C Algoritmi implementabili

~~Note~~: (1) Introducere

Întreaga lucrare este în fond, plină de algoritmi. Co

Considerăm că, la locul ~~respectat~~ respectiv s-au explicat pe rând

1) procedeele generale de scriere a ecuațiilor în staționare în forme standard (cap 3 pag 87, cap 4 pag 223)
2) procedeele generale de scriere a ec. de regim tranzitoriu a circuitelor cu tr. (cap 5 pag 368, 397)
3) Rezolvarea ecuațiilor ~~rez~~ statice prin diverse metode (cap 3 ~~pag 138~~) (algoritmi)
4) Rez. ecuației diferențiale a rețelei prin alg de integrare numerică (cap 5 pag 394, 400)
5) Metode de găsire a marginilor sol. ecuației staționare (cap 3 pag ~~182~~ 62)

etc.

În cele ce urmează însă, mă voi ocupa cu alte tipuri de algoritmi, proprii metodelelor A.C.F, adică de aceia care verifică îndeplinirea condițiilor unor teoreme. Această verificare poate să fie lucrată manual, în cazuri simple, dar deseori am văzut că se cerea verificări mai complicate, care pot fi foarte ușor făcute pe calculator, prin simple programe ~~destul de simple~~ ușoare, în marea lor majoritate, operații ~~simple~~ elementare cu matrici și determinanți. (nu se cere în general rezolvarea ~~unor~~ nici unei ecuații) înainte de a trece la listarea acestui tip de algoritmi, (să le spunem algoritmi A.C.F) vom insista asupra utilității lor multiple:

– pe de o parte se pot obține în timp relativ scurt să determinăm rezultate calitative privind ecuațiile curenților, fără a depinde de împrăștierea caracteristicilor tranz. și diodelor, care rezultatele calitative, rezultă din partea acestor elemente numai să respecte o anumită „turnură", numită „ipoteză de calitate". Astfel, de exemplu, dacă valoarea G e real dominantă pe coloane (lucru banal de verificat, după obținerea ecuațiilor, eventual tot de către calculator), atunci $T^{-1}G \in P_0$ și avem certitudinea unicității soluției ecuației.

Dacă dispunem de valoarea exactă a parametrilor, este din nou păcat să încărcăm calculatorul cu probleme insolvabile sau, pe care programul lui intern nu le poate rezolva. Și în acest sens avem numeroase teoreme, și ar fi de dorit ca utilizatorul să știe pe ce tip de algoritm lucrează o anumită subrutină, pentru a verifica (manual sau pe calculator) că problema sa îndeplinește condițiile cerute de teoremele de convergență a algoritmului.

Deocamdată, algoritmul poate fi „ajutat" să converge mai rapid, prin diverse metode, dintre care să dau exemplu alegerea unui punct „bun" de plecare (v pag 187).

De aceea, voi da mai jos principiul pe care trebuie construite programele de calcul a apartenenței unor matrici la diverse clase, utile aplicării imediate a uneia sau alteia din teoremele A.C.F.

② Verificarea apartenenței la P_0 (P)

Clasa P_0 este cea mai importantă, în lumina teoremelor din lucrare. Cum nu este în general evident din inspectare, dacă o matrice aparține sau nu acestei clase, să urmărim lucrarea uneia din condițiile echivalente de apart. la P_0 (pag 250).

Verificarea condițiilor 2, 3, 4 este desigur mai dificilă (2, 3 reclamă de fapt o infinitate de încercări). Definiția (rel 6 5 pag 247) pare și ea dificil de probat. Totuși să notăm că definiția ~~se~~ sugerează următoarea metodă (manuală):

- se adaugă la elementele ~~diagonale~~ diag. principale a lui A o matrice diagonală $d_1, \dots d_n$. Se dezvoltă

$$\det(A+D) = P(d_1, \dots d_n)$$

(polinomială în d_i)

și nu mai rămâne decât să verificăm că toți coeficienții polinomiali sunt ≥ 0, și că măcar unul e nenul.

Un program bazat pe proprietatea (1) poate însă fi

conceput, destul de simplu. :

~~Pe pagina~~ Am alcătuit un astfel de program, care urmează

să apară cît pe cît posibil mai elegant ~~posibil~~ grandoarea

to pe care o reprezintă faptul că minorii care trebuiesc

calculați, ~~[illegible]~~ (în număr de $2^n - 1$) sînt de aceeași ordine.

Ideea de bază a organigramei ~~din pagina alăturată~~ din figură este

următoarea :

1) — Zona ~~inferioară~~ a organigramei creează o recurență

de tipul :

$$\left.\begin{matrix} 0 & 0 & \dots & 1 \\ 0 & 0 & \dots & 10 \\ 0 & 0 & \dots & 11 \\ - & - & - & - \\ 1 & 1 & \dots & 1 \end{matrix}\right\} 2^n - 1 \text{ vectori linie } V$$

în acest scop ea exploatează ușor scrierea numărului

N în baza doi : cu $N = 1, \dots, 2^n - 1$, de fiecare dată cele n

"cifre" din baza 2 ce-l reprezintă pe N, vor fi chiar

componentele vectorului V.

În acest mod, vectorul V va putea duce, rînd

pe rînd către o mulțime distinctă de indici care să intre

în calculul minorului corespunzător. (~~acesta în care~~ format din

matricea ~~inițială, eliminînd liniile care au indicele~~ 0

~~în V sau ...~~

2) - Zona ~~superioară~~ inferioară, pornind de la indicii dictați

de V, va acționa în modul următor :

— Dacă V(j) = ~~1~~ 1 rămîne linia j neschimbată

— Dacă V(j) = 0, face linia j toată nulă cu excepția elementului de pe diagonală, pe care-l lasă egal cu 1. Pentru a vedea clar modul cum acționează acest procedeu să privim tabloul:

$$
\begin{array}{c|ccccc|}
V(j) & & & & & \\
1 & a_{11} & \cdots & a_{1j} & \cdots & a_{1n} \\
1 & - & - & - & - & - \\
1 & & & & & \\
0 & a_{j1} & \cdots & a_{jj} & \cdots & a_{jn} \\
1 & & & & & \\
1 & a_{n1} & \cdots & a_{nk} & \cdots & a_{nn}
\end{array}
$$

Vectorul $V(j) = (1\ 1 \ldots 1\ \overset{j}{0}\ 1 \ldots 1 \ldots 1)$ va acționa astfel: – lasă liniile k, cu $k \neq j$ neschimbate

– înlocuiește linia j ca în figură:

$$
\begin{pmatrix}
a_{11} & \cdots & \cdots & a_{1j} & & & \\
 & & & & & & \\
a_{j-1\,1} & & & & & & \\
0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\
a_{j+1\,1} & & & & & & \\
\vdots & & & & & & \\
a_{n1} & & & & & &
\end{pmatrix}
$$

Acum, dacă dezvoltăm determinantul tabloului după linia j, se obține exact $1 \cdot \Delta_{jj}$ (minorul diagonal principal respectiv).

Orice alt „0" din $V(j)$ $(j = 1 \ldots n)$ va acționa similar. Este evident că în acest mod, parcurgem ordonat, fără repetiție mulțimea tuturor minorilor principali $(2^n - 1)$ și <u>numai pe ei</u>.

Prin acest subalgoritm, calculatorul va calcula de fiecare dată, un determinant de același ordin: n și problema ~~...~~ aranjării organigramei se simplifică resortului. Desigur că pot fi și alte variante, care de exemplu să verifice determinanții în ordinea crescândă a ordinelor, cu economie de timp.. Totuși, aprecierea programului pt astfel de algoritmi ar putea avea probleme diferite de altă factură (cazul costului programului, memoria ocupată etc.).

Să remarcăm că în programul propus calculatorul nu are în nici un moment blocate mai mult de $n \times n = n^2$ celule pt A + încă n pentru $V(j)$ + 1 pentru N + 1 pt i + 1 pt K + 1 pt D + n^2 pt B.

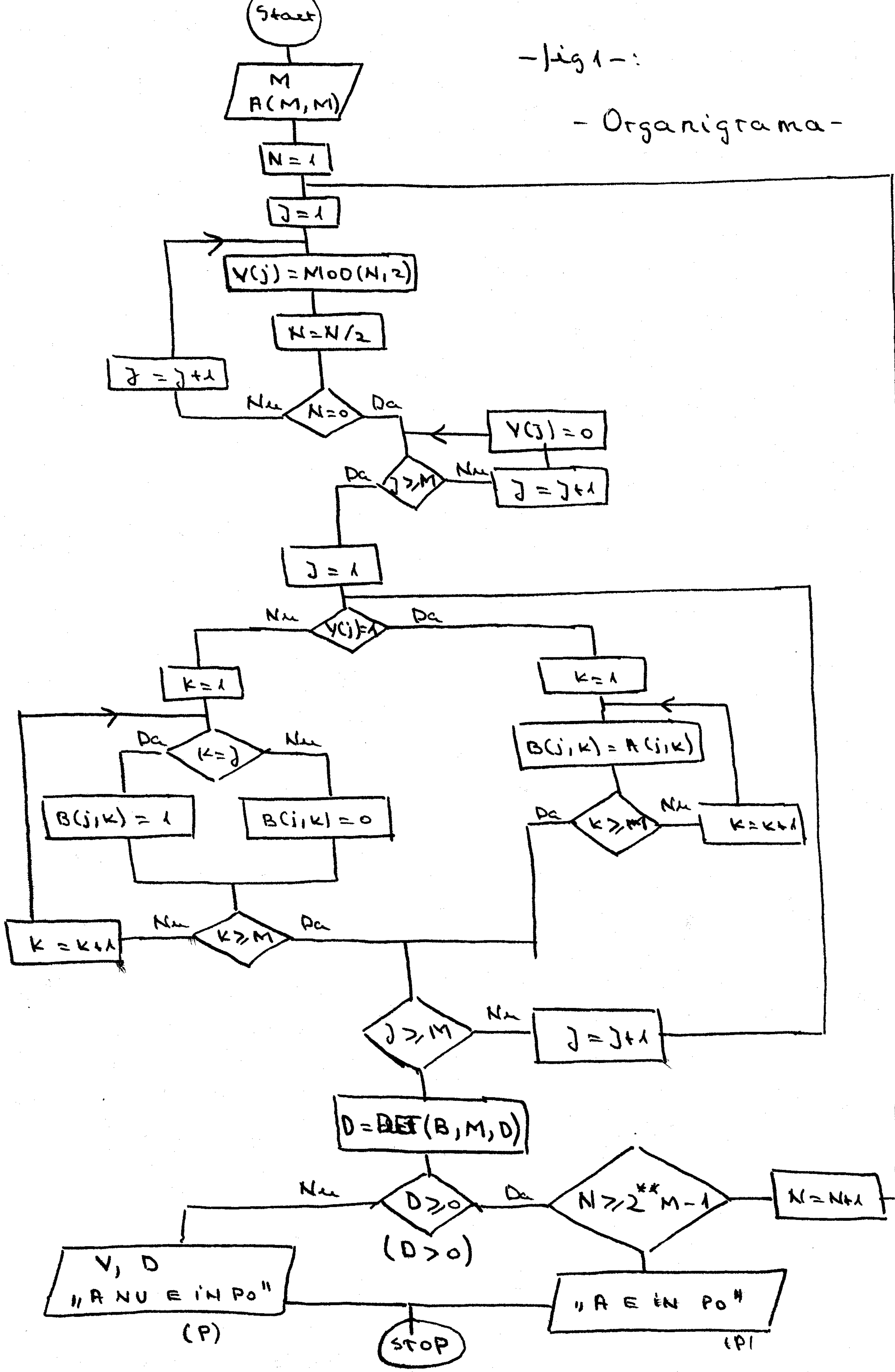
Peste tot, înlocuind condiția ca $\det B \geq 0$ cu $\det B > 0$ programul poate verifica fără nici o altă modificare apartenența la clasa P.

Programul a fost scris utilizând o instrucțiune DIMENSION care să permită calcularea matricilor de ordin 30 (care vor apare în circuite cu până la 15 tranzistori). Pt în rest, M-ul este variabil în READ, ceea ce dă o totală flexibilitate utilizatorului din punct de vedere al ordinului matricii pe care o testează.

Deocamdată se presupune că există o subrutină care calculează determinantul unei matrici de ordin M, numită DET(B, M, D) la care se apelează la fiecare pas.

-fig 1-:

- Organigrama -



```
12345 6 78910
          JØB  PO, AN:----, PN: IOAN
          CØMPILE FØRTRAN
      SØUBRØUTINE  DET(B,M,D)
          .
          .
      RETURN 50
      END
      DIMENSIØN  A(30,30), B(30,30), V(30)
      READ(105,1) M
    1 FØRMAT(I2)
      READ(105,2) ((A(I,J), J=1,M), I=1,M)
    2 FØRMAT(10F8.3)
      L = 2**M - 1
      DØ 17  N=1,L
      J = 1
    7 V(J) = MOD(N,2)
      N = N/2
      IF(N.NE.0) GØ TØ 5
    3 IF(J.GE.M) GØ TØ 4
      J = J+1
      V(J) = 0
      GØ TØ 3
    5 J = J+1
      ~~V(J)=0~~
      GØ TØ 7
    4 DØ 10  J=1,M
      V(J) = 1
      IF(V(J).EQ.1) GØ TØ 13
      DØ 20  K=1,M
      IF(K.EQ.J) GØ TØ 11
      B(J,K) = 0
      GØ TØ 20
      α
```

Subrutină pentru calculul determinanților

6 α

```
11    B(j,K)=1
20    CØNTINUE
      GØ TØ 10
13    DØ 30  K=1,M
30    B(j,K)=A(j,K)
10    CONTINUE
      CALL DET(B,M,D)
50    IF(D.GE.0) GØ TØ 17
      WRITE(108,33) D,(V(j)),j=1,M)
33    FØRMAT(5X,F8.3,5X,30I1)
17    CØNTINUE
      WRITE(108,22)
22    FORMAT(2X,'A ESTE IN PO')
      STOP
      END
          LINK
          RUN

          DATE

          EOJ
```

– Bibliografie –

I Articole

Articolele de mai jos sînt grupate în lucrarea „Nonlinear Networks" – A. Wilson – I.E.E. – 1973.

1) Introducere (Wilson)

2) „Circuite neliniare" (R.J. Duffin)

3) „Circuite neliniare RLC" (Desoer, Katzenelson)

4) „Condiții necesare și suficiente pentru convertibilitatea globală a unor operatori ce apar în analiza circuitelor" (Sandberg)

5) „Existența și unicitatea soluțiilor ecuațiilor circuitelor DC neliniare" (Sandberg)

6) „Circuite neliniare monotone" (Desoer și Wu)

7) „Despre soluțiile ecuațiilor circuitelor rezistive neliniare" (Wilson)

8) „O teorie a circuitelor neliniare" (Brayton și Moser)

9) „Asupra ecuațiilor dinamice a unei clase de circuite neliniare RLC" (Chua și Rohrer)

10) „Unele teoreme despre proprietățile soluțiilor DC a circuitelor neliniare" (Sandberg și Wilson)

11) „Unele proprietăți teoretice a circuitelor D.C cu tranzistoare neliniare" (Sandberg și Wilson)

12) „Teoreme noi despre ecuațiile circuitelor neliniare cu tranzistoare" (Wilson)

13) „Teoreme în analiza circuitelor neliniare cu tranzistoare" (Sandberg)

14) „Condiții pentru existența unei inverse globale a operatorilor circuitelor neliniare cu disp. semic." (Sandberg)

15) „Existența soluțiilor pentru ecuațiile circuitelor cu tranzistoare, rezistențe și surse de tensiune" (Sandberg și Wilson)

16) „Un model al tranzistorului controlat în sarcină" (Gummel)

17) „Unele teoreme asupra răspunsului dinamic a circuitelor cu tranzistoare, neliniare" (Sandberg)

18) „Teoreme despre calculul răspunsului tranzitoriu a circuitelor neliniare conținînd tranzistoare și diode" (Sandberg)

19) „Asupra ecuațiilor circuitelor neliniare" (T. Stern)

20) „Forma normală și stabilitatea unei clase de circuite neliniare cuplate" (Varaya și Liu)

21) „Teorema funcției inverse globale" (Wu și Desoer)

22) „Homeomorfismul global al funcțiilor cu valori vectoriale" (Chua și Lam)

23) „Unele rezultate asupra existenței și unicității soluțiilor ecuațiilor neliniare (Fujisawa, Kuh)

II [0] „Analiza calitativă fundamentală a circuitelor neliniare" (la care lucrez în prezent)

III Lucrări:

1) André Angot: „Complemente de matematici pentru inginerii din electronică și telecomunicații" E.T. București 1966

2) Norman Balabanian, T. A. Bickart „Teoria modernă a circuitelor" E.T. București 1974.

3) Gh. Cartianu ș.a. „Semnale, circuite și sisteme" E.D.P. București 1980

4) Adrian Corduneanu „Ecuații diferențiale cu aplicații în electrotehnică" E. „Facla" Timișoara 1981

5) D. Dascălu ș.a. „Dispozitive și circuite electronice" E.D.P. București 1982

6) —„— (probleme) E.D.P. București 198[illegible]

7) G. Dodescu, M. Toma „Metode de calcul numeric" E.D.P. București

8) D. Dascălu, L. Turic „Circuite electronice" E.D.P. București 1981

9) E.A. Gray, C.L. Searle „Bazele electronicii moderne" (vol I, II) EDP București 1973

10) G. Marin „Dispozitive și circuite electronice" - curs lit.

11) G.M. Fihtenholț „Curs de calcul diferențial și integral" vol I, III E.T. București 1964

12) W. Pfeiffer „Tehnica impulsului" E.T. București 1982

13) Gh. Șabac „Matematici speciale" (v I) EDP Buc. 1964

14) Stelian Niculescu „Fortran. Inițiere în programare structurată" E.T. 1979 București