



1 SUMMARY

To evaluate the **multi-dimensioned integral**

$$f_1 \int_{a_1}^{b_1} f_2(x_1) \int_{a_2(x_1)}^{b_2(x_1)} f_3(x_1, x_2) \cdots \int_{a_m(x_1, x_2, \dots, x_{m-1})}^{b_m(x_1, x_2, \dots, x_{m-1})} f(x_1, x_2, \dots, x_m) dx_m dx_{m-1} \dots dx_1$$

where $1 \leq m \leq 6$. The user must specify the type of quadrature to be used for each dimension, choosing from: Simpson's Rule, Gauss quadrature or a quadrature based on Chebyshev polynomials.

A relative accuracy can be requested and a subroutine to compute the values of the limits and integrand must be provided.

ATTRIBUTES — **Version:** 1.0.0. **Types:** QB01A; QB01AD. **Calls:** LIMITS (a user subroutine). **Original date:** March 1963. **Origin:** F.R.Hopgood, Harwell.

2 HOW TO USE THE PACKAGE

2.1 The argument list

The single precision version

```
CALL QB01A(ERROR, MAXCO, ITYPE, MMAX, F, ICOUNT)
```

The double precision version

```
CALL QB01AD(ERROR, MAXCO, ITYPE, MMAX, F, ICOUNT)
```

ERROR is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to the relative accuracy required in the final result, e.g. the value 0.01 would mean that a 1% error can be tolerated.

The subroutine considers the problem as a number of one dimensional quadratures and each quadrature is normally carried out to the required relative accuracy. Consequently, if there is significant cancellation in the innermost quadrature the required accuracy can be achieved, but if there is cancellation in an outer quadrature this can cause the overall relative accuracy to be less good than that which was asked for.

MAXCO is an INTEGER array of length 6 words which must be set by the user to control the printing. MAXCO(I) is the number of integrands under the I-th integral sign which will be printed out and the first MAXCO(I) integrands which are evaluated will be printed. The value of I and one of the letters: C (Chebyshev), S (Simpson) or G (Gauss) indicating the quadrature type is printed before the integrands with the current value of X(I).

ITYPE is an INTEGER array which must be set by the user to select the integration method to be used at each stage, i.e. ITYPE(I) selects the integration method for the I-th quadrature. The possibilities are:

ITYPE < 0 specifies Chebyshev integration: where a 2^r order Chebyshev polynomial is fitted to $2^r + 1$ points and an error estimate is based on the ratio of the first and last two coefficients in the polynomial. If the required accuracy cannot be achieved with $r \leq 7$ there is a print out.

ITYPE = 0 specifies Simpson's Rule. A rough estimate of the integral value is first obtained so that an estimate of the absolute accuracy may be made. The interval length is halved or doubled as is necessary to obtain the required accuracy with a minimum of work. A table of integrands is stored so that no integrand is evaluated twice.

$0 < \text{ITYPE} = n \leq 16$ specifies an n-point Gauss formula. No error estimate is made in this case because an error estimate requires that approximately $2n$ function values must be evaluated and the method then

has no advantages over the Chebyshev method.

MMAX is an INTEGER variable which must be set by the user to the dimension of the integral, i.e. to the value of m in the definition of the problem given in section 1. **Restriction:** $m \leq 6$.

F is a REAL (DOUBLE PRECISION in the D version) variable which will be set by the subroutine to the value of the integral.

ICOUNT is an INTEGER array of length 6 which is set by the subroutine. On return ICOUNT(I) will give the number of integrands which have been evaluated under the I-th integral sign.

2.2 The subroutine to evaluate integrands and limits

The user must provide a subroutine to evaluate the integrands and limits. The subroutine must take the form:

The single precision version

```
SUBROUTINE LIMITS(M,A,B,G,X,F,FA)
REAL X(6),FA(6)
```

The double precision version

```
SUBROUTINE LIMITS(M,A,B,G,X,F,FA)
DOUBLE PRECISION A,B,G,X(6),F,FA(6)
```

Note that single and double precision versions have the same name.

M is an INTEGER variable set by the caller to m the integration level. It will always be in the range $1 \leq m \leq \text{MMAX}$.

A is a REAL (DOUBLE PRECISION in the D version) variable. If $1 \leq m \leq \text{MMAX}$ it must be set to the value of the lower integration limit $a_m(x_1, x_2, \dots, x_{m-1})$, otherwise it is ignored.

B is a REAL (DOUBLE PRECISION in the D version) variable. If $1 \leq m \leq \text{MMAX}$ it must be set to the value of the upper integration limit $b_m(x_1, x_2, \dots, x_{m-1})$, otherwise it is ignored.

G is a REAL (DOUBLE PRECISION in the D version) variable. If $1 \leq m \leq \text{MMAX}$ it must be set to the constant to be added to the integral at the current level

$$\int_{a_m}^{b_m} f_{m+1} \left\{ \int \dots \right\} dx_{m+1}$$

Normally G would be set to zero but the case may occur when an integral, from α to y say, has a subrange, α to β which can be evaluated analytically. In this case G could be set to the analytic integral and the limits would then be set to β and y .

X is a REAL (DOUBLE PRECISION in the D version) array which will contain the current values of x_1, x_2, \dots, x_{m-1} .

F is a REAL (DOUBLE PRECISION in the D version) variable. If $m = \text{MMAX} + 1$ the subroutine must set F to the value of the innermost integrand $f(x_1, x_2, \dots, x_m)$.

FA is a REAL (DOUBLE PRECISION in the D version) array of length 6. If $1 \leq m \leq \text{MMAX}$ the subroutine must set F(M) to the value of the integrand $f_m(x_1, x_2, \dots, x_{m-1})$, otherwise it is not to be set.

3 GENERAL INFORMATION

Workspace: None.

Use of common: None.

Other routines called directly: calls a user provided subroutine called LIMITS (both single and double versions have the same name).

Input/output: printing occurs on unit 6, see argument MAXCO.

Restrictions: integration dimension limited to 6.

4 METHOD

Three integration methods are used: Chebyshev, Simpson's and Gauss, see argument `ITYPE` for some more detail.