PHI2 – ***double precision*** FUNCTION, supplied by the user *External Procedure*

PHI2 must return the upper limit of the inner integral for a given value of y.

Its specification is:

```
double precision FUNCTION PHI2 (Y)
double precision               Y
```

1: Y – ***double precision*** *Input*

On entry: the value of y for which the upper limit must be evaluated.

PHI2 must be declared as EXTERNAL in the (sub)program from which D01DAF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

5: F – ***double precision*** FUNCTION, supplied by the user *External Procedure*

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

Its specification is:

```
double precision FUNCTION F (X, Y)
double precision            X, Y
```

1: X – ***double precision*** *Input*
2: Y – ***double precision*** *Input*

On entry: the co-ordinates of the point (x, y) at which the integrand must be evaluated.

F must be declared as EXTERNAL in the (sub)program from which D01DAF is called. Parameters denoted as *Input* must **not** be changed by this procedure.

6: ABSACC – ***double precision*** *Input*

On entry: the absolute accuracy requested.

7: ANS – ***double precision*** *Output*

On exit: the estimate of the integral.

8: NPTS – INTEGER *Output*

On exit: the total number of function evaluations.

9: IFAIL – INTEGER *Input/Output*

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

On final 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 $= 1$

This indicates that 255 points have been used in the outer integral and convergence has not been obtained. All the inner integrals have, however, converged. In this case ANS may still contain an approximate estimate of the integral.

IFAIL $= 10 \times n$

This indicates that the outer integral has converged but n inner integrals have failed to converge with the use of 255 points. In this case ANS may still contain an approximate estimate of the integral, but its reliability will decrease as n increases.

IFAIL $= 10 \times n + 1$

This indicates that both the outer integral and n of the inner integrals have not converged. ANS may still contain an approximate estimate of the integral, but its reliability will decrease as n increases.

7 Accuracy

The absolute accuracy is specified by the variable ABSACC. If, on exit, IFAIL $= 0$ then the result is most likely correct to this accuracy. Even if IFAIL is non-zero on exit, it is still possible that the calculated result could differ from the true value by less than the given accuracy.

8 Further Comments

The time taken by D01DAF depends upon the complexity of the integrand and the accuracy requested.

With Patterson's method accidental convergence may occasionally occur, when two estimates of an integral agree to within the requested accuracy, but both estimates differ considerably from the true result. This could occur in either the outer integral or in one or more of the inner integrals.

If it occurs in the outer integral then apparent convergence is likely to be obtained with considerably fewer integrand evaluations than may be expected. If it occurs in an inner inegral, the incorrect value could make the function $F(y)$ appear to be badly behaved, in which case a very large number of pivots may be needed for the overall evaluation of the integral. Thus both unexpectedly small and unexpectedly large numbers of integrand evaluations should be considered as indicating possible trouble. If accidental convergence is suspected, the integral may be recomputed, requesting better accuracy; if the new request is more stringent than the degree of accidental agreement (which is of course unknown), improved results should be obtained. This is only possible when the accidental agreement is not better than machine accuracy. It should be noted that the routine requests the same accuracy for the inner integrals as for the outer integral. In practice it has been found that in the vast majority of cases this has proved to be adequate for the overall result of the double integral to be accurate to within the specified value.

The routine is not well-suited to non-smooth integrands, i.e., integrands having some kind of analytic discontinuity (such as a discontinuous or infinite partial derivative of some low-order) in, on the boundary of, or near, the region of integration. **Warning**: such singularities may be induced by incautiously presenting an apparently smooth interval over the positive quadrant of the unit circle, R

$$I = \int_R (x+y)\,dx\,dy.$$

This may be presented to D01DAF as

$$I = \int_0^1 dy \int_0^{\sqrt{1-y^2}} (x+y)\,dx = \int_0^1 \left(\tfrac{1}{2}\left(1-y^2\right) + y\sqrt{1-y^2}\right) dy$$

but here the outer integral has an induced square-root singularity stemming from the way the region has been presented to D01DAF. This situation should be avoided by re-casting the problem. For the example given, the use of polar co-ordinates would avoid the difficulty:

$$I = \int_0^1 dr \int_0^{\frac{\pi}{2}} r^2(\cos v + \sin v)\,dv.$$

9 Example

The following program evaluates the integral discussed in Section 8, presenting it to D01DAF first as

$$\int_0^1 \int_0^{\sqrt{1-y^2}} (x+y)\,dx\,dy$$

and then as

$$\int_0^1 \int_0^{\frac{\pi}{2}} r^2(\cos v + \sin v)\,dv\,dr.$$

Note the difference in the number of function evaluations.

9.1 Program Text

```
*     D01DAF Example Program Text
*     Mark 14 Revised. NAG Copyright 1989.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      DOUBLE PRECISION ABSACC, ANS, YA, YB
      INTEGER          IFAIL, NPTS
*     .. External Functions ..
      DOUBLE PRECISION FA, FB, P1, P2A, P2B
      EXTERNAL         FA, FB, P1, P2A, P2B
*     .. External Subroutines ..
      EXTERNAL         D01DAF
*     .. Executable Statements ..
      WRITE (NOUT,*) 'D01DAF Example Program Results'
      YA = 0.0D0
      YB = 1.0D0
      ABSACC = 1.0D-6
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'First formulation'
      IFAIL = 1
*
      CALL D01DAF(YA,YB,P1,P2A,FA,ABSACC,ANS,NPTS,IFAIL)
*
      WRITE (NOUT,99999) 'Integral =', ANS
      WRITE (NOUT,99998) 'Number of function evaluations =', NPTS
```

```
      IF (IFAIL.GT.0) WRITE (NOUT,99997) 'IFAIL = ', IFAIL
      WRITE (NOUT,*)
      WRITE (NOUT,*) 'Second formulation'
      IFAIL = 1
*
      CALL D01DAF(YA,YB,P1,P2B,FB,ABSACC,ANS,NPTS,IFAIL)
*
      WRITE (NOUT,99999) 'Integral =', ANS
      WRITE (NOUT,99998) 'Number of function evaluations =', NPTS
      IF (IFAIL.GT.0) WRITE (NOUT,99997) 'IFAIL = ', IFAIL
      STOP
*
99999 FORMAT (1X,A,F9.4)
99998 FORMAT (1X,A,I5)
99997 FORMAT (1X,A,I2)
      END
*
      DOUBLE PRECISION FUNCTION P1(Y)
*     .. Scalar Arguments ..
      DOUBLE PRECISION             Y
*     .. Executable Statements ..
      P1 = 0.0D0
      RETURN
      END
*
      DOUBLE PRECISION FUNCTION P2A(Y)
*     .. Scalar Arguments ..
      DOUBLE PRECISION              Y
*     .. Intrinsic Functions ..
      INTRINSIC                     SQRT
*     .. Executable Statements ..
      P2A = SQRT(1.0D0-Y*Y)
      RETURN
      END
*
      DOUBLE PRECISION FUNCTION FA(X,Y)
*     .. Scalar Arguments ..
      DOUBLE PRECISION             X, Y
*     .. Executable Statements ..
      FA = X + Y
      RETURN
      END
*
      DOUBLE PRECISION FUNCTION P2B(Y)
*     .. Scalar Arguments ..
      DOUBLE PRECISION              Y
*     .. External Functions ..
      DOUBLE PRECISION              X01AAF
      EXTERNAL                      X01AAF
*     .. Executable Statements ..
      P2B = 0.5D0*X01AAF(0.0D0)
      RETURN
      END
*
      DOUBLE PRECISION FUNCTION FB(X,Y)
*     .. Scalar Arguments ..
      DOUBLE PRECISION             X, Y
*     .. Intrinsic Functions ..
      INTRINSIC                    COS, SIN
*     .. Executable Statements ..
      FB = Y*Y*(COS(X)+SIN(X))
      RETURN
      END
```

9.2 Program Data

None.

9.3 Program Results

```
D01DAF Example Program Results

First formulation
Integral =   0.6667
Number of function evaluations =  189

Second formulation
Integral =   0.6667
Number of function evaluations =   89
```