Raabo.

A/S REGNECENTRALEN
SCANDINAVIAN INFORMATION PROCESSING SYSTEMS

SYSTEM LIBRARY

RCSL NO: 53-D48
EDITION: July 1969
AUTHOR : S.E. Christiansen

adapint

KEY-WORDS: RC 4000, Software, adapint, Integration, Algol Procedure, ISO Tape

---

ABSTRACT: The procedure adapint calculates the integral of a function f(x) given in an interval (a,b) by means of 7-point formula and adaptive control of the subdivisions of (a,b) with respect to the desired accuracy.

INFORMATION DEPARTMENT
DK-2500 VALBY · BJERREGAARDSVEJ 5 · PHONE: (01) 46 08 88 · TELEX: 64 64 rcinf dk · CABLES: INFOCENTRALEN

real procedure adapint(a, b, f, x, delta, order)

1. Function and parameters.

Call parameters:

a, b	real values. The endpoints of the interval over which the integration is carried out. It is allowed to have b < a.
delta	real value. The permitted error relative to Ia.

Return parameter:

adapint	real procedure. The approximation of the integral of f(x) obtained by the procedure.

Other parameters:

f	real. The function f(x) given as an expression in x. f and x are used as 'Jensen parameters'.
x	real. The independent variable used in the expression f(x). x need not be initialized. Upon exit x = sign(e-Ia×delta)×e, where e is an estimate of the abs error. So x > 0 indicates a failure and x <= 0 a success.

2. Method.

The real procedure adapint calculates the integral of a function f(x) from a to b within a prescribed accuracy given by the parameter: delta. This is achieved by making further subdivisions of those subintervals, where the error is too large - and only of those. These subdivisions are stopped when the desired accuracy is obtained or when the number of subdivisions reaches its permitted upper bound. In all cases the procedure delivers on exit an approximation to the integral and an indication of success or failure.

The procedure is particularly useful when the function f(x) exhibits an almost singular behaviour within the interval (a, b) like 1/sqrt(x+1₁₀-6) over (0, 1). etc. In such cases it is almost always possible to get through by a proper choice of the governing parameter: delta.

The method uses a 7-point formula, and all subdivisions are nondestructive (i.e. all function evaluations are used). The successive subdivisions are carried out so that the squares of the estimates of the absolute error in all subintervals finally obtained are uniformly distributed. The calculation is considered successful if e <= Ia × delta, where e = abs(the estimate of the absolute error) and Ia is the integral of abs f over (a, b). Ia is computed by the procedure but not delivered as return value. Upon exit x is assigned the value sign(e-Ia×delta)×e. So a success is indicated by x = 0 or x = -e<0, and a failure by x = e>0. Since Ia usually is not known in advance, it is necessary to have a realistic estimate of Ia before running the procedure, so that delta can be properly fixed. This estimate may be obtained either by an honest guess or by means of the procedure itself. It must be noticed that the estimate of the error (=e) made by the procedure usually is 10 - 100 times as large as the actual error.

Ex. The call adapint(0,5,exp(x),x,$_{10}$-4) gives adapint = 147.4131649 (true value = 147.413159...) and x = -1.2$_{10}$-3, success (actual abs error = 5.8$_{10}$-6).

3. References.

The present algorithm is the result of many experiments made at Regnecentralen during the last years and is not described in the literature. For a similar algorithm, see:

[1] H. O'Hara, and Francis J. Smith: The evaluation of definite integrals by interval subdivision. The Computer Journal, Vol 12.2 (May 1969), p. 179-182.

4. Algol procedure

```
adapint = set 4
adapint = algol index.no message.yes

external
real procedure adapint(a,b,f,x,delta); message adapint version 1.10.69;
value a,b,delta; real a,b,f,x,delta;
```

```
begin array A(1:60); boolean selection,ex; integer p,Sign;
  real e,fe,fa,fb,h,x1,x2,x4,x6,x7,f1,f2,f4,f6,f7,base,r,s,t,
  sa,sb,hmin,sum,eps,dev,dd;
  Sign:=sign(b-a); h:=abs(b-a); hmin:=h/18200; selection:=true;
  eps:=(6615/192×delta)××2/(if h=0 then 1 else h); sum:=dd:=0;
  x:=x4:=(a+b)/2;f4:=f; x:=a;fa:=f;x:=e:=b;fe:=fb:=f;
  x:=x2:=(a+x4)/2;f2:=f;x:=x6:=(b+x4)/2;f6:=f;
  s:=abs(4×f4+fa+fb)×2; base:=s×h; p:=-3; goto TEST;
  STORE:
  if abs(3×fb-8×f7+6×f6-f4)<abs(3×fa-8×f1+6×f2-f4) then
  begin
    A(p):=b;A(p+1):=fb;A(p+2):=f7;A(p+3):=f6;
    b:=e:=a;fb:=fe:=fa;a:=x4;fa:=f4;
    x6:=x1;f6:=f1;x4:=x2;f4:=f2;s:=2×sa
  end
    else
  STORE2:
  begin
    A(p):=a;A(p+1):=fa;A(p+2):=f1;A(p+3):=f2;
    a:=x4;fa:=f4;x4:=x6;f4:=f6;x6:=x7;f6:=f7;s:=2×sb
  end;
  x:=x2:=(a+x4)/2;f2:=f;
  TEST:
  x:=x1:=(a+x2)/2;f1:=f;x:=x7:=(b+x6)/2;f7:=f;h:=abs(b-a);
  sa:=abs(4×f2+fa+f4);sb:=abs(4×f6+fb+f4);base:=(sa-s+sb)×h+base;
  r:=f2+f6;s:=f1+f7;t:=fa+fb;
  dev:=(84×r-64×s+15×t-70×f4)××2×h; ex:=dev>base××2×eps;
  if (h-hmin)×₁₀9>abs x4∧p<56∧ex then
  begin
    p:=p+4; goto if selection then STORE else STORE2
  end;
  sum:=sum+(2016×r+2048×s+549×t+4004×f4)×h;
  dd:=dev×h+dd; if ex then eps:=(16×eps+dev/base××2)/4;
  if p>0 then
  begin
    selection:=false;
    t:=A(p); if (t-a)×(a-e)<0 then
```

```
        begin
          b:=e; fb:=fe; e:=a; fe:=fa
        end
          else
        begin
          b:=a;fb:=fa
        end;
        a:=t; fa:=A(p+1); f2:=A(p+2); f4:=A(p+3);
        x4:=(a+b)/2; x:=x6:=(x4+b)/2; f6:=f; x2:=(a+x4)/2;
        s:=abs(4×f4+fa+fb)×2; p:=p-4; goto TEST
      end from STORE;
      adapint:=Sign×sum/13230; dd:=16/6615×sqrt(dd);
      x:=if dd>delta×base/12 then dd else -dd
    end adapint;

    end
```