ASUPRA PROGRAMĂRII LA MAŞINILE ELECTRONICE DE CALCUL A FORMULELOR DE CUADRATURI NUMERICE

DE

L. NEGRESCU şi T. RUS

(Cluj)

Lucrare prezentată la sesiunea de comunicări ştiinţifice în domeniul automaticii, din 8 — 9 noiembrie 1962, Bucureşti

1. Dacă pentru funcţia $f(x)$ definită şi integrabilă pe segmentul $[a, b]$ există o primitivă $F(x)$, atunci

$$\int_a^b f(x)dx = F(b) - F(a). \tag{1}$$

De multe ori, însă, nu se poate găsi o primitivă a funcţiei $f(x)$. În acest caz se pune problema găsirii valorii aproximative a acestei integrale. O primă metodă de calcul aproximativ a integralei (1) ar fi folosirea definiţiei integralei definite ca limită a sumei integrale (2)

$$S_n = \sum_{i=1}^{n} f(x_i)\Delta x_i \tag{2}$$

Această metodă are deficienţa că convergenţa sumei S către valoarea integralei (1) este în general slabă, şi prin urmare necesită un număr foarte mare de termeni, ceea ce practic se realizează foarte greu.

Din această cauză se pune problema aproximării funcţiei de sub integrala (1) cu ajutorul unei funcţii interpolatoare $\varphi(x)$. Folosind asemenea funcţii interpolatoare generale, problema pusă se poate trata mai general introducînd o funcţie pondere.

Să presupunem că avem de calculat integrala.

$$\int_a^b f(x)p(x)dx \tag{3}$$

unde $p(x)$ este o funcție oarecare care satisface condiția $p(x) > 0$, pe intervalul $[a, b]$. Ea se numește funcție pondere. În cele ce urmează vom reprezenta funcția $f(x)$ sub forma

$$f(x) = \varphi(x) + R(x). \tag{4}$$

unde $\varphi(x)$ este un polinom de interpolare iar $R(x)$ este restul. Atunci.

$$\int_a^b f(x)dx = \int_a^b \varphi(x)dx + \int_a^b R(x)dx \tag{5}$$

Prima parte a membrului drept al acestei formule este formula de integrare numerică căutată, iar a doua parte este restul acestei formule.

Polinomul de interpolare $\varphi(x)$ se poate reprezenta sub forma

$$\varphi(x) = f(x_0)\Phi_0(x) + f(x_1)\Phi_1(x) + \ldots + f(x_n)\Phi_n(x), \tag{6}$$

unde $x_0, x_1, \ldots, x_n$ sînt nodurile de interpolare.

Dacă notăm

$$\int_a^b p(x)\Phi_i(x)dx = c_i, \quad i = 0, 1, \ldots n \tag{7}$$

și dacă neglijăm restul, formula de cuadratură numerică devine

$$\int_a^b p(x)f(x)dx \approx \sum_{i=2}^{n} c_i f(x_i). \tag{8}$$

În această lucrare vom face un studiu general al formulelor de cuadratură numerice din punct de vedere al posibilității folosirii mașinilor electronice la rezolvarea acestor probleme.

Pentru a obține precizia scontată a rezultatului, vom neglija restul formulelor și vom proceda în felul următor :

Fie $c_0, c_1, c_2, \ldots, c_p$ rezultatul calculului integralei (1) cu ajutorul mașinii electronice după o anumită formulă de cuadratură, folosind n noduri. Mărim numărul n de un număr convenabil λ de ori și aplicăm din nou același procedeu. Dacă rezultatul este $c_0, c_1, c_2, \ldots, c_r$, atunci zecimalele care coincid cu zecimalele primului rezultat sînt cifrele exacte ale valorii integralei (1). Dacă această precizie este insuficientă, mărim din nou numărul nodurilor, continuînd așa mai departe pînă cînd rezultatul apare cu precizia dorită.

2. În cele ce urmează vom reaminti pe scurt formulele de cuadratură numerică fără rest, care ne vor sta în continuare în atenție.

Formula lui Newton-Cotes. Se obține dacă considerăm ca polinom de interpolare $\varphi(x)$, polinomul lui Lagrange cu noduri echidistante.

Fie de calculat $\int_c^d f(x)\,dx$. Funcția $f(x)$ o vom aproxima cu polinomul de interpolare al lui Lagrange pe nodurile

$$x = a + ih, \quad i = 1, 2, \ldots, n$$

unde sau a coincide cu c și atunci $d = a + (n + 1)h$, sau $a + h = c$ și $d = a + nh$.

În primul caz mulțimea nodurilor de interpolare nu conține punctele c și d, iar intervalul de intergrare se împarte în $n + 1$ părți egale, iar în al doilea caz punctele c și d aparțin mulțimii nodurilor de interpolare și intervalul de integrare se împarte în $n - 1$ părți egale.

Formula de cuadratură numerică obținută în primul caz se numește de tip deschis, iar în al doilea caz de tip închis. Dacă punem

$$c = a + (1 - k)h \quad \text{și} \quad d = a + (n + k)h$$

atunci formula de tip deschis se obține pentru $k = 1$, iar de tip închis pentru $k = 0$.

Notăm cu $F(y) = f(a + hy)$.

Atunci

$$\int_c^d f(x)dx = h \int_{1-k}^{n+k} F(y)dy$$

Polinomul de interpolare al lui Lagrange construit pentru $f(x)$ pe nodurile x_i prin schimbarea făcută, are următoarea formă :

$$L(y) = \sum_{i=1}^{n} \frac{(y-1)(y-2)\ldots(y-i+1)\ldots(y-n)}{(i-1)(i-2)\ldots 2 \cdot 1(-1)(-2)\ldots(i-n)} F(i).$$

Înlocuind în formula (5) și ținînd seama de restul formulei de interpolare a lui Lagrange, obținem

$$\int_c^d f(x)dx = (d-c)\sum_{i=1}^{n} I_{i,k}^{(n)} f(a + ih) +$$

$$+ h\int_{1-k}^{n+k} (y-1)(y-2)\ldots(y-n)F(y\,;\,1, 2, \ldots, n)dy, \tag{9}$$

unde

$$I_{i,k}^{(n)} = \frac{(-1)^{n-i}}{(n-1+2k)(i-1)!\,(n-1)!} \int_{1-k}^{n+k} \frac{(y-1)(y-2)\ldots(y-n)}{y-i}\,dy.$$

Numerele $I_{i,k}^{(n)}$, posedă următoarele proprietăţi :

1) $I_{i,k}^{(n)} = I_{n-i+1,k}^{(n)}$

2) $\sum_{i=1}^{n} I_{i,k}^{(n)} = 1.$

După studiul restului obţinem următoarele formule ale lui Newton-Cotes

$$\text{I} \quad \int_c^d f(x)dx = (d-c)\sum_{i=1}^{2m-1} I_{i,k}^{2m-1} f(a+ih) + R_1(f)$$

$$h = \frac{d-c}{2m-2+2k}, \quad n = 2m-1$$

$$\text{II} \quad \int_c^d f(x)dx = (d-c)\sum_{i=1}^{2m} I_{i,k}^{(2m)} f(a+ih) + R_2(f)$$

$$h = \frac{d-c}{2m-1+2k}, \quad n = 2m.$$

Cazurile particulare $n = 2$ şi $n = 3$ ale formulei de tip închis sînt respectiv formula trapezului şi a lui Simpson, adică

$$\int_c^d f(x)dx = \frac{d-c}{2}[f(d)+f(c)] + R'(f) \tag{10}$$

$$\int_c^d f(x)dx = \frac{d-c}{6}\left[f(c) + 4f\left(\frac{c+d}{2}\right) + f(d)\right] + R''(f). \tag{11}$$

Formula lui Gauss. Prin schimbarea funcţiei de sub integrală cu un polinom algebric de interpolare, construit pe n noduri de interpolare, se obţine o astfel de formulă de cuadratură, pentru care restul se anulează dacă funcţia de sub integrală este un polinom de grad nu mai mare ca $n-1$.

Aşa cum se ştie, în cazul formulei lui Newton-Cotes, cu un număr par de noduri, restul se anulează dacă funcţia de sub integrală este un polinom de gradul n.

Se poate întîmpla ca pentru o altă distribuţie a nodurilor, gradul polinomului să crească.

Prin folosirea aceluiaşi număr de noduri vom spune că o formulă de integrare numerică este mai precisă decît alta, dacă gradul polinomului care înlocuit în locul funcţiei de sub integrală ce dă restul nul, este mai mare.

Formula de integrare numerică a lui Gauss se obţine punînd condiţia ca gradul polinomului care înlocuit sub integrală să dea restul zero, să fie maxim. Ea are următoarea formă

$$\int_a^b p(x)f(x)dx = c_1^{(n)}f(x_1) + c_2^{(n)}f(x_2) + \ldots + c_n^{(n)}f(x_n). \tag{12}$$

Pentru determinarea coeficienţilor $c_i^{(n)}(i = 1, 2, \ldots, n)$ se ajunge la un sistem de ecuaţii care ne arată că gradul maxim de precizie al formulei este $2n-1$.

Nodurile formulei lui Gauss expusă în lucrarea [1] din studiul ei, rezultă că sînt rădăcinile polinomului lui Legendre de gradul n.

$$L_n(x) = \frac{n!}{(2n)!}\frac{d^n}{dx^n}[(x-a)^n(x-b)^n]. \tag{13}$$

Dacă se face schimbarea de variabilă

$$x = \frac{a+b}{2} + \frac{b-a}{2}t,$$

atunci se reduce intervalul de integrare $[a, b]$ la $[-1, 1]$, iar $L_n(x)$ are atunci forma

$$L_n(x) = \frac{1}{2^n n!}\frac{d^n}{dx^n}[(x^2-1)^n] \tag{14}$$

În polinoamele $L_n(x)$, ale căror proprietăţi sînt bine cunoscute, avem relaţia de recurenţă

$$(n+1)L_{n+1}(x) = (2n+1)xL_n(x) + nL_{n-1}(x) \tag{15}$$

care permite calculul succesiv al coeficienţilor polinomului $L_n(x)$ cunoscînd că $L_0(x) = 1$ şi $L_1(x) = x$. Prin urmare determinarea nodurilor formulei de integrare a lui Gauss este echivalentă cu determinarea celor n rădăcini reale şi distincte ale polinomului lui Legendre.

Dacă se consideră $p(x) = 1$, atunci formula lui Gauss devine

$$\int_{-1}^{1} f(x)dx = \sum_{i=1}^{n} c_i^{(n)}f(x_i) + R(f) \tag{16}$$

unde

$$c_i^{(n)} = \frac{(n!)^4(b-a)^{2n+1}}{[(2n!)]^2(x_k-a)(b-x_k)L_n'^2(x_k)} \tag{17}$$

$$i = 1, 2, \ldots, n.$$

$b = 1$, $a = -1$, $L_n'(x)$ fiind derivata polinomului lui Legendre de gradul n.

Se poate arăta simplu că $c_k^{(n)} = c_{n-k+1}^{(n)}$.

Formula lui Hermite. Dacă luăm în calitate de $p(x)$ funcţia

$$p(x) = \frac{1}{\sqrt{1-x^2}}$$

iar în calitate de interval de integrare intervalul $[-1, 1]$, atunci se obţine formula de integrare numerică

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,dx = \sum_{i=1}^{n} c_i^{(n)} f(x_i) + R(f) \tag{18}$$

în ipoteza că integrala din membrul stîng există. De data aceasta x_i vor fi zerourile polinomului lui Cebîşev de grad n

$$\frac{1}{2^{n-1}}\,T_n(x) = \frac{1}{2^{n-1}} \cos\,(n \arccos x),$$

prin urmare

$$x_i = \cos\frac{2i-1}{2n}\,\pi$$

iar

$$c_i^{(n)} = \frac{\pi}{n}\ (i = 1, 2, \ldots n).$$

Formula obţinută este un caz particular al formulei lui Gaus, şi poartă numele lui Hermite. Ea are forma

$$\int_{-1}^{1} \frac{f(x)}{\sqrt{1-x^2}}\,dx = \frac{\pi}{n}\sum_{i=1}^{n} f(x_i) + R(f). \tag{19}$$

Ne vom opri aci cu consideraţiile generale asupra formulelor de integrare numerică pe care le vom avea în vedere, urmînd ca în continuare să construim schemele logice operatoriale ale programelor algoritmilor descrişi mai sus. Nu vom preciza pentru aceasta o anumită maşină dată, deoarece programul se face cu uşurinţă pentru oricare maşină dacă se ştie schema logică operatorială a programului unui algoritm dat.

3. Să considerăm formula lui Newton-Cotes

$$\int_{c}^{d} f(x)dx = (d-c)\sum_{i=1}^{n} I_{i,k}^{(n)} f(a+ih) + R(f).$$

unde asupra funcţiei $f(x)$ facem ipoteza că este calculabilă printr-un algoritm $\mathcal{A}$ şi că rezultatul aplicării algoritmului $\mathcal{A}$ la funcţia $f(x)$ în nodurile $x_1, x_2, \ldots, x_n$ sînt numerele $a_1^{(r)}, a_2^{(r)} \ldots a_n^{(r)}$ *).

*) Nu vom preciza algoritmul $\mathcal{A}$ de calcul al funcţiei $f(x)$ deoarece el are un caracter particular şi se precizează de la caz la caz.

Pentru calculul integralei după această formulă este necesar în primul rînd să calculăm coeficienţii $I_{i,k}^{(n)}$. Pentru aceasta vom preciza algoritmul de integrare al polinomului de sub integrală

$$P_i(y) = \frac{(y-1)(y-2)\ldots(y-n)}{y-i}$$

Coeficienţii polinomului $(y-i)P_i(y)$ sînt daţi de sumele următoare

$$a_0 = S_n^{(0)} = 1$$
$$a_1 = S_n^{(1)} = 1 + 2 + \ldots + n$$
$$a_2 = S_n^{(2)} = 1\cdot 2 + 1\cdot 3 + \ldots + 1\cdot n + 2\cdot 3 + \ldots + 2\cdot n + \ldots + (n-1)n$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$a_n = S_n^{(n)} = 1\cdot 2 \ldots (n-1)n.$$

Se poate arăta uşor că avem

$$S_n^{(2)} = n\,S_{n-1}^{(1)} + S_{n-1}^{(2)}$$

şi în general, prin inducţie, că

$$S_n^{(p)} = n\,S_{n-1}^{(p-1)} + S_{n-1}^{(p)}.$$

Se vede deci că coeficienţii polinomului $(y-i)P_i(y)$ se găsesc oarecum analog cu coeficienţii binomiali din regula următoare:

Dacă s-au calculat coeficienţii polinomului $(y-i)P_i^{(k)}(y)$, unde prin k am notat gradul polinomului $P_i(y)$, atunci coeficienţii polinomului $(y-i)P_i^{(k+1)}(y)$ se determină astfel:

Fie

$$a_0, a_1, a_2, \ldots, a_{k-1}, a_k$$

coeficienţii daţi. Coeficienţii calculaţi vor fi

$$a_0, -(k+1)a_0 + a_1, \quad -(k+1)a_1 + a_2, \ldots, \quad -(k+1)a_{k-1} + a_k, \quad -(k+1)a_k.$$

Pentru calculul coeficienţilor polinomului $\frac{P_i(y)}{y-i}$ vom proceda prin împărţire cu ajutorul schemei lui Horner.

Subliniem de la început că parametrii schemelor logice operatoriale care vor urma au o valoare locală, adică dacă $i, j, k, \ldots$ sînt parametrii în schema (S), cu valorile precizate pentru operatorul A, atunci ei se pot folosi şi pentru operatorul A′ din aceeaşi schemă, valorile lor fiind însă precizate relativ la acest operator.

Schema logică operatorială S_0 a programului algoritmului descris mai sus va fi

$$\overset{3}{\downarrow}\overset{1}{\downarrow} A_1^{(k)} p(k \leqq j) \overset{2}{\uparrow} F(j) O \overset{1}{\uparrow}\overset{2}{\downarrow} p(j < n) \overset{4}{\uparrow} [[\beta^{(0)}] - [1] \Rightarrow (\beta^{(0)})] \times$$

$$\times F^{-j}(k) F(j) O \overset{3}{\uparrow}\overset{4}{\downarrow}\overset{5}{\downarrow} A_2^{(i,k)} p(k \leqq n) \overset{6}{\uparrow} F(k) O \overset{5}{\uparrow}\overset{6}{\downarrow} S_1$$

unde operatorii $A_1^{(k)}$, $A_2^{(ik)}$ sînt determinaţi de relaţiile

$$A_1^{(k)} \equiv [\beta^{(0)}][\alpha_k] + [\alpha_{k+1}] \Rightarrow (\alpha_{k+1}) ; \ (k = 0, 1, \ldots n)$$

$$A_2^{(i,k)} \equiv [i][\beta_k] + [\alpha_{k+1}] \Rightarrow (\beta_{k+1}) ; \ [\beta_0] = [\alpha_0]$$

iar k este parametrul ciclului care calculează la un j dat coeficienţii polinomului $(y - i)P_i^{(k)}(y)$. Am folosit mai sus notaţiile următoare:

$[\alpha]$ = numărul conţinut în celula α.

(α) = numărul celulei

În felul acesta se vede că $[\alpha_0] = a_0$, $[\alpha_k] = a_k$, unde avem iniţial $a_0 = 1$, $a_1 = -1$, iar $[\beta^{(0)}] = -2$.

Referitor la operatorul $A_2^{(i,k)}$ avem iniţial $i = 1$, iar (β_k) sînt un şir de celule care păstrează coeficienţii polinomului $\frac{P_i(y)}{(y_i - i)}$.

Rezultatul calculului după schema logică operatorială (S_0) îl formează pe de o parte coeficienţii polinomului $\prod_{i=1}^{n} (y - i)$, iar pe de altă parte coeficienţii polinomului $\frac{\prod_{i=1}^{n} (y - i)}{y - i}$.

Schema logică operatorială al programului algoritmului de calcul al coeficienţilor $I_{i,k}^{(n)}$ este de forma

$$\overset{3}{\downarrow} [[\beta_0] \Rightarrow (\alpha^{(j)})] [[\alpha^{(j)}] : [n] \Rightarrow (\alpha^{(j)})] \overset{1}{\downarrow} A_1^{(l,j)} p(l \leqq n - 1) \overset{2}{\uparrow} \times$$

$$\times F(l) O \overset{1}{\uparrow}\overset{2}{\downarrow} p(j < 2) \overset{4}{\uparrow} F(j) O \overset{3}{\uparrow}\overset{4}{\downarrow} [[\alpha^{(2)}] - [\alpha^{(1)}] \Rightarrow (\alpha)] \times$$

$$\times \overset{5}{\downarrow}\overset{7}{\downarrow} A_2^{(k)} p(k < i - 1) \overset{6}{\uparrow} F(k) O \overset{5}{\uparrow}\overset{6}{\downarrow} [[n - 1] \Rightarrow (i - 1)] p(j < 1) \overset{8}{\uparrow} \times \qquad (S_1)$$

$$\times F(j) O \overset{7}{\uparrow}\overset{8}{\downarrow} A_3^{(i)} p(i < n) \overset{9}{\uparrow} F(i) O \overset{(S_0)}{\uparrow}\overset{9}{\downarrow} S_2.$$

În această schemă s-a folosit parametrul j pentru a delimita calculul polinomului integrat între cele două limite. Pentru $j = 1$ se calculează polinomul în limita inferioară, iar pentru $j = 2$, în cea superioară. Operatorii $A_1^{(l,j)}$, $A_2^{(k)}$ şi $A_3^{(i,k)}$ sînt definiţi de relaţiile

$$A_1^{(l,j)} \equiv [\alpha^{(j)}][y^{(j)}] + [\alpha_l] : [n - l] \Rightarrow (\alpha^{(j)})$$

$$[\alpha^{(j)}] \cdot [y^{(j)}] \Rightarrow (\alpha^{(j)}),$$

unde $[y^{(j)}] = 0$ pentru $j = 1$ în cazul formulei de tip deschis, şi $[y^{(j)}] = 1$ pentru $j = 1$ în cazul formulei de tip închis, iar $[y^{(j)}] = n + 1$ în primul caz pentru $j = 2$ şi $[y^{(j)}] = n$ pentru $j = 2$ în al doilea caz.

$$A_2^{(k)} \equiv [\delta_j] \ [k] \Rightarrow (\delta_j) ; \ j = 0, 1 ; \quad [\delta_j] = 1 \text{ iniţial.}$$

Pentru $j = 0$ se realizează $(i - 1)!$, iar pentru $j = 1$ se realizează $(n - i)!$

$$A_3^{(i,k)} \equiv [\delta_0] \ [\delta_1] \Rightarrow (\delta_0)$$

$$[1] : [\delta_0] \Rightarrow (\delta_0)$$

$$[-1]^{n-i} \ [\delta_0] \Rightarrow (\gamma_i^{(k)})$$

$$[\gamma_i^{(k)}] \ [\alpha] \Rightarrow (\gamma_i^{(k)})$$

În acest fel, în şirul de celule $(\gamma_i^{(k)})$ apar coeficienţii $I_{i,k}^{(n)}$.

Algoritmul de calcul al integralei folosind formula lui Newton-Cotes de tip închis $(k = 0)$, sau de tip deschis $(k = 1)$, are următoarea schemă logică operatorială a programului:

$$\overset{1}{\downarrow} A_r^{(i)} p(i \leqq n) \overset{2}{\uparrow} F(i) O \overset{1}{\uparrow}\overset{2}{\downarrow} A_r p(|[\gamma] - [M]| \geqq \varepsilon) \overset{3}{\uparrow} \times \qquad (S_2)$$

$$\times [[\gamma] \Rightarrow (M)] F^{-n}(i) F(\lambda n) O \overset{c_1}{\uparrow} \overset{3}{\downarrow} Op,$$

unde operatorii $A_r^{(i)}$, A_r sînt definiţi în felul următor:

$$A_r^{(i)} \equiv [\gamma_i^{(k)}][a_i^{(r)}] + [r] \Rightarrow (\gamma)$$

$$A_r \equiv [\gamma][d - c] \Rightarrow (\gamma).$$

M este un număr arbitrar ales pentru compararea preciziei rezultatului aşa cum am amintit în prima parte, $F(\lambda n)$ este operatorul de mărire a numărului n de un număr λ de ori, iar c_1 este prima instrucţiune a programului.

Deşi formulele trapezului şi a lui Simpson sînt cazuri particulare, este bine ca să se studieze din punct de vedere al programării, separat, deoarece

prezintă avantajul simplităţii lor. Să scriem acum formula trapezului sub forma următoare :

$$\int_a^b f(x)dx = \frac{b-a}{2n}\,[f(x_0) + f(x_n) + 2(f(x_1) + \ldots + f(x_{n-1}))] + R(f). \qquad (20)$$

Schema logică operatorială a programului algoritmului descris de această formulă are forma

$$\overset{4}{\downarrow} A_1 \overset{1}{\downarrow} A_2^{(i)}\, p(i \leqq n-1) \overset{2}{\uparrow} F(i) O \overset{1}{\uparrow} A_3\, p(|[\alpha] - [M]| \geqq \varepsilon) \overset{3}{\uparrow} \times$$
$$\times\, [[\alpha] \Rightarrow (M)]\, F^{-n+1}(i)\, F(\lambda n)\, O \overset{4}{\uparrow} \overset{3}{\downarrow} Op \qquad (S_3)$$

operatorii A_1, $A_2^{(i)}$, A_3 fiind definiţi de relaţiile :

$$A_1 \equiv [a_0^{(r)}] + [a_n^{(r)}] \Rightarrow (\alpha)$$
$$A_2^{(i)} \equiv [a_i^{(r)}] + [\beta] \Rightarrow (\beta)$$
$$A_3 \equiv [\beta][2] \Rightarrow (\beta)$$
$$[\alpha] \cdot [\beta] \Rightarrow (\alpha)\,;\ [\alpha]\,([d-c] : [2n]) \Rightarrow (\alpha).$$

Cînd precizia nu este satisfăcută pentru un n dat, şi-l multiplicăm pe acesta de λ ori, se iveşte problema calculului unor noi valori ale funcţiei $f(x)$ în aceste puncte bine precizate, pe lîngă cele de mai sus. Această problemă este rezolvată de către algoritmul $\mathcal{A}$ de care am vorbit mai sus.

Formula lui Simpson o vom scrie de asemenea pentru cazul a $2n+1$ noduri $x_0, x_1, \ldots, x_{2n+1}$ sub forma

$$\int_a^b f(x)dx = \frac{b-a}{6n}\,\{[f(x_0) + f(x_{2n})] + 2[f(x_2) + f(x_4) + \ldots +$$
$$+ f(x_{2n-2})] + 4[f(x_1) + f(x_3) + \ldots + f(x_{2n+1})]\} + R(f). \qquad (21)$$

Schema logică operatorială a programului algoritmului descris de această formulă este :

$$\overset{5}{\downarrow} A_1 \overset{1}{\downarrow} A_2^{(2i)}\, p(i \leqq n-1) \overset{2}{\uparrow} F(i) O \overset{1}{\uparrow} \overset{2}{\downarrow} A_3 F^{-(n-1)}(i) \overset{3}{\downarrow} A_4^{(2i-1)} \times$$
$$\times\, p(i < n) \overset{4}{\uparrow} F(i) O \overset{3}{\uparrow} \overset{4}{\downarrow} A_5\, p(|[\alpha] - [M]| \geqq \varepsilon) \overset{6}{\uparrow} [\alpha] \Rightarrow (M)] \times$$
$$\times\, F^{-n}(i) F(\lambda n) O \overset{5}{\uparrow} \overset{6}{\downarrow} Op. \qquad (S_4)$$

operatorii A_1, $A_2^{(2i)}$, A_3, $A_4^{(2i-1)}$ şi A_5 fiind definiţi de relaţiile următoare :

$$A_1 \equiv [a_0^{(r)}] + [a_{2n}^{(r)}] \Rightarrow (\alpha)$$
$$A_2^{(2i)} \equiv [a_{2i}^{(r)}] + [\beta_1] \Rightarrow (\beta_1) \quad [\beta_1] = 0 \text{ iniţial},\ i = 1, 2, \ldots n-1$$
$$A_3 \equiv [2] \cdot [\beta_1] \Rightarrow (\beta_1)$$
$$A_4^{(2i-1)} \equiv [a_{2i-1}^{(r)}] + [\beta_2] \Rightarrow (\beta_2) \quad [\beta_2] = 0 \text{ iniţial},\ i = 1, 2, \ldots, n$$
$$A_5 \equiv \{[\alpha] + [\beta_1] + [4][\beta_2]\} \cdot \{[d-c] : [6n]\} \Rightarrow (\alpha)$$

În continuare vom avea în vedere formule de integrare numerică de precizie maximă. Fie pentru aceasta formula lui Gauss (16) [1] cu coeficienţii (17). Aşa cum am văzut, avem nevoie pentru această formulă în primul rînd de zerourile polinomului lui Legendre. Acestea le vom calcula folosindu-ne de o metodă iterativă a lui Laguerre [2]. Aceasta constă în următoarele :

Fie polinomul $P(x)$ dat sub forma

$$P(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + a_n \qquad (22)$$

şi $x_1, x_2, \ldots, x_n$ rădăcinile lui, despre care ştim că sînt reale şi distincte. Vom căuta un polinom

$$A_0 X^n + A_1 X^{n-1} + \ldots + A^{n-1} X + A_n \qquad (23)$$

care are proprietatea că rădăcinile lui sînt în următoarea relaţie cu rădăcinile polinomului (22), $X_i = x_i^m$, unde $m = 2^k$.

Numerele A_i $(i = 0, 1, \ldots, n)$ se determină din relaţia

$$A_i = (-1)^i B^i, \qquad (24)$$

unde numerele B se determină din următoarea relaţie de recurenţă :

$$B_i^{(k+1)} = 2\sum_{s=1}^{i} (-1)^s B_{i-s}^{(k)} B_{i+s}^{(k)} + B_i^{(k)2} \qquad (25)$$

ştiind că

$$B_i^{(1)} = 2\sum_{s=1}^{i} (-1)^s a_{i-s} a_{i+s} + a_i^2. \qquad (26)$$

Dacă $m=2^k$ este mare, atunci rădăcinile polinomului (23) sînt foarte mult distanțate una de alta. După formulele lui Viete, avem

$$X_1 \simeq \frac{B_1}{B_0}$$

$$X_1X_2 \simeq \frac{B_2}{B_0}\ ;\ X_2 \simeq \frac{B_2}{B_1}$$

. . . . . .

$$X_1X_2 \ldots X = \frac{B_i}{B_0}\ ;\ X_i \simeq \frac{B_i}{B_{i-1}}$$

. . . . . .

iar $$x_i = \sqrt[m]{X_i}, \quad i = 1, 2, \ldots n.$$

Deoarece m este o putere a lui 2, rădăcina se poate extrage succesiv destul de ușor. Inconvenientul acestei metode este faptul că numerele, B_i sînt mari, astfel că ele pot depăși diapazonul mașinii, după un număr relativ mic de pași. De aceea, cu o soluție inițială determinată cu ajutorul acestei metode, vom proceda în continuare pentru găsirea soluției cu precizia cerută, cu ajutorul metodei iterative a lui Laguerre, dată de relația

$$X = x - \frac{nP(x)}{P'(x) + \sqrt{H(x)}} \tag{27}$$

unde $H(x) = (n-1)^2P'^2(x) - n(n-1)P(x)P''(x)$.

Dacă x este o valoare aproximativă a unei rădăcini a polimonului $P(x)$, atunci X dat de relația (27) este o valoare mai apropiată de aceeași rădăcină.

Cu această metodă, în cele ce urmează vom evalua zerourile polinomului lui Lependre cu precizia ε pe intervalul $[-1, 1]$.

Să considerăm polinomul lui Legendre dat de relația de recurență (15), unde $L_0(x) = 1$, iar $L_1(x) = x$. Vom înscrie coeficienții polinomuiui $L_{n-1}(x)$ celulele (α_i), $i = 0, 1, \ldots, n-1$, iar coeficienții polinomului L_{n-2} în celulele (b_i), $i = 0, 1, \ldots, n-2$. Coeficienții polinomului $L_n(x)$ îi vom obține atunci în celulele (α_i), $i = 0, 1, \ldots, n$, urmînd ca ceiulele (a_i) (b_i), după calcularea coeficienților polimonului $L_n(x)$, să fie eliberate spre a fi disponibile.

Schema logică operatorială a programului algoritmului după care se calculează coeficientul polinomului $L_n(x)$, este de forma

$$\overset{3}{\uparrow} \mathrm{A}_1^{(k)} \mathrm{A}_2^{k)} \overset{1}{\downarrow} \mathrm{A}_3^{(k,i)} p(i < k+1) \overset{2}{\uparrow} \mathrm{F}(i)\, \mathrm{O} \overset{1}{\uparrow} \overset{2}{\downarrow} p(k < n) \overset{4}{\uparrow} \times$$

$$\times \mathrm{F}^{-k}(i)[[a_i] \Rightarrow (b_i)]\,[[a_i] \Rightarrow (a_i)]\,\mathrm{O} \overset{3}{\uparrow} \overset{4}{\downarrow} S_6, \qquad S_5$$

unde operatorii $A_1^{(k)}$, $A_2^{(k)}$, $A_3^{(k,i)}$ sînt definiți de relațiile

$$\mathrm{A}_1^{(k)} \equiv \{[2k+1] : [k+1]\} \cdot [a_0] \Rightarrow (\alpha_0)$$

$$\mathrm{A}_1^{(k)} \equiv \{[2k+1) : [k+1]\} \cdot [a_1] \Rightarrow (\alpha_1)$$

$$\mathrm{A}_1^{(k)} \equiv \{[2k+1] : [k+1)\}[a_i] - \{[k] : [k+1]\} \cdot [b_{i-2}] \Rightarrow (\alpha_i),$$
$$i = 2, 3, \ldots, k.$$

Se vede că după fiecare ciclu după k, executat de acești operatori i se mărește, și că la ciclul următor în calitate de $L_{k-1}(x)$ se folosește polinomul cu coeficienții noi calculați, iar în calitate de $L_{k-2}(x)$ polinomul care la ciclul anterior a jucat rolul lui L_{k-1}. În acest fel, în șirul de celule (α_i), la sfîrșitul calculelor vom avea coeficienții polinomului lui Legendre de gradul $n_1, a_0, a_1, \ldots, a_n$.

Schema logică operatorială a programului algoritmului după care se calculează zerourile acestui polinom, are forma

$$\overset{16}{\downarrow} [[\alpha_0]^2 \Rightarrow (\beta_0)] \overset{3}{\downarrow} \overset{1}{\downarrow} \mathrm{A}_1^{(i,k)} p(k \leqq i) \overset{2}{\uparrow} \mathrm{F}(k)\ [[\alpha][-1] \Rightarrow (\alpha)] \times$$

$$\times \mathrm{O} \overset{1}{\uparrow} \overset{2}{\downarrow} p([B_i] < [\beta]) \overset{4}{\uparrow} \overset{5}{\downarrow} p(i \leqq n) \overset{6}{\uparrow} [[-1] \Rightarrow (\alpha)]\, \mathrm{F}^{-i}(k) \times$$

$$\times \mathrm{F}(i)\, \mathrm{O} \overset{3}{\uparrow} \overset{4}{\downarrow} [[B_i] \Rightarrow [\beta)]\, \mathrm{O} \overset{5}{\uparrow} \overset{6}{\downarrow} [[N] + [1] \Rightarrow (N)] \overset{7}{\downarrow} \overset{11}{\downarrow} p([x_i^{(0)}] = [x_i^{(n)}] \times$$

$$\times \overset{8}{\uparrow} p(i \leqq n) \overset{13}{\uparrow} \mathrm{F}(i)\, \mathrm{O} \overset{7}{\uparrow} \overset{8}{\downarrow} [[x_i^{(0)}] \Rightarrow (x_i^{(1)})]\, \mathrm{A}_2^{(i)} [[N] \Rightarrow (P)] \times \qquad (S_6)$$

$$\times \overset{9}{\downarrow} \mathrm{A}_3^{(i)} [[P] - [1] \Rightarrow (P) p([P] > 0) \overset{10}{\uparrow} \mathrm{O} \overset{9}{\uparrow} \overset{10}{\downarrow} \times$$

$$\times p(|[x_i^{(1)}] - [x_i^{(0)}]| < \varepsilon) \overset{14}{\uparrow} [[x_i^{(1)}] \Rightarrow (x_i^{(0)})][[M] + [1] \Rightarrow (M)] \times$$

$$\times \overset{14}{\downarrow} p(i \leqq n) \overset{12}{\uparrow} \mathrm{F}(i) \mathrm{O} \overset{11}{\uparrow} \overset{12}{\downarrow} \overset{13}{\downarrow} p([M] = [n]) \overset{15}{\uparrow} \mathrm{O} \overset{S_7}{\uparrow} \times$$

$$\times \overset{15}{\downarrow} p([\beta] < d) \overset{17}{\uparrow} [[c_1] + [n+1] \Rightarrow [c_1]][c_2] + [n+1] \Rightarrow (c_2)] \times$$

$$\times \mathrm{F}(i) \mathrm{O} \overset{16}{\uparrow} \overset{17}{\downarrow} S_7.$$

Operatorii $A_1^{(k)}$, A_2, $A_3^{(i)}$, $A_4^{(i,k)}$ sînt definiți de relațiile

$$A_1^{(k)} \equiv [\alpha]\cdot[\alpha_{i-k}][\alpha_{i+k}] + [B_i] \Rightarrow (B_i)\,; \quad [\alpha] = -1 \text{ inițial}, \quad [B_1] = 0 \quad ,,$$

$$A_2 \equiv [2]\cdot[B_i] \Rightarrow (B_i) \qquad k = 1, 2, \ldots, i.$$

$$A_3^{(i)} \equiv [B_i] : [B_{i-1}] \Rightarrow (x_i^{(1)}) \qquad i = 1, 2, \ldots, n.$$

$$A_4^{(i,k)} \equiv \sqrt{[x_i^{(1)}]} \equiv (x_i^{(1)})$$

extrage rădăcina succesiv de un număr k de ori, unde k se determină cu ajutorul unui numărător.

Inițial în celulele

$$(x_1^{(0)}),\ (x_2^{(0)}),\ \ldots,\ (x_n^{(0)}) \text{ și } (x_1^{(0)}),\ (x_2^{(1)}),\ \ldots,\ (x_n^{(1)})$$

vom înscrie două numere care nu sînt rădăcini ale polinomului considerat și nici nu aproximează rădăcinile acestui polinom. În cazul nostru ar putea fi de exemplu $[x_1^{(0)}] = 2$, $[x_i^{(0)}] = 3$; $i = 1, 2, \ldots, n$. După primul ciclu, care calculează numerele $B_i^{(1)}$, în celulele $(x_i^{(1)})$ se obțin numerele $[x_i]$ rădăcini ale polinomului $L_n(x)$ într-o primă aproximație, iar în celulele $(x_i^{(0)})$ numerele $[x_i^{(0)}] = 3$. În acest fel, trecerea la algoritmul descris de metoda lui Laguerre se face atunci și numai atunci cînd $\max_i \{B_i\} \geqq d$, unde prin d am notat diapazonul mașinii. Dacă nu este nevoie de această trecere, condiția logică $p([M] = [n])$ ne spune că am obținut precizia cerută pentru toate rădăcinile și se trece la calcularea integralei. Dacă însă condiția logică $p([M] = [n])$ nu este îndeplinită, atunci trecerea depinde de condiția amintită, adică $\max_i \{B_i\} \geqq d$. Prin c_1 și c_2 am notat numerele celulelor care conțin prima, respectiv a doua comandă a programului. Algoritmul descris de metoda lui Laguerre se aplică numai la acele zerouri care nu sînt suficient de precis calculate și care se găsesc în celulele $(x_i^{(1)})$. Punerea lor în evidență se face relativ ușor, verificînd condiția logică $p([x_i^{(0)}] = [x_i^{(1)}])$. Schema logică operatorială a programului care realizează acest lucru, este de forma

$$\overset{1}{\downarrow} p([x_i^{(0)}] = [x_i^{(1)}]) \overset{2}{\uparrow} \overset{3}{\downarrow} p(i \leqq n) \overset{4}{\uparrow} F(i)[[N] + [1] \Rightarrow (N)] \times$$

$$\times\ O \overset{1}{\uparrow} \overset{2}{\downarrow} [[x_i^{(1)}] \Rightarrow (x_j)][[i] \Rightarrow (n_j)] O \overset{3}{\uparrow} \overset{4}{\downarrow} S_7. \qquad (S'_6)$$

În felul acesta, algoritmul descris de metoda lui Laguerre se va aplica numai asupra numerelor $[x_j]$, urmînd ca rezultatele să fie înscrise în celulele $(x_{n_j}^{(0)})$.

Pentru executarea acestui algoritm, avem nevoie de un algoritm de calcul al derivatelor polinomului și al valorilor lor în punctele $[x_j]$. Programul acestui algoritm are următoarea schemă logică operatorială:

$$\overset{3}{\downarrow} \overset{1}{\downarrow} A_1^{(i,r)} p(i \leqq n - r) \overset{2}{\uparrow} F(i)\, O \overset{1}{\uparrow} \overset{2}{\downarrow} p(r < k) \overset{4}{\uparrow} F^{-(n-r)}(i) \times$$

$$\times\ F(r)\, O \overset{3}{\uparrow} \overset{4}{\downarrow} S'_7. \qquad (S_7)$$

Operatorul $A_1^{(i,r)}$ se definește în felul următor:

$$A_1^{(i,r)} \equiv [\alpha_i^{(r)}][n - i] \Rightarrow (\alpha_i^{(r)}) \qquad i = 0, 1, 2, \ldots\ n - r$$

$$r = 0, 1, \ldots, k - 1$$

Se vede că pentru $r = 0$, avem de-a face cu celulele $(\alpha_0^{(0)})$, $(\alpha_1^{(0)})$, $\ldots$ $\ldots$, $(\alpha_n^{(0)})$, în care sînt păstrați coeficienții polinomului dat. Cînd r ia valorile $1, 2, \ldots, k$, în celulele $\{(\alpha_i^{(r)})\}$ se obțin coeficienții derivatelor succesive pînă la ordinul k al polinomului dat.

În cazul nostru, $r = 0, 1, 2$.

Programul algoritmului după care se calculează valorile polinomului și al derivatelor sale succesive în punctele $[x_j]$, este dat de schema logică operatorială (S'_7).

$$\overset{3}{\downarrow} \overset{5}{\downarrow} [\alpha_0^{(r)}] \Rightarrow (\alpha^{(r)})] \overset{1}{\downarrow} A_1^{(i,r,j)} p(n \leqq i) \overset{2}{\uparrow} F(i)\, O \overset{1}{\uparrow} \times$$

$$\times\ \overset{2}{\downarrow} p(r \leqq k) \overset{4}{\uparrow} F^{-(n-r-1)}(i)\, F(r)\, O \overset{3}{\uparrow} \overset{4}{\downarrow} p(j < n - m) \overset{6}{\uparrow} \times$$

$$\times\ F^{-k}(r)\, F^{-(n-j+1)}(i)\, F(j)\ O \overset{5}{\uparrow} \overset{6}{\downarrow} S_8, \qquad (S'_7)$$

unde operatorul $A_1^{(i,r,j)}$ este definit de relațiile

$$A_1^{(i,r,j)} \equiv [\alpha^{(r,i,j)}]\,[x_j] + [\alpha_i^{(r)}] \Rightarrow (\alpha^{(r,j)})$$

$$r = 0, 1, 2, \ldots, k$$

$$j = 1, 2, \ldots, n - m; \quad i = 0, 1, 2, \ldots, n,$$

unde prin m am notat numărul rădăcinilor calculate cu precizia cerută cu ajutorul programului dat de schema logică operatorială (S_6). În felul acesta, în celulele $\{(\alpha^{(r,j)})\}$ vom avea calculate valorile polinomului dat și primele sale k derivate în punctele $[x_j]$.

Vom putea da acum schema logică operatorială a programului care realizează algoritmul de găsire a zerourilor unui polinom după metoda lui Laguerre. Această schemă are forma

$$\overset{4}{\downarrow} \overset{5}{\downarrow} A_1^{(j)} A_2^{(j)} A_3^{(j)} A_4^{(j)} A_5^{(j)} A_6^{(j)} p(|[x_j] - [\gamma^{(2)}]| < \varepsilon) \overset{3}{\uparrow} \times \qquad (S_8)$$

$$\times\ [[x_j] \Rightarrow (x_j^{(0)})] p(j < n - m) \overset{2}{\uparrow} F(j)\, O \overset{4}{\uparrow} \overset{3}{\downarrow} [[\gamma^{(2)}] \Rightarrow (x_j)] O \overset{6}{\uparrow} \overset{2}{\downarrow} S_9$$

Operatorii $A_1^{(j)}A_2^{(j)}A_3^{(j)}A_4^{(j)}A_5^{(j)}A_6^{(j)}$ sînt definiți de relațiile

$$A_1^{(j)} \equiv [n-1]^2 \cdot [\alpha^{(1,j)}]^2 - [n] \cdot [n-1][\alpha][\alpha^{(0,j)}] \Rightarrow (\gamma^{(1)})$$

$$A_2^{(j)} \equiv \sqrt{\gamma^{(1)}} \Rightarrow (\gamma^{(1)})$$

$$A_3^{(j)} \equiv [\gamma^{(1)}] + [\alpha^{(1,j)}] \Rightarrow [\gamma^{(1)}]$$

$$A_4^{(j)} \equiv [n][\alpha^{(0,j)}] \Rightarrow (\gamma^{(2)})$$

$$A_5^{(j)} \equiv [\gamma^{(2)}] : [\gamma^{(1)}] \Rightarrow (\gamma^{(2)})$$

$$A_6^{(j)} \equiv [x_j] - [\gamma^{(2)}] \Rightarrow (\gamma^{(2)}).$$

Rezultatul acestui program îl constituie zerourile polinomului lui Legendre calculate cu precizia ε cerută care se găsesc în celulele $(x_1^{(0)})$, $(x_2^{(0)})$, ..., $(x_n^{(0)})$.

Coeficienții (17), în cazul nostru, se pot scrie sub forma

$$c_i^{(n)} = \frac{1}{2}\left[\left(\frac{2}{n+1}\right)\left(\frac{4}{n+2}\right)\cdots\left(\frac{2(n-1)}{2n-1}\right)\right]^2 \frac{1}{(1-x_j^2)L_n'^2(x_j^{(0)})}.$$

Schema logică operatorială a programului algoritmului descris de această formulă, este de forma

$$\downarrow^1 \downarrow^4 A_1^{(i)} p(i \leqq n-1) \uparrow^2 F(i)\, O \uparrow^1 \downarrow^2 A_2 A_3^{(i)} p(i < n) \uparrow^3 \times$$

$$\times F^{-(n-1)}(i)\, F(j)\, O \uparrow^4 \downarrow^3 S_{10}, \qquad (S_9)$$

unde

$$A_1^{(i)} = \{[2i] : [n+i]\} \cdot [\alpha] \Rightarrow (\alpha) \qquad [\alpha] = 1 \text{ inițial}$$
$$i = 1, 2, \ldots, n-1$$

$$A_2 = [\alpha]^2 : [2] \Rightarrow (\alpha)$$

$$A_3^{(i)} \equiv \{1 - [x_i^{(0)}]^2\}[\alpha^{(1,j)}]^2 \Rightarrow (\beta);\ [\alpha] : [\beta] \Rightarrow (\nu^{(i)}).$$

Rezultatul acestui program îl constituie coeficienții $c_i^{(n)}$, $i = 1, 2, \ldots, n$ înscriși în celulele $(\nu^{(i)})$, $i = 1, 2, \ldots, n$.

Schema logică operatorială a programului algoritmului descris de formula (16) a lui Gauss, este următoarea:

$$\downarrow^1 A_1^{(i)} p(i \leqq n) \uparrow^2 F(i)\, O \uparrow^1 \downarrow^2 p(|[\gamma] - [M]| \geqq \varepsilon) \uparrow^3 \times$$

$$\times F^{-n}(i)\, F(\lambda n)\, O \uparrow^{S_8} \downarrow^3 Op, \qquad (S_{10})$$

unde

$$A_1^{(i)} \equiv [a_i^{(r)}] \cdot [\nu^{(i)}] + [\gamma] \Rightarrow (\gamma)$$
$$[\gamma] = 0 \text{ inițial}$$

iar

$$\lambda \leqq 2.$$

Formula lui Hermite. Revenind acum la formula (19), se observă că nodurile x_i, $i = 1, 2, \ldots, n$, se calculează din relația

$$x_i = \cos\frac{2i-1}{2n}\pi.$$

Să presupunem că (S') este schema logică operatorială a programului standard al algoritmului de calcul al funcției cos x, prezent în biblioteca de subprogram standard a majorității mașinilor electronice. Dacă notăm cu y_i mărimea $\frac{2i-1}{2n}\pi$, atunci schema logică operatorială a programului algoritmului de calcul al integralei după formula lui Hermite este următoarea:

$$\downarrow^1 \downarrow^3 S'(y_i)\ \mathcal{A}(x_i)\, A_1^{(i)} p(i < n) \uparrow^2 F(i)\ O \uparrow^1 \downarrow^2 p(|[\alpha] - [M]| \geqq \varepsilon) \uparrow^4 \times$$

$$\times F^{-n}(i)\, F(\lambda n)\, O \uparrow^3 \downarrow^4 Op, \qquad (S_{11})$$

unde $\mathcal{A}(x)$ este algoritmul $\mathcal{A}$ aplicat funcției $f(x)$ în punctul x_i, iar

$$A_1^{(i)} \equiv [a_1^{(r)}] + [\alpha] \Rightarrow (\alpha) \qquad [\alpha] = 0 \text{ inițial}$$
$$[\alpha]\{[\pi] : [n]\} \Rightarrow (\alpha).$$

О ПРОГРАММИРОВАНИИ НА ЭЛЕКТРОННЫХ ВЫЧИСЛИТЕЛЬНЫХ МАШИНАХ ФОРМУЛ ЧИСЛЕННОЙ КВАДРАТУРЫ

КРАТКОЕ СОДЕРЖАНИЕ

В труде исследуются формулы численной квадратуры с точки врения возможности использования электронной вычислительной машины для приблизительной оценки интегралов при помощи этих формул. Труд состоит из трёх частей.

В первой части делается введение, оправдывая возможность пренебрежения остатком этих формул.

Во второй части дается обзор главных формул численной квадратуры без остатка и даётся их явная форма, при которой возможно применение вычислительной машины для вычисления интегралов при их помощи.

В последней части строятся операторные логические схемы программов алгоритмов, описанных этими формулами.

SUR LA PROGRAMMATION POUR LES MACHINES ÉLECTRONIQUES DE CALCUL DES FORMULES DE QUADRATURES NUMÉRIQUES

RÉSUMÉ

On étudie les formules de quadratures numériques du point de vue de la possibilité d'utiliser à leur aide la machine électronique à l'évaluation approximative des intégrales. Le travail se compose de trois parties :

La première partie est une introduction. On y justifie la possibilité de négliger le reste de ces formules.

Dans la deuxième partie on passe en revue les principales formes de quadratures numériques sans reste, et on donne leur forme explicite et acceptable permettant d'utiliser, moyennant ces formules, la machine électronique pour le calcul des intégrales.

Dans la partie finale, on construit les schémas logiques opérationnels des programmes des algorithmes décrits par ces formules.

BIBLIOGRAFIE

1. Березин С., Жидков Н. П., *Методы вычислений, I.* Физ-матгиз, Москва, 1959.
2. Durand E., *Solutions numériques des équations algébriques.* I. Masson et. Cie, Paris, 1960.
3. Ляпунов А. А., *О логических схемах программ.* Проблемы Кибернетики, I, 46 (1958).

Primit la 11. XI. 1962.