Ivo Hrubec

Algoritmy. 9. UNIVERZAL. Univerzální integrační formule. 10. Romint. Rombergova metoda.

Aplikace matematiky, Vol. 12 (1967), No. 5, 415–418

Persistent URL: `http://dml.cz/dmlcz/103119`

ALGORITMY

9. UNIVERZAL

UNIVERZÁLNÍ INTEGRAČNÍ FORMULE

IVO HRUBEC, Matematický ústav ČSAV, Žitná 25, Praha 1

Určitý integrál

$$\int_a^b f(x)\,\mathrm{d}x$$

je transformován na

$$\frac{b-a}{2}\int_{-\pi/2}^{\pi/2} f\left(\frac{b-a}{2}\sin\varphi + \frac{b+a}{2}\right)\cos\varphi\,\mathrm{d}\varphi\,,$$

jenž je aproximován univerzální formulí:

$$I_n = \frac{b-a}{4n}\sum_{j=0}^{n} d_{jn}\left(f\left(\frac{b+a}{2} + \frac{b-a}{2}\sin\frac{\pi j}{2n}\right) + f\left(\frac{b+a}{2} - \frac{b-a}{2}\sin\frac{\pi j}{2n}\right)\right),$$

kde

$$d_{jn} = (2 - \delta_{0j} - \delta_{jn})\left(1 + 2\sum_{k=1}^{n}\frac{(-1)^k}{1-4k^2}\cos\frac{\pi jk}{n}\right), \quad \delta_{ik}\text{-Kroneckerovo } \delta.$$

V pracech [1], [2] je ukázáno, že tato formule je asymptoticky optimální pro širokou třídu Hilbertových prostorů 2π-periodických funkcí.

real procedure *Univerzal* (a, b, f, eps, ok);
value a, b, eps;
real a, b, eps; **Boolean** *ok*; **real procedure** f;
begin comment procedura provádí výpočet integrálu funkce $f(x)$ v mezích od a do b. Postupně se počítají aproximace z 16, 32, 48, ..., 512 dělících bodů. Je-li dvakrát po sobě relativní rozdíl sousedních aproximací menší než *eps*, výpočet skončí a proměnná *ok* nabude hodnoty **true**. Jestliže nebylo dosaženo žádané přesnosti *eps*, je *ok* **false**. Maximální počet vyvolání procedury f je asi 34000. V závislosti na rychlosti počítače je možno seznam cyklu proměnné n nahradit jiným seznamem, jenž vytvoří rostoucí posloupnost přirozených čísel n;

```
integer i, j, n; Boolean bo;
real p, c, s, u, v, h, q, qq, ff;
b: = (b − a)/2; a: = a + b; qq: = 0;
ok: = bo: = true;
 for n: = 8 step 8 until 512 do
 begin q: = 0;
        for j: = 0 step 1 until n do
        begin h: = 1.5707963267 × j/n;
              s: = sin(h); c: = cos(h); ff: = f(a + b × s) + f(a − b × s);
              s: = 2 × s × c; c: = 2 × c × c − 1; u: = v: = 0; h: = 2;
              for i: = if n > 128 then 128 else n step −1 until 0 do
              begin p: = u×s+v×c; u: = u×c−v×s+h/(4×i×i−1);
                    v: = p; h: = −h
              end i;
              q: = q+ff×(2+h×(u−h))×(if j=0 ∨ j=n then 0·5 else 1)
        end j;
        q: = q/n;
        if abs(q − qq) > eps × abs(q) then bo: = true else
        if bo then bo: = false else goto Z;
        qq: = q
 end n;
 ok: = false;
Z: Univerzal: = q × b/2
end Univerzal;
```

Procedura byla vyzkoušena v ELLIOTT 4100 ALGOLu, převedena do ICT FORTRANu a provedeny rozsáhlejší výpočty; např. pro $\int_{-1}^{1} e^{5x}\,dx$ maximální přesnosti 10^{-20} bylo dosaženo již při 32 vyvoláních integrované funkce.

Literatura

[1] *I. Babuška:* Über optimale Formeln zur numerischen Berechnung linearer Funktionale. Aplikace matematiky 10, 441—443.

[2] *I. Babuška:* Über optimale Quadraturformeln im Raume periodischer Funktionen. Aplikace matematiky 11, 259—265.

[3] *J. Kautský:* Program INTBAB (nepublikováno).

10. ROMINT

ROMBERGOVA METODA

IVO HRUBEC, Matematický ústav ČSAV, Žitná 25, Praha 1

Pro výpočet $\int_a^b f(x)\,dx$ se nejprve určí hodnoty $T(h_j)$ $(j = 0, 1, 2, ..., m)$ počítané lichoběžníkovým pravidlem s pevným krokem $h_j = (b - a) \times 2^{-j}$. Body $(h_j, T(h_j))$ proložíme interpolační polynom $T_m'(h)$ a hodnotu $T_m'(0)$ pokládáme za aproximaci $T(0)$ Účinnost metody je zaručena za předpokladu, že $T(h)$ lze vyjádřit ve tvaru $\sum_{i=0}^{\infty} a_i h^{2i}$.

real procedure *Romint* (*a*, *b*, *f*, *eps*, *max*);
value *a*, *b*, *eps*; **integer** *max*;
real *a*, *b*, *eps*; **real procedure** *f*;
begin comment procedura provádí výpočet integrálu funkce $f(x)$ v mezích od *a* do *b*. Parametr *max* udává při vstupu do procedury maximální řád extrapolace. Bude-li dvakrát po sobě splněna nerovnost $abs(t[i] - t[i-1]) \leqq eps \times \times abs(t[i])$, výpočet skončí a *max* nabude hodnoty *i*. Počet vyvolání procedury *f* je pak $2^{\max} + 1$;

```
    integer i, j; real p, q, d;
    Boolean bo; array t[1: max + 1];
    real procedure suma (k);
    value k; integer k;
    begin integer i, n; real s; own real h, tt;
        if k = 0 then
        begin h: = b − a; tt: = 0; s: = (f(a) + f(b))/2
        end else
        begin n: = 2↑k; h: = h/2; s: = 0;
            for i: = 1 step 2 until n do
            s: = s + f(a + i × h)
        end;
        tt: = suma: = s × h + tt/2
    end lichoběžníkové pravidlo;
    q: = t[1]: = suma(0); bo: = false;
    for i: = 1 step 1 until max do
    begin p: = t[i + 1]: = suma (i); d: = 1;
        for j: = i step −1 until 1 do
        begin d: = 4×d; p: = t[j]: = p+(p−t[j])/(d − 1) end;
```

```
q: = abs(p-q); if q > eps × abs(p) then bo: = false else
if ¬ bo then bo: = true else begin max: = i; go to Z end;
q: = p
end i;
Z: Romint: = p
end Romint;
```

Algoritmus je upraven tak, aby výměnou procedury *suma* bylo možno použít proceduru *Romint* pro řešení diferenciálních rovnic [1], k numerickému derivování, atd. V tomto případě má procedura pro určení první derivace v bodě b tvar:

```
real procedure suma(k); value k; integer k;
begin real h; h: = .5↑k × a; suma: = (f(b + h) - f(b - h))/h/2
end derivace;
```

Algoritmus byl vyzkoušen v ICT ALGOLu na příkladě $\int_{-1}^{1} e^x \, dx$. Přesné řešení je 2·3504023873; výsledky jsou uvedeny v tabulce:

eps	*max*	*Romint*
0.1	3	2.3504024941
0.001	4	2.3504023873
0.00001	5	2.3504023873

Literatura

[1] *F. L. Bauer, H. Rutishauser, E. Stiefel:* New Aspects in Numerical Quadrature, Proc. of Symp. in Appl. Math. 15 (1963), str. 199—218.

[2] *F. L. Bauer:* Algorithm 60 — Romberg Integration, Communications of ACM 3 (1961), str. 255.