D01DAFP

NAG Parallel Library Routine Document

Note: before using this routine, please read the Users' Note for your implementation to check for implementation-dependent details. You are advised to enclose any calls to NAG Parallel Library routines between calls to Z01AAFP and Z01ABFP.

1 Description

D01DAFP attempts to evaluate the double integral

$$I = \int_a^b \int_{\phi_1(y)}^{\phi_2(y)} f(x,y)\,dx\,dy \tag{1}$$

to a specified absolute accuracy by repeated applications of the method described by Patterson [1]. In (1), a and b are constants and $\phi_1(y)$ and $\phi_2(y)$ are functions of the variable y. See Section 6.1 for a description of the method.

2 Specification

```
   SUBROUTINE D01DAFP(ICNTXT, A, B, PHI1, PHI2, F, ACCREQ, RESULT,
  1                   ABSERR, NFUN, IFAIL)
   DOUBLE PRECISION   A, B, PHI1, PHI2, F, ACCREQ, RESULT, ABSERR
   INTEGER            ICNTXT, NFUN, IFAIL
   EXTERNAL           PHI1, PHI2, F
```

3 Usage

3.1 Definitions

The following definitions are used in describing the data distribution within this document:

m_p	–	the number of processor rows in the processor grid.
n_p	–	the number of processor columns in the processor grid.
p	–	$m_p \times n_p$, the total number of processors in the Library Grid.

3.2 Global and Local Arguments

The following global **input** arguments must have the same value on entry to the routine on each processor and the global **output** arguments will have the same value on exit from the routine on each processor:

Global input arguments: A, B, ACCREQ, IFAIL

Global output arguments: RESULT, ABSERR, NFUN, IFAIL

The remaining arguments are local.

4 Arguments

1: ICNTXT — INTEGER *Local Input*

On entry: the Library context, usually returned by a call to the Library Grid initialisation routine Z01AAFP.

Note: the value of ICNTXT **must not** be changed.

2: A — DOUBLE PRECISION *Global Input*

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

3: B — DOUBLE PRECISION *Global Input*

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

4: 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 *Local 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 D01DAFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

5: 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 *Local 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 D01DAFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

6: 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 *Local Input*
2: Y — DOUBLE PRECISION *Local Input*

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

F must be declared as EXTERNAL in the (sub)program from which D01DAFP is called. Arguments denoted as *Input* must **not** be changed by this procedure.

7: ACCREQ — DOUBLE PRECISION *Global Input*

On entry: the absolute accuracy requested. IF ACCREQ is negative, then the absolute value is used.

8: RESULT — DOUBLE PRECISION *Global Output*

On exit: an estimate of the integral, I.

9: ABSERR — DOUBLE PRECISION *Global Output*

On exit: an estimate of the absolute error.

10: NFUN — INTEGER *Global Output*

On exit: the total number of evaluations of function $f(x, y)$ used in computing the integral.

11: IFAIL — INTEGER *Global Input/Global Output*

The NAG Parallel Library provides a mechanism, via the routine Z02EAFP, to reduce the amount of parameter validation performed by this routine. For a full description refer to the Z02 Chapter Introduction.

On entry: IFAIL must be set to 0, −1 or 1. For users not familiar with this argument (described in the Essential Introduction) the recommended values are:

IFAIL = 0, if multigridding is **not** employed;
IFAIL = −1, if multigridding is employed.

On exit: IFAIL = 0 (or −9999 if reduced error checking is enabled) unless the routine detects an error (see Section 5).

5 Errors and Warnings

If on entry IFAIL = 0 or −1, explanatory error messages are output from the root processor (or processor {0,0} when the root processor is not available) on the current error message unit (as defined by X04AAF).

5.1 Full Error Checking Mode Only

IFAIL = −2000

The routine has been called with an invalid value of ICNTXT on one or more processors.

IFAIL = −1000

The logical processor grid and library mechanism (Library Grid) have not been correctly defined, see Z01AAFP.

IFAIL = $-i$

On entry, the ith (global) argument did not have the same value on all logical processors (see Section 3.2).

5.2 Any Error Checking Mode

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 RESULT may still contain an approximate estimate of the integral.

IFAIL = 10×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 RESULT may still contain an approximate estimate of the integral, but its reliability will decrease as IFAIL increases.

IFAIL = 10×N+1

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

6 Further Comments

6.1 Algorithmic Detail

This routine 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

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

Both the outer integral I and the inner integrals $F(y)$ are evaluated by the method, described by Patterson [1] and [2], 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.

6.2 Parallelism Detail

This routine is based on the master/slave paradigm. The inner integrals and outer integral are evaluated by the slave and master processors respectively. When an inner integral satisfies the requested accuracy on a slave processor, it requests more work from the master.

Because of the master/slave nature of the algorithm, no speed-up is achieved by running this routine on two processors. The more expensive the integrand the better the performance of D01DAFP scales with the number of processors.

6.3 Accuracy

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 integral, 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 integrand over the positive quadrant of the unit circle, R,

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

This may be presented to D01DAFP as

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

but here the outer integral has an induced square-root singularity stemming from the way the region has been presented to D01DAFP. This situation should be avoided by re-casting the problem. For the example given, the use of polar coordinates would avoid the difficulty:

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

7 References

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

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

8 Example

The following program evaluates the integral

$$I = \int_0^{\pi/2} \int_0^{\pi/2} \sin y \sum_{k=0}^{200} \frac{\sqrt{1 - R(k)^2 \sin^2 x \sin^2 y}}{1 - R(k)^2 \sin^2 y}\, dx\ dy,$$

where $R(k) = 4.99975 \times 10^{-3}k$.

8.1 Example Text

```
*     D01DAFP Example Program Text
*     NAG Parallel Library Release 2. NAG Copyright 1996.
*     .. Parameters ..
      INTEGER          NOUT
      PARAMETER        (NOUT=6)
*     .. Local Scalars ..
      DOUBLE PRECISION A, ABSERR, ACCREQ, B, RESULT
      INTEGER          ICNTXT, IFAIL, MP, NFUN, NP
      LOGICAL          ROOT
*     .. External Functions ..
      DOUBLE PRECISION F, PHI1, PHI2, X01AAF
      LOGICAL          Z01ACFP
      EXTERNAL         F, PHI1, PHI2, X01AAF, Z01ACFP
*     .. External Subroutines ..
      EXTERNAL         D01DAFP, Z01AAFP, Z01ABFP
*     .. Executable Statements ..

      ROOT = Z01ACFP()
      IF (ROOT) WRITE (NOUT,*) 'D01DAFP Example Program Results'

      MP = 2
      NP = 2
      IFAIL = 0
*
*     Initialize Library Grid
*
      CALL Z01AAFP(ICNTXT,MP,NP,IFAIL)
*
      A = 0.0D0
      B = 0.5D0*X01AAF(0.0D0)
      ACCREQ = 1.0D-6
      IFAIL = -1
```

```
*
*     Integrate function f(x,y)
*
      CALL D01DAFP(ICNTXT,A,B,PHI1,PHI2,F,ACCREQ,RESULT,ABSERR,NFUN,
     +             IFAIL)
*
      IF (IFAIL.GE.0) THEN
         IF (ROOT) THEN
            WRITE (NOUT,*)
            WRITE (NOUT,99999) 'IFAIL                 =', IFAIL
            WRITE (NOUT,99999) 'Number of tasks       =', MP*NP
            WRITE (NOUT,99998) 'Computed result       =', RESULT
            WRITE (NOUT,99997) 'Computed error        =', ABSERR
            WRITE (NOUT,99999) 'No. of function eval. =', NFUN
         END IF
      END IF

      IFAIL = 0
      CALL Z01ABFP(ICNTXT,'No',IFAIL)

      STOP

99999 FORMAT (1X,A,I12)
99998 FORMAT (1X,A,F12.4)
99997 FORMAT (1X,A,E12.2)
      END

      DOUBLE PRECISION FUNCTION PHI1(Y)
*
*     This function provides the lower limit
*     for the value at y
*
*     .. Scalar Arguments ..
      DOUBLE PRECISION              Y
*     .. Executable Statements ..

      PHI1 = 0.0D0
      RETURN
      END
*
      DOUBLE PRECISION FUNCTION PHI2(Y)
*
*     This function provides the upper limit
*     at the value y
*
*     .. Scalar Arguments ..
      DOUBLE PRECISION              Y
*     .. External Functions ..
      DOUBLE PRECISION              X01AAF
      EXTERNAL                      X01AAF
*     .. Executable Statements ..

      PHI2 = 0.5D0*X01AAF(0.0D0)
      RETURN
      END
*
      DOUBLE PRECISION FUNCTION F(X,Y)
*
```

```
*    This function is the function to be integrated
*
*     .. Scalar Arguments ..
      DOUBLE PRECISION            X, Y
*     .. Local Scalars ..
      DOUBLE PRECISION            A, AA, B, BB, RK, RK2, SUM
      INTEGER                     K
*     .. Intrinsic Functions ..
      INTRINSIC                   DBLE, SIN, SQRT
*     .. Executable Statements ..

      SUM = 0.0D0
      B = SIN(Y)
      A = SIN(X)*B
      AA = A*A
      BB = B*B
      DO 20 K = 0, 200
         RK = 4.99975D-3*DBLE(K)
         RK2 = RK*RK
         SUM = SUM + SQRT(1.0D0-RK2*AA)/(1.0D0-RK2*BB)
   20 CONTINUE
      F = B*SUM
      RETURN
      END
```

8.2 Example Data

None.

8.3 Example Results

```
D01DAFP Example Program Results

IFAIL                 =           0
Number of tasks       =           4
Computed result       =    628.2235
Computed error        =    0.30E-10
No. of function eval. =        8225
```