

# ALGORITM PENTRU ANALIZA CIRCUITELOR ELECTRICE DE MARE COMPLEXITATE

prof.dr. Marius Preda  
prof.dr. ing. Fănică Spinei  
dr. ing. Horia Andrei

Institutul Politehnic București  
Cat. Electrotehnică

**Rezumat.** Se prezintă o metodă eficientă de calcul al funcțiilor de rețea bazată pe utilizarea metodei topologice, aplicată circuitului echivalent simplificat. Algoritm de calcul are următoarele părți principale:

- obținerea unui model adecvat al rețelei;
- pasivizarea automată a porțiilor inactive;
- reordonarea porțiilor de acces pentru care se determină funcțiile de transfer;
- determinarea topologică a funcțiilor de rețea.

Aplicarea metodei propuse permite analiza directă, unitară și sistematică a circuitelor de mare complexitate.

## 1. Problema abordată și studiul ei calitativ

Cele mai eficiente metode de analiză a regimurilor tranzitorii ale unor circuite neliniare de mare complexitate au la bază descompunerea circuitului, într-o parte rezistivă liniară reciprocă și partea de circuit complementară acestuia. Partea complementară are în aplicațiile curente un număr de elemente mult mai redus decât partea rezistivă liniară și reciprocă. Problema abordată în cele ce urmează este o generalizare a problemei circuitului liniar rezistiv de mare complexitate. Generalizarea constă în renunțarea la caracterul rezistiv al elementelor liniare și reciproce. În lucrare se tratează problema circuitului de mare complexitate liniar și reciproc, fără cuplaje magnetice în formă operațională, pentru regimul tranzitoriu, respectiv, complexă pentru regimul sinusoidal. Problema tratată implică descrierea structurii topologice a circuitului, descrierea parametrilor elementelor circuitului și efectuarea calculului necesare determinării funcțiilor de transfer căutate. Lucrarea prezintă modalități noi și eficiente, atât pentru partea de descriere a structurii topologice, cât și pentru determinarea funcțiilor de transfer.

## 2. Modelul utilizat pentru descrierea simultană a structurii topologice și a parametrilor circuitului format exclusiv din elemente dipol

Modalitatea uzuală de descriere a structurii topologice a unui circuit electric conex cu  $L$  laturi și  $N$  noduri constă în indicarea pentru fiecare latură a câte două numere naturale. Aceste numere precizează nodul de

început și nodul terminal al laturii respective. În plus, se indică pentru fiecare latură câte un număr complex care este valoarea parametrului circuitului în formă operațională. Având în vedere posibilitatea înlocuirii laturilor în derivație cu câte o latură echivalentă, ne referim în continuare la circuite complexe fără laturi în derivație. Cazul cel mai complex, studiat în cele ce urmează, este circuitul în poligon complet. Acest circuit la  $N$  noduri are un număr de laturi  $L=N*(N-1)/2$ . Ca urmare, folosind modalitatea uzuală de descriere a proprietăților acestui circuit se folosesc  $N*(N-1)$  întregi pentru descrierea structurii topologice și încă  $L$  numere complexe pentru descrierea valorilor parametrilor circuitului. Pentru determinarea funcțiilor de rețea ar mai fi necesare: formarea ecuațiilor circuitului, rezolvarea acestor ecuații și, în fine, calculul funcțiilor de rețea.

**Modelul triunghiular al circuitului** propus utilizează, pentru descrierea simultană a structurii topologice și a parametrilor circuitului studiat, un singur vector reprezentativ complex cu un număr de elemente egal cu numărul  $L$  al laturilor. Memoria utilizată pentru stocarea elementelor acestui vector (într-o fracțiune din ce în ce mai redusă) va fi folosită și pentru calcule necesare determinării funcțiilor de transfer.

Modelul triunghiular al circuitului este un tablou cu  $N-1$  linii, asociate nodurilor  $1, 2, \dots, N-1$ , și  $N-1$  coloane, asociate nodurilor  $2, 3, \dots, N$ . Elementul  $a_{jk}$  de pe linia  $j$  și coloana  $k$  are valoarea complexă a parametrului elementului ce formează latura circuitului ce unește nodul  $j$  cu nodul  $k$ . În figura 1a, este reprezentat un exemplu simplu de circuit, iar în figura 1b este redat tabloul model triunghiular al acestui circuit.

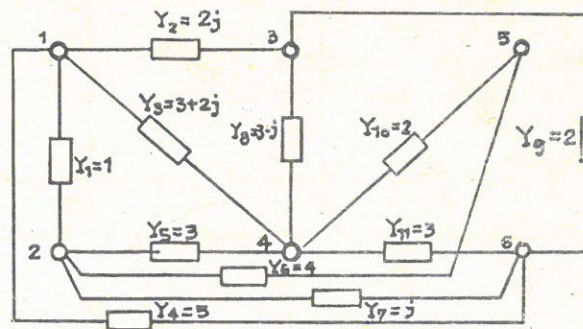


Fig. 1,a

	2	3	4	5	6
1	1	$2j$	$3+2j$	0	3
2		0	3	4	$j$
3			$3j$	0	2
4				$+2$	3
5					0

Fig. 1,b



Circuitul luat ca exemplu are  $N=6$  noduri și  $L=15$  laturi. Vectorul reprezentativ complex, notat  $V$  este  $V = (1 \ 2j \ 3+2j \ 0 \ 5 \ 0 \ 3 \ 4 \ j \ 3-j \ 0 \ 2 \ 2 \ 3 \ 0)$ .

Numărul de ordine în vectorul complex reprezentativ al laturii având parametrul complex pe linia  $j$  și coloana  $k$  se poate exprima simplu (și avantajos pentru algoritmul de calcul utilizat). Acest număr se exprimă în forma  $NO(j,k) = NA(j) + k$  cu  $NA(1)=-1, NA(2)=NA(1)+(N-2), NA(3)=NA(2)+(N-3), \dots, NA(h)=NA(h-1)+(N-h), \dots NA(N-1)=N*(N-3)/2$ .

### 3. Pasivizările implicate de determinarea unei funcții de rețea și reflectarea lor asupra vectorului reprezentativ complex.

În cazul general al unui circuit cu mai multe intrări și ieșiri, determinarea unei funcții de rețea implică menținerea unei singure surse și pasivizarea tuturor celorlalte surse. În figura 2.a, se reprezintă un circuit la care se evidențiază sursele  $i, 0$  corespunzătoare porții de intrare respectiv porții de ieșire. Ca exemplu, în aceeași figură se mai arată și două porți  $j, k$  care se pasivizează. Poarta  $j$  are o sursă ideală de tensiune, iar poarta  $k$  are o sursă ideală de curent. În figura 2.b este prezentat modelul triunghiular al

circuitului. Pasivizarea surselor ideale de tensiune corespunde înlocuirii lor cu câte un scurtcircuit. Pasivizarea surselor ideale de curent corespunde înlocuirii lor cu un gol. Aceste înlocuiri sînt ilustrate corespunzător în figura 2.c. Înlocuirea unui element dipolar cu un scurtcircuit conduce la suprapunerea celor două noduri  $f, g$ , ale elementului dipolar și, deci, la generarea unor laturi în paralel care se vor înlocui cu câte o latură echivalentă. Suprapunerea nodurilor și înlocuirea laturilor în paralel cu câte o latură echivalentă se reflectă asupra modelului triunghiular și asupra vectorului complex, reprezentativ prin două operații simple. Aceste operații sînt eliminarea liniei și coloanei nodului eliminat, respectiv adunarea anumitor elemente. Pentru a face univoc procedeul de suprapunere, se face o convenție de eliminare a nodului cu numărul de ordine mai mare la fiecare pereche de noduri suprapuse. Figura 2.d ilustrează realizarea celor două operații amintite și mai indică renumerotarea unor noduri.

### 4. Reducerea vectorului complex reprezentativ al circuitului multipolar

Vectorul complex, reprezentativ al multipolului obținut după pasivizarea surselor făcută la punctul 3 poate fi redus substanțial, printr-un algoritm simplu, fără a pierde informațiile necesare determinării

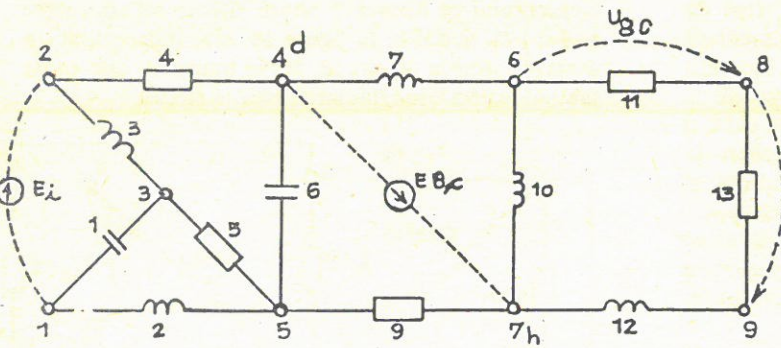


Fig. 2, a

			d					h
	2	3	4	5	6	7	8	9
1		x	a	x		i		
2		x	b			j		
3			c	x		k		
4 d				d	e	f	g	h
5						l		
6						m	x	
7 h							p	q
8								x

Fig. 2, b

			d					h
	2	3	4	5	6	7	8	9
1		x	a+i	x		i		
2		x	b+j			j		
3			c+k	x		k		
4 d				d+l	e+m	f	g+p	h+t
5						l		
6						m	x	
7 h							p	q
8								x

Fig. 2, c

	2	3	4	5	6	7	8	
1		x	a+i	x				
2		x	b+j					
3			c+k	x				
4				d+l	e+m	g+p	h+q	
5								
6								
7							x	

Fig. 2, d



funcțiilor de transfer corespunzătoare acestei pasivizării. Această reducere corespunde, evident, înlocuirii multipolului cu tetrapolul echivalent având cele patru noduri de acces ale porții de intrare, respectiv ieșire. Forma simplificată a algoritmului de reducere se obține printr-o modificare ușor de realizat a modelului triunghiular, respectiv a vectorului complex reprezentativ al circuitului multiport. Aceasta se face prin mutarea parametrilor laturilor corespunzătoare tetrapolului echivalent pe ultimele șase locuri din modelul triunghiular. Procedul este exemplificat în figura 3 pentru un circuit cu numai 10 noduri la care latura porții de intrare avea inițial nodurile 2,5, iar latura de ieșire avea nodurile 4, 8. După această modificare a modelului triunghiular, obținerea tetrapolului echivalent corespunde eliminării succesive a primei linii a tabloului triunghiular curent (respectiv scurtării vectorului reprezentativ) și recalculării simple a elementelor acestui tablou. Recalcularea la eliminarea primei linii curente se face astfel:

- se face suma  $S$  a elementelor situate pe această linie;
- se înlocuiesc pe rând elementele  $Y_{jk}$  ale fiecărei linii inferioare  $j$ , situate pe coloana  $k$  printr-o nouă valoare

$$Y'_{jk} = Y_{jk} + Y_{1j} * Y_{1k} / S$$

Prezentarea detaliată a algoritmului se face în anexa I.

## 5. Determinarea topologică a funcțiilor de rețea pe baza formei reduse a vectorului reprezentativ complex

Calculul diverselor tipuri de funcții de rețea se poate face, fără a mai utiliza și rezolva mai multe sisteme de ecuații, cu ajutorul unei formule topologice generale [B3].

Formula topologică utilizată exprimă o funcție de rețea operațională  $F_{jh}$  definită în condiții inițiale de zero ca

$F_{jh} = N_j / M_h$  la  $M_k = 0$ , pentru  $K = 1, 2, \dots, h-1, h+1, \dots, n_i$ , cu:

$M_h$  - Transformata Laplace a mărimii de intrare de la poarta  $h$ ;

$N_j$  - transformata Laplace a mărimii de ieșire de la poarta  $j$ ;

$n_i$  - numărul porților circuitului.

Expresia formulei topologice [B3] este:

$$F_{jh} = -T_1 / T_p, \text{ cu:}$$

$T_1$  suma valorilor algebrice ale arborilor comuni, care conțin latura unitate a porților  $j$  și  $h$  în graful circuitului;

$T_p$  suma valorilor algebrice ale arborilor comuni în graful circuitului cu porților  $j$  și  $h$  pasivizate.

În cele de mai sus, arborii comuni sînt arbori formați din laturi cu același nume din mulțimea laturilor comandate, respectiv, din mulțimea laturilor de comandă. Laturile de comandă au adăugat la nume indicele  $C$ , iar cele comandate au indicele  $c$ . Arborii comuni care conțin laturile surselor cu anumite nume sînt arbori în graful circuitului cu surse pasivizate pentru laturile cu nume neconținute.

Valoarea unui arbore comun format din anumite laturi este produsul parametrilor acelor laturi. Laturile corespunzătoare porții de intrare și porții de ieșire au valoarea 1. Laturile corespunzătoare unui element pasiv au ca valoare admitanța (operațională) a elementului. În fine, laturile corespunzătoare unei surse comandate (latura cu sursa și latura de comandă a sursei) au ca parametru, parametrul de comandă al sursei.

Ca exemplu, se consideră circuitul din figura 4. Se consideră la poarta  $i$  de intrare o mărime de intrare tensiune.

Ca urmare, la intrare se pune o latură  $O_c$  cu o sursă comandată de tensiune. Se consideră la poarta  $o$  de ieșire o tensiune. Ca urmare, la ieșire apare o latură de comandă  $O_C$  pentru sursa comandată de la intrare.

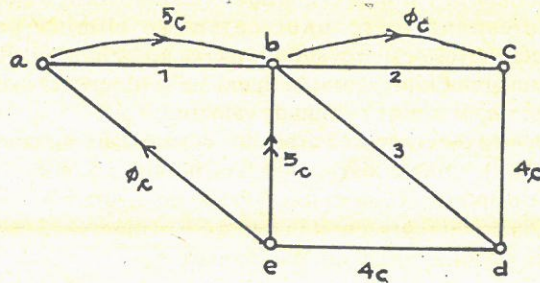


Fig. 4

Laturile corespunzătoare unor surse de tensiune se marchează cu o săgeată indicînd sensul de referință al tensiunii electromotoare a sursei. Laturile corespunzătoare unor surse de curent se marchează cu o săgeată dublă indicînd sensul de referință al intensității curentului sursei. Laturile corespunzătoare unor elemente dipolare pasive nu au săgeți care să indice sensuri de referință. Pentru tensiunile de comandă se consideră cite o latură cu săgeată simplă, respectiv dublă, după cum mărimea de comandă este o tensiune, respectiv un curent. Aceste săgeți sînt utile din două motive, și anume, pentru a indica natura surselor respectiv a mărimii de comandă (natură care determină modul de pasivizare) și, în al doilea rînd, pentru a indica un sens care intervine în determinarea semnului la însumarea algebrică a valorii arborilor comuni.

Pentru determinarea arborilor comuni, care conțin numai laturile unitate, notate cu  $O_c$  respectiv  $O_C$ , se



va folosi graful din figura 5.a. În aceasta, laturile c, C ale celorlalte surse comandate sînt pasivizate.

Pasivizarea laturilor sursă (notate cu indicele c) se face prin eliminarea lor. O latură corespunzătoare unei surse de tensiune pasivizate se elimină prin înlăturarea

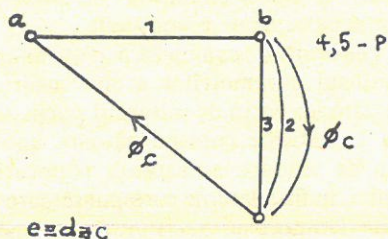


Fig. 5,a

ei și prin suprapunerea nodurilor laturii. O latură corespunzătoare unei surse de curent pasivizate se elimină prin simpla ei înlăturare.

Pasivizarea laturilor de comandă (notate cu indicele C) se face prin eliminarea lor de o manieră inversă. O latură corespunzătoare unei tensiuni, la pasivizare, se elimină prin simpla ei înlăturare. La pasivizare, o latură corespunzătoare unui curent se elimină prin înlăturarea ei și prin suprapunerea nodurilor laturii.

Din graful reprezentat în figura 5.a se observă că există un singur arbore comun de valoare  $1 \cdot Y_1$ .

Pentru determinarea arborilor comuni care nu conțin laturile unitate, dar conțin laturile sursei 5, notate cu  $5_c$  respectiv  $5_C$ , se va folosi graful din figura 5.b.

Din graful reprezentat în figura 5.b se observă că există un singur arbore comun de valoare  $Y_5$ .

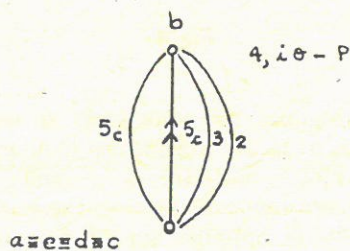


Fig. 5,b

Dificultatea problemei determinării arborilor comuni (care sînt adesea în număr foarte mare) și, mai ales, problema determinării factorului de semn pentru fiecare pereche de arbori comuni limitează, din punct de vedere pragmatic, aplicabilitatea metodelor de analiză topologică la circuite cu număr redus de laturi (de ordinul zecilor).

Folosirea vectorului reprezentativ redus al circuitului conduce la circuite cu dimensiuni reduse la care determinarea funcțiilor de rețea se poate face într-o formă generală, simplă și eficient. Acest lucru este dovedit de algoritmul în limbaj C++ prezentat la anexa I.

## 6. Concluzii

Combinarea metodelor topologice cu metodele de reducere prin echivalență permite elaborarea unor algoritmi eficienți de analiză a circuitelor electrice cu număr mare de elemente. Descrierea simultană a proprietăților topologice și a mărimilor caracteristice ale elementelor dipolare lineare și reciproce cu ajutorul vectorului reprezentativ, definit în lucrare, poate contribui la creșterea eficienței algoritmilor de analiză a circuitelor electrice, cu număr mare de elemente.

## Bibliografie

1. CHUA, L.O., LIN, P.M. - *Computer - aided analysis of electronic circuits*, Prentice Hall, Englewood Cliffs, 1975.
2. SCHWARZ, A.P. - *Computer aided design of microelectronic circuits and systems*, Academic Press, Londra, 1987.
3. PREDA, M., CRISTEA, P. - *Bazele electrotehnicii*, vol. II, Editura Didactică și Pedagogică, București, 1980.
4. CHUA, L.O., DESOER, C.A., KUH, E.S. - *Linear and nonlinear circuits*, Mc Graw-Hill, New York, 1987.
5. CALLA, W.J. - *Fundamentals of computer aided circuit simulation*, Kluwer Academic Publisher, Boston/Lancaster, 1988.



## ANEXA I

```

/*****/
/* PROGRAM FTRANSFERB-CPP
*****/
/* program transfk; 6.01.91 */
# include <studio.h>
# include <iostream.h>
# include <conio.h>
# include <time.h>
# include <dos.h>
# include <io.h>
# include <string.h>
# include <fcntl.h>
# include <sys\stat.h>
const int nc=170, mc169, pc=14365, nc2=250;
int P,PELN,PELNT;
float T [pc+1],VM1 [nc2],VM2 [nc2];
int M,N,NLM,NLT;
float T1, TPR,TPA,TPB,TPG,TPO;
float GI,RI,RT,GT,AU,AI;
FILE *FT;
/*vm, T2:array[2..mc] of r;*/
float K,A,RR,SS;
/*****/
/*****/
avoid CALCULT(avoid)
{
float P1,P2,P3,P4,P5,P6;
float A,B;
float T1,TPR,TPA,TPB,TPG,TPO;
float GI,RI,RT,GT,AU,AI;
P1=T[6];P2=T[5];P3=T[4];
P4=T[3];P5=T[2];P6=T[1];
printf ("\n T: %f %f %f %f %f %f
",P1,P2,P3,P4,P5,P6);
A=P2+P3;B=P4+P5;

T1=P2*p5-P3*P4;
TPG=P2+P3+P4+P5;
TPA=P6*TPG+A*B;
TPB=P1*TPG+A*B;
TPR=P6*TPB+P1*(P2+P4)*(P3+P5)+
+P2*P3*B+P4*P5*A;
TPO=P6*(P2+P3+P4+P5)+(P2+P4)*(P3+P5);
GT=-T1/TPG;
AU=-T1/TPA;
AI=-T1/TPB;
RT=-T1/TPR;
RI=TPO/TPR;
GO=TPR/TPO;
cout << "CONDUCTANTA DE TRANSFER:" <<
GT << "\n";
cout << "AMPLIFICAREA ÎN TENSIUNE:" <<
AU << "\n";
cout << "AMPLIFICAREA ÎN CURENT:" <<
AI << "\n";

```

```

cout << "REZISTENȚA DE TRANSFER:" << RT
<< "\n";
cout << "CONDUCTANTA DE TRANSFER:" <<
GT << "\n";
cout << "REZISTENȚA DE INTRARE:" << RI
<< "\n";
cout << "CONDUCTANTA DE INTRARE:" <<
GI << "\n";
}
/*****/
/* void WRT (int NO)
{inf X,Z;
printf ("% \n");
for (x=1;X==NO;X++)
{
printf ("%d ",T[X];
for (Z=1;Z== X;Z++O) printf (" ");
}
}
*/
/*****/
/*****/
/*PROGRAM PRINCIPAL*/
/*****/
main ()
{ int X,Y,Z,L,C,L1,C1,NB,NO,X5;
int CLC,CL,CC;
float H,K,A,SS;
// main
clrscr ();
printf ("PROGRAM PASCAL DE CALCUL AL
FUNCTIILOR DE TRANSFER \n");
printf (" - PRINCIPIU NOU - \n by DR. DOC.
MARIUS PREDA \n");
printf (" BUC; 19.02.1990/26.06.90 \N");
printf ("N= ");
cin >> N;
cout < "\n NL.MENȚINUTE= ";cin >> NLM;
FT=fopen("E:\FT1.CPP","wb+");
NLT=N-1;
M=N-1;
if ((N%2) == 0) {P=(N/2)*M;} else {P=M/2)*N;}
if ((M%2) == 0){PELN=(M/2) * (M -1) +1 -pc;}
else {PELEN=((M-1)/2)*M + 1 -pc;}
PELNT=mc*(mc-1)/2;
/*scrierea fișierului FT cu conductantele */
/*initialisation of T*/
CC=(mc*nc)/2;
cout << "CC=" < CC < "Â";
if (N<nc) { CC=P;}
cout << "CC=" << CC << "\n";
cout << "P==" << "P-CC=" << "P-CC << "\n";
for (Y=1;Y++)
{
T[Y]=1.0;
}
}

```



```

cout << "GATA T" << "\n";
fseek (FT,OL,SEEK_SET);
if (N>nc)
}
    for (x=0;x P-CC;X++)
    {
        K=1;
        fwrite(&K,sizeof(float),1,FT);
    }
    cout << "GATA FT" << "\n;
}
/* cout<<"NO,PARAMETRU: ";cin >>NO>> A;
while (NO>0)
{
    fseek (FT,NO*sizeof(float), SEEK_SET);
    fwrite(&A,sizeof(float),1,FT);
    printf("NO,PAR");
    cin >>NO>>A;
}
*/
    clock_t sta:t, end;
start = clock();
    CL=PELN;
for (L=N;L=nc+1; L--)
{ // L
    /* CITIREA liniei L din file FT */
    printf ("In FT L= %d \n",L;
    CC=CL;
    fseek (FT,CC*sizeof(float), SEEK_SET);
    for (X5=1; X51;X5++)
        {fread(&H,sizeof(H),1,FT); VM1[X5]=H;
        };
    /* calcule */
    SS=0;
    for (C=1;C<=1;C++)
    { SS=SS+VM1[C];}
    /*citirea liniei L1 din FT în VM2 */
    CLC=CL-L+2;
    for (L1=L-1;L1= nc+1;L1--)
    {
        fseek(FT,CLC*sizeof(float), SEEK_SET);
        for (C1=1;C<=L1-1;C1++)
            {fread(&H,sizeof(H),1,FT); VM2[C1]=H;
            /* printf("X5= %d v2= %f", X5,H); */
            }
        /* calculul liniei curente */
        A=VM1[L1];
        for (C1=1; C11-1;c1++)
            {VM2[C1]=VM2[C1]+a/SS*VM1[C1];}
        /* rescrierea liniei curente L1 în fişierul FT */
        fseek(FT,CLC*sizeof(float),SEEK_SET);
        for (X5=1;X5<=L1-1;X5++)
            { H=VM2[X5];
            /* printf("X5= %d v22= %f",X5,H); */

```

```

        fwrite(&H,sizeof(H),1,FT);}
        CLC+CLC-L1+2;
    }
    /*citirea liniilor L1 din T*/
    CLC=PELNT;
    for (L1=nc;L1=2;L1--)
        A=VM[L1];CC=CLC;
    for (C1=1;C<=L1-1;C1++)
        {T[C1+CC]=T[C1+CC]+A/SS*VM1[C1];}
        CLC=CLC-L1+2;
    }
    CL=CL-L+2;
} // L-END
fclose(FT); printf("PELNT %d ",PELNT);
/*-----*/
/*calculul liniilor L din T */
if (N=mc) {NLT=N;} else {NLT=nc;}
PELNT=((NLT-2)*(NLT-1))/2;
CL=PELNT;
for (L=NLT;L>=NLM+2;L--)
{ //L-2
    printf("DIN T: L= %d ",L);
    SS=0; for (C1=1;C1<=L-1;C1++)
    {
        A=T[CL+C1];SS=SS+A;
        VM1[C1]=A;
    };
    /*calculul liniei curente L1 din T1*/
    CLC=CL-L+2;
    for (L1=L-1;L1>=2;L1--)
    {
        A=VM1[L1];CC=CLC;
        for (C1=1;C1<=L1-1;C1++)
            {T[CC+C1]+C1f=T[CC+C1]+
            +A/SS*VM1[C1];}
            CLC=CLC-L1+2;
        };
        CL=CL-L+2;
    } // L-2 end
    CALCULT();
    cout << "Nr.Noduri: " << N << "\n";
    cout << "NR.LATURI: " << P << "\n";
    cout << "TRIUNGHI MENTINUT: \n";
    end = clock();
    printf("The time was: %f sec \n", (end -start)
    / CLK_TCK);
    printf("STOP ");
}
// main
/*****
/*****

```