8. Skaitinis integravimas ir diferencijavimas

Sprendžiant inžinerinius ir mokslo uždavinius, skaitinio integravimo ir diferencijavimo veiksmai atliekami gana dažnai. Pavyzdžiui, skaitinio integravimo prireikia tada, kai integralo neįmanoma apskaičiuoti tiksliai arba kai pointegralinė funkcija pateikta reikšmių lentele. Analizuojant inžinerinius duomenis, dažnai tenka skaičiuoti liestinės krypties koeficientą kuriame nors taške, todėl taikomi skaitinio diferencijavimo metodai.

8.1. Skaitinio integravimo metodai

Aptarsime apibrėžtinių integralų apskaičiavimo uždavinį.

Uždavinio formulavimas. *Duotas integralas* $R=\int_a^b f(x)dx$ *; čia* a *ir* b *— integravimo rėžiai, o* $f(x)$ *— pointegralinė funkcija, pateikta arba analiziškai, arba reikšmių lentele. Reikia apskaičiuoti R reikšmę tikslumu* ε.

Visų skaitinio integravimo metodų esmė yra ta, kad pointegralinė funkcija $f(x)$ keičiama aproksimuojančiąja funkcija $F(x)$ ir laikoma, jog $R=\int_a^b f(x)dx \approx \int_a^b F(x)dx$. Paprastai parenkama tokia aproksimuojančioji funkcija $F(x)$, kad integralas $\int F(x)dx$ būtų lengvai integruojamas analiziškai.

Istoriškai plačiausiai naudojama aproksimuojančioji funkcija $F(x)$ yra n-tosios eilės interpoliacinis polinomas, einantis per taškus $(x_i; f(x_i))$, $i=\overline{0,n}$. Keičiant interpoliavimo mazgų skaičių, t.y. keičiant n reikšmę, bei įvairiai parenkant pačius mazgus x_i, gaunamos skirtingos skaitinio integravimo formulės, kurios vadinamos kvadratūrinėmis formulėmis.

Jei interpoliavimo mazgai pasirenkami laisvai, tai turime Niutono ir Koteso kvadratūrines formules.

Jei interpoliavimo mazgai apskaičiuojami taip, kad kvadratūrinė formulė tiksliai integruotų galimai aukštesnės eilės polinomus, tai turime Gauso tipo kvadratūrines formules.

Jei integravimo rėžiai a ir b priklauso interpoliavimo mazgams, tai turime uždaro tipo kvadratūrinę formulę, priešingu atveju, - atviro tipo kvadratūrinę formulę.

Pastaruoju metu pointegralinė funkcija dažnai keičiama splainu, paprastai kubiniu, ir gaunamos kvadratūrinės splainų formulės.

Integralas $\int_a^b F(x)dx$, čia $F(x)$ — pointegralinę funkciją $f(x)$ aproksimuojanti funkcija, apskaičiuojamas tiksliai pagal Niutono ir Leibnico formulę. Šio veiksmo rezultatas yra pointegralinės funkcijos reikšmių $f(x_i)$ $(i=\overline{0,n})$ tiesinis darinys.

Prieš pradėdami nagrinėti konkrečias kvadratūrines formules, trumpai aptarsime jų taikymo strategiją. Literatūroje, aprašančioje skaičiavimo metodus, kvadratūrinės formulės labai dažnai pateikiamos su konkrečia jų taikymo strategija. Pavyzdžiui, norint apskaičiuoti $\int_a^b f(x)dx$ Simpsono metodu, kvadratūrinė formulė išvedama taip. Pirmiausia intervalas $[a;b]$ taškais $a = x_0, x_1, \ldots, x_{N-1}, x_N = b$ padalijamas į $N = 2m$ **lygių** dalių ir kiekvienoje jų $[x_{i-1}; x_{i+1}]$ taikoma Simpsono kvadratūrinė formulė

$$h/3\left(y_{i-1} + 4y_i + y_{i+1}\right),$$

kurios $h = (b-a)/N$, o $y_i = f(x_i)$. Tada visi rezultatai sumuojami ir gaunama apibendrinta Simpsono kvadratūrinė formulė:

$$h/3\left(y_0 + 4y_1 + 2y_2 + 4y_3 + \ldots + 2y_{N-2} + 4y_{N-1} + y_N\right).$$

Matome, kad ji susijusi su integravimo strategija: integravimo intervalas dalijamas į N ***lygių*** dalių.

Jei integravimo intervale yra sričių, kuriose pointegralinė funkcija $f(x)$ kinta sparčiau nei kitose integravimo intervalo srityse, tai taikyti tokią strategiją būtų neprotinga. Šiuo atveju, norint apskaičiuoti integralą tam tikru tikslumu, reikėtų visame integravimo intervale imti tokį integravimo žingsnį *h*, kuris garantuotų reikiamą tikslumą srityse, kuriose funkcija kinta sparčiausiai, nepaisant to, kad srityse, kuriose funkcija kinta lėčiau, reikiamas tikslumas gali būti pasiektas esant didesniam integravimo žingsniui. Vadinasi, taikant šią strategiją, bus naudojama per daug pointegralinės funkcijos reikšmių, todėl padidės integravimo paklaidos.

Paprastai norimą integravimo tikslumą stengiamasi pasiekti imant kuo mažiau pointegralinės funkcijos reikšmių. Todėl pastaruoju metu taikoma adaptyviojo integravimo strategija.

Atsižvelgdami į šias pastabas, kvadratūrines formules toliau nagrinėsime nesiedami jų su minėtomis integravimo strategijomis, o integravimo strategijų algoritmus aptarsime atskirai.

8.1.1. Niutono ir Koteso kvadratūrinės formulės

Kaip jau minėta, Niutono ir Koteso kvadratūrinės formulės yra interpoliacinio tipo, kai pointegralinė funkcija $f(x)$ keičiama interpoliaciniu polinomu, kurį nusako iš anksto fiksuoti vienodai nutolę vienas nuo kito taškai x_i.

Tarkime, kad reikia apskaičiuoti integralą $R = \int_a^b f(x)dx$. Tada *n*-tosios eilės Niutono ir Koteso kvadratūrinė formulė išvedama taip:

1) integravimo intervalas $[a;b]$ *taškais* $a = x_0, x_1, \ldots, x_{n-1}, x_n = b$ *dalijamas į n lygių dalių; čia* $x_i = x_0 + ih$, $i = \overline{0, n}$, *o* $h = (b-a)/n$ *;*

2) pointegralinė funkcija $f(x)$ *keičiama n-tosios eilės interpoliaciniu polinomu* $F(x)$, *einančiu per taškus* $(x_i; y_i)$, $i = \overline{0, n}$ *; čia* $y_i = f(x_i)$*;*

3) $$R = \int_a^b f(x)dx \approx \int_a^b F(x)dx = \sum_{i=0}^{n} w_i y_i . \qquad (8.1)$$

Svorinius koeficientus w_i patogu apskaičiuoti Hemingo metodu.

(8.1) formulė turi tiksliai integruoti funkcijas $1, x, x^2, \ldots, x^k, \ldots, x^n$. Tada, simboliu m_k pažymėję $\int_a^b x^k dx$ (literatūroje m_k paprastai vadinamas integravimo operatoriaus k-tuoju momentu), iš (8.1) formulės suformuosime lygčių sistemą

$$\begin{cases} w_0 + w_1 + \ldots + w_n = m_0, \\ w_0 x_0 + w_1 x_1 + \ldots + w_n x_n = m_1, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ w_0 x_0^k + w_1 x_1^k + \ldots + w_n x_n^k = m_k, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ w_0 x_0^n + w_1 x_1^n + \ldots + w_n x_n^n = m_n. \end{cases} \qquad (8.2)$$

(8.2) sistemos matricos A determinantas yra Vandermondo determinantas, t. y.

$$\det(A) = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_0 & x_1 & \ldots & x_n \\ \ldots & \ldots & \ldots & \ldots \\ x_0^n & x_1^n & \ldots & x_n^n \end{bmatrix}.$$

Jis nelygus nuliui, jei $x_i \neq x_j$. Vadinasi, (8.2) lygčių sistema yra suderinta ir turi vienintelį sprendinį. R. V. Hemingas siūlo šią sistemą spręsti atvirkštinės matricos metodu ir pateikia paprastą analizinį jos apskaičiavimo metodą. Jį čia ir aptarsime.

Apibrėžkime polinomą

$$\omega_i(x) = (x - x_0)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n). \qquad (8.3)$$

Sudauginę ir sutraukę panašiuosius narius, turime:

$$\omega_i(x) = \sum_{k=0}^{n} c_k x^k, \quad c_n = 1.$$

Kadangi $\omega_i(x_j) = 0$, kai $j \neq i$, ir $\omega_i(x_i) \neq 0$, tai matricos A^{-1} i-toji eilutė bus $\frac{1}{\omega_i(x_i)}(c_0, c_1, \ldots, c_n)$. Padauginę A^{-1} i-tąją eilutę iš A j-tojo stulpelio, turime:

$$\frac{\sum_{k=0}^{n} c_k x_j^k}{\omega_i(x_i)} = \frac{\omega_i(x_j)}{\omega_i(x_i)} = \begin{cases} 1, \text{ kai } j = i, \\ 0, \text{ kai } j \neq i. \end{cases}$$

Vadinasi,

$$w_i = \frac{1}{\omega_i(x_i)} \sum_{k=0}^{n} c_k m_k. \qquad (8.4)$$

(8.4) formulė nusako koeficiento w_i išraišką, priklausančią nuo taškų x_i ir momentų m_k. Ši formulė ypač patogi, kai taškai x_i nesudaro aritmetinės progresijos, o taip dažnai būna tada, kai funkcija $f(x)$ pateikta lentele.

1 pavyzdys. Tarkime, kad turime taškus $x_0 = -1, x_1 = 1$ ir $x_2 = 2$. Hemingo metodu apskaičiuokime koeficientus w_0, w_1 ir w_2, su kuriais formulė $w_0 y_0 + w_1 y_1 + w_2 y_2$ tiksliai integruoja polinomus iki antrosios eilės imtinai.

Apskaičiuojame polinomų $\omega_i(x)$ reikšmes, kai $i = \overline{0,2}$, ir momentus m_0, m_1, m_2:

$$\omega_0(x) = (x-1)(x-2) = x^2 - 3x + 2, \quad \omega_0(-1) = 6,$$
$$\omega_1(x) = (x+1)(x-2) = x^2 - x - 2, \quad \omega_1(1) = -2,$$
$$\omega_2(x) = (x+1)(x-1) = x^2 - 1, \quad \omega_2(2) = 3;$$
$$m_0 = \int_{-1}^{2} dx = 3, \quad m_1 = \int_{-1}^{2} x dx = \frac{3}{2}, \quad m_2 = \int_{-1}^{2} x^2 dx = 3.$$

Pagal (4) formulę apskaičiuojame koeficientus w_0, w_1 ir w_2:

$$w_0 = \frac{1}{6}\left(2 \cdot 3 - 3 \cdot \frac{3}{2} + 1 \cdot 3\right) = \frac{3}{4},$$
$$w_1 = -\frac{1}{2}\left(-2 \cdot 3 - 1 \cdot \frac{3}{2} + 1 \cdot 3\right) = \frac{9}{4},$$
$$w_2 = \frac{1}{3}\left(3 \cdot (-1) + 0 \cdot \frac{3}{2} + 1 \cdot 3\right) = 0.$$

Vadinasi, formulė $q = \frac{3}{4} y_0 + \frac{9}{4} y_1$, esant taškams –1, 1, 2, tiksliai integruoja polinomus iki antrosios eilės imtinai.

Išvesdami Niutono ir Koteso kvadratūrines formules, $\int_a^b f(x)dx$ pertvarkysime taip, kad svoriniai koeficientai w_i priklausytų tik nuo *n*.

Pavartoję keitinį $z = x - a$, turime:

$$\int_a^b f(x)dx = \int_0^{b-a} f(z+a)dz = \int_0^{nh} f(z+a)dz.$$

Pažymėkime: $z = ht$. Tada

$$\begin{aligned} \int_a^b f(x)dx &= \int_0^{nh} f(z+a)dz = h\int_0^n f(a+ht)dt = \\ &= Ah\big(B_0 f(a) + B_1 f(a+h) + \ldots + \\ &\quad + B_i f(a+ih) + \ldots + B_n f(b)\big) + K_n; \end{aligned} \tag{8.5}$$

čia $A \cdot B_i = w_i$, o K_n — kvadratūrinės formulės liekamasis narys.

Apskaičiuoti koeficientus w_i bus paprasčiau, jei (8.5) formulėje imsime

$$a = \begin{cases} -\frac{n}{2}h, & \text{kai } n \text{ — lyginis skaičius,} \\ -\frac{(n-1)}{2}h, & \text{kai } n \text{ — nelyginis skaičius} \end{cases}$$

ir $h = 1$. Tada iš (8.5) formulės gausime:

$$\int_a^{a+n} f(x)dx = A\big(B_0 f(a) + \ldots + B_i f(a+i) + \ldots + B_n f(a+n)\big). \quad (8.6)$$

Taikant (8.6) formulę, Hemingo metodu patogu apskaičiuoti koeficientus A ir B_i.

2 pavyzdys. Apskaičiuokime Simpsono kvadratūrinės formulės koeficientus. Šiuo atveju $n = 2$ ir taškai yra –1, 0, 1.

Pirmiausia apskaičiuojame momentus m_0, m_1 ir m_2:

$$m_0 = \int_{-1}^{1} dx = 2, \; m_1 = \int_{-1}^{1} x dx = 0, \; m_2 = \int_{-1}^{1} x^2 dx = \frac{2}{3}.$$

Paskui randame polinomų $\omega_0(x)$, $\omega_1(x)$ ir $\omega_2(x)$ reikšmes:

$$\omega_0(x) = x(x-1) = x^2 - x, \; \omega_0(-1) = 2,$$

$$\omega_1(x) = (x+1)(x-1) = x^2 - 1, \; \omega_1(0) = -1,$$

$$\omega_2(x) = (x+1)x = x^2 + x, \; \omega_2(1) = 2.$$

Tada pagal (4) formulę apskaičiuojame koeficientus w_0, w_1 ir w_2:

$$w_0 = \frac{1}{2}\left(0 \cdot 2 + (-1) \cdot 0 + 1 \cdot \frac{2}{3}\right) = \frac{1}{3},$$

$$w_1 = -\frac{1}{1}\left(-1 \cdot 2 + 0 \cdot 0 + 1 \cdot \frac{2}{3}\right) = \frac{4}{3},$$

$$w_2 = \frac{1}{2}\left(0 \cdot 2 + 1 \cdot 0 + 1 \cdot \frac{2}{3}\right) = \frac{1}{3}.$$

Vadinasi, taikydami Simpsono metodą, turime tokius koeficientus A ir B_i: $A = 1/3$, $B_0 = 1$, $B_1 = 4$, $B_2 = 1$.

Niutono ir Koteso kvadratūrinių formulių liekamuosius narius patogu apskaičiuoti remiantis interpoliacinio polinomo liekamojo nario išraiška. Kadangi praktiniu požiūriu kvadratūrinės formulės liekamojo nario analizinės išraiškos gavimas yra mažiau svarbus nei liekamojo nario reikšmės įvertinimas, tai čia kvadratūrinių formulių liekamųjų narių išvedimo nenagrinėsime.

Toliau pateikta Niutono ir Koteso kvadratūrinių formulių ir jų paklaidų lentelė. Joje $x \in [a; b]$, o k_8, k_9 ir k_{10} — koeficientai.

Pavyzdžiui,

$$k_8 = \frac{-606208}{33 * 10!},$$

o

$$k_{10} = \frac{-538540000}{273 * 12!}.$$

8.1 lentelė. Niutono ir Koteso kvadratūrinių formulių bei jų liekamųjų narių lentelė integralui

$$\int_a^b f(x)\,dx = Ah\big(B_0 f(a) + B_1 f(a+h) + \ldots + B_i f(a+ih) + \ldots + B_n f(b)\big) + K_n$$

apskaičiuoti

n	A	B_0	B_1	B_2	B_3	B_4	B_5	K_n
1	1/2	1	1					$-(1/12)h^3 f^{(2)}(x)$
2	1/3	1	4	1				$-(1/90)h^5 f^{(4)}(x)$
3	3/8	1	3	3	1			$-(3/80)h^5 f^{(4)}(x)$
4	2/45	7	32	12	32	7		$-(8/945)h^7 f^{(6)}(x)$
5	5/288	19	75	50	50	75	19	$-(275/12096)h^7 f^{(6)}(x)$
6	1/140	41	216	27	272	27	216	$-(9/1400)h^9 f^{(8)}(x)$
7	7/17280	751	3577	1323	2989	2989	1323	$-(8183/518400)h^9 f^{(8)}(x)$
8	4/14175	989	5888	–928	10496	–4540	10496	$k_8 h^{11} f^{(10)}(x)$
9	9/89600	2857	15741	1080	19344	5778	5778	$k_9 h^{11} f^{(10)}(x)$
10	5/299376	16067	106300	–48525	272400	–260550	427368	$k_{10} h^{13} f^{(12)}(x)$

Pirmosios eilės Niutono ir Koteso formulė ($n = 1$) vadinama trapecijų formule, o antrosios eilės ($n = 2$) — Simpsono formule.

Kokias išvadas galima padaryti iš 8.1 lentelės?

1. Niutono ir Koteso kvadratūrinių formulių koeficientai w_i yra simetriški centro atžvilgiu, t. y. $w_i = w_{n-i}$ $\left(i = 0, \left[\frac{n}{2}\right]\right)$, ir jų suma lygi n.

2. Lyginės eilės Niutono ir Koteso kvadratūrinės formulės tiksliai integruoja visus polinomus iki (n + 1)-osios eilės imtinai, tuo tarpu nelyginės eilės kvadratūrinės formulės — tik iki n-tosios eilės imtinai. Ši lyginės eilės formulių savybė atsiranda dėl to, kad Niutono ir Koteso kvadratūrinėse formulėse naudojami vienodai nutolę vienas nuo kito taškai. Pavyzdžiui, Simpsono kvadratūrinė formulė tiksliai integruoja polinomus iki trečiosios eilės imtinai. Tačiau antrosios eilės kvadratūrinė formulė $q = \frac{3}{4}y_0 + \frac{9}{4}y_1$, kuri buvo apskaičiuota su taškais –1, 1, 2, tiksliai integruoja polinomus tik iki antrosios eilės imtinai. Vadinasi, praktiškai reikia naudoti lyginės eilės kvadratūrines formules.

3. Lyginės eilės Niutono ir Koteso kvadratūrinės formulės liekamasis narys yra proporcingas h^{k+3} (čia k — lyginis skaičius) ir į proporcingumo koeficientą kaip daugiklis įeina $(k+2)$-osios eilės pointegralinės funkcijos išvestinė. Nelyginės eilės kvadratūrinės formulės liekamasis narys proporcingas h^{l+2} (čia l — nelyginis skaičius) ir proporcingumo koeficientas turi daugiklį, lygų pointegralinės funkcijos $(l+2)$-ajai išvestinei.

Reikia pabrėžti, kad integravimo intervale taikant kvadratūrinę formulę kelis kartus, suminio liekamojo nario h laipsnis yra vienetu mažesnis. Pavyzdžiui, Simpsono kvadratūrinę formulę integravimo intervale $[a;b]$ taikykime m kartų, t. y. intervalą $[a;b]$ dalykime į $2m$ dalių ir kiekviename intervale $[x_{i-1},x_{i+1}]$ $\left(\text{čia } i=\overline{1,m-1}\right)$ taikykime Simpsono formulę. Tada suminį liekamąjį narį apskaičiuosime pagal formulę

$$K_m = -\frac{1}{90}h^5 f^{(4)}(x)\cdot m = -\frac{1}{90}h^4\frac{b-a}{2m}f^{(4)}(x)\,m = -\frac{1}{180}(b-a)h^4 f^{(4)}(x).$$

Kadangi praktiškai apskaičiuojant integralus, kvadratūrinės formulės integravimo intervale taikomos daug kartų, tai, norėdami įvertinti metodo paklaidą, h laipsnį imsime vienetu mažesnį nei 8.1 lentelėje.

8.1.2. Metodo paklaidos įvertinimas

Kaip nustatyti metodo paklaidą? 8.1 lentelėje pateiktos liekamųjų narių išraiškos turi daugiau teorinę vertę, nes, taikant jas, reikia įvertinti pointegralinės funkcijos aukštesnės eilės išvestines. Todėl praktikoje integravimo metodo paklaida dažniausiai apskaičiuojama Ričardsono ir Rombergo metodu.

Tarkime, kad kvadratūrinės formulės metodo paklaida proporcinga h^p ir proporcingumo koeficientas turi daugiklį, lygų pointegralinės funkcijos p-tosios eilės išvestinei. Simboliais R_h ir $R_{h/2}$ pažymėkime integralo R reikšmes, kai naudojamas atitinkamai integravimo žingsnis h ir $h/2$. Tada, laikydami, kad pointegralinės funkcijos p-tosios eilės išvestinė integravimo intervale yra pastovi, galime užrašyti:

$$\begin{cases} R = R_h + Ch^p, \\ R = R_{h/2} + C(h/2)^p. \end{cases}$$

Iš šios lygčių sistemos apskaičiuojame $C(h/2)^p$:

$$C(h/2)^p = \frac{R_{h/2}-R_h}{2^p-1}. \qquad (8.7)$$

(8.7) formulė literatūroje vadinama ***Rungės taisykle.***

Remdamiesi (8.7) formule, turėsime:

$$R = R_{h/2} + \frac{R_{h/2}-R_h}{2^p-1}. \qquad (8.8)$$

Vadinasi, jei pointegralinės funkcijos p-tosios eilės išvestinė integravimo intervale yra pastovi, iš (8.7) formulės gaunama tiksli metodo paklaida, priešingu atveju $\left|\dfrac{R_{h/2}-R_h}{2^p-1}\right|$ įvertina integravimo metodo paklaidą. Taigi (8.8) kvadratūrinės formulės eilė yra vienetu didesnė nei kvadratūrinės formulės, pagal kurią apskaičiavome R_h ir $R_{h/2}$.

Pavyzdžiui, Simpsono metodu galime tiksliai apskaičiuoti , reikšmę integralo, kurio pointegralinė funkcija yra 4-osios eilės polinomas.

Išnagrinėkime pavyzdį, t. y. Simpsono metodu tiksliai apskaičiuokime

$$\int_0^4 \left(5x^4-16x^3+1\right)dx.$$

Tiksli šio integralo reikšmė lygi 4.

Dabar apskaičiuokime jo reikšmę pagal (8.8) formulę.

Pirmiausia Simpsono kvadratūrinę formulę pritaikome visam integravimo intervalui [0; 4]. Turime mazgus: $x_0 = 0; x_1 = 2; x_2 = 4$ ir *h=2.* Apskaičiavus turėsime:

$$R_h = \frac{h}{3}\left(f(0) + 4f(2) + f(4)\right) = \frac{2}{3}\left(1 + 4(-47) + 257\right) = \frac{140}{3}.$$

Dabar intervalą [0; 4] padalysime į keturias lygias dalis mazgais: $x_0 = 0; x_1 = 1; x_2 = 2; x_3 = 3; x_4 = 4$ ir Simpsono formulę taikysime intervalams: [0; 2] ir [2; 4]. Šiuo atveju *h=1*. Turėsime:

$$R_{h/2} = \frac{h}{3}\left(f(0) + 4f(1) + 2f(2) + 4f(3) + f(4)\right) =$$

$$= \frac{1}{3}\left(1 + 4(-10) + 2(-47) + 4(-26) + 257\right) = \frac{20}{3}.$$

Pritaikę (8.8) formulę, turėsime:

$$R = R_{h/2} + \frac{R_{h/2} - R_h}{2^4 - 1} = \frac{20}{3} + \frac{\frac{20}{3} - \frac{140}{3}}{15} = \frac{20}{3} - \frac{8}{3} = 4.$$

Kaip ir tikėjomės, gautoji reikšmė yra tiksli integralo reikšmė.

***Įdėtųjų formulių metodas* (*nulio aisyklė*) .** Pagrindinis Rungės taisyklės trūkumas yra tas, kad integralo reikšmę reikia apskaičiuoti du kartus: su žingsniu h ir su žingsniu $h/2$. Todėl pastaruoju metu integravimo paklaidos įvertinimui naudojamas toliau aptariamas įdėtųjų formulių metodas.

Tarkime, kad Q_{n1} yra *d* eilės inerpoliacinė kvadratūrinė formulė, sukonstruota duotiems *n*+1 interpoliavimo mazgams, o Q_{n2} - žemesnės eilės nei *d* kvadratūrinė formulė, sukonstruota, pašalinus vieną arba kelis minėtus interpoliavimo mazgus. Simboliu *Qf* pažymėkime kvadratūrinės formulės *Q* skaitinio integravimo rezultatą, integruojant funkciją $f(x)$. Tada skirtumą

$$Ef = \left|Q_{n1}f - Q_{n2}f\right|,$$

galima laikyti kvadratūrinės formulės Q_{n1} paklaida.

Apibrėžimas. *Formulė* $Nf = \sum_{i=1}^{n} u_i f(x_i)$, *kurioje bent vienas koeficientas* u_i *nelygus nuliui ir* $\sum_{i=1}^{n} u_i = 0$, *vadinama nulio taisykle.*

Sakoma, kad nulio taisyklė turi eile d , jei $Nf = 0$ *visiems polinomams f, kurių laipsnis nedidesnis nei d ir* $Nf \neq 0$, *kai polinomo f laipsnis didesnis už d.*

Nulio taisyklė N ir interpoliacinė kvadratūrinė formulė Q, nusakyta tais pačiais interpoliavimo mazgais, yra ekvivalenčios (equally strong) , jei $\|N\|_2 = \|Q\|_2$, *čia*

$$\|N\|_2 = \left(\sum_{i=1}^{n} u_i^2\right)^{\frac{1}{2}}, \ \|Q\|_2 = \left(\sum_{i=1}^{n} w_i^2\right)^{\frac{1}{2}}$$ *ir* w_i *yra formulės Q koeficientai.*

Aišku, kad Ef yra nulio taisyklė (formulė), kurios eilė sutampa su kvadratūrinės formulės Q_{n2} eile.

Pirmasis nulio taisyklę apibrėžė J. N. Lyness [21], o apibendrintą nulio taisyklių teoriją 1991 pateikė J. Berntsen ir T. O. Espelid [19].

8.2 lentelėje patalpinti lyginės eilės Niutono ir Koteso įdėtųjų kvadratūrinių formulių ir nulio taisyklės koeficientai [20].

8.2 lentelėje stulpelis „n“ žymi Niutono ir Koteso kvadratūrinės formulės eilę, stulpelis „p“ – įdėtųjų formulių eiles, stulpeliai „w_i“ žymi formulių koeficientus, o - „Δ“ žymi formulių teorines paklaidas. Eilutės „δ_i“ parodo nulio taisyklių koeficientus.

Pastaba. 8.2 lentelėje patalpintos nulio taisyklės buvo sukonstruotos atmetus antrą iš kairės interpoliavimo mazgą.

8.2. lentelė. Niutono ir Koteso įdėtųjų kvadratūrinių formulių ir nulio taisyklės koeficientai

n	p	w_0	w_1	w_2	w_3	w_4	w_5	Δ
2	2	$\frac{1}{3}$	$\frac{4}{3}$	$\frac{1}{3}$				$\frac{-4}{15}\frac{f^{(4)}(\xi)}{4!}h^5$
	1	1	0	1				$\frac{-4}{3}\frac{f^{(2)}(\xi)}{2!}h^3$
	δ_2	$\frac{-2}{3}$	$\frac{4}{3}$	$\frac{-2}{3}$				
4	4	$\frac{14}{45}$	$\frac{64}{45}$	$\frac{24}{45}$	$\frac{64}{45}$	$\frac{14}{45}$		$\frac{-128}{21}\frac{f^{(6)}(\xi)}{6!}h^7$
	3	$\frac{2}{9}$	$\frac{16}{9}$	0	$\frac{16}{9}$	$\frac{2}{9}$		$\frac{32}{15}\frac{f^{(4)}(\xi)}{4!}h^5$
	δ_4	$\frac{4}{45}$	$\frac{-16}{45}$	$\frac{24}{45}$	$\frac{-16}{45}$	$\frac{4}{45}$		
6	6	$\frac{41}{140}$	$\frac{216}{140}$	$\frac{27}{140}$	$\frac{272}{140}$	$\frac{27}{140}$	$\frac{216}{140}$	$\frac{-1296}{5}\frac{f^{(8)}(\xi)}{8!}h^9$
	5	$\frac{14}{50}$	$\frac{81}{50}$	0	$\frac{110}{50}$	0	$\frac{81}{50}$	$\frac{324}{35}\frac{f^{(6)}(\xi)}{6!}h^7$
	δ_6	$\frac{9}{700}$	$\frac{-54}{700}$	$\frac{135}{700}$	$\frac{-180}{700}$	$\frac{135}{700}$	$\frac{-54}{700}$	
8	8	$\frac{3956}{14175}$	$\frac{23552}{14175}$	$\frac{-3712}{14175}$	$\frac{41984}{14175}$	$\frac{-18160}{14175}$	$\frac{41984}{14175}$	$\frac{-606208}{33}\frac{f^{(10)}(\xi)}{10!}h^{11}$
	7	$\frac{1908}{6615}$	$\frac{10496}{6615}$	0	$\frac{16128}{6615}$	$\frac{-4144}{6615}$	$\frac{16128}{6615}$	$\frac{-118784}{315}\frac{f^{(8)}(\xi)}{8!}h^9$
	δ_8	$\frac{-928}{99225}$	$\frac{7424}{99225}$	$\frac{-25984}{99225}$	$\frac{51968}{99225}$	$\frac{-64960}{99225}$	$\frac{51968}{99225}$	
10	10	$\frac{80335}{299376}$	$\frac{531500}{299376}$	$\frac{-242625}{299376}$	$\frac{1362000}{299376}$	$\frac{-1302750}{299376}$	$\frac{2136840}{299376}$	$\frac{-538540000}{273}\frac{f^{(12)}(\xi)}{12!}h^{13}$
	9	$\frac{11690}{40824}$	$\frac{65125}{40824}$	0	$\frac{97500}{40824}$	$\frac{-23250}{40824}$	$\frac{106110}{40824}$	$\frac{-6470000}{99}\frac{f^{(10)}(\xi)}{10!}h^{11}$
	δ_{10}	$\frac{-16175}{898128}$	$\frac{161750}{898128}$	$\frac{-727875}{898128}$	$\frac{1941000}{898128}$	$\frac{-3396750}{898128}$	$\frac{4076100}{898128}$	

Turėdami nulio formulių koeficientus, paprastai apskaičiuosime kvadratūrinės formulės paklaidą. Pavyzdžiui, ketvirtos eilės (penkių taškų) Niutono ir Koteso kvadratūrinės formulės paklaidos modulis apskaičiuojamas pagal formulę:

$\delta = \left|\left(4y_0 - 16y_1 + 24y_2 - 16y_3 + 4y_4\right)/45\right|$

8.1.3. Integravimo strategijos

Kaip jau buvo minėta, praktikoje tenka spręsti tokį integravimo uždavinį:

apskaičiuoti $\int_a^b f(x)dx$ tikslumu ε.

Integruojant skaitiniu būdu, susiduriama su dviejų rūšių paklaidomis:

1) integravimo metodo paklaidomis;
2) apvalinimo paklaidomis.

Kvadratūrinių formulių integravimo metodo paklaidas ir jų įvertinimą aptarėme 8.1.2 paragrafe. Apvalinimo paklaidos susijusios su pointegralinės funkcijos reikšmių apskaičiavimu. Kuo daugiau šios funkcijos reikšmių naudojama integruojant, tuo didesnė apvalinimo paklaida.

Kaip matyti iš 8.1 ir 8.2 lentelių, integravimo metodo paklaida yra proporcinga h^p; čia h — integravimo žingsnio ilgis, o p — teigiamas sveikasis skaičius. Kaip matome, kad mažinant h , metodo paklaida mažėja, tuo tarpu apvalinimo paklaida, kuri proporcinga $1/h$ (žr. [22]) - didėja. Aišku, kad kiekvienai kvadratūrinei formulei ir pointegralinei funkcijai galima parinkti tokią optimalią integravimo žingsnio reikšmę, su kuria metodo ir apvalinimo paklaidos yra tos pačios eilės. Vadinasi, kai integravimo žingsnio ilgis pasidaro mažesnis už šią reikšmę, apvalinimo paklaida viršija metodo paklaidą ir toliau mažinti integravimo žingsnį nėra prasmės.

Metodo, kuriuo būtų galima iš anksto apskaičiuoti konkrečios kvadratūrinės formulės ir pointegralinės funkcijos optimalų integravimo žingsnį, nėra. Todėl iš šio samprotavimo galime daryti išvadą, kad reikia rinktis tokią integravimo strategiją, kuri leistų norimą integravimo tikslumą pasiekti esant mažiausiam pointegralinės funkcijos reikšmių skaičiui.

Iš pradžių aptarsime ne itin racionalią, tačiau dar plačiai taikomą strategiją, kai visame integravimo intervale naudojamas to paties ilgio integravimo žingsnis, paskui — kur kas racionalesnę adaptyviojo integravimo strategiją.

Vienodo žingsnio visame integravimo intervale strategija. Tarkime, kad norime apskaičiuoti $R = \int_a^b f(x)dx$ reikšmę tikslumu ε, integravimui taikydami n-os eilės ($(n+1)$ taško) kvadratūrinę formulę Q_{n+1}. Tada vienodo žingsnio integravimo strategijos k-sis žingsnis aprašomas taip.

k-sis žingsnis, k=0,1,2,…, iki bus pasiektas norimas tikslumas

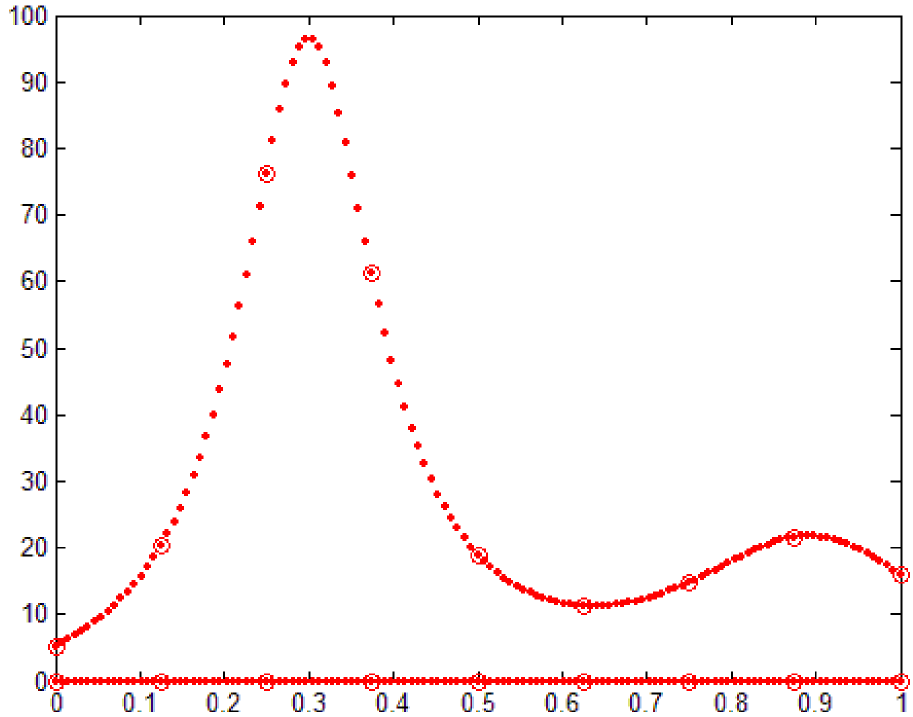
1. *Integravimo intervalą (a,b) dalome į 2^k vienodo ilgio subintervalų.*
2. *Kiekvienam subintervalui pritaikome kvadratūrinę formulę Q_{n+1},t.y. kiekviename subintervale apskaičiuojame apibrėžtinio integralo reikšmę.*
3. *Gautas reikšmes sumuojame ir turime apytikslę integralo reikšmę $R_{\frac{h}{2}}$.*

4. *Pasinaudodami* $(k-1)$ *žingsnyje gautu rezultatu* R_h, *pagal Rungės taisyklę apskaičiuojame integravimo paklaidą.*
5. *Jei gautoji paklaida mažesnė už norimą tikslumą, tai pabaiga; priešingu atveju pereiname prie* $(k+1)$ *žingsnio.*

Žemiau pateikta funkcija ***niutcot9***, kuri, taikydama devynių taškų Niutono ir Koteso formulę bei vienodo žingsnio integravimo strategiją, apskaičiuoja integralo $R = \int_a^b f(x)dx$ reikšmę nurodytu tikslumu.

```
function [r,er,nn]=niutcot9(fun,a,b,ar);
% NIUTCOT9 funkcija apskaičiuoja apibrėžtinį integralą tikslumu „ar“.
% Naudojama 9-ių taškų Niutono ir Koteso kvadratūrinė formulė
% ir vienodo žingsnio visame integravimo intervale strategija.
% k-me žingsnyje apskaičiuotos pointegralinės funkcijos
% reikšmės (k+1)-me žingsnyje iš naujo neskaičiuojamos.
% Paklaida įvertinama pagal Rungės taisyklę.
% Įėjimo parametrai
% fun - pointegralinės funkcijos vardas,
% a,b - integravimo rėžiai,
% ar  - norimas integralo tikslumas.
% Išėjimo parametrai
% r   - apskaičiuota integralo reikšmė,
% er  - faktinė paklaida, apskaičiuota pagal Rungės taisyklę,
% nn  - sunaudotas pointegralinės funkcijos reikšmių skaičius.

n=8; % intervalu skaičius kvadratūrinėje formulėje
kmax=2048; % maksimalus subintervalų skaičius
% Atminties išskirimas
mp=n*kmax+1; x=zeros(1,mp); xp=zeros(1,mp);
fx=zeros(1,mp); fxp=zeros(1,mp); fxpp=zeros(1,mp);
if a == b
    r=0; return
end;
if a < b
    sgr=1;
else
    z=a; a=b; b=z; sgr=-1
end;
%9-ių taškų Niutono ir Koteso kvadratūrinės formulės koeficientai
c=14175; w=[3956,23552,-3712,41984,-18160];
w=[w w(end-1:-1:1)]; w=w/c;
k=1; % k - subintervalų skaičius
t=true; % jei t==true,tai integravimo tikslumas nepasiektas
kf=w; % apibendrintos kvadratūrinės formulės koeficientų masyvas
x=linspace(a,b,n+1); % interpoliavimo mazgai
fx=feval(fun,x); % pointegralines funkcijos reikšmės mazguose x
h=(b-a)/n; % integravimo žingsnio ilgis
rh=sum(w.*fx)*h; % integralo reikšmė r(h)
plot(x,fx,'ro')
nz=numel(x); yzero=zeros(1,nz);
hold on; plot(x,yzero,'ro')
while t & (k <= kmax)
    k=2*k; h=(b-a)/(n*k); w=[w(1:end-1) 2*w(end) w(2:end)];
```

```
        x=linspace(a,b,n*k+1); nz=numel(x); yzero=zeros(1,nz);
        xp=x(2:2:end); fxp=feval(fun,xp);% papildomos f-jos reikšmės
        m=n*k+1; fxpp(1:2:m)=fx; fxpp(2:2:m)=fxp; fx(1:m)=fxpp(1:m);
        rh2=sum(w.*fx)*h; % integralo reikšmė r(h/2)
        delta=abs(rh2-rh)/1023;% paklaidos įvertis (Rungės taisyklė)
        hold on; plot(x,fx,'r.')
        hold on; plot(x,yzero,'r.')
        % Ar norimas tikslumas pasiektas?
        if delta <= ar
            t=false;
        else
            rh=rh2;
        end
    end;
    % Baigiamasis žingsnis
    if k <= kmax
        r=rh2+(rh2-rh2)/1023; er=delta; nn=n*k+1;
    else
        disp('prie duoto kmax norimo tikslumo nepasiekėme')
    end
```



8.1 Kuprų funkcijos integravimas, taikant vienodo žingsnio integravimo strategiją

8.1 paveiksle pavaizduotas integravimo intervalo skaidymas skaičiuojant

$$\int_0^1 \left(\left((x-0{,}3)^2 + 0{,}01 \right)^{-1} + \left((x-0{,}9)^2 + 0{,}04 \right)^{-1} - 6 \right) dx$$

tikslumu $\varepsilon = 10^{-6}$ su funkcija ***niutcot9***.

Pastaba. *Pointegralinė funkcija literatūroje yra žinoma kaip funkcija „humps“ (kuprų funkcija).*

Adaptyviojo integravimo strategija. Adaptyviojo integravimo strategijos idėja labai paprasta: intervalas $[a;b]$ dalijamas į trumpesnius subintervalus ten, kur pointegralinė funkcija kinta sparčiau, ir į ilgesnius subintervalus ten, kur ji kinta lėčiau. Kiekvienam subintervalui taikoma pasirinkta kvadratūrinė formulė, o subintervalas imamas tokio ilgio, kad jame integravimo paklaida būtų ne didesnė už $\frac{H_i}{b-a}\varepsilon$; čia H_i — i-jo subintervalo ilgis. Tada integravimo paklaida visame intervale $[a;b]$ bus ne didesnė kaip $\sum\limits_i \frac{H_i}{b-a}\varepsilon = \frac{\varepsilon}{b-a}\sum\limits_i H_i = \varepsilon$.

1) Aišku, kad ši idėja bus veiksminga tik tada, jei intervalas $[a;b]$ į subintervalus, tenkinančius minėtą reikalavimą, bus automatiškai skaidomas integravimo metu. Integravimo paklaidą patogu apskaičiuoti tiek pagal ***Rungės***, tiek ir pagal ***nulio*** taisykles.

Adaptyviojo integravimo strategija, kai paklaida apskaičiuojama pagal Rungės taisyklę.

1) Apskaičiuojamas integravimo intervalo ilgis $H = b-a$*;*

2) Kvadratūrinė formulė taikoma intervale $[a;b]$ *ir gaunama integralo reikšmė* R_h*; čia h žymi integravimo žingsnį;*

3) Apskaičiuojama $R_{h/2} = R^k_{h/2} + R^d_{h/2}$*; čia* $R^k_{h/2}$ *ir* $R^d_{h/2}$ *— integralo R reikšmės atitinkamai integravimo intervalo kairiajame ir dešiniajame pusintervalyje, kai integravimo žingsnis* $h/2$*;*

4) Jei $\left|\frac{R_{h/2}-R_h}{2^p-1}\right| > \frac{H}{b-a}\varepsilon$*, tai nagrinėjamame intervale integravimo tikslumas yra nepakankamas. Tada dešiniojo pusintervalio integralo reikšmę* $R^d_{h/2}$*, integravimo abscises bei jas atitinkančias pointegralinės funkcijos reikšmes įsimename; integravimo intervalą imame lygų kairiajam pusintervaliui, t. y.* $R_h := R^k_{h/2}$*,* $H = H/2$*, ir grįžtame į 3 punktą.*

Jei $\left|\frac{R_{h/2}-R_h}{2^p-1}\right| \le \frac{H}{b-a}\varepsilon$*, tai nagrinėjamame intervale, kurio ilgis H, integravimo tikslumas yra pasiektas ir* $R_H = R_{h/2} + \frac{R_{h/2}-R_h}{2^p-1}$.

Toliau, jei yra, nagrinėjame dešinįjį pusintervalį, t. y. $R_h := R^d_{h/2}$*, surenkame įsimintas jo integravimo abscises bei tas abscises atitinkančias pointegralinės funkcijos reikšmes, integravimo intervalą imame lygų dešiniajam pusintervaliui, apskaičiuojame intervalo integravimo ilgį H ir grįžtame į 3 punktą.*

Jei nagrinėtų dešiniųjų pusintervalių nėra, tai integralo R reikšmė tikslumu ε *lygi visų* R_H *sumai.*

Adaptyviojo integravimo strategija, kai paklaida apskaičiuojama pagal nulio taisyklę.

1) $R = 0$. *Integravimo_intervalas=* $[a,b]$.
2) *Pagal nulio taisyklę* ***integravimo_intervale*** *apskaičiuojame integravimo paklaidą „**delta**“.*
3) *Jei* ***delta*** **>|** ***integravimo_intervalas*epsilon / (b-a)*****|**, *tai tikslumas nepasiektas. Tada* ***integravimo_ intervalą*** *dalome į dvi lygias dalis;*
   įsimename dešiniojo pusintervalio mazgus ir pointegralinės funkcijos reikšmes tuose mazguose;
   integravimo_intervalas=kairysis pusintervalis ir grįžtame į 2) punktą.

   Priešingu atveju (tikslumas tenkinamas) pagal kvadratūrinę formulę ***integravimo_intervale*** *apskaičiuojame integralo reikšmę* R_d*;*

   $R = R + R_d$*;*

   integravimo_intervalas=paskutinis dešinysis pusintervalis (jei yra) ir grįžtame į antrą punktą; jei tokių pusintervalių nėra, tai pabaiga.

Žemiau pateikta funkcija ***nc8***, kuri, taikydama devynių taškų Niutono ir Koteso formulę bei antrąją adaptyviojo integravimo strategiją, apskaičiuoja integralo $R = \int_a^b f(x)dx$ reikšmę nurodytu tikslumu.

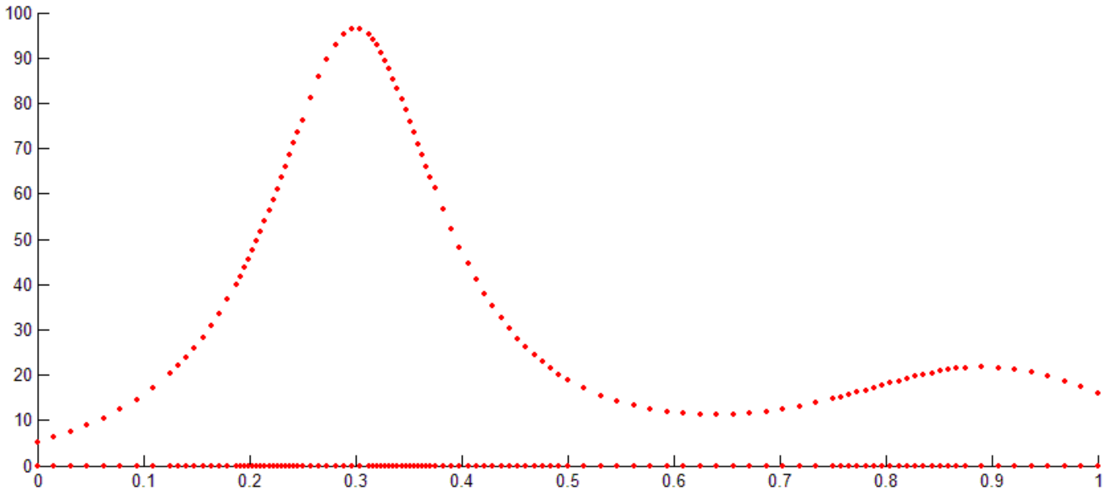
```
function [r,er,flag,nn]=nc8(fun,a,b,ar);
% NCM8 funkcija apskaičiuoja apibrėžtinį integralą.
% Naudojama 9-ių taškų Niutono ir Koteso kvadratūrinė formulė
% ir adaptyviojo integravimo strategija.
% Paklaida apskaičiuojama pagal normuotą nulio taisyklę.
% Įėjimo parametrai
% fun - pointegralinės funkcijos vardas,
% a,b - integravimo rėžiai,
% ar  - norimas integralo tikslumas.
% Išėjimo parametrai
% r    - apskaičiuota integralo reikšmė,
% er   - faktinė paklaida, apskaičiuota pagal Rungės taisyklę,
% flag – parodo keliuose subintervaluose nebuvo pasiektas
%        norimas tikslumas,
% nn   - sunaudotas pointegralinės funkcijos reikšmių skaičius.

n=8; yzero=zeros(1,n+1);
% 9 taškų Niutono ir Koteso kvadratūrinės formulės koeficientai
c=14175; w=[3956,23552,-3712,41984,-18160]; w=[w w(end-1:-1:1)];
w=w/c;
% Paklaidos įverčio koeficientai
cd=928/99225; dw=[1, -8, 28, -56, 70]; dw=[dw dw(end-1:-1:1)];
dw=dw*cd;
% Grubus integralo reikšmės apskaičiavimas
xx=linspace(a,b,n+1); % interpoliavimo mazgai
fx=feval(fun,xx); % pointegralinės funkcijos reikšmės mazguose
h=(b-a)/n; % integravimo žingsnio ilgis
is=sum(w.*fx)*h; % grubi integralo reikšmė
if is==0
    is=b-a
end;
is=abs(is); is=is*ar/eps; noise=is*1e-17; % triukšmo lygis
```

```
dgnr= 4.71256635677993; % normuojantis daugiklis
levmax=80; % maksimalus lygių skaičius
nmax=20000;% maksimalus leistinas funkcijos reikšmių skaičius
%Pradinių reikšmių suteikimas kintamiesiems
flag=0; r=0; er=0; lev=0; poz=1;
if a<=b
    sgn =1;
else
    sgn =-1; t=a; a=b; b=t;
end
if a==b
    nn=0; r=0; return
end
% Pagrindiniai skaičiavimai
x=xx(1:2:end); y=feval(fun,x); nn=5;
while poz ==1
    % Apskaičiuojame papildomus taškus
    xp=(x(1:4)+x(2:5))/2; yp= feval(fun,xp); nn=nn+4;
    step=(x(5)-x(1))/8;
    % Nulio taisyklei reikalingi x ir y masyvai
    xx=zeros(1,n+1); yy=zeros(1,n+1);
    xx(1:2:end)=x; xx(2:2:end-1)=xp; yy(1:2:end)=y; yy(2:2:end-1)=yp;
    % Paklaidos apskaičiavimas
    delta=sum(dw.*yy)*step;
    delta=delta*dgnr; % paklaida normuojama
    if abs(delta)<noise
        delta=0;
    end;
    % Tikslumo sąlygos tikrinimas i-tame intervale
     if(abs(delta) >ar*(xx(9)-xx(1))/(b-a))&(lev<levmax)
    % Tikslumas netenkinamas; Įsiminti dešinės pusės elementus
        lev=lev+1; fsave(lev,:)=yy(5:9); xsave(lev,:)=xx(5:9);
    % Surinkti kairiojo pusintervalio elementus
        x(1:5)=xx(1:5); y(1:5)=yy(1:5);
    else % tikslumas tenkinamas ir lygis neviršija levmax
        hold on; plot(xx,yy,'r.'); hold on; plot(xx,yzero,'r.');
        r=r+sum(yy.*w)*step; er=er+abs(delta);
        if lev>=levmax % lygis viršija levmax
            flag=flag+1;
        end
    % Rasti kitą intervalą
        if lev<=0
            poz=0;
        else
            x=xsave(lev,:); y=fsave(lev,:); lev=lev-1;
        end
    end % if
end % while
% Baigiamosios operacijos
r=r*sgn;
if er~=0
    while (abs(r)+er)==abs(r)
        er=er*2;
    end
end
```

8.2 paveiksle pavaizduotas integravimo intervalo skaidymas skaičiuojant

$$\int_0^1 \left(\left((x-0{,}3)^2 + 0{,}01 \right)^{-1} + \left((x-0{,}9)^2 + 0{,}04 \right)^{-1} - 6 \right) dx$$

tikslumu $\varepsilon = 10^{-7}$ su funkcija ***nc8***.



8.2 Kuprų funkcijos integravimas, taikant adaptyviojo integravimo strategiją

8.2. Gauso kvadratūrinės formulės

8.2.1. Gauso-Ležandro kvadratūrinė formulė

Niutono ir Koteso kvadratūrinėse formulėse taškai x_i, apibrėžiantys pointegralinę funkciją aproksimuojantį polinomą, yra vienodai nutolę vienas nuo kito ir iš anksto fiksuoti, o kvadratūrinė formulė turi išraišką

$$\int_a^b f(x)dx = \sum_{i=1}^{n} w_i f(x_i). \tag{8.9}$$

Šioje formulėje ir toliau šiame paragrafe patogiau nagrinėti kvadratūrines formules, kai taškų indeksas *i* kinta nuo 1 iki *n*.

Atsisakykime iš anksto fiksuotų taškų ir uždavinį formuluokime taip: reikia rasti tokius taškus x_i ir svorinius koeficientus w_i, su kuriais (8.9) formulė tiksliai integruotų kiek galima aukštesnės eilės polinomus.

Kadangi turime 2*n* laisvai kintamų parametrų, tai galime reikalauti, kad (8.9) formulė tiksliai integruotų visus polinomus iki (2*n* – 1)-osios eilės imtinai. Nagrinėjamo uždavinio nesusiaurinsime, vietoj integravimo rėžių *a* ir *b* paėmę rėžius -1 ir 1, taigi galime nagrinėti formulę

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} w_i f(x_i). \tag{8.10}$$

Integralą $\int_a^b f(x)dx$ nesunku paversti integralu $\int_{-1}^{1} \varphi(t)\,dt$, taikant keitinį

$$x = \frac{1}{2}(b-a)t + \frac{1}{2}(b+a). \qquad (8.11)$$

Tada

$$\int_a^b f(x)dx = \frac{1}{2}(b-a)\int_{-1}^{1} f\left(\frac{1}{2}(b-a)t + \frac{1}{2}(b+a)\right)dt. \qquad (8.12)$$

Pirmiausia reikia atsakyti į klausimą „Ar šis uždavinys turi sprendinį ir ar jis yra vienintelis?“ Į šį klausimą atsako žemiau pateiktos teoremos.

1 teorema. (8.10) kvadratūrinė formulė tiksliai integruoja visus polinomus iki ($2n$ – 1)-osios eilės imtinai tada ir tik tada, kai:

1) daugianaris $\omega(x) = (x-x_1)(x-x_2)\ldots(x-x_n)$ yra ortogonalus bet kuriam polinomui $q(x)$, kurio laipsnis $k \le n-1$, t. y.

$$\int_{-1}^{1} \omega(x)\,q(x)\,dx = 0;$$

2) ta formulė yra interpoliacinio tipo kvadratūrinė formulė, t. y. kai

$$w_i = \int_{-1}^{1} \frac{\omega(x)}{(x-x_i)\omega'(x_i)}dx.$$

2 teorema. Egzistuoja vienintelis polinomas $\omega(x)$, tenkinantis 1 teoremos pirmąją sąlygą.

3 teorema. Jei $\omega(x)$ yra n-tojo laipsnio polinomas, tenkinantis 1 teoremos pirmąją sąlygą, tai intervale $[-1;1]$ jis turi n tikrųjų šaknų.

Iš šių teoremų darome išvadą, kad (8.10) Gauso kvadratūrinė formulė egzistuoja ir yra vienintelė. Ši formulė dar vadinama ***Gauso-Ležandro kvadratūrine formule***.

Polinomas, tenkinantis 1 teoremos pirmąją sąlygą, yra Ležandro polinomas $P_n(x)$, kurį patogu reikšti rekurenčiomis formulėmis:

$$P_0(x) = 1,$$
$$P_1(x) = x, \qquad (8.13)$$
$$P_n(x) = \frac{1}{n}\big((2n-1)xP_{n-1}(x) - (n-1)P_{n-2}(x)\big), \quad n \ge 2.$$

Kaip randami Gauso kvadratūrinės formulės taškai x_i ir svoriniai koeficientai w_i? Šiuos parametrus galima apskaičiuoti dviem metodais.

Pirmasis metodas. Taškai x_i sutampa su Ležandro polinomo šaknimis, o svoriniai koeficientai w_i apskaičiuojami pagal formulę

$$w_i = \frac{2}{(1-x_i^2)\big(P_n'(x_i)\big)^2}. \qquad (8.14)$$

1 pavyzdys. Apskaičiuokime Gauso kvadratūrinės formulės taškus bei svorinius koeficientus, kai $n = 3$.

Pirmiausia randame 3-iosios eilės Ležandro polinomą:

$$P_0(x) = 1; \quad P_1(x) = x; \quad P_2(x) = \frac{1}{2}(3x \cdot x - 1) = \frac{3}{2}x^2 - \frac{1}{2},$$

$$P_3(x) = \frac{1}{3}\left(5x\left(\frac{3}{2}x^2 - \frac{1}{2}\right) - 2x\right) = \frac{5}{2}x^3 - \frac{3}{2}x.$$

Vadinasi, ieškomieji taškai yra tokie: $x_1 = -\sqrt{\frac{3}{5}}$, $x_2 = 0$, $x_3 = \sqrt{\frac{3}{5}}$.

Tada $w_1 = w_3 = \dfrac{2}{\left(1 - \frac{3}{5}\right)\left(\frac{15}{2} \cdot \frac{3}{5} - \frac{3}{2}\right)^2} = \dfrac{2}{\frac{2}{5} \cdot 9} = \dfrac{5}{9}$;

$$w_2 = \frac{2}{(1-0)\left(\frac{15}{2} \cdot 0 - \frac{3}{2}\right)^2} = \frac{8}{9}.$$

Antrasis metodas (Hemingo metodas). Šis metodas yra bendresnis ir remiasi tuo, kad (8.10) formulė turi tiksliai integruoti visus polinomus iki $(2n - 1)$-osios eilės imtinai. Tada iš (8.10) formulės gauname lygčių sistemą

$$\begin{cases} w_1 + w_2 + \ldots + w_n = m_0, \\ w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = m_1, \\ w_1 x_1^2 + w_2 x_2^2 + \ldots + w_n x_n^2 = m_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ w_1 x_1^{2n-1} + w_2 x_2^{2n-1} + \ldots + w_n x_n^{2n-1} = m_{2n-1}; \end{cases} \tag{8.15}$$

čia $m_k = \int\limits_{-1}^{1} x^k dx$.

R. V. Hemingas pasiūlė originalų šios sistemos sprendimo metodą. Tarkime, kad žinome taškus x_i, tenkinančius (8.15) sistemą. Apibrėžkime polinomą

$$\pi(x) = \prod_{i=1}^{n}(x - x_i) = \sum_{k=0}^{n} c_k x^k; \text{ čia } c_n = 1.$$

Aišku, kad $\pi(x_i) = 0$, $i = \overline{1, n}$.

Padauginkime (8.15) sistemos pirmąją lygtį iš c_0, antrąją — iš c_1 ir t. t., $(n + 1)$-ąją — iš c_n ir šias lygtis sudėkime. Atsižvelgdami į tai, kad $\pi(x_i) = 0$, gauname:

$$\sum_{k=0}^{n} c_k m_k = 0.$$

Procesą pakartokime pradėdami antrąja (8.15) sistemos lygtimi. Turėsime:

$$\sum_{k=0}^{n} c_k m_{k+1} = 0.$$

Tai pakartoję n kartų, turėsime tiesinių lygčių sistemą

$$\sum_{k=0}^{n} c_k m_{k+j} = 0,\ j = \overline{0, n-1}. \qquad (8.16)$$

Jeigu jos determinantas nelygus nuliui, tai, ją išsprendę, gausime polinomą, kurio šaknys yra Gauso kvadratūrinės formulės taškai. Turėdami juos ir paėmę (8.15) sistemos n pirmųjų lygčių, iš gautos tiesinių lygčių sistemos apskaičiuosime svorinius koeficientus w_i.

Kai (8.16) lygčių sistemos determinantas lygus nuliui, galimas vienas iš dviejų atvejų: arba (8.16) lygčių sistema yra nesuderinta ir kartu nėra tokios kvadratūrinės formulės, arba matricos rangas žemesnis už n. Šiuo atveju prie (8.16) lygčių sistemos prijungiamos papildomos lygtys, kurios gaunamos iš (8.15) sistemos, polinomo laipsnį imant aukštesnį nei $2n-1$. Tai reiškia, kad gaunama tikslesnė kvadratūrinė formulė.

2 pavyzdys. Apskaičiuokime Gauso kvadratūrinę formulę, kai $n = 3$. Pirmiausia randame momentus:

$$m_0 = \int_{-1}^{1} dx = 2;\quad m_1 = \int_{-1}^{1} x dx = 0;\quad m_2 = \int_{-1}^{1} x^2 dx = \frac{2}{3};$$

$$m_3 = \int_{-1}^{1} x^3 dx = 0;\quad m_4 = \int_{-1}^{1} x^4 dx = \frac{2}{5};\quad m_5 = \int_{-1}^{1} x^5 dx = 0.$$

Apskritai $m_k = \begin{cases} 0, & \text{kai } k \text{ — nelyginis skaičius,} \\ \dfrac{2}{k+1}, & \text{kai } k \text{ — lyginis skaičius.} \end{cases}$

Tada gauname (8.16) lygčių sistemą

$$\begin{cases} c_0 m_0 + c_1 m_1 + c_2 m_2 = -m_3, \\ c_0 m_1 + c_1 m_2 + c_2 m_3 = -m_4, \\ c_0 m_2 + c_1 m_3 + c_2 m_4 = -m_5, \end{cases} \text{ arba } \begin{cases} 2c_0 + \dfrac{2}{3} c_2 = 0, \\ \dfrac{2}{3} c_1 = -\dfrac{2}{5}, \\ \dfrac{2}{3} c_0 + \dfrac{2}{5} c_2 = 0. \end{cases}$$

Jos sprendinys yra toks: $c_1 = -\frac{3}{5}$, $c_0 = c_2 = 0$, vadinasi, daugianaris $\pi(x) = x^3 - \frac{3}{5} x$.

Šio polinomo šaknys, o kartu Gauso kvadratūrinės formulės taškai yra: $x_1 = -\sqrt{\frac{3}{5}}$, $x_2 = 0$ ir $x_3 = \sqrt{\frac{3}{5}}$.

Svorinius koeficientus apskaičiuojame iš sistemos

$$\begin{cases} w_1 + w_2 + w_3 = 2, \\ w_1 x_1 + w_2 x_2 + w_3 x_3 = 0, \\ w_1 x_1^2 + w_2 x_2^2 + w_3 x_3^2 = \dfrac{2}{3}. \end{cases}$$

Žinodami, kad $x_1 = -\sqrt{\dfrac{3}{5}}$, $x_2 = 0$ ir $x_3 = \sqrt{\dfrac{3}{5}}$, turime tokį šios sistemos sprendinį: $w_1 = w_3 = \dfrac{5}{9}$, $w_2 = \dfrac{8}{9}$.

Antrasis metodas yra bendresnis ir gali būti taikomas išvedant kvadratūrines formules, kurių koeficientai w_i tenkina papildomas tiesines sąlygas. Pavyzdžiui, jei pareikalautume, kad koeficientai w_i tenkintų sąlygą $w_1 = w_2 = \; = \ldots = w_n = w$, gautume Čebyšovo kvadratūrines formules, kurios egzistuoja tik tada, kai $n = 1, 2, \ldots, 9$.

Žemiau patalpinta funkcija ***gauskvdrkoef***, kuri įgalina apskaičiuoti n taškų Gauso kvadratūrinės formulės mazgus ir svorinius koeficientus.

```
function [x, w]=gauskvdrkoef(n)
% GAUSKVDRKOEF apskaičiuoja Gauso kvadratūrinės formulės
% mazgus ir svorinius koeficientus.

format long e
% n-os eiles Ležandro polinomo koeficientų apskaičiavimas
p0=zeros(1,n+1); p1=zeros(1,n+1); p=zeros(1,n+1);
p0(1)=1; p1(2)=1; k=2;
while k<=n
    p(1:k+1)=([0 p1(1:k)]*(2*k-1)-p0(1:k+1)*(k-1))/k;
    p0=p1; p1=p; k=k+1;
end
pn=p(end:-1:1);
% Mazgų apskaičiavimas
xp=roots(pn); x=sort(xp);
% Svorinių koeficientų apskaičiavimas
 m=numel(x); b=zeros(m,1); c=zeros(m,1);
 for k=1:n
     b=pn(k)+b.*x; c=b+c.*x;
 end;
w=2./((1-x.^2).*(c.^2));
```

Pavyzdžiui, surinkus komandą : << `[x, w]=gauskvdrkoef(10)`

apskaičiuotume dešimties taškų Gauso kvadratūrinės formulės mazgus ir koeficientus, kurie patalpinti 8.3 lentelėje.

8.3 lentelė. Gauso kvadratūrinė formulė, kai n = 10

x	w
±0,148874338981631	2.955242247147529e-001
±0,433395394129247	2.692667193099963e-001
±0,679409568299024	2.190863625159796e-001
±0,865063366688985	1.494513491505762e-001
±0,973906528517172	6.667134430869286e-002

8.2.2. Gauso-Ermito kvadratūrinė formulė

Gauso-Ermito kvadratūrinė formulė yra naudojama apskaičiuoti integralui $\int_{-\infty}^{\infty} e^{-x^2} f(x)dx$ pagal formulę: $\int_{-\infty}^{\infty} e^{-x^2} f(x)dx = \sum_{i=1}^{n} w_i f(x_i) + E_n$, čia E_n - formulės liekamasis narys, o mazgai x_i ir svoriniai koeficientai w_i apskaičiuojami taip, kad formulė $\int_{-\infty}^{\infty} e^{-x^2} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$ tiksliai integruotų visus polinomus iki $(2n-1)$-os eilės imtinai.

Gauso-Ermito formulės parametrus x_i ir w_i galima apskaičiuoti anksčiau išnagrinėtu Hemingo metodu. Tačiau yra žinoma, kad mazgai x_i yra Ermito polinomo $H_n(x)$ šaknys, o koeficientai apskaičiuojami pagal formulę: $w_i = \dfrac{2^{n-1} n! \sqrt{\pi}}{n^2 \left(H_{n-1}(x_i)\right)^2}$.

Ermito polinomas, kaip ir kiti ortogonalieji polinomai, gali būti nusakomas rekurentine formule:

$$H_n(x) = 2xH_{n-1}(x) - 2(n-1)H_{n-2}(x), n \geq 2, H_0(x) = 1; H_1(x) = 2x.$$

Žemiau pateikta funkcija ***gaushermitkvf***, skirta apskaičiuoti Gauso-Ermito formulės parametrams: x_i ir w_i.

```
function [x, w]=gaushermitkvf(n)
% GAUSHERMITKVF apskaičiuoja Gauso-Ermito kvadratūrinės formulės
% mazgus ir svorinius koeficientus.

format long e
% n-os eiles Ermito polinomo koeficientų apskaičiavimas
p0=zeros(1,n+1); p1=zeros(1,n+1); p=zeros(1,n+1);
p0(1)=1; p1(2)=2; k=2;
while k<=n
    p(1:k+1)=([0 p1(1:k)]-(k-1)*p0(1:k+1))*2;
    p0=p1; p1=p;
    k=k+1;
end
pn1=p0(end-1:-1:1); pn=p(end:-1:1);
% Mazgų x_i apskaičiavimas
xp=roots(pn); x=sort(xp);
% Svorinių koeficientų w_i apskaičiavimas
 m=numel(x); b=zeros(m,1);
 n1=numel(pn1);
 for k=1:n1
     b=pn1(k)+b.*x;
 end;
 cn1=2^(n-1)*prod(1:n)*sqrt(pi)/n^2;
w=cn1./b.^2;
```

Surinkę komandą : << `[x, w]= gaushermitkvf(10)` apskaičiuotume dešimties taškų Gauso-Ermito kvadratūrinės formulės mazgus ir koeficientus, kurie patalpinti 8.4 lentelėje.

8.4 lentelė. Gauso-Ermito kvadratūrinė formulė, kai n = 10

x	w
±0.34290132722370	6.108626337353257e-001
±1.03661082978951	2.401386110823140e-001
±1.75668364929988	3.387439445548182e-002
±2.53273167423279	1.343645746781175e-003

±3.43615911883774	7.640432855232011e-006

8.2.3. Gauso-Legero kvadratūrinė formulė

Gauso-Legero kvadratūrinė formulė yra naudojama apskaičiuoti integralui $\int_{-\infty}^{\infty} e^{-x} f(x)dx$ pagal formulę: $\int_{-\infty}^{\infty} e^{-x} f(x)dx = \sum_{i=1}^{n} w_i f(x_i) + E_n$, čia E_n- formulės liekamasis narys, o mazgai x_i ir svoriniai koeficientai w_i, kaip ir anksčiau, apskaičiuojami taip, kad formulė $\int_{-\infty}^{\infty} e^{-x} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i)$ tiksliai integruotų visus polinomus iki $(2n-1)$-os eilės imtinai.

Gauso-Legero formulės mazgai x_i yra Legero polinomo $L_n(x)$ šaknys, o koeficientai apskaičiuojami pagal formulę: $w_i = \dfrac{1}{x_i\left(\dfrac{d}{dx}L_n(x_i)\right)^2}$.

Legero polinomas, kaip ir kiti ortogonalieji polinomai, gali būti nusakomas rekurentine formule:

$L_n(x) = \big((2n-1-x)L_{n-1}(x) - (n-1)L_{n-2}(x)\big)/n,$ kai $n \geq 2,$ o $L_0(x) = 1; L_1(x) = 1-x.$

Žemiau pateikta funkcija ***gauslegerkvf*** , skirta apskaičiuoti Gauso-Legero formulės parametrams: x_i ir w_i.

```
function [x, w]=gauslegerkvf(n)
% GAUSLEGERKVF apskaičiuoja Gauso-Legero kvadratūrinės formulės
% mazgus ir svorinius koeficientus.

format long e
% n-os eiles Legero polinomo koeficientų apskaičiavimas
p0=zeros(1,n+1); p1=zeros(1,n+1); p=zeros(1,n+1);
p0(1)=1; p1(1)=1; p1(2)=-1; k=2;
while k<=n
    p(1:k+1)=([0 -p1(1:k)]+(2*k-1)*p1(1:k+1)-(k-1)*p0(1:k+1))/k;
    p0=p1; p1=p;
    k=k+1;
end
pn=p(end:-1:1);
% Mazgų xi apskaičiavimas
xp=roots(pn); x=sort(xp);
% Svorinių koeficientų wi apskaičiavimas
 m=numel(x); b=zeros(m,1); c=zeros(m,1);
 for k=1:n
     b=pn(k)+b.*x;
     c=b+c.*x;
 end;
w=1./(x.*c.^2);
```

Surinkę komandą : << `[x, w]= gauslegerkvf(5)` apskaičiuotume dešimties taškų Gauso-Legero kvadratūrinės formulės mazgus ir koeficientus, kurie patalpinti 8.5 lentelėje.

8.5 lentelė. Gauso-Legro kvadratūrinė formulė, kai n = 5

x	*w*
2.635603197181412e-001	5.217556105828091e-001
1.413403059106516e+000	3.986668110831767e-001
3.596425771040715e+000	7.594244968170834e-002
7.085810005858870e+000	3.611758679921840e-003
1.264080084427574e+001	2.336997238577805e-005

8.2.4. Lobato ir Rado kvadratūrinės formulės.

Antrasis metodas tinka ir uždaro tipo Gauso kvadratūrinėms formulėms gauti. Jei pareikalautume, kad integravimo intervalo kraštiniai taškai priklausytų taškams x_i, o kiti $n-2$ taškai x_i ir visi svoriniai koeficientai w_i $\left(i=\overline{1,n}\right)$ būtų apskaičiuojami taip, kad kvadratūrinė formulė tiksliai integruotų visus polinomus iki $(2n-3)$-iosios eilės imtinai, tai gautume Lobato ir Rado kvadratūrines formules. R. Lobato (R. Lobatto) jas nagrinėjo 1852 metais, o R. Rado (R. Radau) — 1880 metais.

Pavyzdžiui, apskaičiuokime Lobato ir Rado kvadratūrinės formulės koeficientus, kai $n=4$, o integravimo intervalas yra $[-1;1]$. Šiuo atveju kvadratūrinę formulę galime užrašyti taip:

$$\int_{-1}^{1} f(x)\,dx \approx w_1 f(-1) + \sum_{i=2}^{n-1} w_i f(x_i) + w_4 f(1).$$

Tada (15) lygčių sistema įgis išraišką

$$\begin{cases} w_1 + w_2 + w_3 + w_4 = m_0, \\ -w_1 + w_2 x_2 + w_3 x_3 + w_4 = m_1, \\ w_1 + w_2 x_2^2 + w_3 x_3^2 + w_4 = m_2, \\ -w_1 + w_2 x_2^3 + w_3 x_3^3 + w_4 = m_3, \\ w_1 + w_2 x_2^4 + w_3 x_3^4 + w_4 = m_4, \\ -w_1 + w_2 x_2^5 + w_3 x_3^5 + w_4 = m_5. \end{cases}$$

Elementariai pertvarkę šią sistemą, eliminuokime iš jos w_1 ir w_4. Turėsime:

$$\begin{cases} w_2\left(x_2^2-1\right) + w_3\left(x_3^2-1\right) = m_2 - m_0, \\ w_2\left(x_2^2-1\right)x_2 + w_3\left(x_3^2-1\right)x_3 = m_3 - m_1, \\ w_2\left(x_2^2-1\right)x_2^2 + w_3\left(x_3^2-1\right)x_3^2 = m_4 - m_2, \\ w_2\left(x_2^2-1\right)x_2^3 + w_3\left(x_3^2-1\right)x_3^3 = m_5 - m_3. \end{cases}$$

Remdamiesi šia sistema, apskaičiuosime (8.16) lygčių sistemos koeficientus:

$$\begin{cases} c_0\left(m_2-m_0\right) + c_1\left(m_3-m_1\right) + m_4 - m_2 = 0, \\ c_0\left(m_3-m_1\right) + c_1\left(m_4-m_2\right) + m_5 - m_3 = 0. \end{cases}$$

Įrašę į ją momentų reikšmes ir išsprendę, apskaičiuosime tokias c_0 ir c_1 reikšmes: $c_0 = -1/5$, o $c_1 = 0$. Tada $\pi(x) = x^2 + 0 \cdot x - 1/5 = 0$. Iš čia $x_1 = -1/\sqrt{5}$, o $x_2 = 1/\sqrt{5}$.

Taigi kai $n = 4$, o integravimo intervalas yra $[-1; 1]$, Lobato ir Rado kvadratūrinės formulės mazginiai taškai yra tokie:

$x_1 = -1$, $x_2 = -1/\sqrt{5}$, $x_3 = 1/\sqrt{5}$ ir $x_4 = 1$.

Turėdami šiuos taškus, galime nesunkiai apskaičiuoti svorinius koeficientus:

$w_1 = 1/6$, $w_2 = w_3 = 5/6$, $w_4 = 1/6$.

Yra žinoma, kad Lobato kvadratūrinės formulės mazgai yra polinomo $(x^2 - 1)P'_{n-1}(x)$, čia $P'_{n-1}(x)$ - $(n-1)$-os eilės Ležandro polinomo išvestinė, šaknys, o svoriniai koeficientai apskaičiuojami pagal formules:

$$w_1 = w_n = \frac{2}{n(n-1)},\ w_i = \frac{2}{n(n-1)(P_{n-1}(x_i))^2}, i = 2,3,\ldots,(n-1).$$

Žemiau patalpinta funkcija ***lobatokvrf*** , kuri įgalina apskaičiuoti n taškų Lobato kvadratūrinės formulės mazgus ir svorinius koeficientus.

```
function [x, w]=lobatokvrf(n)
% LOBATOKVRF apskaičiuoja Lobato kvadratūrinės formulės
% mazgus ir svorinius koeficientus

format long e
% (n-1)-os eiles Ležandro polinomo koeficientų apskaičiavimas
p0=zeros(1,n); p1=zeros(1,n); p=zeros(1,n); pp=zeros(1,n);
p0(1)=1; p1(2)=1; k=2;
while k<=n-1
    p(1:k+1)=([0 p1(1:k)]*(2*k-1)-p0(1:k+1)*(k-1))/k;
    p0=p1; p1=p;
    k=k+1;
end
% (n-1)-os eiles Ležandro polinomo išvestinės apskaičiavimas
pd=[0 linspace(1,n-1,n-1)]; pp=p.*pd; pn=pp(end:-1:2);
% Mazgų apskaičiavimas
xp=roots(pn); xp=sort(xp); x=[-1; xp; 1];
% Svorinių koeficientų apskaičiavimas
 m=numel(xp); b=zeros(m,1); p=p(end:-1:1);
 for k=1:n
     b=p(k)+b.*xp;
 end;
w=2./(n*(n-1)*(b.^2)); w0=2/(n*(n-1)); w=[w0; w; w0];
```

Pavyzdžiui, surinkus komandą : << `[x, w]=lobatokvrf(10)` apskaičiuotume dešimties taškų Lobato kvadratūrinės formulės mazgus ir koeficientus, kurie patalpinti 8.6 lentelėje.

8.6 lentelė. Lobato kvadratūrinė formulė, kai n = 10

x	w
±1.00000000000000	2.222222222222222e-002
±0.91953390816646	1.333059908510710e-001
±0.73877386510550	2.248893420631273e-001

±0.47792494981044	2.920426836796835e-001
±0.16527895766639	3.275397611838974e-001

8.2.5. Gauso tipo kvadratūrinių formulių paklaidos.

Yra žinoma, kad Gauso kvadratūrinės formulės liekamasis narys integralui $\int_a^a f(x)dx$ apskaičiuojamas pagal formulę

$$K_n = (b-a)^{2n+1}\frac{(n!)^4 \cdot f^{(2n)}(z)}{((2n)!)^3(2n+1)},\ z \in [a;b],$$

o Lobato kvadratūrinės formulės liekamojo nario išraiška yra

$$K_n = -\frac{n(n-1)^3 2^{2n-1}((n-2)!)^4 \cdot f^{(2n-2)}(z)}{(2n-1)((2n-2)!)^3},\ z \in [a;b].$$

Gauso-Ermito kvadratūrinės formulės liekamasis narys apibrėžiamas formule: $E_n = \frac{n!\sqrt{\pi}}{2^n(2n)!} f^{(2n)}(\xi), -\infty \le \xi \le \infty.$

Gauso-Legero kvadratūrinės formulės liekamasis narys apibrėžiamas formule: $E_n = \frac{n!\Gamma(n+1)}{(2n)!} f^{(2n)}(\xi), -\infty \le \xi \le \infty$, o gama funkcija yra $\Gamma(\alpha) = \int_0^\infty e^{-t} t^{\alpha-1} dt$.

Šias formules praktiškai taikyti nepatogu, nes reikia įvertinti pointegralinės funkcijos $2n$-tosios arba ($2n$—2)-tosios eilės išvestinę. Todėl liekamieji nariai įvertinami kitais metodais.

Gauso tipo kvadratūrinės formulės liekamąjį narį, kaip ir Niutono bei Koteso kvadratūrinių formulių liekamuosius narius, galima rasti Ričardsono ir Rombergo metodu (pagal Rungės formulę). Tačiau Gauso formulės turi vieną didelį trūkumą: dvi Gauso formulės Q_{n1}^G ir Q_{n2}^G, kai $n2 \neq n1$, neturi nei vieno bendro mago (išskyrus, gal būt, mazgą intervalo centre). Todėl apskaičiuojant liekamąjį narį pagal Rungės formulę reikia skaičiuoti per daug poinegralinės funkcijos reikšmių. Dėl šios priežasties Gauso formulėms paklaida apskaičiuojama pagal nulio formules, taikant A. S. Kronrodo 1965 metais pasiūlytą metodą.

Kronrodo metodas. Tarkime, kad $x_1, x_2, \ldots, x_n$ yra n taškų Gauso kvadratūrinės formulės Q_{n1}^G mazgai. Tada $(2n+1)$ taškų kvadratūrinės formulės Q_{n2}^G mazgai apskaičiuojami taip: imami jau žinomi kvadratūrinės formulės Q_{n1}^G mazgai $x_1, x_2, \ldots, x_n$ ir dar kiekviename intervale $(a, x_1), (x_1, x_2), \ldots, (x_n, b)$ apskaičiuojami nauji mazgai taip, kad formulė Q_{n2}^G būtų galimai didesnio tikslumo, t.y. tiksliai integruotų galimai aukštesnio laipsnio polinomus. Tuo būdu formulės Q_{n2}^G eilė $d = 3n+1$, jei n yra lyginis skaičius, ir $d = 3n+2$, jei n yra nelyginis skaičius. Tada kvadratūrinės formulės Q_{n2}^G paklaida gali būti apskaičiuota pagal formulę

$$Ef = \left| Q_{n2}^G f - Q_{n1}^G f \right|.$$

Tokį paklaidos apskaičiavimo metodą naudoja MATLAB‘o funkcija ***quadl***.

8.3. MATLAB‘o funkcija quadl .

Funkcija ***quadl*** yra skirta apskaičiuoti nurodytu tikslumu apibrėžtinio integralo $R = \int_a^b f(x)dx$ reikšmę . Į šią funkciją dažniausiai naudojami kreipiniai yra:

[*q*, *fcnt*] = *quadl(fun, a, b)* arba [*q*, *fcnt*] = *quadl(fun, a, b, tol)*, čia *fun* – vardas funkcijos, apskaičiuojančios pointegracinę funkciją, *a,b* – integravimo rėžiai, *tol* – norimas tikslumas; *q* – apskaičiuota integralo reikšmė, o *fcnt* – skaičiavime panaudotų pointegralinės funkcijos reikšmių skaičius.

Jei kreipinyje į funkciją *quadl* tikslumas nenurodytas (pirmasis kreipinys), tai pagal nutylėjimo principą priimama, kad $tol = 10^{-6}$.

Funkcija *quadl* naudoja adaptyviojo integravimo strategiją, realizuoja keturių taškų Lobato kvadratūrinę formulę (ji atitinka anksčiau aptartą formulę Q_{n1}^G) ir šios formulės simetrinį Kronrodo praplėtimą iki septynių taškų (tai atitinka anksčiau aptartą formulę Q_{n2}^G). Be to, funkcija *quadl* grubiam integralo reikšmės įvertinimui naudoja formulės Q_{n2}^G tolesnį simetrinį Kronrodo praplėtimą iki trylikos taškų. Grubus integralo reikšmės įvertis *is* (skaičiuojamas pradžioje vieną kartą) naudojamas kaip sudėtinė skaičiavimo pabaigos sąlygos dalis: *if is* + $\left|Q_{n2}^G f - Q_{n1}^G f\right|$ = *is* kuriame tai subintervale, tai reiškia, kad toliau skaidyti šį subintervalą į smulkesnes dalis – neverta.

Praleisdami kvadratūrinių formulių išvedimą (žr. [23 Gander]), čia tik pateiksime pačias kvadratūrines formules.

Keturių taškų Lobato formulė:

$$Q_{n1}^G f = \frac{1}{6}\big(f(-1) + f(1)\big) + \frac{5}{6}\left(f(-\frac{1}{\sqrt{5}}) + f(\frac{1}{\sqrt{5}})\right).$$

Keturių taškų Lobato formulės simetrinis Kronrodo praplėtimas iki 7-ių taškų:

$$Q_{n2}^G f = \frac{11}{210}\big(f(-1) + f(1)\big) + \frac{72}{245}\left(f(-\sqrt{\frac{2}{3}}) + f(\sqrt{\frac{2}{3}})\right) + \frac{125}{294}\left(f(-\frac{1}{\sqrt{5}}) + f(\frac{1}{\sqrt{5}})\right) + \frac{16}{35}f(0).$$

Septynių taškų Lobato formulės simetrinis Kronrodo praplėtimas iki 13-os taškų:

$$Q_{n3}^G f = A\big(f(-1) + f(1)\big) + B\big(f(-x_1) + f(-x_1)\big) + C\left(f(-\sqrt{\frac{2}{3}}) + f(\sqrt{\frac{2}{3}})\right) +$$

$$+ D\big(f(-x_2) + f(x_2)\big) + E\left(f(-\frac{1}{\sqrt{5}}) + f(\frac{1}{\sqrt{5}})\right) + F\big(f(-x_3) + f(x_3)\big) + Gf(0),$$ čia

x_1=0.942882415695479719056; x_2=0.641853342345781305781;

x_3=0.236383199662149880282;

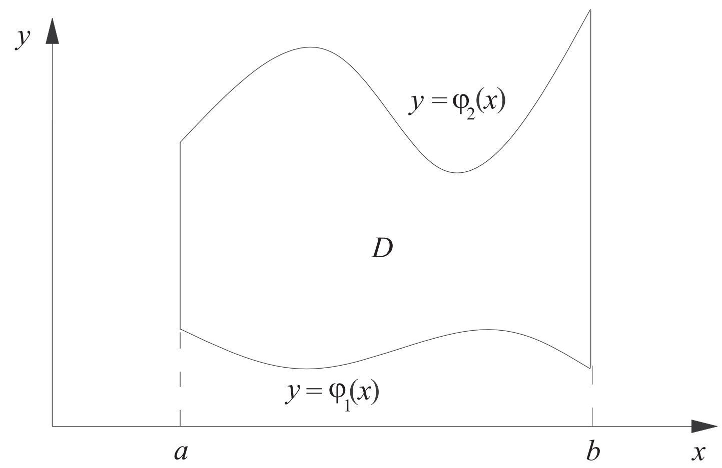
A=0.015827191973480183087; B=0.094273840218850045531;
C=0.155071987336585396254; D=0.188821573960182454420;
E=0.199773405226858526792; F=0.224926465333339527016;
G=0.242611071901407733800.

8.4. Kubatūrinės formulės

Sprendžiant tiek inžinierinius, tiek matematinius uždavinius, tenka apskaičiuoti dvilypio integralo reikšmę įvairiose integravimo srityse. Skaitinio integravimo formulės, sprendžiančios šį uždavinį, vadinamos kubatūrinėmis formulėmis. Toliau aptarsime dvilypio integralo apskaičiavimo kreivakraštėje, stačiakampėje ir trikampėje srityse uždavinį. Integravimas trikampėje srityje ypač aktualus sprendžiant baigtinių elementų metodu diferencialines lygtis su dalinėmis išvestinėmis.

8.4.1. Dvilypio integralo apskaičiavimas kreivakraštėje srityje

Nagrinėsime tokį uždavinį: reikia nurodytu tikslumu ε apskaičiuoti dvilypį integralą $R = \iint\limits_{(D)} f(x, y)\, dx\, dy$; čia sritis D nusakoma nelygybėmis $a \le x \le b$, $\varphi_1(x) \le y \le \varphi_2(x)$ (žr. 8.3 pav.).



8.3 pav. Integravimo sritis

Kartotinio integralo reikšmė randama nuosekliai integruojant pagal kiekvieną kintamąjį. Šia idėja ir remsimės ieškodami dvilypio integralo reikšmės:

$$R = \iint\limits_{(D)} f(x, y)\, dx\, dy = \int\limits_a^b dx \int\limits_{\varphi_1(x)}^{\varphi_2(x)} f(x, y)\, dy .$$

Pažymėkime:

$$F(x) = \int\limits_{\varphi_1(x)}^{\varphi_2(x)} f(x, y)\, dy . \qquad (8.17)$$

Tada $$R = \int\limits_a^b F(x)\, dx . \qquad (8.18)$$

Vadinasi, (8.17) ir (8.18) formulės dvilypių integralų skaičiavimą paverčia paprastųjų integralų skaičiavimu.

(8.18) integralą apskaičiuokime taikydami kurią nors kvadratūrinę, pavyzdžiui, Simpsono, formulę. Tada

$$R = \frac{h}{3}\left(F(x_0) + 4F(x_1) + F(x_2)\right);$$

čia $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$, $h = (b-a)/2$, o

$$F(x_i) = \int_{\varphi_1(x_i)}^{\varphi_2(x_i)} f(x_i, y)\, dy,\ i = 0, 1, 2.$$

$F(x_i)$ yra vienlypis apibrėžtinis integralas, kurį taip pat apskaičiuojame pagal pasirinktą kvadratūrinę formulę. Kitaip tariant, dvilypio integralo skaičiavimas iš esmės nesiskiria nuo paprastojo integralo skaičiavimo. Ieškodami paprastojo integralo $\int_a^b f(x)\, dx$ reikšmės, nepriklausomai nuo integravimo strategijos turėjome „orakulą“ (pointegralinės funkcijos apskaičiavimo procedūrą), kuris kiekvienai x_i reikšmei priskirdavo $f(x_i)$ reikšmę. Apskaičiuodami dvilypį integralą $\int_a^b F(x) dx$, taip pat turime turėti „orakulą“, kuris kiekvienai x_i reikšmei priskirtų $F(x_i)$ reikšmę. Aišku, kad šiuo atveju „orakulas“ bus procedūra, kuri pagal pasirinktą integravimo strategiją apskaičiuos integralo $\int_{\varphi_1(x_i)}^{\varphi_2(x_i)} f(x_i, y)\, dy$ reikšmę norimu tikslumu.

Pavyzdys. Remdamiesi Simpsono formule, apskaičiuokime integralą

$$R = \int_0^1 dx \int_0^{1-x} (x+y)\, dy.$$

Tiksli šio integralo reikšmė yra $1/3$. Kadangi $F(x) = \int_0^{1-x} (x+y) dx$ yra kvadratinis polinomas, tai iš Simpsono kvadratūrinės formulės gauname tikslią R reikšmę. Taigi

$$R = h/3\left(F(x_0) + 4F(x_1) + F(x_2)\right) = 1/6\left(F(0) + 4F(1/2) + F(1)\right);$$

$$F(0) = \int_0^1 y\, dy = 1/6\left(0 + 4\cdot(1/2) + 1\right) = 1/2;$$

$$F(1/2) = \int_0^{1/2} (1/2 + y) dy = 1/12\left(1/2 + 4\cdot(3/4) + 1\right) = 9/24;$$

$$F(1) = \int_0^0 (1+y) dy = 0.$$

Vadinasi, $R = 1/6(1/2 + 4\cdot 9/24 + 0) = 1/3$. Kaip ir tikėjomės, apskaičiavome tikslią dvilypio integralo reikšmę.

Dabar išnagrinėkime, kaip apskaičiuojamas dvilypis integralas tikslumu ε. Tam tikslui laikykimės adaptyviojo integravimo strategijos. Pasirinkę Simpsono kvadratūrinę formulę, uždavinio nesusiaurinsime. Pasirinkus kitas kvadratūrines formules, samprotavimai bus analogiški. Taikant adaptyviojo integravimo strategiją, teks apskaičiuoti $\int_{x_i}^{x_{i+1}} F(x)\,dx$. Kaip anksčiau buvo nagrinėta, norėdami apskaičiuoti integralą $R = \int_a^b F(x)\,dx$ tikslumu ε, integralą $R_i = \int_{x_i}^{x_{i+1}} F(x)\,dx$ turime apskaičiuoti tikslumu $\frac{H_i}{b-a}\varepsilon$; čia $H_i = x_{i+1} - x_i$. Kyla klausimas: kokiu tikslumu turime apskaičiuoti $F(x_k) = \int_{\varphi_1(x_k)}^{\varphi_2(x_k)} f(x_k, y)\,dy$ (čia $x_k \in [x_i; x_{i+1}]$) reikšmę, kad formulės $R_i = \frac{h_i}{3}\left(F(x_i) + 4F\left(\frac{x_i + x_{i+1}}{2}\right) + F(x_{i+1})\right)$ apvalinimo paklaida būtų mažesnė nei $\frac{H_i}{b-a}\varepsilon$? Bendru atveju tikslaus atsakymo į šį klausimą nėra. Praktikoje dažniausiai integralo $F(x_k) = \int_{\varphi_1(x_k)}^{\varphi_2(x_k)} f(x_k, y)\,dy$ reikšmė apskaičiuojama $\frac{H_i 10^{-1}}{b-a}\varepsilon$ tikslumu.

Dvilypis integralas stačiakampėje srityje. Jei integravimo sritis D yra stačiakampis: $a \le x \le b, c \le y \le d$ tai, integralą $R = \iint_{(D)} g(x,y)dxdy$, įvedę pakeitimus: $x = \frac{1}{2}(b-a)u + \frac{1}{2}(b+a), y = \frac{1}{2}(d-c)u + \frac{1}{2}(d+c)$, galime perrašyti:

$$R = \iint_{(D)} g(x,y)dxdy = \frac{1}{4}(b-a)(d-c)\int_{-1}^{1} du \int_{-1}^{1} f(u,v)dv, \text{ čia}$$

$$f(u,v) = g\left(\frac{1}{2}(b-a) + \frac{1}{2}(b+a), \frac{1}{2}(d-c) + \frac{1}{2}(d+c)\right).$$

Esant stačiakampei integravimo sričiai, galime integruoti nepriklausomai pagal kintamuosius u ir v. Tuo būdu, pasirinkę kvadratūrinę formulę Q ir pradėdami nuo vidinio integralo, turėsime: $\int_{-1}^{1} f(u,v)dv \approx \sum_{i=1}^{n} w_i f(u, v_i)$, čia n rodo kvadratūrinės formulės mazgų skaičių.

Toliau integruodami pagal *u,* turėsime:

$$R \approx \int_{-1}^{1}\left(\sum_{i=1}^{n} w_i f(u,v_i)\right)du = \sum_{j=1}^{n} w_j \left(\sum_{i=1}^{n} w_i f(u_j, v_i)\right) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^{*} f(u_j, v_i), \text{ čia } w_{ij}^{*} = w_i w_j,$$

o taškų (u_j, v_i) koordinates apibrėžia pasirinktoji kvadratūrinė formulė Q. Integruojant dvilypius integralus, pasirinktoji kvadratūrinė formulė Q paprastai yra Gauso tipo kvadratūrinė formulė.

Jei kvadratūrinė formulė Q tiksliai integruoja visus polinomus, kurių laipsnis nedidesnis nei p, tai pastaroji formulė tiksliai integruoja išraiškų $u^{p_1} v^{p_2}$ tiesinį darinį, kai $p_1, p_2 \le p$.

8.4.2. Dvilypio integralo apskaičiavimas trikampėje srityje

Dvilypio integralo apskaičiavimo trikampėje srityje uždavinys afinėje erdvėje sprendžiamas žymiai paprasčiau, nei stačiakampėje koordinačių sistemoje (Euklido erdvėje). Todėl, prieš pradėdami nagrinėti integralo apskaičiavimą trikampėje srityje, pirmiausia aptarsime afiniąją erdvę ir taško *baricentrines koordinates*.

Afinioji erdvė. Afiniąją erdvę galime įsivaizduoti kaip tiesę, plokštumą ar įprastinę trimatę Euklido erdvę. Afiniosios erdvės taškai $A_0, A_1, \ldots, A_n$ apibrėžia tašką

$$A = \alpha_0 A_0 + \alpha_1 A_1 + \ldots + \alpha_n A_n, \quad \alpha_0 + \alpha_1 + \ldots + \alpha_n = 1,$$

kurio koordinatės randamos sudedant atitinkamas taškų A_i koordinates, padaugintas iš koeficientų α_i, kurių suma lygi 1.

Kai n = 1, taškas $A = \alpha_0 A_0 + \alpha_1 A_1$ dalija atkarpą $A_0 A_1$ santykiu α_1 / α_0. Šis santykis dar vadinamas *paprastuoju* trijų kolinearių taškų A_0, A_1, A_2 *santykiu* ir žymimas $r(A_0, A_1, A_2) = \alpha_1 / \alpha_0 = A_0 A / A A_1$.

Kai n = 2, taškas $A = \alpha_0 A_0 + \alpha_1 A_1 + \alpha_2 A_2$ dalija trikampį $A_0 A_1 A_2$ į tris trikampius, kurių plotas atitinkamai proporcingas koeficientams α_i:

$$\Delta_0 : \Delta_1 : \Delta_2 = \alpha_0 : \alpha_1 : \alpha_2;$$

čia Δ_0, Δ_1 ir Δ_2 yra atitinkamai trikampių $A_1 A A_2$, $A_0 A A_2$ ir $A_0 A A_1$ plotas.

Ir apskritai jei taškai $A_0, A_1, \ldots, A_n$ nėra vienoje tiesėje, tai bet kurį plokštumos tašką A galima vieninteliu būdu užrašyti kaip afinųjį jų darinį. Koeficientai α_i vadinami taško A *baricentrinėmis koordinatėmis*. Šias koordinates 1827 m. pirmą kartą pavartojo vokiečių matematikas A. F. Mėbijus (A. F. Möbius), atsakydamas į klausimą, kokia turi būti trikampio viršūnių masė, kad jų masės centras sutaptų su pasirinktu tašku.

Jei visi $\alpha_i \geq 0$, tai afinusis darinys vadinamas *iškiluoju* ir yra taškų A_i „viduje“: dviejų taškų atveju — tai visi tuos taškus jungiančios tiesės atkarpos taškai, trijų taškų atveju — tai visi vidiniai trikampio taškai, įskaitant ir kraštinėms priklausančius taškus. Taškų $A_0, A_1, \ldots, A_n$ *iškilusis apvalkalas* — tai visų iškilųjų jo darinių aibė.

Integralo transformacija. Afinioji transformacija įgalina bet kokį trikampį, kurio viršūnės yra taškai $A(a_x, a_y)$, $B(b_x, b_y)$, $C(c_x, c_y)$, transformuoti į trikampį, kurio viršūnių koordinatės yra: (0,0), (0,1) ir (1,0).

Aptarkime šią transformaciją.

Taškas $P = uA + vB + wC$, kai $u + v + w = 1$ ir $0 \leq u, v, w \leq 1$ yra bet kuris trikampio taškas afinėje erdvėje. Kadangi $w = 1 - u - v$, tai $P = u(A - C) + v(B - C) + C$.

Atlikime keitinius: $x = u(x_a - x_c) + v(x_b - x_c) + x_c$, $y = u(y_a - y_c) + v(y_b - y_c) + y_c$.

Tada $R = \iint_T f(x, y)dxdy = \det(J) \iint_{T^*} g(u, v)dudv$, čia T- trikampis ABC, T^* - trikampis, kurio viršūnių koordinatės yra (0,0), (0,1) ,(1,0) , $g(u, v) = f\big(u(x_a - x_c) + v(x_b - x_c) + x_c, u(y_a - y_c) + v(y_b - y_c) + y_c\big)$, o $\det(J)$ yra

Jakobio matricos $J = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix}$ determinantas. Kadangi $\det(J) = \begin{pmatrix} (x_a - x_c)(x_b - x_c) \\ (y_a - y_c)(y_b - y_c) \end{pmatrix} = 2S_T$, čia S_T– trikampio ABC plotas, tai $R = \iint_T f(x,y)dxdy = 2S_T \iint_{T^*} g(u,v)dudv$. Vadinas, nesiaurindami klausimo , toliau nagrinėsime uždavinį: apskaičiuoti $R = \iint_{T^*} g(u,v)dudv = \int_0^1 du \int_0^{1-u} g(u,v)dv = \int_0^1 dv \int_0^{1-v} g(u,v)du$.

Kubatrūrinės formulės konstravimas. Kubatūrinės formulės trikampyje yra konstruojamos taip pat, kaip ir vienmačiu atveju: kubatūrinė formulė užrašoma tiesiniu pointegracinės funkcijos reikšmių, apskaičiuotų pasirinktuose taškuose, dariniu, t.y.

$$R = \iint_{T^*} g(u,v)dudv = \int_0^1 du \int_0^{1-u} g(u,v)dv = \int_0^1 dv \int_0^{1-v} g(u,v)du \approx \sum_{i=1}^{n} p_i g(u_i, v_i).$$

Kaip ir vienmačiu atveju, svorinius koeficientus p_i ir taškus (u_i, v_i) apskaičiuosime taip.\, kad kubatūrinė formulė integruotų galimai aukštesnės eilės paviršius, kurie yra vienanarių $1, u, v, u^2, uv, v^2, u^3, u^2v, uv^2, v^3, u^4, u^3v, u^2v^2, v^3u, v^4, \dots$ ir t.t. n-to laipsnio paviršius yra $\frac{(n+1)(n+2)}{2}$ skaičiaus vienanarių tiesinis darinys.

Kubatūrinėje formulėje kiekvienas taškas duoda tris parametrus: p_i, u_i, v_i. Vadinasi, norint sukonstruoti kubatūrinę formulę, kuri tiksliai integruotų visus paviršius, kurių laipsnis nedidesnis nei n, taškų skaičius k turi būti toks, kad galiotų nelygybė

$$\frac{(n+1)(n+2)}{2} \le 3k.$$

Paėmus vieną tašką, galime sukonstruoti kubatūrinę formulę, kuri tiksliai integruotų visus paviršius iki pirmo laipsnio imtinai. Šiuo atveju kubatūrinė formulė bus tokia:

$R = \int_0^1 du \int_0^{1-u} g(u,v)dv \approx p_1 g(u_1, v_1)$ ir ji turi tiksliai integruoti vienanarius $1, u, v$.

Reiškinys $m_{rs} = \int_0^1 u^r du \int_0^{1-u} v^s dv$ vadinamas integravimo momentu. Nesunku apskaičiuoti, kad $m_{00} = \frac{1}{2}, m_{10} = m_{01} = \frac{1}{6}$. Tada tašką (u_1, v_1) ir svorinį koeficientą p_1 apskaičiuosime iš lygčių sistemos:

$$\begin{cases} p_1 = m_{00}, \\ p_1 u_1 = m_{10}, \\ p_1 v_1 = m_{01}. \end{cases}$$

Sistemos sprendinys yra: $p_1 = \frac{1}{2}, u_1 = v_1 = \frac{1}{3}, w_1 = 1 - u_1 - v_1 = \frac{1}{3}$.

Jei imtume du taškus, tai galime tikėtis sukonstruoti kubatūrinę formulę, kuri tiksliai integruotų visus paviršius iki antro laipsnio imtinai, nes nežinomų parametrų p_i, u_i, v_i ir lygčių, rišančių šiuos parametrus, skaičiai – sutampa. Tačiau ši sistema neturi sprendinio. Todėl, norint sukonstruoti kubatūrinę formulę, kuri tiksliai integruotų visus paviršius iki antro laipsnio imtinai, imami trys taškai. Tada turime šešias lygtis ir devynis nežinomuosius. Simetrinis šios sistemos sprendinys yra:

$p_1 = p_2 = p_3 = \frac{1}{6}$, o $\left(\frac{1}{2}, \frac{1}{2}, 0\right)$, $\left(\frac{1}{2}, 0, \frac{1}{2}\right)$, $\left(0, \frac{1}{2}, \frac{1}{2}\right)$ - taškų baricentrinės koordinatės (žr.8.4 pav).

Pastaba. Apskaičiuojant kubatūrines formules trikampėje srityje, reikia prisiminti, kad $u_i, v_i \geq 0, u_i + v_i \leq 1$.

Reziumė. Įvertinant aukščiau pateiktą nagrinėjimą, galime reziumuoti:

$R = \iint\limits_T f(x,y)dxdy = 2S_T \iint\limits_{T^*} g(u,v)dudv \approx S_T \sum_{i=1}^{n} p_i^* f(x_i, y_i)$, čia S_T – trikampio ABC plotas, $p_i^* = 2p_i$, $x_i = u_i x_a + v_i x_b + w_i x_c, y_i = u_i y_a + v_i y_b + w_i y_c$.

Pavyzdys. *Taikydami keturių taškų formulę, apskaičiuokime dvilypį integralą*

$$R = \int_0^1 dy \int_0^{1-y} (x+y)^{\frac{1}{2}} dx.$$

Tiksli šio integralo reikšmė yra *0.4*. Pritaikę 4-ių taškų (taškų koordinatės: (1/3;1/3), (0.2;0.2), (0.2;0.6), (0.2;0.6)) formulę turėsime:

$$R = \int_0^1 dy \int_0^{1-y} (x+y)^{\frac{1}{2}} dx \approx S_T \sum_{i=1}^{4} p_i^* (x_i + y_i)^{\frac{1}{2}} = \frac{1}{2}\left(\frac{-27}{48}\left(\frac{2}{3}\right)^{\frac{1}{2}} + \frac{25}{48}(0.4)^{\frac{1}{2}} + \frac{25}{48}(0.8)^{\frac{1}{2}} + \frac{25}{48}(0.8)^{\frac{1}{2}}\right) =$$

=0.400909793.

Kaip matome, rezultato paklaida $\Delta R \approx 0.00091$.

8.7 lentelėje pateikti populiariausių kubatūrinių formulių trikampėje srityje parametrai, o 8.4 paveiksle parodyta dviejų formulių geometrinė interpretacija [24,25].

Didelio tikslumo kubatūrinių formulių konstravimas. Iš anksčiau pateikto nagrinėjimo matyti, kad sukonstruoti didelio tikslumo kubatūrines formules trikampyje nėra paprasta, nes reikia rasti sudėtingos netiesinių lygčių sistemos sprendinį. Todėl, tokios formulės konstruojamos transformuojant trikampę sritį į stačiakampę ir, po to, toje srityje taikant didelio tikslumo Gauso kvadratūrines formules [26].

Aptarkime šią transformaciją. Turime: $R = \int_0^1 dx \int_0^{1-x} f(x,y)dy.$ Įveskime pakeitimą: $x = u, y = (1-u)v$. Šis pakeitimas trikampę sritį transformuoja į kvadratą: $\{(u,v) \mid 0 \leq u, v \leq 1\}$. Tuo būdu $R = \int_0^1 dx \int_0^{1-x} f(x,y)dy. = \int_0^1 du \int_0^1 f(u,(1-u)v)(1-u)dv,$ nes

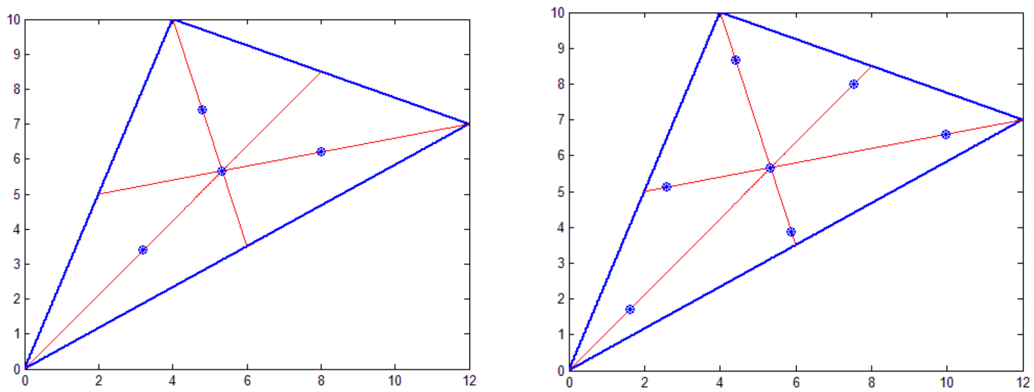
$dxdy = \det J dudv = (1-u)dudv$. Pastarasis integralas toliau transformuojamas į standartinį stačiakampį: $\{(\xi,\eta) \mid -1 \le \xi,\eta \le 1\}$, atlikus pakeitimus:

$u = (1+\xi)/2, v = (1+\eta)/2$.

8.7 lentelė. Kubatūrinių formulių taškai ir svoriniai koeficientai

n	*Paviršiaus laipsnis*	*Taškų baricentrinės koordinatės*			*Svoriniai koeficientai* p_i^*
		u	*v*	*w*	
1	1	1/3	1/3	1/3	1
3	2	1/2	1/2	0	1/3
		0	1/2	1/2	1/3
		1/2	0	1/2	1/3
4	3	1/3	1/3	1/3	-27/48
		0.6	0.2	0.2	25/48
		0.2	0.6	0.2	25/48
		0.2	0.2	0.6	25/48
7	4	1/3	1/3	1/3	0.225 000 0000
		α_1	β_1	β_1	0.132 394 1527
		β_1	α_1	β_1	0.132 394 1527
		β_1	β_1	α_1	0.132 394 1527
		α_2	β_2	β_2	0.125 939 1805
		β_2	α_2	β_2	0.125 939 1805
		β_2	β_2	α_2	0.125 939 1805

α_1= 0.059 715 071 7, β_1=0.470 142 064 1,
α_2=0.797 426 985 3, β_2=0.101 286 507 3.



8.4. pav. 4-ių taškų (kairėje) ir 7-ių taškų(dešinėje) kubatūrinės formulės taškai

Kadangi $dudv = \det J d\xi d\eta = \frac{1}{4} d\xi d\eta$. Tuo būdu $R = \int_0^1 dx \int_0^{1-x} f(x,y)dy. = \int_0^1 du \int_0^1 f(u,(1-u)v)(1-u)dv = \int_{-1}^1 d\xi \int_{-1}^1 f(\frac{1+\xi}{2}, \frac{(1-\xi)(1+\eta)}{4}) \frac{(1-\xi)}{8} d\eta.$

Pastarajam integralui galime taikyti n taškų Gauso kvadratūrines formules ir turėsime: $R = \int_{-1}^1 d\xi \int_{-1}^1 f(x(\xi,\eta), y(\xi,\eta)) \frac{(1-\xi)}{8} d\eta \approx \sum_{i=1}^n \sum_{j=1}^n \frac{(1-\xi_i)}{8} w_i w_j f(x(\xi_i,\eta_j), y(\xi_i,\eta_j))$, čia ξ_i, η_j yra Gauso kvadratūrinės formulės taškai atitinkamai ξ, η kryptimis, o w_i ir w_j - koeficientai.

Paskutinę formulę galima perrašyti:

$$R \approx \sum_{k=1}^{N=n\times n} c_k f(x_k, y_k),$$

čia $c_k = \frac{(1-\xi_i)}{8} w_i w_j$, $x_k = \frac{(1+\xi_i)}{2}$, $y_k = \frac{(1-\xi_i)(1+\eta_j)}{4}$, $k = 1,2,3,\ldots,n$, $i, j = 1,2,3,\ldots,n$.

Pavyzdys. Apskaičiuokime ankstesniame pavyzdyje pateiktą integralą $R = \int_0^1 dy \int_0^{1-y} (x+y)^{\frac{1}{2}} dx$, taikydami trijų taškų Gauso kvadratūrinę formulę.

Įvykdę komandas:

```
>> [xk,yk,ck]=kubfortrikamp(3);
>> r=sum(sqrt((xk+yk)).*ck)
```

turėsime rezultatą:

```
r = 0.40017997986424.
```

Kaip matome, rezultato paklaida $\Delta r \approx 0.00018$. Vadinasi, rezultatas yra tikslesnis, nei skaičiuojant su anksčiau pateikta keturių taškų formule. Tačiau, reikia pastebėti, kad šiuo atveju veiksmų skaičius yra žymiai didesnis, nei keturių taškų formulės veiksmų skaičius.

Žemiau pateikta funkcija ***kubfortrikamp***, įgalinanti pagal išnagrinėtą metodą apskaičiuoti didelio tikslumo kubatūrines formules, o 8.8 lentelėje patalpinti kubatūrinės formulės trikampyje taškai ir svoriniai koeficientai, kai *n=3*.

```
function [xk,yk,ck]=kubfortrikamp(n)
% KUBFORTRIKAMP apskaičiuoja kubatūrinės formulės
% mazgus ir svorinius koeficientus trikampyje.
% Įėjimo parametrai
% n  - Gauso kvadratūrinės formulės taškų skaičius.
% Išėjimo parametrai
% (xk,yk) - kubatūrinės formulės taškai,
% ck      - kubatūrinės formulės koeficientai.

format long e
% n-os eilės Ležandro polinomo koeficientų apskaičiavimas
p0=zeros(1,n+1); p1=zeros(1,n+1); p=zeros(1,n+1);
p0(1)=1; p1(2)=1; k=2;
while k<=n
```

```
    p(1:k+1)=([0 p1(1:k)]*(2*k-1)-p0(1:k+1)*(k-1))/k;
    p0=p1; p1=p;
    k=k+1;
end
pn=p(end:-1:1);
% n taškų Gauso formulės mazgų apskaičiavimas
xp=roots(pn); x=sort(xp);
% n taškų Gauso formulės svorinių koeficientų apskaičiavimas
 m=numel(x); b=zeros(m,1); c=zeros(m,1);
 for k=1:n
     b=pn(k)+b.*x;
     c=b+c.*x;
 end;
w=2./((1-x.^2).*(c.^2));
% Kubatūrinės formulės trikampyje taškai
xtr=(1+x)/2; vn=ones(1,n); xtr=(xtr*vn)'; xk=xtr(:);
ytr=(((1-x)*(1+x)')/4)'; yk=ytr(:);
% Kubatūrines formulės trikampyje svoriniai koeficientai
wi=(1-x).*w/8; ck=(wi*w')'; ck=ck(:);
```

8.8 lentelė. Kubatūrinės formulės trikampyje taškai ir svoriniai koeficientai, kai n=3

x_k	y_k	c_k
1.127016653792584e-001	1.000000000000000e-001	6.846437767135356e-002
1.127016653792584e-001	4.436491673103709e-001	1.095430042741657e-001
1.127016653792584e-001	7.872983346207417e-001	6.846437767135356e-002
5.000000000000000e-001	5.635083268962918e-002	6.172839506172841e-002
5.000000000000000e-001	2.500000000000000e-001	9.876543209876543e-002
5.000000000000000e-001	4.436491673103709e-001	6.172839506172841e-002
8.872983346207417e-001	1.270166537925832e-002	8.696116155806972e-003
8.872983346207417e-001	5.635083268962915e-002	1.391378584929115e-002
8.872983346207417e-001	1.000000000000000e-001	8.696116155806972e-003

8.5. ***Skaitinis diferencijavimas***

Išnagrinėkime du funkcijos $y = f(x)$ išvestinės apskaičiavimo uždavinio aspektus.

1. Funkcija $y = f(x)$ pateikta reikšmių lentele (x_i, y_i). Dažniausiai tai būna eksperimento rezultatai ir y_i reikšmės turi paklaidą. Skirtingai nei integravimas, šios paklaidos labai iškreipia diferencijavimo rezultatus.

2. Žinoma funkcijos $y = f(x)$ analizinė išraiška.

Funkcijos, pateiktos reikšmių lentele, diferencijavimas. Tarkime, kad funkcija $y = f(x)$ pateikta reikšmių lentele (x_i, y_i); čia $i = \overline{0, N}$. Norime apskaičiuoti tos funkcijos išvestinę taške $x \in [x_0; x_N]$.

Galimi du sprendimo būdai.

Pirmasis būdas.

1. Remdamiesi lentele (x_i, y_i), funkciją $y = f(x)$ aproksimuojame funkcija $y = F(x)$. Dažniausiai funkcija $F(x)$ yra arba polinomas, arba splainas.

2. Laikome, kad $f'(x) \approx F'(x)$, $x \in [x_0; x_N]$. Tačiau reikia pabrėžti, kad nors $F(x) \approx f(x)$, kai $x \in [x_0; x_N]$, bet $F'(x)$ gali labai skirtis nuo $f'(x)$.

Antrasis būdas. Šis būdas gali būti taikomas, kai reikia apskaičiuoti išvestines lentelės taškuose ir lentelės žingsnis yra pastovus bei lygus *h*. Tada taikomi dešinieji, centriniai arba kairieji skirtumai.

8.9, 8.10 ir 8.11 lentelėse pateiktos išvestinių bei jų liekamųjų narių (skliausteliuose) formulės, kai naudojami atitinkamai dešinieji, centriniai ir kairieji skirtumai. Šios formulės išvedamos remiantis funkcijos $f(x)$ Teiloro eilute arba interpoliacinių polinomų, einančių per atitinkamus lentelės taškus, išvestinėmis. Pavyzdžiui, pasinaudodami trečiaisiais dešiniaisiais skirtumais, išveskime formulę y_0' apskaičiuoti:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)}{2}(x - x_0)^2 + \frac{f'''(x_0)}{6}(x - x_0)^3 + \dots;$$

$$f(x_0 + h) = y_1 = y_0 + y_0'h + \frac{y_0''}{2}h^2 + \frac{y_0'''}{6}h^3 + \dots;$$

$$f(x_0 + 2h) = y_2 = y_0 + y_0' 2h + y_0''\frac{4h^2}{2} + \frac{y_0'''}{6}8h^3 + \dots .$$

Iš paskutinių dviejų lygčių eliminavę y_0'', turime:

$$y_0' = \frac{1}{2h}(4y_1 - y_2 - 3y_0) + O\left(\frac{h^2}{3}y_0'''\right).$$

Funkcijos diferencijavimas, kai yra žinoma jos analizinė išraiška. Tarkime, kad funkcija $y = y(x)$ yra diferencijuotina intervale $[a; b]$ ir žinoma analizinė jos išraiška. Norint skaitiniu būdu apskaičiuoti tos funkcijos išvestinę taške $x \in [a; b]$, patogiausia taikyti Ričardsono ir Rombergo metodą.

8.9 lentelė. Išvestinės skaičiavimas naudojant dešiniuosius skirtumus

Išvestinės	*Antrieji skirtumai*	*Tretieji skirtumai*	*Ketvirtieji skirtumai*	*Penktieji skirtumai*
y_0'	$\frac{1}{h}\left(y_1 - y_0\right) - \left(\frac{h}{2} y_0'\right)$	$\frac{1}{2h}\left(-y_2 + 4y_1 - 3y_0\right) + \left(\frac{h^2}{3} y_0'''\right)$	$\frac{1}{6h}\left(2y_3 - 9y_2 + 18y_1 - 11y_0\right) - \left(\frac{h^3}{4} y_0^{(4)}\right)$	$\frac{1}{12h}\left(-3y_4 + 16y_3 - 36y_2 + 48y_1 - 25y_0\right) + \left(\frac{h^4}{5} y_0^{(5)}\right)$
y_0''		$\frac{1}{h^2}\left(y_2 - 2y_1 + y_0\right) - \left(h y_0'''\right)$	$\frac{1}{h^2}\left(-y_3 + 4y_2 - 5y_1 + 2y_0\right) + \left(\frac{11h^2}{12} y_0^{(4)}\right)$	$\frac{1}{12h^2}\left(11y_4 - 56y_3 + 114y_2 - 104y_1 + 35y_0\right) + \left(\frac{5h^3}{6} y_0^{(5)}\right)$
y_0'''			$\frac{1}{h^3}\left(y_3 - 3y_2 + 3y_1 - y_0\right) - \left(\frac{3h}{2} y_0^{(4)}\right)$	$\frac{1}{2h^3}\left(-3y_4 + 14y_3 - 24y_2 + 18y_1 - 5y_0\right) + \left(\frac{21h^2}{12} y_0^{(5)}\right)$

8.10 lentelė. Išvestinės skaičiavimas naudojant centrinius skirtumus

Išvestinės	*Tretieji skirtumai*	*Penktieji skirtumai*	*Septintieji skirtumai*
y_0'	$\frac{1}{2h}\left(y_1 - y_{-1}\right) - \left(\frac{h^2}{6} y'''\right)$	$\frac{1}{12h}\left(-y_2 + 8y_1 - 8y_{-1} + y_{-2}\right) + \left(\frac{h^4}{30} y_0^{(5)}\right)$	$\frac{1}{60h}\left(y_3 - 9y_2 + 45y_1 - 45y_{-1} + 9y_{-2} - y_{-3}\right) - \left(\frac{h^6}{140} y_0^{(7)}\right)$
y_0''	$\frac{1}{h^2}\left(y_1 - 2y_0 + y_{-1}\right) - \left(\frac{h^2}{12} y_0^{(4)}\right)$	$\frac{1}{12h^2}\left(-y_2 + 16y_1 - 30y_0 + 16y_{-1} - y_{-2}\right) - \left(\frac{h^4}{90} y_0^{(6)}\right)$	$\frac{1}{180h^2}\left(2y_3 - 27y_2 + 270y_1 - 490y_0 + 270y_{-1} - 27y_{-2} + 2y_{-3}\right) - \left(\frac{h^6}{560} y_0^{(8)}\right)$
y_0'''		$\frac{1}{2h^3}\left(y_2 - 2y_1 + 2y_{-1} - y_{-2}\right) - \left(\frac{h^2}{4} y_0^{(5)}\right)$	$\frac{1}{8h^3}\left(-y_3 + 8y_2 - 13y_1 + 13y_{-1} - 8y_{-2} + y_{-3}\right) + \left(\frac{7h^4}{120} y_0^{(7)}\right)$

8.11 lentelė. Išvestinės skaičiavimas naudojant kairiuosius skirtumus

Išvestinės	*Antrieji skirtumai*	*Tretieji skirtumai*	*Ketvirtieji skirtumai*	*Penktieji skirtumai*
y_0'	$\frac{1}{h}\left(y_0 - y_{-1}\right) + \left(\frac{h}{2}y_0''\right)$	$\frac{1}{2h}\left(3y_0 - 4y_{-1} + y_{-2}\right) + \left(\frac{h^2}{3}y_0'''\right)$	$\frac{1}{6h}\left(11y_0 - 18y_{-1} + 9y_{-2} - 2y_{-3}\right) + \left(\frac{h^3}{4}y_0^{(4)}\right)$	$\frac{1}{12h}\left(25y_0 - 48y_{-1} + 36y_{-2} - 16y_{-3} + 3y_{-4}\right) + \left(\frac{h^4}{5}y_0^{(5)}\right)$
y_0''		$\frac{1}{h^2}\left(y_0 - 2y_{-1} + y_{-2}\right) + \left(hy_0'''\right)$	$\frac{1}{h^2}\left(2y_0 - 5y_{-1} + 4y_{-2} - y_{-3}\right) + \left(\frac{11}{12}h^2 y_0^{(4)}\right)$	$\frac{1}{h^2}\left(35y_0 - 104y_{-1} + 114y_{-2} - 56y_{-3} + 11y_{-4}\right) + \left(\frac{5}{6}h^3 y_0^{(5)}\right)$
y_0'''			$\frac{1}{h^3}\left(y_0 - 3y_{-1} + 3y_{-2} - y_{-3}\right) + \left(\frac{3h}{2}y_0^{(4)}\right)$	$\frac{1}{2h^3}\left(5y_0 - 18y_{-1} + 24y_{-2} - 14y_{-3} + 3y_{-4}\right) + \left(\frac{7}{4}h^2 y_0^{(5)}\right)$

Diferencijavimas intervalo centre. Tarkime, kad Ričardsono ir Rombergo metodu reikia apskaičiuoti $y(x)$ išvestinę taške x, be to, $[x-h;\, x+h] \subset [a;\, b]$. Tam tikslui galime taikyti centrinius skirtumus.

Šio metodo esmę paaiškinsime imdami du argumento pokyčius: h_k ir h_{k+1} ($h_k < h$ ir $h_{k+1} < h$). Tada

$$y'(x) = \frac{y(x+h_k) - y(x-h_k)}{2h_k} - \frac{y'''(x)}{3!}h_k^2 - \dots\,.$$

$\frac{y(x+h_k) - y(x-h_k)}{2h_k}$ pažymėję T_{0k} ir laikydami, kad $y'''(x)$ yra pastovi, galime užrašyti:

$$y'(x) = T_{0k} + Ch_k^2\,.$$

Analogiškai

$$y'(x) = T_{0,\,k+1} + Ch_{k+1}^2\,.$$

Iš šių lygčių galime apskaičiuoti konstantą C:

$$C = \frac{T_{0,\,k+1} - T_{0k}}{h_k^2 - h_{k+1}^2}\,.$$

Vadinasi, formulė

$$T_{1k} = T_{0,\,k+1} + \frac{T_{0,\,k+1} - T_{0k}}{h_k^2 - h_{k+1}^2} \cdot h_{k+1}^2 \qquad (8.19)$$

bus tikslesnė nei T_{0k} arba $T_{0,\,k+1}$. Nesunku įsitikinti, kad (8.19) formulės liekamasis narys proporcingas h_k^4 ir į proporcingumo koeficientą įeina $y^{(5)}(x)$. Vadinasi, (8.19) formulė tiksliai diferencijuoja visus polinomus iki 4-osios eilės imtinai nepriklausomai nuo h_k ir h_{k+1} reikšmių.

Pavyzdžiui, apskaičiuokime $y = x^4$ išvestinę, kai $x = 1$. Aišku, kad $y'\big|_{x=1} = 4$. Dabar raskime šią išvestinės reikšmę pagal (35) formulę, kai $h_k = 1$, o $h_{k+1} = 1/2$. Tada $T_{0k} = 8$, o $T_{0,\,k+1} = 5$ ir

$$y'(x) = T_{1k} = 5 + \frac{5-8}{1-1/4} \cdot \frac{1}{4} = 4 .$$

Išvestinės tikslinimo procesą analogiškai galime tęsti. Ričardsono ir Rombergo metodas paprastai apibūdinamas formulėmis.

Tarkime, kad turime žingsnį h_1 ir skaičių n. Dažniausiai $n = 5$. Apibrėžkime žingsnius h_k pagal formulę

$$h_k = \frac{n-k+1}{n} h_1, \quad k = \overline{1, n} . \tag{8.20}$$

Tada

$$T_{0k} = \frac{y(x+h_k) - y(x-h_k)}{2h_k}, \quad k = \overline{1, n},$$

$$T_{mk} = T_{m-1,\,k+1} + \frac{T_{m-1,\,k+1} - T_{m-1,\,k}}{a_{mk} - 1}, \tag{8.21}$$

$$a_{mk} = \left(1 + \frac{m}{n - (k + (m-1))}\right)^2, \; m = \overline{1, n-1}, \; k = \overline{1, n-m} .$$

Aišku, kad $T_{n-1,\,1}$ yra tiksliausia išvestinės reikšmė. T_{mk} liekamasis narys yra $O\left(h_k^{2m+2}\right)$.

Skaičiavimas pagal (8.21) formulę pavaizduotas 8.12 lentelėje.

8.12 lentelė. Ričardsono ir Rombergo metodas, kai naudojami centriniai skirtumai

Paklaida	$O(h_k^2)$	$O(h_k^4)$	$O(h_k^6)$	$O(h_k^8)$	$O(h_k^{10})$
Žingsnio ilgis	0	1	2	3	4
h_1	T_{01}	T_{11}	T_{21}	T_{31}	T_{41}
$h_2 = 0{,}8h_1$	T_{02}	T_{12}	T_{22}	T_{32}	
$h_3 = 0{,}6h_1$	T_{03}	T_{13}	T_{23}		
$h_4 = 0{,}4h_1$	T_{04}	T_{14}			
$h_5 = 0{,}2h_1$	T_{05}				

Diferencijavimas intervalo galuose. Šiuo atveju skaitinis diferencijavimas yra analogiškas diferencijavimui intervalo centre. Svarbiausias skirtumas yra tas, kad reikia naudotis ne centriniais, bet kairiaisiais arba dešiniaisiais skirtumais. Be to, norint gauti tokį pat tikslumą, pravartu imti dvigubai didesnį skaičių n.

Parašykime darbo formules, kuriose naudojami dešinieji skirtumai ir $n = 10$.

$$T_{0k} = \frac{y(x+h_k) - y(x)}{h_k};$$

čia

$$h_k = \frac{n-k+1}{n} h_1, \ k = \overline{1, n}. \tag{8.22}$$

$$T_{mk} = T_{m-1,\,k+1} + \frac{T_{m-1,\,k+1} - T_{m-1,\,k}}{a_{mk} - 1}, \tag{8.23}$$

$$a_{mk} = 1 + \frac{m}{n - (k + (m-1))}, \ m = \overline{1, n-1}, \ k = \overline{1, n-m}.$$

T_{mk} metodo paklaida yra $O\left(h_k^{m+1}\right)$.