CALCULUL VALORILOR ŞI VECTORILOR PROPRII ALE UNOR MATRICI CU AJUTORUL CALCULATOARELOR ELECTRONICE DACICC-1 ŞI DACICC-200

DE

M. FERNEA, L. GHERMAN, M. MIHOC, L. NEGRESCU

(Cluj)

În această lucrare se prezintă o adaptare pentru calculatoarele DACICC-1 şi DACICC-200 a algoritmului lui Jacobi pentru calculul valorilor şi vectorilor proprii ale unor matrici simetrice. Pe baza algoritmului adaptat s-au întocmit programe pentru cele două calculatoare, care au fost utilizate în rezolvarea unor probleme din domeniul chimiei.

1. În multe aplicaţii cu caracter tehnico-ştiinţific intervine studiul valorilor şi vectorilor proprii ale unor matrici. Astfel de probleme referitoare la matrici de diferite ordine apar şi în studiul unor substanţe organice, cum sînt de exemplu hidrocarburile.

În literatura de specialitate se cunosc mai multe metode pentru determinarea valorilor şi vectorilor proprii cum ar fi : metoda lui Jacobi [1], a lui Krîlov [2], sau metoda dată de Givens [1]. Metodele folosite cu succes la un calculator electronic sînt metodele iterative care permit obţinerea rezultatelor cu o precizie dinainte stabilită, chiar şi pentru matrici de rang mare.

În lucrarea de faţă se prezintă o adaptare pentru calculatoarele DACICC-1 şi DACICC-200 a algoritmului lui Jacobi pentru calculul valorilor şi vectorilor proprii ale unor matrici simetrice.

Pe baza algoritmului adaptat s-au întocmit programe pentru cele două calculatoare care au fost utilizate în rezolvarea unor probleme din domeniul chimiei.

2. Metoda lui Jacobi se bazează pe următoarea teoremă : *Dacă A este o matrice simetrică de rangul n, atunci există o matrice ortogonală Q astfel ca*

$$Q^T A Q = D,$$

ST. CERC. MAT., TOM 23, NR. 7, P. 1083–1090 BUCUREŞTI, 1971

unde D este o matrice ale cărei elemente de pe diagonala principală sînt valorile proprii ale matricei A, elementele extradiagonale fiind nule, iar Q^T reprezintă transpusa matricei Q.

Algoritmul constă în determinarea unui şir $\{S_k\}$ de matrici ortogonale, cu proprietatea :

$$\lim_{k\to\infty} S_1 S_2 \ldots S_k = Q. \tag{1}$$

Vom nota

$$T_k = S_k^T S_{k-1}^T \ldots S_1^T A S_1 S_2 \ldots S_k, \qquad (k = 1, 2, \ldots), \tag{2}$$

unde $T_0 = A$.

Fie

$$v_k = \left(\sum_{\substack{i=1 \\ i\neq j}}^{n} \sum_{j=1}^{n} [t_{ij}^{(k)}]^2 \right)^{1/2} \tag{3}$$

şi

$$w_k = \sum_{i=1}^{n} \sum_{j=1}^{n} [t_{ij}^{(k)}]^2 \tag{4}$$

Se aleg matricile S_k astfel ca să fie îndeplinite condiţiile :

$$w_{k+1} = w_k \text{ şi } w_{k+1} < v_k \text{ pentru toţi } k \text{ şi } \lim_{k\to\infty} v_k = 0. \tag{5}$$

De aici rezultă :

$$\lim_{k\to\infty} T_k = D. \tag{6}$$

Să considerăm elementele matricei S_k astfel :

$$S_{pp}^{(k)} = S_{qq}^{(k)} = \cos\theta_k, \quad S_{ii}^{(k)} = 1 \text{ dacă } i \neq p$$

şi $i \neq q$ $(p, q = 1, \ldots, n)$

$$S_{pq}^{(k)} = -S_{qp}^{(k)} = \sin\theta, \quad S_{ij}^{(k)} = 0 \text{ în celelalte cazuri.} \tag{7}$$

Din (2) rezultă că $T_k = S_k^T T_{k-1} S_k$, elementele matricii T_k calculîndu-se cu ajutorul formulelor :

$$\begin{aligned} t_{pj}^{(k)} &= t_{pj}^{(k-1)} \cos\theta_k - t_{qj}^{(k-1)} \sin\theta_k, \\ t_{qj}^{(k)} &= t_{pj}^{(k-1)} \sin\theta_k + t_{qj}^{(k-1)} \cos\theta_k, \qquad (j \neq p \text{ sau } j \neq q) \end{aligned} \tag{8}$$

$$\begin{aligned} t_{ip}^{(k)} &= t_{ip}^{(k-1)} \cos\theta_k - t_{iq}^{(k-1)} \sin\theta_k, \\ t_{iq}^{(k)} &= t_{ip}^{(k-1)} \sin\theta_k + t_{ip}^{(k-1)} \cos\theta_k, \qquad (i \neq p \text{ sau } i \neq q) \end{aligned} \tag{9}$$

$$\begin{aligned} t_{pp}^{(k)} &= t_{pp}^{(k-1)} \cos^2\theta_k + t_{qq}^{(k-1)} \sin^2\theta_k - 2t_{pq}^{(k-1)} \sin\theta_k \cos\theta_k, \\ t_{qq}^{(k)} &= t_{pp}^{(k-1)} \sin^2\theta_k + t_{qq}^{(k-1)} \cos^2\theta_k + 2t_{pq}^{(k-1)} \sin\theta_k \cos\theta_k, \\ t_{pq}^{(k)} &= t_{qp}^{(k)} = \frac{1}{2} [t_{pp}^{(k-1)} - t_{qq}^{(k-1)}] \sin 2\theta_k + t_{pq}^{(k-1)} \cos 2\theta_k, \end{aligned} \tag{10}$$

$$t_{ij}^{(k)} = t_{ij}^{(k-1)} \qquad (i \neq p \text{ şi } j \neq q), \tag{11}$$

unde $i, j = 1, \ldots, n$.

Transformările (8—(11) se aplică asupra acelor elemente $t_{p,q}^{(k)}$ pentru care elementul $t_{p,q}^{(k-1)}$ verifică relaţia :

$$|t_{p,q}^{(k-1)}| > \nu_k, \tag{12}$$

unde $\nu_k = \frac{\nu_{k-1}}{\sigma}$, σ fiind o constantă dată (pentru accelerarea procesului de convergenţă se ia σ egal cu rangul matricii A).

În acest caz alegem parametrul θ_k astfel ca $t_{p,q}^{(k)} = 0$.

Dacă relaţia (12) nu este îndeplinită, atunci :

$$t_{p,q}^{(k)} = t_{p,q}^{(k-1)}.$$

Din (10) rezultă :

$$\operatorname{tg} 2\theta_k = \frac{-t_{p,q}^{(k-1)}}{\frac{1}{2}[t_{pp}^{(k-1)} - t_{qq}^{(k-1)}]}.$$

În cazul cînd

$$t_{pp}^{(k-1)} = t_{qq}^{(k-1)},$$

vom lua

$$\operatorname{tg} 2\theta_k = -1.$$

Pentru simplificarea calculului la calculatoarele electronice se utilizează formulele următoare :

$$\cos\theta_k = \left(\frac{\omega + |\mu|}{2\omega} \right)^{1/2} \qquad \sin\theta_k = \frac{\operatorname{sgn}(\mu)\,\lambda}{2\omega\cos\theta_k},$$

unde

$$\lambda = -t_{pq}^{(k-1)} \quad \text{şi} \quad \mu = 1/2[t_{pp}^{(k-1)} - t_{qq}^{(k-1)}].$$

Aceste transformări se fac asupra tuturor elementelor $t_{p,q}^{(k-1)}$ care satisfac relaţia (12).

Convergenţa procesului iterativ este asigurată de faptul că transformările (8)—(11) verifică condiţiile (5).

Fie $\rho > 0$ un număr dat, care se alege ţinîndu-se cont de precizia pe care dorim s-o avem în obţinerea rezultatelor, şi fie :

$$\nu_F = \left(\frac{\rho}{\sigma}\right) \nu_1,$$

unde $\nu_1 = v_1$, iar σ reprezintă rangul matricei.

Algoritmul se opreşte la iteraţia k dacă :

$$\nu_k < \nu_F. \tag{13}$$

Pentru calculul vectorilor proprii se utilizează formula :

$$T_k = R_k^T A R_k, \tag{14}$$

unde

$$R_k = S_1 S_2 \ldots S_k,$$

iar elementele matricii R_k se calculează după formulele :

$$\begin{aligned} r_{ip}^{(k)} &= r_{ip}^{(k-1)} \cos\theta_{k-1} - r_{iq}^{(k-1)} \sin\theta_{k-1}, \\ r_{iq}^{(k)} &= r_{ip}^{(k-1)} \sin\theta_{k-1} + r_{iq}^{(k-1)} \cos\theta_k, \\ r_{ij}^{(k)} &= r_{ij}^{(k-1)} \text{ pentru } j \neq p \text{ sau } j \neq q, \\ i,\ j &= 1, \ldots, n. \end{aligned} \tag{15}$$

Schema logică a algoritmului este dată în figurile 1 şi 2.

3. Utilizînd algoritmul de mai sus s-au efectuat programe pentru calculatorul DACICC-1 şi DACICC-200.

La introducerea datelor în calculatorul DACICC-1 se foloseşte un procedeu special, care permite numai memorarea elementelor a_{ij} pentru care $i \leqq j$.

Elementele matricii se introduc sub forma :

$$a_{11}, a_{12}, \ldots, a_{1n}, a_{22}, a_{23}, \ldots, a_{2n}, \ldots, a_{n-1,n-1}, a_{n-1,n}, a_{n,n}. \tag{16}$$

Un element oarecare a_{ij}, al matricei A, utilizat în calcul la un moment dat se determină din şirul (16) după algoritmul dat de următoarea schemă logică (fig. 3), unde $[a_{11}]$ este adresa primului element din şirul (16) iar $[a_{ij}]$ este adresa elementului a_{ij}.

După cum se vede, şirul (16) permite efectuarea calculelor memorînd doar $\frac{\sigma(\sigma+1)}{2}$ elementele în loc de σ^2 elemente.

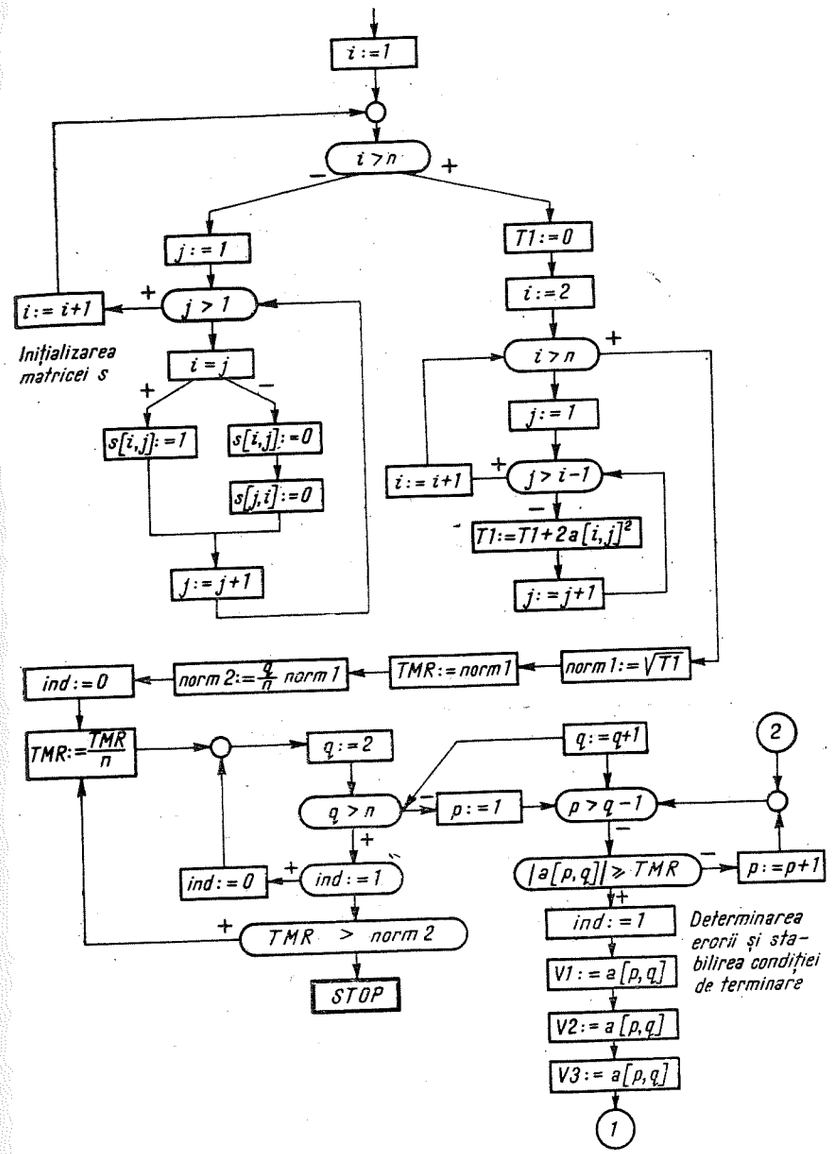


Fig. 1

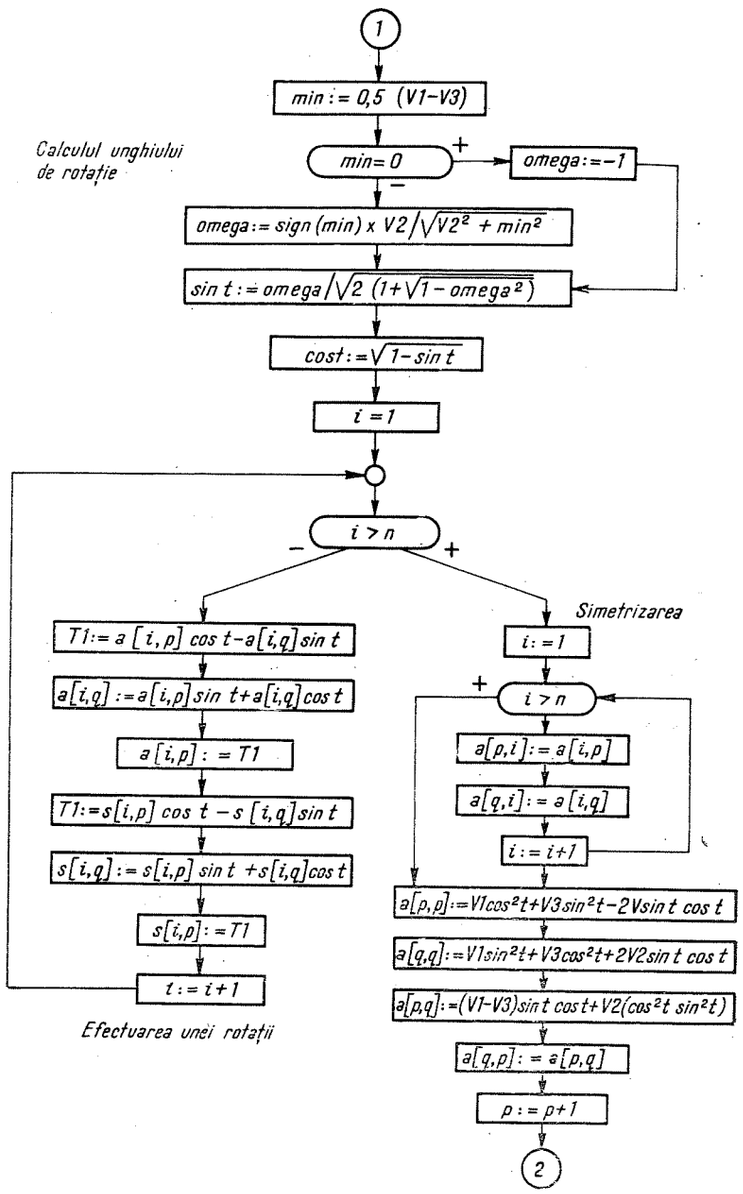


Fig. 2

Acest fapt permite sporirea capacităţii memoriei, dînd posibilitatea să se calculeze valorile proprii ale unor matrici a căror ordin este cel mult 32.

Programul la la calculatorul DACICC-1 este întocmit în cod-maşină cuprinzînd 500 instrucţiuni. Acest program utilizează subrutinele stan-

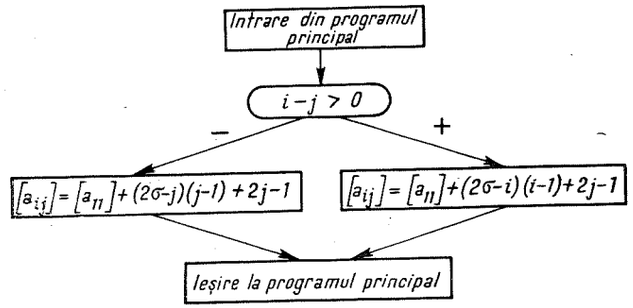


Fig. 3

dard care permit efectuarea calculator în virgulă mobilă. Pentru matrici de rang mai mare decît 14 se utilizează şi memoria externă pe tambur magnetic.

La calculatorul DACICC-200 s-au întocmit două programe: programul VLPR şi subrutina JCB, care permit calculul valorilor şi vectorilor proprii pentru matrici de rangul maxim $n = 35$.

Programul VLPR are o lungime de 1000 de cuvinte şi este scris în codul calculatorului. Subrutina JCB a fost întocmită în limbajul DACICC-FORTRAN. Ele nu utilizează supervizorul, folosesc doar programele de convertire.

4. La calculatorul DACICC-1 s-a efectuat calculul valorilor proprii ale unor matrici ce intervin în studiul spectrului în infraroşu ale unor substanţe, a structurii decaceanulindicobaltului (II), a anilinei etc. Rangul acestor matrici fiind $4 \leqq n \leqq 8$ calculul valorilor proprii a durat între 5′ şi 20′, durata introducerii datelor şi scoaterea rezultatelor fiind neglijabilă în raport cu timpul necesar efectuării calculului. Calculul continuă pînă la obţinerea a două matrici T_k şi T_{k+1} ale căror elemente de pe diagonala principală coincid.

La calculatorul DACICC-200 programele VLPR şi JCB au fost aplicate la diverse matrici de diferite ordine. Ca exemplificare amintim aplicarea lor la matrici de ordinul 32 care intervin în studiul tetramenului 1, 3, 5, 7 şi a tetramenului simetric 1, 3, 6,8 utilizaţi în studiul hidrocarburilor.

Programul VLPR, aplicat la o astfel de matrice permite obţinerea rezultatelor respective cu o eroare mai mică decît 10^{-3} în cca. 5 minute de calcul. În acest timp se mai adaugă timpii de introducere şi extragere a datelor, primul fiind neglijabil (20 secunde), iar al doilea este cca 4 minute.

Subrutina JCB aplicată la aceeaşi martice a obţinut rezultatele cu o eroare mai mică decît 10^{-7} în aproximativ o oră de calcul, timpii de introducere şi extragere a datelor fiind cei menţionaţi anterior.

Din aceste exemple se observă că programele de mai sus dau un randament mare în calculul valorilor proprii la calculatorul DACICC-200.

Primit la redacţie la 5.02.1970

Institutul de calcul,
Cluj

CALCUL DES VALEURS ET DES VECTEURS PROPRES DE CERTAINES MATRICES À L'AIDE DES ORDINATEURS DACICC-1 ET DACICC-200.

(RÉSUMÉ)

Dans ce travail on aborde un problème de calcul effectif des valeurs et des vecteurs propres des matrices à l'aide de certains ordinateurs à performances moyenne et petite. Dans ce but, on a adapté pour les ordinateurs DACICC-1 et DACICC-200, l'algorithme de Jacobi au cas des matrices symétriques. A l'aide de l'algorithme adapté on a élaboré des programmes pour les deux ordinateurs, qui ont été utilisés à la résolution de certains problèmes du domaine de la chimie. Les programmes ont été conçus de manière à permettre l'utilisation la plus rationnelle de la mémoire du calculateur.

BIBLIOGRAFIE

1. RALSTON, A., A *first course in numerical analysis*. McGraw-Hill, New York, 1965.
2. ДЕМИДОВИЧ, Б. И. и МАРОН, И. А., *Основы вычислительной математики*, Физматгиз, Москва, 1960.
3. RALSTON, A., *Mathematical methods for digital computers*, John Wiley & Sons, New York, 1962.
4. АГЕЕВ, М. И., *Алгоритмы*, II, В.Ц. АНСССР, Москва, 1966.