# UN ALGORITM PENTRU REZOLVAREA UNEI ECUAȚII TRANSCENDENTE ȘI PROGRAMAREA LUI CU METODA OPERAȚIONALĂ

DE

EMIL MUNTEANU și LADISLAU NÉMETI
(Cluj)

*Lucrare prezentată la Consfătuirea tehnico-științifică asupra mașinilor electronice de calcul din 13—15 ianuarie 1960, București*

În această lucrare se prezintă un algoritm de calcul pentru rezolvarea ecuației transcendente

$$\rho_1 + \rho_2 - a = p, \tag{1}$$

ecuație care intervine în calculul corijării roților dințate. Forma explicită a funcțiilor $\rho_1$, $\rho_2$ și $a$ este următoarea :

$$\rho_1 = \sqrt{\left[1 + \frac{z_1}{2} + Z\lambda - x_2\right]^2 - \frac{z_1^2}{4}\cos^2\alpha_0}\,,$$

$$\rho_2 = \sqrt{\left[1 + \frac{z_2}{2} + Z\lambda - x_1\right]^2 - \frac{z_2^2}{4}\cos^2\alpha_0}\,,$$

$a = Z \cos\alpha_0 \operatorname{tg}\alpha$, unde $\alpha$ este definit de relația

$$x_1 + x_2 = ZX.$$

Mărimea $Z$ și funcțiile $X$ și $\lambda$ sînt definite în felul următor :

$$Z = \frac{z_1 + z_2}{2}, \quad X = \frac{\operatorname{inv}\alpha - \operatorname{inv}\alpha_0}{\operatorname{tg}\alpha_0}, \quad \text{și } \lambda = \frac{\cos\alpha_0}{\cos\alpha} - 1,$$

unde $\operatorname{inv}\alpha = \operatorname{tg}\alpha - \alpha$, iar $p$ și $\alpha_0$ sînt constante.

Variabilele curente sînt $x_1$ și $x_2$. Parametrii $z_1$ și $z_2$ reprezintă numărul de dinți a celor două roți și iau toate combinațiile valorilor : 8, 10, 12, 16, 20, 25, 32, 40, 50, 63, 80, 100, 125, astfel ca $z_1 \leqslant z_2$, iar $\alpha$ este unghiul de angrenare.

Ecuația (1) reprezintă în planul $(x_1, x_2)$ o familie de ovale care depinde de parametrul $Z$. Dacă intersectăm un oval al familiei, cu o familie de drepte paralele reprezentate de ecuația

$$(D_{\alpha_i})\, x_1 + x_2 = ZX, \tag{2}$$

ceea ce revine la eliminarea lui $x_1$ sau $x_2$ din ecuațiile (1) și (2), obținem o ecuație de gradul doi. Eliminarea lui $x_2$ de exemplu, duce la următoarea ecuație

$$Ax_1^2 + Bx_1 + C = 0, \tag{3}$$

unde coeficienții $A$, $B$, $C$, sînt funcții de $z_1$, $z_2$ și $\alpha$, de următoarea formă :

$$A = b^2 - (k_1 + k_2)^2,$$

$$B = (k_1 - k_2)A + (k_1 + k_2)\,(z_1^2 - z_2^2)\frac{\cos^2 \alpha_0}{4},$$

$$C = h_1\, h_2 - \frac{1}{2}\,(b^2 - h_1 - h_2)^2,$$

unde

$$k_1 = 1 + \frac{z_1}{2} + Z\,(\lambda - X),\ k_2 = 1 + \frac{z_2}{2} + Z\lambda,$$

iar

$$h_1 = k_1^2 - z_1^2\,\frac{\cos^2 \alpha_0}{4},\ h_2 = k_2^2 - z_2^2\,\frac{\cos^2 \alpha_0}{4},$$

$$b = a + p.$$

Coordonatele punctelor curbelor familiei (1) sînt date evident de soluțiile ecuației (3).

Algoritmul propus în această lucrare analizează tocmai disjuncția cazurilor de intersecție a unui oval din familia (1) cu familia de drepte paralele (2). Pentru a nu mări volumul calculelor și pentru a avea totuși suficiente puncte necesare construirii curbelor familiei (1), s-a ales pentru variația parametrului $\alpha$ următoarea expresie :

$$\alpha_i = \sqrt[3]{\frac{0{,}155\, i}{\sqrt{Z}}}$$

Fie $d_i = \left(\frac{B}{2}\right)^2 - AC$, discriminantul ecuației (3). O remarcă specială trebuie să facem asupra lui $d_0$, și anume, dacă $d_0 < 0$, la al doilea pas cu siguranță $d_1$ va fi pozitiv. Pentru problema propusă nu se consideră cazul cînd $\alpha < 0$.



Vom considera în general două cazuri care determină alegerea indicelui $i$ pentru pasul următor :

a) $\alpha_{i-1} < \alpha_i$. În acest caz $d_i$ poate fi mai mare decît 0, mai mic sau egal cu 0. În oricare din aceste cazuri, pasul următor este dat de semnul lui $d_{i-1}$ și anume :

Dacă $d_i > 0$ și $d_{i-1} > 0$, pentru pasul următor $i \to i + 1$, dacă însă $d_i > 0$ și $d_{i-1} < 0$, pentru pasul următor $i \to i + N$ și $N \to -\left|\frac{N}{2}\right|$ inițial $N = 1$.

În cazul cînd $d_i < 0$ atunci și $d_{i-1} > 0$, pentru pasul următor $i \to i + n$, $n \to -\left|\frac{n}{2}\right|$, inițial $n = 1$, iar cazul $d_i < 0$ și $d_{i-1} < 0$, nu poate avea loc.

b) $\alpha_{i-1} > \alpha_i$. Și în acest caz deosebim aceleași subcazuri de studiere a semnului lui $d_{i-1}$.

Dacă $d_i > 0$, $d_{i-1} > 0$, atunci pentru pasul următor $i \to i + N$ și $N \to -\left|\frac{N}{2}\right|$, iar în cazul cînd $d_i > 0$ și $d_{i-1} < 0$, pentru pasul următor $i \to i + n$ și $n \to \left|\frac{n}{2}\right|$.

Dacă $d_i < 0$ și $d_{i-1} > 0$, atunci pentru pasul următor $i \to i + N$ și $N \to \left|\frac{N}{2}\right|$, iar dacă $d_i < 0$ și $d_{i-1} < 0$, pentru pasul următor $i \to i + n$ și $n \to -\left|\frac{n}{2}\right|$. În toate cazurile cînd $d_i > 0$, înainte de a trece la studiul semnului se calculează și se tipăresc valorile variabilelor $x_1^i$ și $x_2^i$, pentru valoarea corespunzătoare a indicelui $i$.

În cazurile cînd $d_i = 0$, adică atunci cînd dreapta $(D_{\alpha_i})$ este tangentă ovalului considerat, se oprește calculul dacă aceasta este tangentă la dreapta, sau se continuă cu alte intersecții, dacă dreapta $(D_{\alpha_i})$ este tangentă la stînga.

Înainte de oprirea mașinii se calculează și se tipăresc coordonatele punctului de tangență.

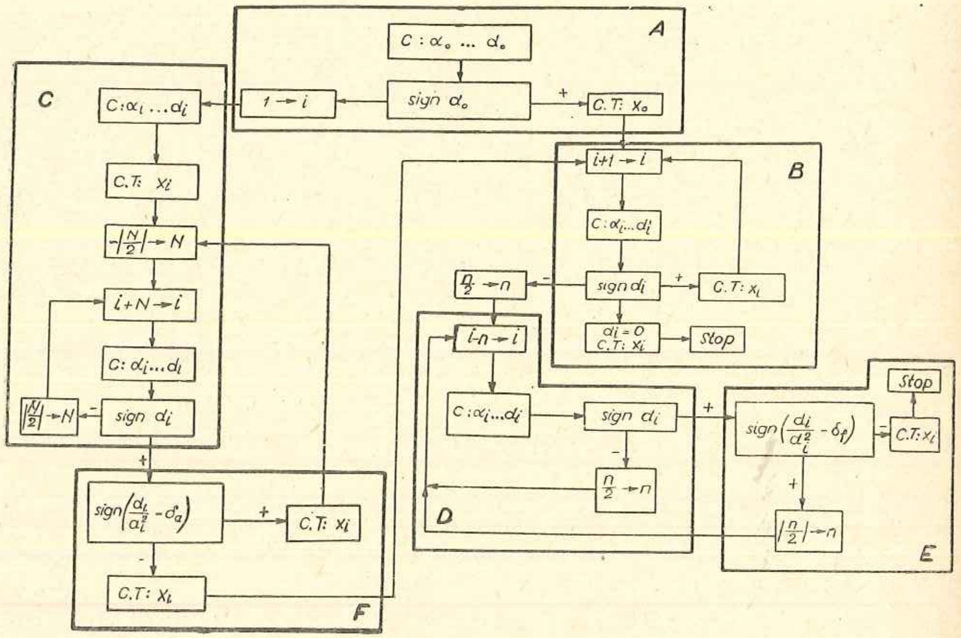
În general vom considera că o secantă $(D_{\alpha_i})$ este aproximativ tangentă la oval, dacă distanța dintre cele două puncte de intersecție ale ei cu ovalul este mai mică decît un număr fix dat $\delta_f$, pentru tangenta din dreapta, sau $\delta_a$, pentru tangenta din stînga.

Schema generală a algoritmului este dată în figură, unde prin $C: \alpha_i \ldots d_i$, înțelegem calculul lui $\alpha_i$ și a coeficienților A, B, C a ecuației (3) și a mărimii $d_i$, iar prin $CT: x_i$, înțelegem calculul și tipărirea coordonatelor $x_1^i$ și $x_2^i$.

Folosind metoda operațională de programare elaborată de A. A. L i a p u n o v [3] și notația introdusă de I. I. I a n o v [1], vom da o schemă generală a programului de calcul pentru mașinile electronice. Pentru aceasta vom introduce următorii operatori aritmetici : $A_1^i$ operatorul aritmetic care realizează calculul lui $\alpha_i$ şi a coeficienţilor $A$, $B$, $C$, a ecuaţiei (3) precum şi mărimea $d_i$, pentru o valoare oarecare a indicelui $i$; $A_5^i$, operatorul aritmetic care realizează calculul şi tipărirea coordonatelor $x_1^i$ şi $x_2^i$. Acelaşi operator îl notăm cu $A_3^i$ în partea a doua a schemei programului.

Condiţiile logice, care determină ramificările în program în funcţie de semnul lui $d_i$ sau în funcţie de faptul că la un moment dat poziţia

Calculul curbelor $\varepsilon' = const$




unei secante ($D_{\alpha_i}$) aproximează destul de bine — în sensul condiţiilor cerute în program — poziţia tangentei la oval, le vom nota astfel:

$$\nu_k = p_k\ (d_i < 0),\quad \nu_k = p_k\left(\frac{d_i}{A} < \delta\right).$$

Un astfel de operator logic este operatorul $\nu_2 = p_2(d_0 < 0)$ operator care realizează împărţirea programului în funcţie de semnul lui $d_0$. Acest operator predă comanda conducerii calculului operatorului următor, dacă răspunsul la condiţia logică $p_2$ este afirmativ, sau predă comanda conducerii calculului porţiunii din program $P_2$, dacă răspunsul este negativ. În schema programului propusă în această lucrare, intervin şapte operatori logici de natura celui descris mai sus.

Un alt tip de operator este $S_k$, care realizează o schimbare a indicelui. Un astfel de operator este $S_3$, care determină indicele pasului următor în



cazul cînd răspunsul la condiţia logică $p_2\,(d_0 > 0)$ este afirmativ, sau de exemplu operatorul $S_6$ care determină indicele pasului următor în cazul cînd $\alpha_{i-1} < \alpha_i$ şi $d_i > 0$, $d_{i-1} < 0$, în acest caz pentru pasul următor $i \to i + N$ şi $-\left|\frac{N}{2}\right| \to N$. Operatorul $S_6$ realizează transformarea $-\left|\frac{N}{2}\right| \to N$, iar operatorul $S_7$ realizează transformarea $i \to i + N$.

Alţi operatori care intervin în programarea algoritmului propus, sînt operatorii de readresare. În cazul algoritmului nostru intervin operatori simpli de readresare, adică operatori care schimbă ultima comandă a operatorilor aritmetici $A_1^i$ şi $A_5^i$, de atîtea ori de cîte ori este necesar ca aceşti operatori să predea conducerea calculului unor alţi operatori diferiţi de cei cărora li s-a predat conducerea calculului într-un ciclu anterior.

Astfel de operatori sînt operatorii $F_k$ şi $G_k$. De exemplu operatorul $F_4$ schimbă ultima comandă a operatorului $A_1^i$ astfel că după executarea operatorului $S_3$ se execută din nou operatorul $A_1^i$, însă de data aceasta conducerea calculului nu mai este predată operatorului $\nu_2$, aşa cum s-a întîmplat în etapa precedentă a calculului, ci operatorului $A_5^i$.

În program mai intervine operatorul $O_{18}$ — operator care realizează oprirea maşinii la sfîrşitul calculului.

Schema scrisă cu ajutorul operatorilor analizaţi mai sus este următoarea:

$$(P_1)\ \overset{8}{\underset{8}{\sqsupset}}\ \overset{4}{\underset{4}{\sqsupset}}\ A_1^i\, \nu_2\, \underset{P_2}{\llcorner}\ S_3 F_4\, \overset{1}{\underset{1}{\sqsubset}}\ \overset{13,12,1}{\underset{13,12,1}{\sqsupset}}\ A_5^i\, S_6\, \overset{10}{\underset{10}{\sqsupset}}\ S_7 F_8\, \overset{1}{\underset{1}{\sqsubset}}\ \overset{1}{\underset{1}{\sqsupset}}\ \nu_9\, \underset{11}{\llcorner}\ S_{10}\, \overset{7}{\underset{7}{\sqsubset}}\ \nu_{11}\, \underset{13}{\llcorner}\ G_{12}\, \overset{5}{\underset{5}{\sqsubset}}\ G_{13}\, \overset{5}{\underset{5}{\sqsubset}}$$

$$(P_2)\ \overset{9}{\underset{9}{\sqsupset}}\ \overset{5}{\underset{5}{\sqsupset}}\ A_1\, \nu_2\, \overset{P_1}{\llcorner}\ \overset{7,16,13}{\underset{17,16,13}{\sqsupset}}\ A_3^i\, S_4 F_5\, \overset{1}{\underset{1}{\sqsubset}}\ \overset{1}{\underset{1}{\sqsupset}}\ \nu_6\, \overset{15}{\ulcorner}\ S_7\, \overset{14,11}{\underset{14,11}{\sqsupset}}\ S_8 F_9\, \overset{1}{\underset{1}{\sqsubset}}\ \nu_{10}\, \underset{12}{\llcorner}\ S_{11}\, \overset{8}{\underset{8}{\sqsubset}}\ \nu_{12}\, \underset{14}{\llcorner}\ G_{13}\, \overset{3}{\underset{3}{\sqsubset}}$$

$$S_{14}\, \overset{8}{\underset{8}{\sqsubset}}\ \nu_{15}\, \underset{17}{\llcorner}\ G_{16}\, \overset{3}{\underset{3}{\sqsubset}}\ G_{17}\, \overset{3}{\underset{3}{\sqsubset}}\ \overset{3}{\underset{3}{\sqsupset}}\ O_{18}.$$

Această schemă de programare are avantajul că scrierea instrucţiunilor cu ajutorul ei se face uşor şi pentru orice maşină.

În cele ce urmează vom da un exemplu de felul cum se scriu instrucţiunile cu ajutorul acestei scheme de programare. În exemplul considerat vom folosi codul maşinii CIFA-1. Din schemă se vede că operatorul $A_1^i$ predă conducerea calculului, la început operatorului $\nu_2$ şi apoi, în urma acţiunii operatorului $F_4$, conducerea este predată operatorului $A_5^i$, iar în urma acţiunii operatorului $F_8$, conducerea calculului se predă operatorului logic $\nu_9$.

Fie operatorul $A_1^i$ realizat prin șirul de instrucțiuni $m_1 + 1$, $m_1 + 2$, ..., $m_2$ și fie numărul $a = 10 - m_2 + 7$ $13 - 000$. Programul pentru operatorii $A_1^i$, $\nu_2$, $S_3$, $F_4$, $A_5^i$ este următorul:

| | | | |
|---|---|---|---|
| $m_1 + 1$ | | | |
| $m_1 + 2$ | $A_1^i$ | | |
| $\vdots$ | | | |
| $m_2$ | | | |
| $m_2 + 1$ | $01 - < d_i >$ | $07 - m_2 + 3$ | $\nu_2$ |
| $m_2 + 2$ | $10 - P_2$ | $13 - 000$ | |
| $m_2 + 3$ | $01 - < 1 >$ | $02 - < i >$ | $S_3$ |
| $m_2 + 4$ | $11 - < i >$ | $13 - 000$ | |
| $m_2 + 5$ | $01 - < a >$ | $11 - m_2$ | $F_4$ |
| $m_2 + 6$ | $10 - m_1 + 1$ | $13 - 000$ | |
| $m_2 + 7$ | | | |
| $\vdots$ | $A_5^i$ | | |
| $m_3$ | | | |

Instrucțiunile $m_2 + 1$ și $m_2 + 2$ realizează operatorul logic $\nu_2$ adică predă conducerea calculului instrucțiunii $m_2 + 3$, dacă răspunsul la condiția logică $p_2$ este afirmativ, sau instrucțiunii $P_2$ dacă răspunsul este negativ. Începînd cu instrucțiunea $m_2 + 3$ începe realizarea operatorului de schimbare a indicelui, $S_3$, care determină indicele pasului următor, dacă răspunsul la condiția logică $p_2$ este afirmativ. Prin realizarea operatorului $S_3$ se mărește $i$ cu o unitate și se predă conducerea calculului operatorului $F_4$. Operatorul $F_4$ realizat de instrucțiunile $m_2 + 5$ și $m_2 + 6$, schimbă ultima instrucțiune a operatorului $A_1^i$, adică pe $m_2$, și pe urmă predă conducerea calculului operatorului $A_1^i$. După executarea calculului, datorită acțiunii operatorului $F_4$, operatorul $A_1^i$ nu mai predă conducerea calculului operatorului $\nu_2$, ci operatorului $A_4^i$ care calculează și tipărește valorile coordonatelor $x_1^i$ și $x_2^i$, predînd apoi conducerea calculului operatorului $S_6$. Operatorul $S_6$ nu mai figurează în exemplul dat.

În schema generală de programare (4), nu s-a introdus automatizarea calculului sumei $z_1 + z_2$ pentru $z_1 \leqslant z_2$. Schema este întocmită pentru un cuplu $(z_1, z_2)$ fixat.

Această schemă a programului permite urmărirea controlului programului mult mai ușor decît în cazul cînd instrucțiunile programului ar fi scrise direct; mai mult, întrucît instrucțiunile se întocmesc pentru fiecare operator, această schemă asigură realizarea unui număr mai mic de instrucțiuni decît în cazul cînd instrucțiunile ar fi scrise direct.



## АЛГОРИФМ ДЛЯ РЕШЕНИЯ ОДНОГО ТРАНСЦЕНДЕНТНОГО УРАВНЕНИЯ И ЕГО ПРОГРАММИРОВАНИЯ ОПЕРАТОРНЫМ МЕТОДОМ

### РЕЗЮМЕ

Работа содержит изложение алгорифма для решения одного трансцендентного уравнения, встречающегося при вычислении поправок зубчатых колес. Уравнение выражает семейство овалов со свойством, что их пересечение с семейством параллельных прямых приводит к решению алгебраических уравнений второй степени.

Этот алгорифм исследует разделение случаев пересечения семейств параллельных прямых с данным семейством овалов. Он обладает преимуществами перед классическими приближенными методами, так как не использует производную.

Для приведенного алгорифма излагается программа вычислений на электронных машинах. В работе приводятся общая схема этой программы, выписанная при помощи операторного метода [4], и пример указаний для машины CIFA-1.

## UN ALGORITHME POUR RÉSOUDRE UNE ÉQUATION TRANSCENDANTE ET SON PROGRAMME EXPRIMÉ PAR LA MÉTHODE OPÉRATIONNELLE

### RÉSUMÉ

Les auteurs présentent un algorithme permettant de résoudre une équation transcendante intervenant dans le calcul de la correction des roues dentées. L'équation représente une famille d'ovales jouissant de la propriété que leurs intersections avec une famille de droites parallèles conduisent à des solutions d'équations algébriques du second degré.

L'algorithme, qui analyse la disjonction des cas d'intersection des familles de droites parallèles avec la famille d'ovales, présente, par rapport aux méthodes classiques d'approximation, l'avantage de pouvoir se passer de l'emploi de la dérivée.

L'auteur a élaboré, pour cet algorithme, un programme de calcul avec les machines électroniques. Il fournit le schéma général de ce programme, exprimé par la méthode opérationnelle (4) et donne un exemple de transcription des instructions pour la machine CIFA-1.

## BIBLIOGRAFIE

1. ЯНОВ Ю. И., *О логических схемах алгорифмов.* Проблемы кибернетики, I, 75—128 (1958).
2. KITOV A. I., *Calculatoare electronice cifrice.* Ed. tehnică, București, 1959.
3. ЛЯПУНОВ А. А., *О логических схемах программ.* Проблемы кибернетики, I, 46—75 (1958).

Primit la 1.IX.1959.