FINANČNÍ INŽENÝRSTVÍ
KBP FFÚ & KSTP FIS

7. Numerical evaluation of definite integrals

Tento učební text byl podpořen z Operačního programu Praha - Adaptabilita

Hana Hladíková

OPPA
PRAHA PRAGUE PRAGA PRAG
EVROPSKÁ UNIE
Evropský sociální fond Praha & EU: Investujeme do vaší budoucnosti

Numerical approximation of definite integral is called **numerical quadrature**, the formulas are called

quadrature formulas. We approximate the value of an indefinite integral $I(f) = \int_b^a f(x)\mathrm{d}x$, where f is continuous on the interval $\langle a, b\rangle$. If, moreover, f is smooth (its first derivative is continuous), then *geometrical interpretation* of $I(f)$ is the area between the graph of f and x-axis on the interval $\langle a, b\rangle$.

Numerical methods are used if

a) the function is given by a tablet,
b) the integral cannot be evaluated directly,
c) the evaluation is too complicated.

Substitute the given function by φ so that the integral of $I(\varphi) = \int_b^a \varphi(x)\mathrm{d}x$ can be evaluated more easily. Let $I(f)$ be the exact value and $I(\varphi)$ its approximation, i.e. $I(f) \approx I(\varphi)$.

Numerical evaluation of definite intagral is stable (unlike numerical derivative). If φ is an approximation of the function f on $\langle a; b\rangle$, then $I(\varphi)$ is an approximation of $I(f)$, because:

$$\left|\int_b^a f(x)\mathrm{d}x - \int_b^a \varphi(x)\mathrm{d}x\right| \leq \int_b^a |f(x) - \varphi(x)|\,\mathrm{d}x \leq (b-a)\overbrace{\sup_{\langle a,b\rangle}(|f(x) - \varphi(x)|}^{\epsilon} \leq (b-a)\epsilon$$

Newton-Cotes quadrature formulas

The given interval $\langle a, b\rangle$ will be divided into n subintervals $\langle x_k, x_{k+1}\rangle$, $a = x_0 < x_1 < x_2 < \cdots < x_{n-1} < x_n = b$. We take *equidistant partition*, where $x_{k+1} - x_k = h = \frac{b-a}{n}$.

The integral is partitioned:

$$I = \int_a^b f(x)\mathrm{d}x = \sum_{k=0}^{n-1}\left(\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x\right).$$

On the interval $\langle x_k, x_{k+N+1}\rangle$ substitute the function f by the Lagrange polynomial N-th degree and evaluate the integral of it. Such a formula is called **basic Newton-Cotes formula** - (basic NC formula).

The sum for the integral $\langle a, b\rangle$ is **composed quadrature formula**.

We call the formula **quadrature formula of the order** n if it approximates exactly polynomial of degree n (for hiher degrees then n we get a nonzero error).

Newton-Cotes quadrature formula - rectangle rule

On the interval $\langle x_k, x_{k+1}\rangle$ the function f will be substituted by a *constant function* $L_0(x) = f\left(\frac{x_{k+1}+x_k}{2}\right)$ going through the points $[\frac{x_{k+1}+x_k}{2}, f\left(\frac{x_{k+1}+x_k}{2}\right)]$. We get **quadrature formula of order** 0.

$$\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x \approx \int_{x_k}^{x_{k+1}} L_0(x)\mathrm{d}x = (x_{k+1} - x_k) f\left(\frac{x_{k+1} + x_k}{2}\right) = h.f\left(\frac{x_{k+1} + x_k}{2}\right) = I_{Rec}(f).$$

For $f\left(\frac{x_{k+1}+x_k}{2}\right) > 0$ the value $I_{Rec}(f)$ is the *area of the rectangle* with sides

$x_{k+1} - x_k$ a $f\left(\frac{x_{k+1}+x_k}{2}\right)$. See Fig. 7.1, where $x_{k+\frac{1}{2}} = \frac{x_{k+1}+x_k}{2}$.

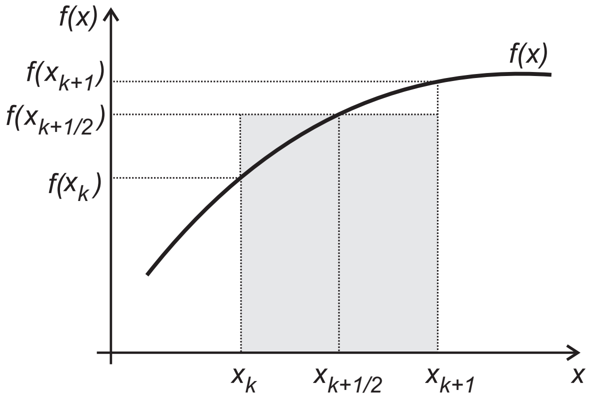

Fig. 7.1 Rectangular rule.

Composed formula:

$$\int_a^b f(x)\mathrm{d}x \approx I_{Rec} = h\sum_{i=0}^{n-1} f\left(\frac{x_{i+1}+x_i}{2}\right)$$

Remark: Let f have on $\langle a, b\rangle$ continuous second derivative, then:

$$I - I_{Rec} = \frac{b-a}{24} f^{''}(\xi)h^2, \text{ where } \xi \in (a, b).$$

If $M_2 = max_{x\in\langle a,b\rangle} \left|f^{''}(x)\right|$, then

$$|I - I_{Obd}| \leq \frac{b-a}{24} M_2 h^2.$$

Newton-Cotes quadrature formula - trapezoid rule

On $\langle x_k, x_{k+1}\rangle$ the function f is substituted by a *linear polynomial* $L_1(x)$ going through the points $[x_k, f(x_k)]$ and $[x_{k+1}, f(x_{k+1})]$. We get **quadrature formula of order** 1

$$\int_{x_k}^{x_{k+1}} f(x)\mathrm{d}x \approx \int_{x_k}^{x_{k+1}} L_1(x)\mathrm{d}x = \frac{(x_{k+1}-x_k)}{2}\left(f(x_k) + f(x_{k+1})\right) =$$

$$= \frac{h}{2}\left(f(x_k) + f(x_{k+1})\right) = I_{Trap}.$$

For $f(x_{k+1}) > 0$ a $f(x_k) > 0$ is $I_{Trap}(f)$ the area of the trapezoid with the sides of the lengths $f(x_{k+1})$ and $f(x_k)$ and the height is $x_{k+1} - x_k$.

Geometrical interpretation is in the Fig. 7.2.

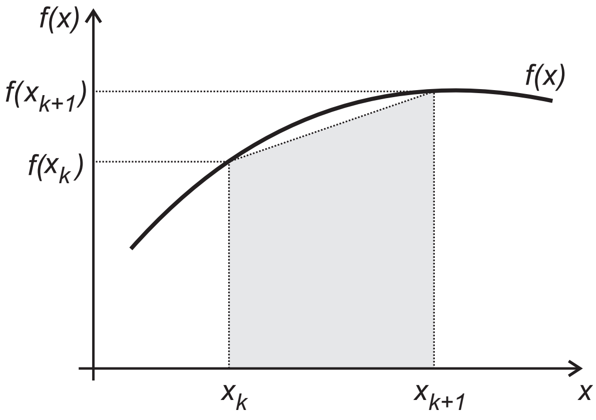

Fig. 7.2 Trapezoid rule .

The composed formula:

$$\int_a^b f(x)\mathrm{d}x \approx I_{Trap} = \frac{h}{2}\left[f(x_0) + 2f(x_1) + 2f(x_2) + \ldots + 2f(x_{n-1}) + f(x_n)\right] =$$

$$= h\left[\frac{1}{2}f(x_0) + \sum_{i=1}^{n-1} f(x_k) + \frac{1}{2}f(x_n)\right].$$

Remark: If on $\langle a, b\rangle$ the function f has a continuous second derivative, then:

$$I - I_{Trap} = -\frac{b-a}{12}f^{''}(\xi)h^2, \text{ kde } \ \xi \in (a, b).$$

If $M_2 = max_{x\in\langle a,b\rangle}\left|f^{''}(x)\right|$, then

$$|I - I_{Lich}| \leq \frac{b-a}{12}M_2h^2.$$

Newton-Cotes quadrature formula - Simpson's rule

On $\langle x_k, x_{k+2}\rangle$ the function f will be substituted by a *quadratic polynomial* $L_2(x)$going through the points $[x_k, f(x_k)]$, $[x_{k+1}, f(x_{k+1})]$ and $[x_{k+2}, f(x_{k+2})]$. We get the **quadrature formula of order** 2.

$$\int_{x_k}^{x_{k+2}} f(x)\mathrm{d}x \approx \int_{x_k}^{x_{k+2}} L_2(x)\mathrm{d}x = \frac{(x_{k+2} - x_k)}{6}\left(f(x_k) + 4f(x_{k+1}) + f(x_{k+2})\right) =$$

$$= \frac{h}{3}\left(f(x_k) + 4f(x_{k+1}) + (x_{k+2})\right) = I_{Simp}.$$

The geometrical interpretation is on the Fig. 7.1.

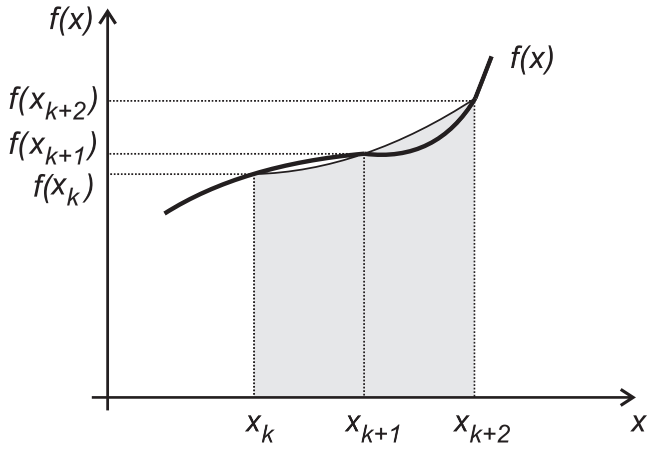

Fig. 7.3 Simpson's rule .

The composed formula for n *even*:

$$\int_a^b f(x)\mathrm{d}x \approx I_{Simp} = \frac{h}{3}[f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \ldots$$

$$\ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)].$$

Remark: If on $\langle a, b\rangle$ the function f has a continuous forth derivative, then:

$$I - I_{Simps} = -\frac{b-a}{90} f^{(4)}(\xi)h^4, \text{ where } \xi \in (a, b).$$

If $M_4 = max_{x\in\langle a,b\rangle} \left|f^{(4)}(x)\right|$, then

$$|I - I_{Simps}| \le \frac{b-a}{90} M_4 h^4.$$

Example 1: Evaluate the approximate value $\int_{-1}^{1} \mathrm{e}^x \mathrm{d}x$ for $n = 4$ using the foregoing rules , estimate the error and compare it to the exact value. $h = \frac{1-(-1)}{4} = 0,5 \Rightarrow x_0 = -1;\ x_1 = -1 + 0,5 = -0,5;\ x_2 = 0;\ x_3 = 0,5;\ x_4 = 1.$

$f(x) = f^{(1)}(x) = f^{(2)}(x) = f^{(4)}(x) = \mathrm{e}^x,\ \mathrm{e}^x < 3,\ x \in \langle -1, 1\rangle$

Exact value:

$$I = \int_{-1}^{1} \mathrm{e}^x \mathrm{d}x = [\mathrm{e}^x]_{-1}^{1} = \mathrm{e}^1 - \mathrm{e}^{-1} = 2,350402387$$

(1) Rectangle rule:

$$I_{Rec} = 0,5\left[\mathrm{e}^{\frac{-1+(-0,5)}{2}} + \mathrm{e}^{\frac{-0,5+0}{2}} + \mathrm{e}^{\frac{0+0,5}{2}} + \mathrm{e}^{\frac{0,5+1}{2}}\right] = 2,326096$$

Error estimation: $I - I_{Obd} \le \frac{b-a}{24} M_2 h^2 \le \frac{2}{24} \cdot 3 \cdot 0,5^2 = 0,0625$

$$|I_{Rec} - I| = 0,024306$$

(2) Trapezoid rule:

$$I_{Trap} = \frac{0,5}{2}\left[\mathrm{e}^{-1} + 2\mathrm{e}^{-0,5} + 2\mathrm{e}^{0} + 2\mathrm{e}^{0,5} + \mathrm{e}^{1}\right] = 2,399166$$

Error estimation: $I - I_{Lich} \leq \frac{b-a}{12} M_2 h^2 \leq \frac{2}{12} \cdot 3 \cdot 0,5^2 = 0,125$

$$|I_{Trap} - I| = 0,048764$$

(3) Simpson's rule:

$$I_{Simps} = \frac{0,5}{3}\left[\mathrm{e}^{-1} + 4\mathrm{e}^{-0,5} + 2\mathrm{e}^{0} + 4\mathrm{e}^{0,5} + \mathrm{e}^{1}\right] = 2,351195$$

Error estimation: $I - I_{Simps} \leq \frac{b-a}{90} M_4 h^4 \leq \frac{2}{90} \cdot 3 \cdot 0,5^4 = 0,0083$

$$|I_{Simps} - I| = 0,0000792$$

MATLAB

trapz(@ fce,a,b) - integral by trapezoid rule of fce on $\langle a, b\rangle$,
quad(@ fce,a,b) - integr al by Simpson's rule of fce on $\langle a, b\rangle$,
dblquad(@ fce,xmin,xmax,ymin,ymax) - dvojný integrál,
triplequad(@ fce,xmin,xmax,ymin,ymax,zmin,zmax) - trojný integrál.

Evaluate $\int_0^2 x^2 \mathrm{d}x$
We can insert a function using the command inline:

```
>> F = inline('(x.^2)')
>> Q = quad(F,0,2);
```

We can define a function:

```
>> Q = quad(@ mojefunkce,0,2);
```

where mojefunkce.m is an M-file:

```
function y = mojefunkce(x)
y = x.^2;
```

Symbolic toolbox: `>> syms x`

```
>> int(x^2,0,2)
ans =
4
```

For finding the antiderivative for a function use **int** with one parameter: **int(expression)**.

```
>> syms x
>> int(x^2-5)
ans =
1/3*x^3-5*x
```
