1 SUMMARY

To **evaluate the integral** $\int_a^b f(x)\,dx$ to a **specified relative or absolute accuracy.**

The subroutine uses an adaptive scheme based on Romberg extrapolation and Trapezoidal rule, and is described under the name CADRE in C. de Boor, ‘CADRE: an algorithm for numerical quadrature’ in ‘Mathematical Software’ Ed. J.R. Rice, Academic Press.

The user must supply a subroutine to evaluate $f(x)$ $a \le x \le b$ and the accuracy required in the integral value. The subroutine returns an estimated upper bound (i.e. sum of absolute values of estimates on sub-intervals) of the actual error; this may be pessimistic in some cases, since no allowance is made for cancellation of errors. The subroutine also attempts to identify any singularities and discontinuities in $f(x)$.

ATTRIBUTES — Version: 1.0.0. **Remark:** Powerful and efficient, but rather large in storage requirement. **Types:** QA05A; QA05AD. Both use double precision arithmetic. **Calls:** FD05. **Original date:** October 1972. **Origin:** Subroutine CADRE, C. de Boor, Purdue Univ., modified for Harwell by A.R.Curtis.

2 HOW TO USE THE PACKAGE

2.1 The argument list

The single precision version

```
CALL QA05A(VAL,FUNC,A,B,AERR,RERR,LEVEL,ERROR,IFLAG)
```

The double precision version

```
CALL QA05AD(VAL,FUNC,A,B,AERR,RERR,LEVEL,ERROR,IFLAG)
```

VAL is a REAL (DOUBLE PRECISION in the D version) variable which the subroutine returns the value of the integral.

FUNC is the name of a user-supplied function subprogram defining $f(t)$ (see §2.2).

A is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to a the lower limit of integration. It is not altered by the subroutine.

B is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to b the upper limit of integration. It is not altered by the subroutine.

AERR is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to the require absolute accuracy in the integral value. Note that the subroutine works to an accuracy requirement based on the maximum of AERR and RERR*ABS(VAL).

RERR is a REAL (DOUBLE PRECISION in the D version) variable which must be set by the user to the require relative accuracy in the integral value, see AERR.

LEVEL is an INTEGR variable which must be set by the user to specify what printed output (if any) is required (see §2.4). LEVEL=1 specifies no output.

ERROR is a REAL (DOUBLE PRECISION in the D version) variable set by the subroutine to an estimated upper bound on the error achieved.

IFLAG is an INTEGER variable set by the subroutine to indicate what difficulties were encountered, if any (see §2.4). Values of IFLAG up to 3 indicate that VAL should be accepted as the result (subject to value of ERROR); the result may well be good even if IFLAG is 4 or 5.

2.2 The function subprogram

The user must supply a `FUNCTION` subprogram to evaluate $f(t)$ for any t in the range $[a,b]$, specifying its name as the second argument and declaring the name `EXTERNAL.` For `QA05AD` this subprogram must return a double precision result and accept a double precision argument. Its only argument is the value of t.

2.3 Labelled common.

This subroutine uses one common block.

The single precision version

```
COMMON/QA05D/IPR
```

The double precision version

```
COMMON/QA05DD/IPR
```

`IPR` is an `INTEGER` variable set to the stream number for printing. It is default to the value 6.

2.4 Printing and diagnostics

The argument `LEVEL` may be given values from 1 to 5, interpreted as follows:

1 No printout from `QA05`.

2 Success or failure messages and a list of the singularities encountered (if any).

3 In addition, all subintervals considered are listed, with the kind of behaviour of $f(t)$ found on each.

4 In addition, all information on which internal decisions are made.

5 In addition all extrapolation tables constructed are listed.

Most users are unlikely to need values of `LEVEL` greater than 3.

The result `IFLAG` indicates to the calling program what difficulties were encountered:

1 All is well.

2 One or more singularities were successfully handled,

3 In one or more subintervals, an estimate of the integral was accepted because the error estimate was small, although no regular behaviour of $f(t)$ was recognised.

4 Failure because of insufficient internal workspace (very unlikely).

5 Failure, too small a subinterval required; may be due to noise on $f(t)$ or to a behaviour which the subroutine cannot cope with.

Even with `IFLAG=4` or `5`, the best available estimate of the integral is returned in `VAL`, and this may well be acceptable because of the stringent tests applied.

4 METHOD

See Carl de Boor: “On writing an automatic integration algorithm” and “CADRE: an algorithm for numerical Quadrature” in “Mathematical Software”, Ed. John R.Rice, Academic press (1971). `QA05A` is virtually identical with `CADRE,` which has been recommended to us by Prof. P. Rabinowitz as the best currently available algorithm. It uses sophisticated tests in an attempt to estimate the actual analytical behaviour of $f(t)$ in each subinterval, and to detect the presence of noise on the function values, and is suspicious of apparently very simple integrands.

Note on highly oscillatory integrands

The subroutine generally behaves well on highly oscillatory integrands, but may give trouble when the number of cycles in the range [a,b] is near (but not at) certain integral values. In such cases, it may help to define

$$c = a + \theta(b - a)$$

Where θ is an *irrational* number, e.g. the best computer representation of $2^{-\frac{1}{2}}$ to call `QA05` twice to evaluate the integral with limits a to c and then with limits c to b, then add the results.

5 EXAMPLE OF USE

Code based on the following:

```
EXTERNAL EXP
  -  -
  -  -
  -  -
LEVEL=1
CALL QA05A(V,EXP,0.,1.,1.E-4,1.E-4,LEVEL,E,I)
  -  -
  -  -
```

will cause `V` to be set to

$$\int_0^1 e^t \, dt = e - 1$$

to a relative accuracy 10^{-4} (since the absolute value is greater than unity, the absolute accuracy requested is ineffective). `I` will be set to 1 and `E` will be set to an estimated upper bound on the actual error.