

UNIVERSITATEA "POLITEHNICA" DIN TIMIȘOARA  
FACULTATEA DE MECANICĂ

ing. ADRIAN SORIN ROȘCA

**CONTRIBUȚII LA STUDIUL PRIN ANALIZĂ  
NUMERICĂ A CÂMPULUI TERMIC LA  
PRELUCRAREA PRIN AȘCHIERE**

**TEZĂ DE DOCTORAT**

619.493  
366 C

BIBLIOTECA CENTRALĂ  
UNIVERSITATEA "POLITEHNICA"  
TIMIȘOARA

Conducător științific:  
Prof.univ.dr.ing. VASILE POPOVICI

TIMIȘOARA 1997

# CUPRINS

INTRODUCERE.....	I
<b>1. STADIUL CERCETĂRILOR PRIVIND STUDIUL CÂMPULUI TERMIC DIN PROCESELE DE AȘCHIERE .....</b>	<b>3</b>
1.1. SOLICITĂRI TERMICE ÎN PROCESELE DE AȘCHIERE .....	3
1.1.1. Fluxul termic .....	3
1.1.2. Tipuri de fenomene termice .....	4
1.1.3. Rigiditate termică .....	6
1.2. SURSE DE CĂLDURĂ ÎN PROCESUL DE AȘCHIERE.....	7
1.2.1. Surse interne din procesul de așchiere.....	7
1.2.2. Mașina unealtă .....	7
1.2.3. Surse externe .....	8
1.3. REPARTIZAREA CĂLDURII .....	8
1.3.1. Scula .....	10
1.3.2. Semifabricatul.....	11
1.3.3. Mașina unealtă.....	12
1.4. METODE DE STUDIU A CÂMPULUI TERMIC DIN PROCESELE DE AȘCHIERE .....	13
1.4.1. Metode analitice .....	13
1.4.2. Metode numerice.....	14
1.4.3. Metode experimentale .....	16
1.4.3.1. Metode clasice .....	16
1.4.3.2. Metoda fotografiei termice.....	17
1.4.3.3. Metoda termografiei electronice.....	20
1.4.3.3.1. Determinarea directă a temperaturii zonei de așchiere la așch- ierea continuă .....	21
1.4.3.3.2. Determinarea câmpului termic la așchieria întreruptă.....	22
1.4.3.4. Metoda cristalelor lichide .....	23
1.4.3.5. Metoda interferometriei holografice .....	24
1.4.3.6. Metoda analogiei electrooptice .....	24
1.4.4. Studiul deformațiilor termice .....	25
1.5. CONCLUZII.....	27
1.6. OBIECTIVELE STUDIULUI.....	28
<b>2. METODA ELEMENTULUI FINIT .....</b>	<b>30</b>
2.1. FORMULĂRI ALE METODEI .....	31
2.2. ETAPE ALE ANALIZEI.....	34
2.2.1. Aproximarea geometrică a domeniului studiat .....	34
2.2.2. Alegerea sau proiectarea elementelor finite.....	35
2.2.2.1. Alegerea aproximării soluției pentru variabila de câmp în interi- orul elementelor.....	35
2.2.2.2. Funcția de formă a elementului.....	36
2.2.2.3. Dezvoltarea ecuațiilor la nivel de element.....	38
2.2.2.4. Evaluarea elementelor izoparametrice.....	39
2.2.3. Asamblarea matricelor elementare .....	42
2.2.4. Aplicarea încărcărilor .....	44
2.2.5. Impunerea condițiilor de contur .....	44
2.2.6. Rezolvarea sistemului algebric echivalent.....	47
2.2.7. Interpretarea rezultatelor .....	48
2.3. CONCLUZII PRIVIND POSIBILITĂȚILE DE UTILIZARE A METODEI ELEMENTULUI FINIT.....	49
<b>3. MODELAREA CÂMPULUI TERMIC .....</b>	<b>51</b>

3.1. ECUAȚIILE TRANSFERULUI TERMIC .....	51
3.1.1. Transferul prin conducție .....	51
3.1.1.1. Ecuația difuziei moleculare .....	51
3.1.1.2. Condiții de contur și condiții inițiale .....	53
3.1.2. Transferul prin convecție .....	54
3.1.3. Transferul prin radiație .....	55
3.2. MODELAREA CÂMPULUI TERMIC CU METODA ELEMENTU- LUI FINIT .....	58
3.2.1. Funcționala pentru transferul de căldură prin conducție .....	58
3.2.2. Proiectarea elementelor finite .....	59
3.2.2.1. Ecuațiile elementelor .....	60
3.2.3. Asamblarea matricelor sistemului .....	64
3.2.3.1. Considerații legate de manevrarea matricelor sistemului .....	64
3.2.4. Încărcări și condiții de contur .....	67
3.2.5. Rezolvarea sistemului .....	68
3.3. CONCLUZII PRIVIND MODELAREA CÂMPULUI TERMIC .....	70
<b>4. CERCETĂRI EXPERIMENTALE .....</b>	<b>72</b>
4.1. STRATEGIA DE EXPERIMENTARE .....	72
4.2. PREZENTAREA STANDULUI .....	75
4.3. ECHIPAMENTE FOLOSITE .....	77
4.3.1. Mașina unealtă .....	77
4.3.2. Scule .....	77
4.3.3. Plăcuțe amovibile .....	78
4.3.4. Materiale .....	79
4.3.5. Scanner infraroșu .....	79
4.3.6. Placă achiziții date PCL711S .....	80
4.3.7. Echipament tensometric .....	81
4.3.8. Organizarea experimentului .....	84
4.4. ETALONAREA MĂSURĂTORILOR TENSOMETRICE .....	86
4.4.1. Stabilirea ecuațiilor de etalonare .....	94
4.5. CONDIȚIILE DE MĂSURARE A TEMPERATURII PRIN TER- MOGRAFIE ELECTRONICĂ .....	98
4.6. CORECȚIA DE EMISIVITATE .....	100
4.6.1. Principiul instalației de determinat emisivitatea .....	100
4.6.2. Soluție constructivă pentru instalația de măsurat emisivitatea .....	100
4.6.3. Determinarea emisivității plăcuțelor de așchiere .....	105
4.7. CONCLUZII .....	109
<b>5. PRELUCRAREA DATELOR EXPERIMENTALE .....</b>	<b>111</b>
5.1. TRATAREA TERMOGRAMELOR .....	111
5.2. EXTRAGEREA TEMPERATURILOR DIN TERMOGRAME .....	127
5.2.1. Obținerea temperaturilor necorectate .....	127
5.2.2. Aplicarea corecției de emisivitate .....	129
5.3. DETERMINAREA CARACTERISTICILOR SURSEI TERMICE .....	131
5.3.1. Algoritm de obținere a încărcărilor la nivel de element .....	131
5.3.2. Obținerea relațiilor de estimare a puterii sursei .....	140
5.4. ESTIMAREA PRECIZIEI RELAȚIILOR OBȚINUTE .....	145
5.4.1. Date experimentale pentru forță și debit .....	146
5.4.2. Regresii liniare pentru forța de așchiere și debitul sursei termice .....	150
5.4.3. Estimarea preciziei relațiilor de forță și debit .....	158
5.4.4. Tabele pentru regresii polinomiale multiple de ordinul 2 ale debitului sursei termice .....	160
5.4.5. Estimarea preciziei relațiilor $Q=f(F,v)$ pentru debit .....	168
5.5. REPREZENTAREA GRAFICĂ A RELAȚIILOR .....	170

<b>6. CONCLUZII.....</b>	<b>175</b>
6.1. DESPRE UTILIZAREA TERMOGRAFIEI ÎN INFRAROȘU .....	175
6.2. DESPRE UTILIZAREA MĂSURĂTORILOR TENSOMETRICE.....	177
6.3. DESPRE APLICAREA METODEI ELEMENTULUI FINIT LA CALCULUL PUTERII SURSEI TERMICE .....	177
6.4. DESPRE INFLUENȚA FACTORILOR DE PRELUCRARE ASUPRA PUTERII SURSEI TERMICE .....	178
<b>7. CONTRIBUȚII .....</b>	<b>179</b>
7.1. ÎN PLAN TEORETIC .....	179
7.2. ÎN PLAN EXPERIMENTAL .....	180
<b>8. ANEXA 1 - PROGRAM ANALIZĂ ELEMENT FINIT.....</b>	<b>183</b>
8.1. CARACTERISTICI .....	183
8.2. FIȘIER HEADER.....	186
8.3. FIȘIER SURSĂ .....	189
<b>9. ANEXA 2 - PROGRAM POSTPROCESARE TERMOGRAME .....</b>	<b>214</b>
9.1. CARACTERISTICI .....	214
9.2. FIȘIER HEADER.....	215
9.3. FIȘIER SURSĂ .....	216
<b>10. ANEXA 3 - PROGRAM ACHIZIȚII DATE.....</b>	<b>230</b>
10.1. CARACTERISTICI.....	230
10.2. FIȘIER HEADER .....	232
10.3. FIȘIER SURSĂ .....	233
<b>11. ANEXA 4 - PROGRAM TRATARE TERMOGRAME .....</b>	<b>252</b>
11.1. CARACTERISTICI.....	252
11.2. FIȘIER HEADER .....	254
11.3. FIȘIER SURSĂ.....	255
<b>12. ANEXA 5.....</b>	<b>275</b>
<b>13. ANEXA 6 - CORECȚIA DE EMISIVITATE .....</b>	<b>276</b>
13.1. FUNCȚIA CARE REZOLVĂ ECUAȚIA DE CORECȚIE .....	276
13.2. FOAIA DE CALCUL PENTRU CORECȚIA DE EMISIVITATE .....	277
<b>14. ANEXA 7 - EXEMPLU DE TRATARE A UNEI TERMOGRAME .....</b>	<b>278</b>
<b>15. ANEXA 8 - PROGRAM REPREZENTARE GRAFICĂ.....</b>	<b>288</b>
15.1. CARACTERISTICI.....	288
<b>16. BIBLIOGRAFIE .....</b>	<b>290</b>



## INTRODUCERE

*Obținerea unui reper în condițiile tehnice impuse de funcționalitatea sa într-un ansamblu, este principalul obiectiv al departamentelor de proiectare constructivă și tehnologică. Deși în ultimul timp este evidentă o diversificare a tehnologiilor neconvenționale corelată cu o creștere a ponderii acestora în ansamblul industriei, totuși prelucrările convenționale prin așchiere ocupă, conform ultimelor statistici, 70% din totalul metodelor de prelucrare folosite pe plan mondial.*

*Studiul factorilor ce influențează precizia de prelucrare la procesele de așchiere este în preocupările specialiștilor încă din deceniul 6, făcând obiectul numeroaselor studii și teme de cercertare. Drăghici Gh. în [24]<sup>1</sup> a grupat acești factori după cum urmează:*

① *Factori dependenți de regimul de așchiere:*

- ◆ *deformația termică a sistemului MDPS;*
- ◆ *uzura sculei măsurată pe direcția normală la suprafața de așchiere;*
- ◆ *deformația elastică a sistemului;*
- ◆ *tensiuni interne din semifabricat în urma prelucrării;*
- ◆ *vibrații introduse de sistemul tehnologic și de procesul în sine;*
- ◆ *forțele de așchiere;*
- ◆ *parametrii regimului de așchiere.*

② *Factori independenți de regimul de așchiere:*

- ◆ *starea mașinii unelte (erori geometrice , mărimea jocurilor din lagăre etc.);*
- ◆ *calitatea semifabricatului (precizia dimensională și a formei);*
- ◆ *erorile de instalare în dispozitiv;*
- ◆ *precizia instrumentelor de control;*
- ◆ *erorile dimensionale și de formă ale sculelor;*

---

<sup>1</sup> Drăghici Gh., *Tehnologia construcțiilor de mașini*, p.156

- ♦ tensiuni remanente din semifabricatele turnate , forjate , sudate;
- ♦ eroarea de reglaj a mașinii unelte;
- ♦ gradul de atenție al executantului.

Estimarea influenței acestor factori asupra preciziei de prelucrare este posibilă numai pentru cei dependenți de regimul de așchiere. În funcție de situația concretă de prelucrare se pot face anumite aprecieri ale comportării sistemului MDSP fără însă a putea oferi o soluție analitică globală, pentru urmărirea simultană a tuturor factorilor și generală, care să modeleze cât mai multe situații.

Dintre factorii din prima grupă, deformația termică a sistemului MDPS este cel care are, în multe cazuri, o influență considerabilă asupra preciziei. Factorul termic produce în procesul de așchiere două efecte:

- efectul asupra preciziei de prelucrare , datorită dilatărilor componentelor sistemului , cu implicații asupra dimensiunilor obținute și a poziției relative dintre componente;

- efectul asupra structurii stratului de suprafață al materialului și eventual al sculei , când temperatura indusă este în apropierea unor puncte de transformare de fază ale materialului;

Deși fenomenele termice sunt în general controlabile datorită fundamentului matematic bine studiat care stă la baza modurilor de transmitere a căldurii, într-un proces de așchiere intervin o multitudine de factori care fac destul de dificilă modelarea fidelă a acestuia.

În eventualitatea realizării unui model analitic al întregului proces de așchiere, trebuie modelate simultan următoarele fenomene:

- deformații elastice ale sistemului de prelucrare;
- deformații plastice ale stratului de așchiat care degenerază în desprinderea așchiei;
- influența câmpului termic având în vedere toate cele trei moduri de transmitere a căldurii;
- oscilațiile autoîntreținute care apar în timpul așchierii.

Dacă la analiza deformațiilor elastoplastice este suficient să se introducă proprietățile de material și forțele, ca date de intrare pentru o simulare numerică, la o analiză termică în primul rând este necesar să se cunoască precis caracteristicile surselor termice, proprietățile de material și dependența lor de temperatură, precum și condițiile de contur specifice.

În decursul celor câteva decenii de când s-au făcut cercetări asupra fenomenelor termice din procesele de așchiere, metodele folosite în studiul experimental și teoretic au fost din cele mai diverse; în primul capitol al

lucrării de față fiind prezentată o recenzie a celor mai importante dintre acestea.

În capitolele 2 și 3 sunt prezentate în detaliu formulările matematice care stau la baza fenomenelor termice, precum și implementarea lor în metoda elementului finit. Aceste elemente au stat la baza programelor de calcul prezentate în anexele 1, 2 și 4, programe pe baza cărora s-au formulat concluziile finale din acest studiu.

În capitolul 4 este prezentată strategia experimentală folosită, care a inclus utilizarea termografiei electronice în infraroșu, monitorizarea forțelor de așchiere și determinarea caracteristicilor de emisivitate pentru corpurile implicate în studiu.

În capitolul 5 este prezentată prelucrarea datelor măsurate, care au fost ulterior introduse în programele de calcul amintite. Rezultatele acestor programe, au fost tratate prin regresie și s-au putut astfel obține relații de estimare a aportului surselor termice din procesele de așchiere, în anumite condiții precizate.

Principalele concluzii desprinse din această cercetare se referă la:

- utilizarea termografiei electronice în infraroșu, ca metodă ce oferă cel mai mare volum de informații pentru investigarea câmpului termic de la suprafața unui corp;
- utilizarea tensometriei la măsurarea forțelor de așchiere;
- aplicarea metodei elementului finit pentru determinarea puterii sursei termice din procesele de așchiere;
- influența diferiților factori asupra acestei surse termice

În finalul lucrării sunt prezentate contribuțiile autorului la studiul câmpului termic, contribuții aduse atât în plan teoretic (prin elaborarea metodologiei și algoritmilor de calcul prin metoda elementului finit pentru problema inversă a transferului de căldură), cât și în plan experimental.

Reușita prezentului demers științific a fost posibilă în primul rând grație competenței îndrumării de care am beneficiat pe tot parcursul cercetării din partea prof. univ. dr. ing. Vasile Popovici.

Îmi exprim pe această cale respectuosul meu omagiu pentru sprijinul acordat de domnia sa, pe parcursul întregii perioade de pregătire, pentru generozitatea în transmiterea gândirii și a rigorii științifice, nutrinđ speranța că prin lucrarea de față am răspuns cel puțin în parte exigențelor domniei sale.

*Corpului profesoral de la catedra de specialitate a Facultății de Mecanică a Universității Politehnica din Timișoara, conducerii acesteia, le mulțumesc pentru condescendența manifestată în această perioadă de pregătire prin doctorat.*

*Mulțumesc membrilor corpului didactic de la Facultatea de Mecanică din Craiova, care m-au susținut și mi-au oferit sugestii în perioada desfășurării acestui studiu.*

*Adresez sincere mulțumiri corpului didactic de la Facultatea Design and Technology din cadrul Universității din Luton Marea Britanie și în mod special domnului profesor Roger Harvey, coordonatorul activității de cercetare din această facultate, în laboratoarele căreia am realizat experimentul.*

*Vreau să închei această scurtă introducere în subiectul cercetării, prin mulțumiri adresate familiei, fără sprijinul și înțelegerea căreia nu ar fi fost posibilă finalizarea acestui studiu.*

*Craiova, septembrie 1997*

*Autorul*

# CAPITOLUL 1

## STADIUL CERCETĂRILOR PRIVIND STUDIUL CÂMPULUI TERMIC DIN PROCESELE DE AŞCHIERE

- *CONSIDERAȚII GENERALE*
- *SOLICITĂRI TERMICE ÎN PROCESELE DE AŞCHIERE*
- *SURSE DE CĂLDURĂ ÎN PROCESUL DE AŞCHIERE*
- *REPARTIZAREA CĂLDURII*
- *METODE DE STUDIU A CÂMPULUI TERMIC DIN PROCESELE DE AŞCHIERE*
- *CONCLUZII*
- *OBIECTIVELE STUDIULUI*

# 1. STADIUL CERCETĂRILOR PRIVIND STUDIUL CÂMPULUI TERMIC DIN PROCESSELE DE AŞCHIERE

## 1.1. SOLICITĂRI TERMICE ÎN PROCESSELE DE AŞCHIERE

### 1.1.1.FLUXUL TERMIC

Efectul câmpului termic asupra unui sistem se traduce prin modificarea poziţiei relative dintre particulele materiale ce alcătuiesc diferitele componente şi subansambluri ale acestuia. Vectorul ce caracterizează această deplasare, pentru două situaţii luate ca reper, este dependent de distribuţia câmpului termic. Această distribuţie la rândul ei depinde de mai mulţi factori:

- proprietăţile de material ale diferitelor subansambluri;
- geometria concretă a acestor subansambluri;
- condiţiile de ambient (temperatura mediului, curenţi de aer);
- fluxul termic produs de sursa de căldură;

Acest din urmă factor este foarte important şi totodată greu de estimat în condiţiile reale ale unui proces de aşchiere.

Pentru definirea fluxului termic, *Bayley* în [6]<sup>1</sup> porneşte de la postulatul fundamental al conducerii căldurii:

$$q = \frac{1}{A} \frac{dQ}{dt} = -k \frac{\partial T}{\partial n} \quad (1)$$

unde:  $Q$  este cantitatea totală de căldură;  $A$ : aria considerată,  $t$  este timpul;  $k$  este coeficientul de conducţie  $T$  este temperatura<sup>2</sup>;  $n$  este direcţia normalei la suprafaţă

Astfel se poate spune că fluxul termic este mărimea care caracterizează puterea termică a unei surse.

În cazul strunjirii, *Pethö* în [76]<sup>3</sup>, îl determină ca fiind lucrul mecanic consumat în timpul  $t$  corespunzător rotirii piesei cu unghiul la centru  $\alpha t$  şi parcurgerii distanţei  $l_1$  de către aşchie pe suprafaţa de degajare (fig.1).

<sup>1</sup> Bayley, F.,J., *Heat transfer*, p. 14

<sup>2</sup> Semnul minus este atribuit pentru a respecta convenţia ca fluxul pozitiv să fie orientat din interiorul către exteriorul suprafeţei considerate

<sup>3</sup> Pethö, L., *Cercetări cu privire la influenţa deformaţiilor elastice şi termice asupra preciziei de prelucrare şi posibilităţi de compensare a acesteia*, Teză de doctorat, I.P. Cluj-Napoca, 1984, p. 12

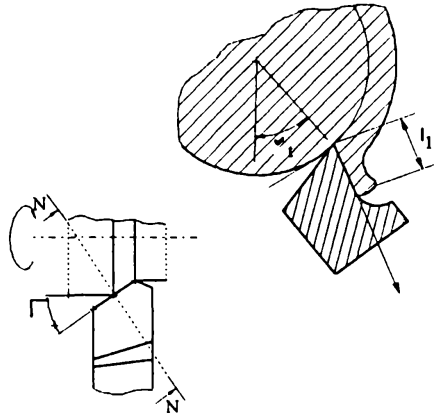


Fig. 1 Estimarea lucrului mecanic

Având în vedere ca ipoteze simplificatoare: neglijarea lucrului mecanic consumat pentru spiralarea și sfărămarea așchii și cel înmagazinat ca energie potențială (care produce tensionarea internă), fluxul termic se poate determina cu relația :

$$\Phi = \frac{F \cdot l_1}{427 \cdot t} \text{ [kcal / h]} \quad (2)$$

unde:  $\Phi$  este fluxul;  $l_1$  este lungimea de contact a așchii cu scula;  $F$  este forța calculată;  $t$  este timpul.

$$F = F_z \left( \sin \gamma + \frac{F_y}{F_z} \cos \gamma \right) \quad (3)$$

unde:  $F_z$  este componenta principală,  $F_y$  este componenta radială,  $\gamma$  este unghiul de degajare

### 1.1.2. TIPURI DE FENOMENE TERMICE

Pentru a putea avea un fundament în studiul unui sistem termic trebuie mai întâi stabilit comportamentul acestuia prin prisma dependenței proprietăților materialului de temperatură, timp, ș.a.m.d.

Conform celor prezentate în [96]<sup>4</sup> un fenomen termic se presupune a fi staționar dacă proprietățile de material și condițiile de ambient nu sunt dependente de timp. Problema este considerată a fi liniar staționară dacă sunt îndeplinite următoarele condiții:

- nu există transfer prin radiație la frontiera sistemului;
- nu există transformări de fază;
- proprietățile de material, coeficientul de convecție, sursele interne de căldură și temperaturile prestabilite din anumite puncte nu sunt dependente de temperatură.

<sup>4</sup> Roșca, A., Metode de analiză a câmpului termic în procesele de aşchiere, Referat nr.2, stagiu de pregătire doctorat, manuscris, U.P. Timișoara, 1995

Atunci când nu sunt îndeplinite toate condițiile de mai sus, problema este considerată a fi *neliniară staționară*.

Fenomenul este considerat a fi *tranzitoriu* dacă proprietățile de material sau condițiile de ambient sunt dependente de timp. Ca și la fenomenele staționare, o problemă este *liniar tranzitorie* dacă sunt îndeplinite simultan condițiile:

- nu există transfer prin radiație la frontiera solidului;
- nu există transformări de fază;
- proprietățile de material, coeficientul de convecție, sursele interne de căldură și temperaturile prestabilite din anumite puncte nu sunt dependente de temperatură.

În caz contrar problema este *neliniară tranzitorie*.

Evoluția fenomenelor termice din procesele de aşchiere continui sau discontinui poate fi reprezentată grafic ca în fig. 2 a respectiv b.

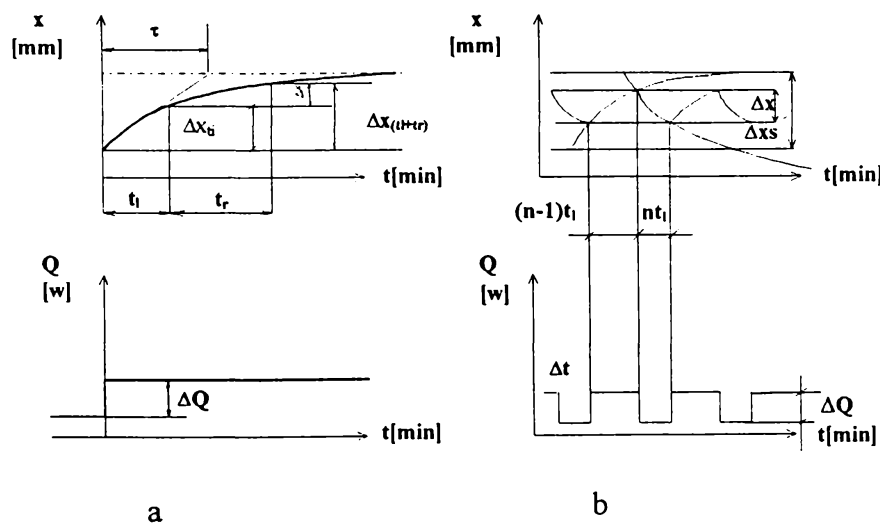


Fig. 2 Variația deplasărilor termice la aşchiera continuă și discontinuă

Pentru aşchiera continuă Sokolovsky apreciază deplasările termice cu relația:

$$\Delta x(t) = \Delta x \left( 1 - e^{-\frac{t}{\tau}} \right) = \Delta x_s V_1 \quad (4)$$

unde:  $V_1$  este funcția de deplasare;  $\tau$  este constanta de timp a mașinii.

În [76], Pethő presupunând că timpul de încălzire la mersul în gol al mașinii-unelte  $t_i$ , este aproximativ egal cu  $\tau$ , rezultă că legea de variație a deplasării termice între două regimuri staționare se reprezintă prin relația:



$$\Delta x = \Delta x_s = \left(1 - e^{-\frac{t}{\tau}}\right) e^{-\frac{t}{\tau}} = \Delta x_s V_2 \quad (5)$$

unde  $t_r$  este timpul scurs între două reglări consecutive al sculei.

La aşchiera discontinuă, regimul termic este cvasistaţionar ca în figura 2.b Dacă timpul  $t_r$  este mai mare sau egal cu durata solicitării termice  $nt_1$ , respectiv  $(1-n)t_1$ , rezultă:

$$\Delta x = \Delta x_s = \left(1 + e^{-\frac{t}{\tau}} - e^{-\frac{nt}{\tau}} - e^{-\frac{(1-n)t}{\tau}}\right) / \left(1 - e^{-\frac{t}{\tau}}\right) = \Delta x_s V_3 \quad (6)$$

Deci stabilitatea termică a sistemului este asigurată când în intervalul de timp  $t_r$ , modificarea maximă a poziţiei relative dintre sculă şi semifabricat datorită variaţiei temperaturii în sistem, este mai mică sau la limită egală cu modificarea admisibilă, adică este îndeplinită condiţia :

$$\Delta x(t_r) \geq \Delta x_a \quad (7)$$

### 1.1.3. RIGIDITATE TERMICĂ

Pentru a aprecia comportamentul sistemului la solicitările termice, prin analogie cu solicitările mecanice, se poate defini rigiditatea termică exprimată ca raport între fluxul termic şi variaţia poziţiei relative a unui punct de referinţă relativ la un interval de timp:

$$j_T = d(\Delta\Phi)/d(\Delta x) \quad [W/m] \quad (8)$$

unde puterea surselor interioare şi exterioare ( $P_i$  respectiv  $P_e$ ), se calculează ca în relaţia (9):

$$\begin{cases} \Delta\Phi_i = \Delta P_i = \alpha S \Delta T_s = \alpha S (T_s - T_e) \quad [W] \\ \Delta\Phi_e = \Delta P_e = \alpha S \Delta T_e = \alpha S (T_e - T_e) \quad [W] \end{cases} \quad (9)$$

unde:  $\alpha$  este coeficientul de transmitere a căldurii;  $S$  este suprafaţa prin care se transmite căldura.

## 1.2. SURSE DE CĂLDURĂ ÎN PROCESUL DE AŞCHIERE

Pentru modelarea corectă, din punct de vedere termic, a procesului de aşchiere, trebuie evidenţiate toate sursele termice care intervin în acest proces. Astfel *Drăghici Gh.* în [24]<sup>5</sup> şi *Dumitraş C.* în [25]<sup>6</sup> evidenţiază următoarele surse care îşi aduc aportul la crearea câmpului termic:

### 1.2.1.SURSE INTERNE DIN PROCESUL DE AŞCHIERE

Datorită proceselor de deformare elasto-plastică suferite de aşchia în formare, apar două fenomene, ambele generatoare de căldură:

⇒ *frecarea intercrystalină*:

Pentru această sursă termică, *Dumitraş* în [23] foloseşte ecuaţia lui Fourier:

$$\frac{\partial T}{\partial \tau} = a \nabla^2 T + \frac{a}{\lambda} q_v \quad (10)$$

care, particularizată pentru acest fenomen devine:

$$\frac{\partial T}{\partial \tau} = \frac{a}{\lambda} q_v \quad (11)$$

*unde: T este funcţia care dă distribuţia câmpului termic;  $\tau$  este timpul;  $\nabla$  este operatorul diferenţial nabra; a este o constantă dependentă de proprietăţile de material-relaţia (12);  $\lambda$  este conductivitatea; c este căldura specifică;  $\gamma$  este masa specifică a materialului piesei;  $q_v$  este intensitatea surselor termice interne repartizate în volumul de aşchie.*

$$a = \frac{\lambda}{c\gamma} \quad (12)$$

⇒ *frecarea externă*

dintre sculă şi semifabricat pe de o parte, respectiv sculă şi aşchie pe de altă parte; acelaşi autor particularizează din nou în relaţia lui Fourier neglijând de data aceasta sursele interne de căldură rezultând relaţia:

$$\frac{\partial T}{\partial \tau} = a \nabla^2 T \quad (13)$$

### 1.2.2.MAŞINA UNEALTĂ

Este normal ca în timpul funcţionării maşinii unelte, subansamblurile electrice, mecanice, hidraulice să transforme o parte a energiei consumate în căldură. Câmpul termic astfel creat are influenţă în primul rând asupra

<sup>5</sup>Drăghici Gh., Op.cit., p.168

<sup>6</sup>Dumitraş C., ş.a., *Aşchiera metalelor şi fiabilitatea sculelor aşchietoare*, p.89

poziţiei relative dintre suportii portscule şi axele geometrice după care se derulează mişcările de avans. Aceste deplasări relative neuniforme produse de câmpul termic al maşinii unelte, au influenţă negativă asupra preciziei cotelor ce se doresc a fi executate, lucru evidenţiat de unii autori prin măsurători experimentale [103], [75], [76].

### 1.2.3.SURSE EXTERNE

În anumite condiţii radiaţia solară precum şi influenţa altor echipamente care prin modul lor de funcţionare produc căldură, pot să aibă influenţe măsurabile în sensul celor de la punctul de mai sus. Astfel pentru maşini unelte instalate în spaţii cu luminatoare s-a constatat că radiaţia solară poate ridica temperatura subsansamblurilor expuse cu până la 10-12°C peste temperatura normală.

### 1.3. REPARTIZAREA CĂLDURII

Factorii care influenţează câmpul termic în procesele de aşchiere pot fi grupaţi în<sup>7</sup>: factori dependenţi de regimul de aşchiere, factori dependenţi de geometria şi dimensiunile sculei şi factori dependenţi de materialul piesei.

Analizând studiile şi lucrările publicate pe tema câmpului termic din procesul de aşchiere, se pot prezenta următoarele informaţii despre distribuţia termică în componentele sistemului tehnologic:

• *Militaru C.* în [64], apreciază pentru aşchiera OL45 cu sculă armată cu P05 folosind un avans de 0.3 [mm/rot] o repartizare a căldurii ca în figura 3.

Este de remarcant faptul că cea mai importantă cantitate de căldură este evacuată în aşchie (cca. 65 %), piesa preluând în jur de 25%, scula cca. 9%,

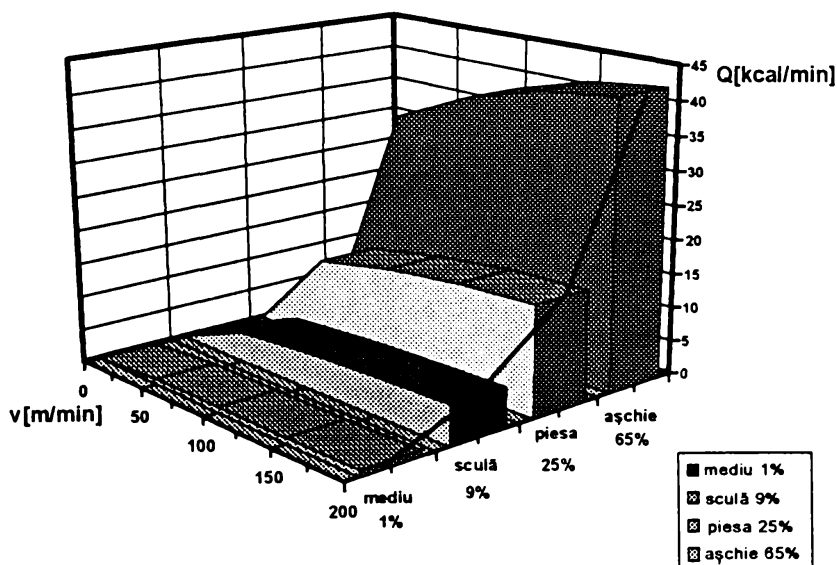


Fig. 3 Repartizarea căldurii în componentele sistemului MDSP

<sup>7</sup> Roşca, A., *Orientări şi direcţii în studiul câmpului termic din procesele de aşchiere*, Referat nr. 1, stagiul de pregătire doctorat, manuscris, U.T. Timișoara 1994

restul disipându-se în mediu. Pentru determinarea cantităţii de căldură degajată în timpul aşchierii, *Drăghici* în [24] şi *Dumitraş* în [25] apreciază cu relaţiile (14) şi (15) lucrul mecanic consumat pentru frecarea intercrystalină şi frecarea externă.

$$L_{m\Phi\gamma\lambda} = \frac{2}{\sqrt{3}} \sigma_c \sqrt{I_{v2}} \quad (14)$$

$$L_{mf\gamma\lambda} = \frac{c}{\sqrt{3I_{v2}}} |v_{\Phi\gamma\lambda} v_{\gamma\lambda}| \quad (15)$$

unde:  $L_{m\Phi\gamma\lambda}$  : lucrul mecanic consumat pentru deformarea plastică şi învingerea frecărilor interne;  $L_{mf\gamma\lambda}$  : lucrul mecanic consumat pentru frecările de pe suprafeţele sculei;  $\sigma_c$ : efortul unitar la curgere al materialului;  $I_{v2}$  : invariantul de ordinul doi al tensorului vitezei de deformare plastică al stratului aşchiat;  $v_{\Phi\gamma\lambda}$  : viteza de deformare din planul de forfecare al aşchierii;  $v_{\lambda\gamma}$ : viteza de curgere a aşchierii pe suprafaţa de degajare a sculei.

•Tot ei în [24], [25], precum şi *Militaru* în [64], prezintă relaţii analitice pentru determinarea cantităţii de căldură repartizată în elementele din zona de aşchiere, cu explicaţiile din figura 4:

$$Q = Q_{as} + Q_{sc} + Q_i + Q_{pf} + Q_{aer} \quad (16)$$

$$Q = \left[ c_{sc} m_{sc} + c_{aer} m_{aer} + c_{pf} (m_{as} + m_i + m_{pf}) \right] \cdot \left[ C_\theta e^{\left( \frac{x_v v^m + y_s s^n + z_t t^q}{m} \right)} - \theta_0 \right] \quad (17)$$

unde indicii au următoarele semnificaţii: *as* - partea de aşchie ce se detaşează; *sc*- sculă; *t*-stratul de aşchiere; *pf*- piesa finită; *aer*- mediul înconjurător; *ci* - căldura specifică pentru fiecare din elementele "i" de mai sus; *mi*: masa aferentă elementelor "i" de mai sus;  $\theta_0$  : temperatura mediului  $\approx 20^\circ\text{C}$ ;  $C_\theta$  : constanta de integrare ce rezultă în urma rezolvării şi aplicării condiţiilor de contur la ecuaţia (18);  $x_v, y_s, z_t$ , coeficienţii unghiulari ai dependenţelor  $x_v = \varphi_1(v)$ ,  $y_s = \varphi_2(s)$ ,  $z_t = \varphi_3(t)$  reprezentate în coordonate dublu logaritmice.

$$\frac{d\theta}{dt} = x_v v^{m-1} dv + y_s s^{n-1} ds + z_t t^{q-1} dt \quad (18)$$

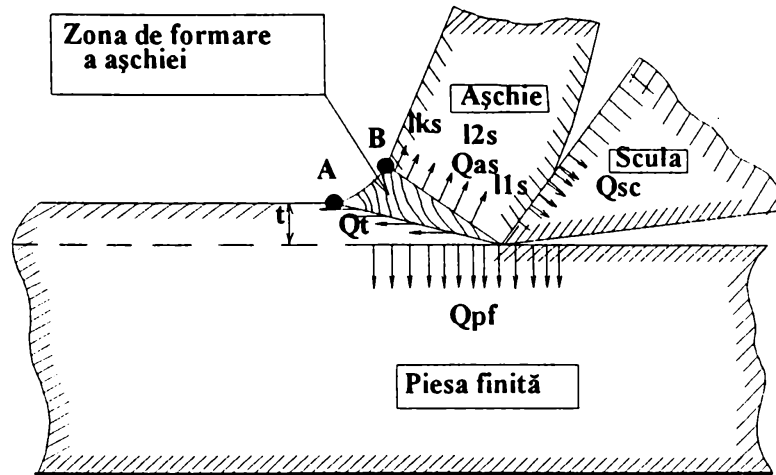


Fig. 4 Repartizarea căldurii la formarea aşchiei

Pentru o mai clară evidenţă a modului cum sunt afectate componentele sistemului MDPS de câmpul termic, se prezintă rezultatele studiilor făcute defalcat pe componentele sistemului de prelucrare.

### 1.3.1.SCULA

•Popescu I. în [82]<sup>8</sup> prezintă distribuţia temperaturii în zona de aşchiere la aşchiere ortogonală a unui semifabricat din oţel cu 0.14%C cu o viteză ce circa 80 [m/min]. Conform distribuţiei din figura 5, se observă că temperatura maximă de cca. 700°C apare în vârful sculei, iar în aşchie se înregistrează o temperatură mai ridicată de cca. 620°C la ieşirea de pe faţa de degajare şi nu în zona rădăcinii acesteia, lucru datorat frecării externe.

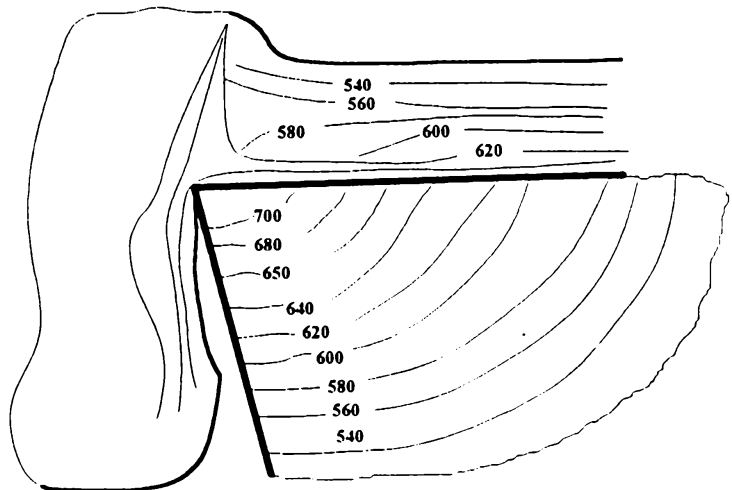


Fig. 5 Distribuţie termică în sculă şi aşchie

•Kudinov A.V. în [53]<sup>9</sup> consideră că temperatura maximă apare de fapt în aşchie în perioada de curgere, într-un strat median conform celor din figura 6 şi nu la contactul cu faţa de degajare:

<sup>8</sup> Popescu, I., *Teoria aşchierii*, p.112

<sup>9</sup> Kudinov, *Dinamica maşinilor unelte*

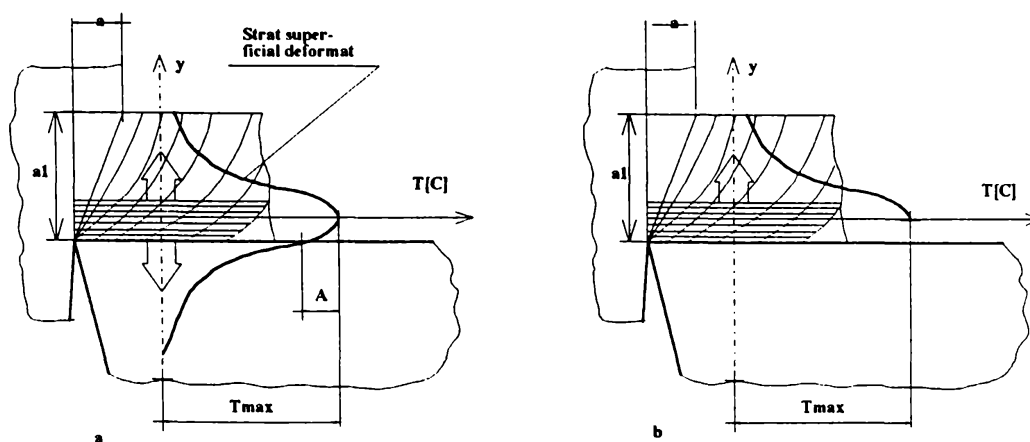


Fig. 6 Temperatura în stratul median de curgere al aşchiei

Acest lucru este explicat de autor prin fenomenul de curgere a materialului care evoluează în planul median chiar și după formarea aşchiei, înregistrându-se astfel diferențe de temperaturi mai mari de  $400^{\circ}\text{C}$  între suprafață și planul median, la partea de ieșire a aşchiei (fig.6 a). Acest lucru favorizează fenomenul de depunere pe tăiș, autorul prezentând o soluție de contracarare prin încălzirea corpului cuțitului la cca  $500-600^{\circ}\text{C}$ . Procesul de formare a aşchiei în această situație arată din punct de vedere termic ca în fig.6 b.

Deși cantitatea de căldură transmisă sculei este relativ mică, (sub 10%) din cantitatea totală, deformațiile termice suferite de aceasta sunt cele mai importante datorită volumului relativ mic pe care scula îl are în raport cu alte componente ale sistemului MDPS.

### 1.3.2.SEMIFABRICATUL

O importantă cantitate de căldură este transmisă semifabricatului (până la 25 %), în funcție de situația concretă ducând la ridicarea temperaturii acestuia cu până la  $50-60^{\circ}\text{C}$ .

Pentru o strunjire exterioară Drăghici Gh. în [24], prezintă o distribuție a câmpului termic (fig. 7) arătând totodată că sursa de căldură se deplasează după o traiectorie elicoidală pe suprafața piesei cu o viteză egală cu rezultanta vitezei de aşchiere și a avansului longitudinal.

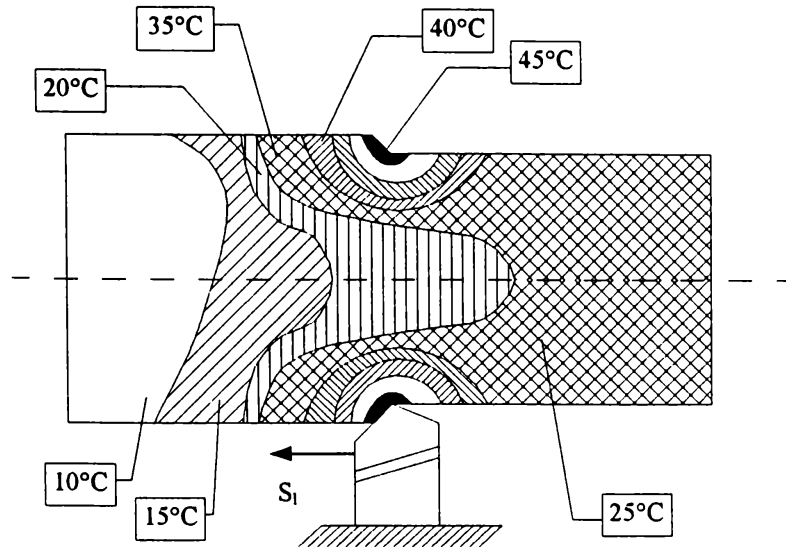


Fig. 7 Distribuție a câmpului termic în piesă

În [76]<sup>10</sup> este prezentat un studiu al lui Sokolovski A.P. pentru variația temperaturii pe suprafața semifabricatului la strunjire exterioară, care arată că la începutul prelucrării deformația termică este mai mare la suprafața semifabricatului decât în interiorul acestuia, iar spre finalul prelucrării deformația termică devine mai mare în interiorul piesei decât la suprafață. Totodată, studiul prezintă faptul că temperatura este mai mare în apropierea păpușii mobile decât în cea a păpușii fixe, în cazul utilizării vârfulor fixe și sensibil egale la utilizarea vârfulor mobile.

### 1.3.3. MAȘINA UNEALTĂ

Mașina unealtă este mai puțin afectată de câmpul termic rezultat din sursele de căldură evidențiate mai sus.

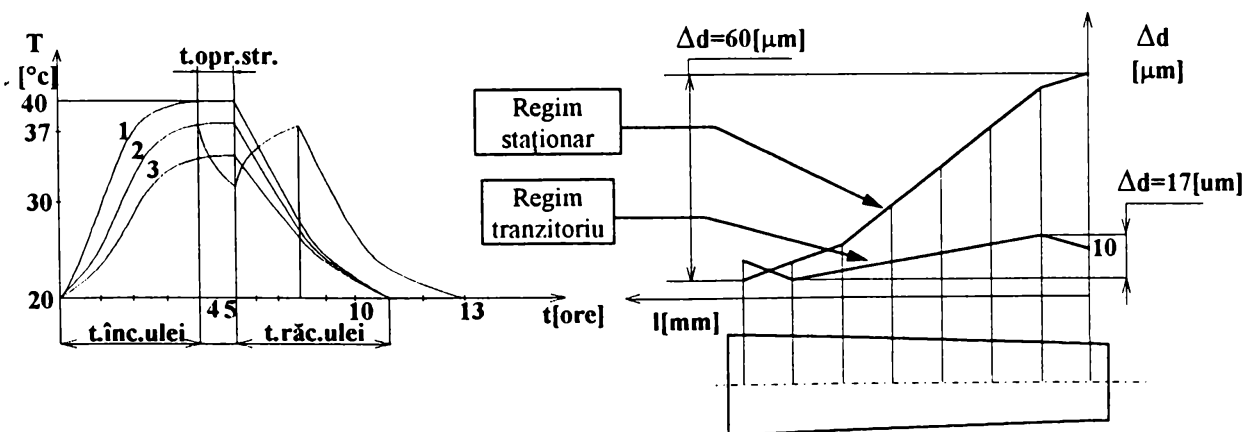


Fig. 8 Variația temperaturii păpușii fixe în timp și variația aceleiași cote prelucrate în dreptul ei

unde:  $t.înc.ulei$ ,  $t.răc.ulei$  este timpul de încălzire respectiv răcire a uleiului;  $t.opr.str.$  este timpul de oprire staționare.

<sup>10</sup>Pethö, L. Op.cit., p. 26

Depalsările suferite de diferitele subansamburi ale acesteia sunt relativ mici și pot fi puse în evidență după câteva zeci de minute de funcționare.

Astfel *Petriceanu Gh.* în [77]<sup>11</sup> a constatat că atât la mersul în gol cât și la mersul în sarcină după 3-4 ore de funcționare temperatura păpușii fixe crește cu 20°C (fig.8), producând deplasări ale axei arborelui principal, după care fenomenul se stabilizează.

## 1.4. METODE DE STUDIU A CÂMPULUI TERMIC DIN PROCESELE DE AȘCHIERE

Așa cum s-a prezentat și în [97], metodele de studiu ale câmpului termic pot fi grupate în: metode analitice, metode numerice și metode experimentale, fiecare dintre ele cu particularizări pentru fiecare componentă a sistemului MDPS, în continuare fiind prezentată o sinteză a lor.

### 1.4.1.METODELE ANALITICE

- *Wulf A.M.* utilizează o relație empirică în funcție de geometria sculei și de parametrii regimului de aşchiere pentru a determina temperatura din zona de aşchiere.

- *Sokolovsky* folosește o relație empirică pentru calculul alungirii cuțitului de strung.

- *Ostafiev* integrează o ecuație de tip laplacian în care apare fluxul termic pentru a determina tensiunile și deformațiile în zona de la vârful sculei.

- *Bogdanov* folosește o relație exponențială pentru a determina temperatura în diverse zone ale cuțitului.

- *Lăzărescu*, în [57], integrează ecuația diferențială  $\Delta T=0$  pentru a obține temperatura în semifabricat la diverse distanțe față de vârful sculei; tot el utilizează și relații empirice pentru a calcula temperatura medie din semifabricat și a determina deformațiile pe diametru și lungime ale semifabricatelor cilindrice.

- *Drăghici*, în [24], folosește relații clasice din rezistența materialelor pentru a calcula reacțiunile și tensiunile interne provocate de solicitarea termică în semifabricate cilindrice strunjite între vârfuri; tot el

---

<sup>11</sup>Petriceanu, Gh. *Contribuții la stabilirea variației temperaturii în elementele sistemului tehnologic al mașinii de rectificat roți dințate cilindrice și deformațiile termice ale acestora*



foloseşte pentru deformaţiile termice ale maşinii unelte o relaţie simplificată în care intervine cota de la planul ghidajelor până la arborele maşinii;

•*Pethö* a prezentat în [76], un studiu al lui *Klager S.*, care pentru determinarea câmpului termic în batiuri, recomandă desfăşurarea în plan a acestuia sub forma unui disc şi integrarea ecuaţiei lui Fourier pentru acest domeniu simplificat.

•*Ju* şi *Liu*, în [45], au efectuat o analiză a câmpului termic şi stării de tensiuni folosind transformata Fourier. Ei au corelat rezultatele obţinute pentru a face o estimare a condiţiilor în care se produc microfisuri în stratul superficial al corpurilor placate, datorat deplasării rapide a sarcinilor pe suprafaţa piesei.

•*Li* şi *Chen*, în [59], au făcut o modelare analitică a unui proces de rectificare pornind de la formularea energetică termomecanică. Ei au folosit o schemă incrementală pentru studiul procesului tranzitoriu.

•*Korneeva* şi *Kamalov* au prezentat, în [50], relaţii de dependenţă a temperaturii de aşchiere pentru strunjirea cu viteze balistice a oţelurilor inoxidabile. Aceste relaţii sunt de forma:

$$\begin{aligned}\theta &= B_1 + B_2\tau + B_3\tau^2 + B_4\tau^3 + B_5V + B_6V^2 + B_7V^3 \\ \tau &= C_1 + C_2\theta + C_3\theta^2 + C_4\theta^3 + C_5V + C_6V^2 + C_7V^3\end{aligned}\quad (19)$$

unde:  $\theta$  este temperatura;  $\tau$  este timpul;  $B$  şi  $C$  sunt coeficienţi prezentaţi tabelar şi care depind de materialul sculei şi de materialul piesei.

#### 1.4.2.METODELE NUMERICE

Cum deformaţiile elastice şi transferul de căldură prin conducţie sunt fenomene guvernate de ecuaţii diferenţiale, în inginerie, pentru rezolvarea acestora pe domenii complexe, se utilizează metode numerice aproximative. Dintre cele mai folosite sunt metoda diferenţelor finite şi metoda elementului finit. În continuare se amintesc câteva dintre cele mai semnificative studii, care au fost sintetizate în [92].

•*Akiyama T.* în [1], a determinat tensiunile şi deformaţiile produse într-un cuţit de strung pentru o stare plană de tensiuni, utilizând metoda elementului finit.

•*Aksenov* a prezentat în [2], un studiu făcut pentru determinarea câmpului termic din procesele de rectificare folosind metoda diferenţelor finite. El a obţinut valoarea vitezei de încălzire la o piesă din oţel 40H în jurul a  $4,3 \times 10^3$  [°C/sec], pentru o rectificare de finisare şi a concluzionat că influenţa cea mai mare asupra acestui parametru o are viteza de rotire a discului.

• *Pethö* în [76], a analizat starea de tensiuni și deformații dintr-un cuțit de strung, utilizând metoda diferențelor finite și metoda elementului finit. El a făcut acest lucru pornind de la cunoașterea în prealabil a distribuției câmpului termic și a suprapus efectele acestuia peste solicitările mecanice din timpul aşchierii.

• *Noscenko* a prezentat în [67] o schemă de calcul cu elemente finite pentru schimb de căldură neliniară în spațiul bidimensional. El a folosit o schemă de integrare în timp pentru aproximarea fenomenului tranzitoriu.

• *Tönshoff* și *Denkena* au prezentat, în [123], rezultatele unei simulări numerice bidimensionale pentru o aşchiere discontinuă. Ei au folosit următoarele ipoteze pentru condițiile limită: distribuția temperaturii pe fața de degajare este uniformă, comportamentul fluxului termic respectă o curbă cu aspect dreptunghiular, amplitudinea fluxului termic este  $100 \text{ W/mm}^2$ , iar perioadele de aşchiere și repaos sunt de 7ms, respectiv 18 ms ceea ce corespunde la o frezare cu cca. 100 m/min. După această simulare au ajuns la concluzia că pentru frezarea oțelurilor turnate, sculele care suportă cel mai bine șocurile termice sunt cele cu TiC.

• În [60], *Lin* și *Pan* prezintă un model termo-elasto-plastic pentru studiul unui proces de aşchiere ortogonală. Pentru acest lucru ei au construit un element finit special și au folosit o autoadaptare a discretizării zonei considerate, pentru a fi în concordanță cu realitatea geometrică apărută ca urmare a detașării aşchiei. Pentru modelarea fenomenelor termice ei au folosit aceleași considerații ca și alți autori. Astfel, pentru a exprima căldura generată de cele două surse locale au folosit o funcție  $\theta = \theta(A_w, A_c, v, \tau)$ , unde:  $A_w$ ,  $A_c$ , sunt aria de frecare sculă-așchie, respectiv aria de frecare sculă-piesă,  $v$  reprezintă viteza relativă de aşchiere iar  $\tau$ , reprezintă efortul unitar de rupere la forfecare a materialului piesei.

• *Strenkowsky* a prezentat în [115] o estimare a procesului de formare a aşchiei pe baza formulării deplasărilor mari. Modelul matematic este asemănător cu cel descris de *Lin* și *Pan* și se bazează pe utilizarea ca și condiții limită a unor date experimentale.

• *Ostafiev*, a prezentat în [69], o scurtă caracterizare a mai multor metode analitice și numerice folosite în studiul câmpului termic. Este prezentată de asemenea o distribuție a câmpului termic în vârful sculei și în zona apropiată din piesă, care este preluată în figura 9.

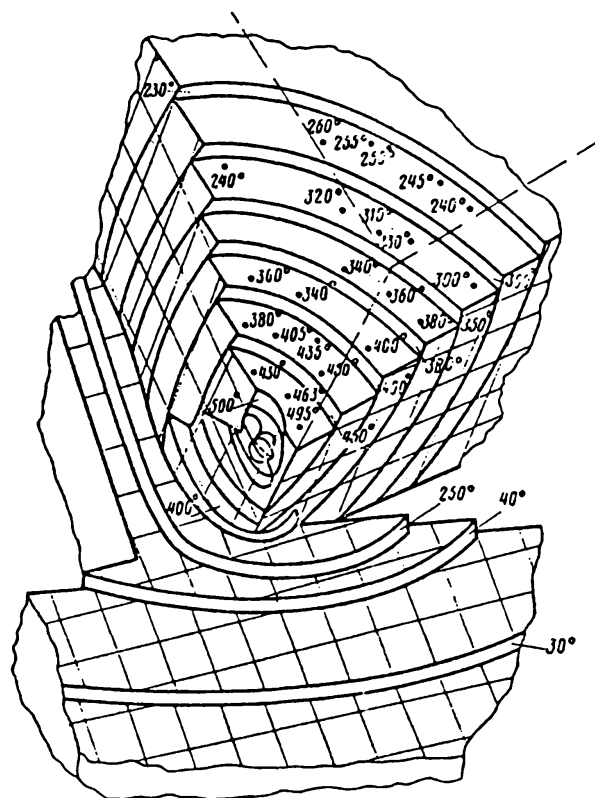


Fig. 9 Distribuție de temperaturi în zona de aşchiere

### 1.4.3.METODE EXPERIMENTALE

#### 1.4.3.1.Metode clasice

Dintre aceste metode cea mai folosită este termocuplul, în multe din procesele stabile oferind informații acceptabile pentru determinarea distribuției câmpului termic.

•*Gurumoorthy*, a folosit în [34], termocuplul pentru măsurarea temperaturii în regim stabilizat și a obținut niște curbe pe baze statistice pentru estimarea duratei de viață a sculelor armate cu plăcuțe ceramice, la diferite temperaturi de exploatare.

•*Zorictuev*, în [141], a prezentat o metodă de estimare a temperaturii locale din zona de aşchiere pe baza unor considerente electrofizice colaterale procesului de aşchiere. El a construit o instalație care măsoară variația conductibilității electrice a contactului sculă-piesă și deci indirect temperatura locală din zona de aşchiere. Ca fundament teoretic el a folosit metoda potențialelor nodale din teoria circuitelor electrice.

•*Kitagawa* în [49], a modificat un cuțit de strung armat cu plăcuță din carbură, pentru a folosi termocuplul natural creat între piesă și un fir de platină cu grosimea de 10  $\mu\text{m}$ , montat într-un tub izolant de cuarț cu dia-

metrul de  $50 \mu\text{m}$  (figura 10). El a corelat informațiile obținute pentru a estima mecanismul uzurii sculei.

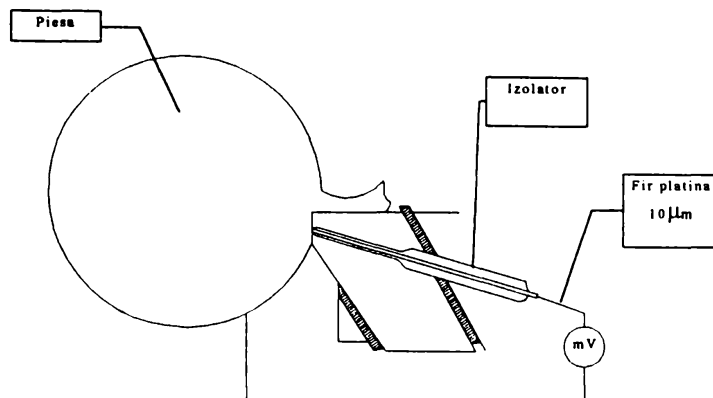


Fig. 10 Termocuplu montat în zona de contact sculă-piesă

•În [70], Ozhan folosește o schemă similară dar cu un termocuplu Ni/NiCr cu diametru de  $0,5 \text{ mm}$ , pentru a măsura temperatura degajată la operația de rectificare plană.

#### 1.4.3.2. Metoda fotografiei termice

Metoda se bazează pe faptul că orice corp cu o temperatură peste zero absolut, emite o cantitate de energie sub formă de radiație electromagnetică, ce este caracterizată de spectrul de emisie și de lungimea de undă corespunzătoare vârfului de emisie din acest spectru. Radiația electromagnetică a cărei lungime de undă este cuprinsă între  $0,7 \mu\text{m}$  și circa  $100 \mu\text{m}$ , denumită radiație infraroșie, constituie mecanismul transferului de căldură prin radiație, datorită efectului termic pe care-l produce asupra corpurilor care pot absorbi această radiație.

Din punct de vedere științific, pentru determinarea temperaturii unui corp fără contact fizic cu acesta, prezintă interes radiațiile din spectrul invizibil corespunzătoare radiației infraroșii.

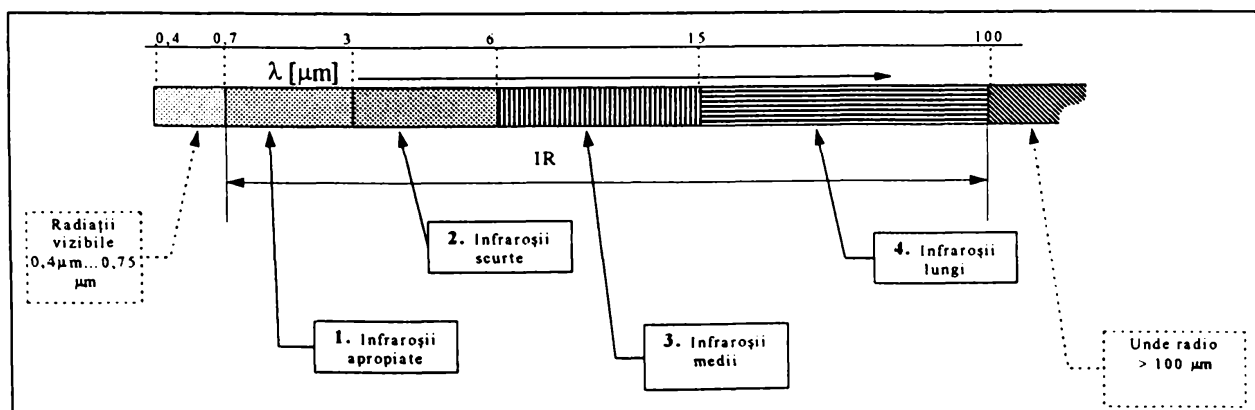


Fig. 11 Porțiunea din spectrul electromagnetic corespunzătoare radiației infraroșii

Literatura de specialitate, împarte zona corespunzătoare acestei radiații, în 4 sectoare (figura 11), delimitarea fiind legată de metodele tehnice folosite pentru detectarea radiației respective.

Evoluția acestor fenomene este analizată pe baza legilor lui Planck, Wien și Boltzmann, aspecte despre acestea fiind comentate în capitolul 3.1.3.

Ținând cont de relația (20), care derivă din teoria cuantică a lui Planck, conform căreia energia radiată este emisă în cuante, relațiile de mai sus pot fi modificate, astfel încât să fie exprimate în funcție de numărul de fotoni și nu de valoarea energiei radiate.

$$Q = \frac{hc}{\lambda} [J] \quad (20)$$

unde:  $Q$  este energia unui foton;  $h$  este constanta lui Planck;  $c$  este viteza luminii [m/sec];  $\lambda$  este lungimea de undă [m].

Acest aspect este util la calibrarea scannerelor infraroșu, care sunt de fapt detectoare de fotoni.

Din reprezentarea grafică a legii lui Plank din figura 20 se poate observa că vârful spectrului radiației emise se deplasează către domeniul radiației vizibile, pe măsură ce crește temperatura. Astfel un corp negru (cu emisivitatea 1) a cărui temperatură este peste 500°C are deja o largă parte a spectrului radiației în zona vizibilă și în condiții de obscuritate devine observabil cu ochiul liber.

Pentru corpurile reale, capacitatea de a emite radiații pe anumite lungimi de undă, este caracterizată de emisivitatea corpului, care la rândul ei este dependentă de materialul respectiv, de geometria corpului și de temperatură. De aceea cunoașterea cât mai exactă a valorii emisivității corpului este esențială la calibrarea imaginilor obținute prin fotografie termică în infraroșu și termografie electronică.

#### *Utilizarea fotografiei pentru înregistrarea radiațiilor infraroșii*

În momentul de față, paleta materialelor fotosensibile disponibile, care pot fi folosite pentru înregistrarea manifestărilor electromagnetice din domeniul ultraviolet, vizibil și infraroșu este distribuită ca în figura 12. Corpurile cu temperatură peste 500°C emit radiație actinică<sup>12</sup> și pot fi fotografiate pe peliculă pancromatică fără surse de lumină.

Cele cu temperatură cuprinsă între 250°C și 500°C au vârful radiației spectrale în zona radiațiilor infraroșii apropiate și o parte din spec-

---

<sup>12</sup> Corp a cărui radiație electromagnetică este capabilă să inițieze o reacție chimică, prin analogie cu actinidele, elemente chimice care au nativ această proprietate

trul respectiv este în aşanumita zonă a radiaţiei actinice, deci care pot impresiona o peliculă fotografică în anumite condiţii.

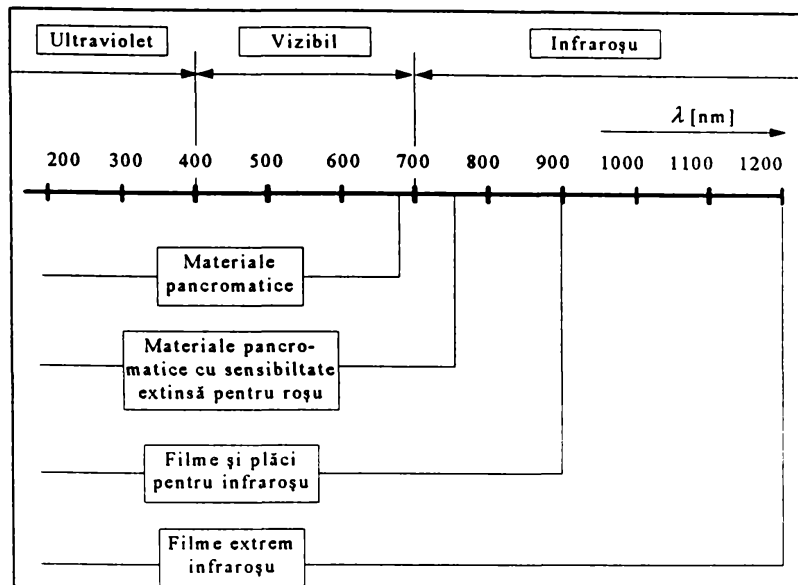


Fig. 12 Recomandări pentru utilizarea materialelor fotosensibile

În fotografia tehnică, folosită pentru estimarea temperaturii unor corpuri, este necesară utilizarea unor filme speciale cu sensibilitate mărită în zona infraroşu şi utilizarea unor filtre optice care să nu permită trecerea radiaţiei vizibile provenită de la surse sau reflectată de alte corpuri, ceea ce poate produce efectul de ceaţă pe peliculă. Corpurile cu temperatura în acest interval, vor produce pe pelicula specială tonuri graduale, care nu sunt vizibile cu ochiul liber, sau care nu pot fi înregistrate pe peliculă pancromatică. Trebuie precizat faptul că limitele fizice ale obiectelor sunt difuze, datorită curenţilor de convecţie creaţi la limitele acestor corpuri, straturile de aer absorbind o parte a radiaţiei emanate de corpul încălzit şi devenind acum la rândul lor sursă radiantă.

Tehnica fotografică în această situaţie nu diferă prea mult faţă de cea normală, în [157] fiind făcute câteva recomandări pentru utilizarea filmului Kodak High Speed Infrared Film 2481. Dacă în aplicaţiile din astronomie timpul de expunere trebuie să fie uneori de câteva ore, la aplicaţiile tehnice valoarea recomandată este în jurul a 1,4 secunde, aspect care impune folosirea unei tripode şi practic nu permite aplicarea procedurii la procese dinamice, cum ar fi operaţiile de aşchiere, sau urmărirea unor corpuri în mişcare. Diafragma preferabilă este  $f/1,9$  la utilizarea filtrului Kodak Wratten 87, însă prin tatonare pot fi găsite şi alte combinaţii între timpul de expunere şi valoarea diafragmei, care să producă imagini utilizabile.



La temperaturi sub  $250^{\circ}\text{C}$ , radiația emanată face teoretic imposibilă reacția chimică de impresionare a peliculei, datorită faptului că energia fotonilor este insuficientă, ea descrescând odată cu creșterea lungimii de undă a radiației așa cum rezultă din relația (20).

Una din problemele care apar la acest tip de fotografie, este cea a sensibilității peliculei și derivă din faptul că fenomenele respective evoluează la limita tehnologiei fotochimice disponibile în momentul de față. Acest lucru se traduce prin imposibilitatea de a obține imagini utilizabile pentru zone ale unor obiecte cu diferență de temperatură mai mare de  $120^{\circ}\text{C}$  (dar în intervalul radiației actinice). Dacă este posibil, este recomandată fotografierea zonelor mai reci într-o expunere mai lungă și a celor mai calde separat într-o expunere mai scurtă. Pentru aprecierea rezoluției termice a acestui procedeu, trebuie știut faptul că la o imagine cu diferențe de temperatură de circa  $80^{\circ}\text{C}$ , se pot distinge cu ochiul liber pe fotografie nuanțe care corespund unor diferențe de temperatură de circa  $15^{\circ}\text{C}$ . Pentru a releva diferențe mai mici de temperatură se recomandă o prelucrare computerizată a fotografiei, prin scanarea ei la o rezoluție minimă de 600 dpi și cel puțin 256 de culori. Prin schimbarea convenabilă a paletii de culori a fișierului bitmap rezultat, se pot pune în evidență mai multe nuanțe care, pe fotografia inițială, nu erau vizibile cu ochiul liber.

O altă problemă care apare este cea a calibrării imaginilor obținute. Pentru acest lucru singura alternativă este fotografierea în condiții similare a unei surse termice cu temperatura cunoscută și cu emisivitate cât mai apropiată de cea a corpului real. Neglijarea emisivității poate produce erori de calibrare de peste 15%, lucru care este de așteptat din expresia legii lui Boltzmann pentru corpuri gri.

Avantajul cel mai important al aplicării fotografiei termice îl constituie prețul relativ scăzut al echipamentelor necesare și buna rezoluție geometrică oferită de o peliculă fotografică. Dar, după cum se observă, utilizarea ei are o serie de limitări datorită fundamentelor fizice care o guvernează și datorită performanțelor echipamentelor implicate.

#### 1.4.3.3. Metoda termografiei electronice

Această metodă se bazează pe aceleași principii fizice ca și fotografia termică, cu deosebirea că elementul sensibil la radiația infraroșie este constituit dintr-o combinație de telur, cadmiu și mercur ce asigură acoperirea întregului spectru de radiație infraroșie. Conform celor prezentate în [97], precum și [18], [144], [148], prin acest procedeu se pot înregistra procese termice deosebit de dinamice cu o mare acuratețe. Legat de acest ultim aspect, s-a constatat că la mai multe echipamente consacrate

pentru termografie electronică, cum ar fi Agema, Cyclop, Termovision, este posibilă afişarea imaginii numai pentru corpuri a căror temperatură se află într-un interval prestabilit înainte de operarea echipamentului (lucru impus de sensibilitatea detectorilor folosiţi în prezent).

#### 1.4.3.3.1. Determinarea directă a temperaturii zonei de aşchiere la aşchiera continuă

În multe situaţii, la prelucrările prin aşchiere, sunt din ce în ce mai folosite scule cu diamant sau cu plăcuţe acoperite cu un strat micrometric de diamant industrial. Tradiţional, aceste scule sunt folosite la aşchiera materialelor neferoase, cum ar fi aliaje de aluminiu, materiale compozite, materiale plastice etc. Utilizarea sculelor diamantate la aşchiera aliajelor feroase este restricţionată de fenomenul de difuzie a carbonului din sculă. Acest fenomen este amplificat de:

- ◆ temperaturile mari din zona de aşchiere;
- ◆ presiunea de contact foarte mare ce apare la vârful sculei;
- ◆ faptul că diamantul este o formă a carbonului, deci acesta apare în zona de aşchiere într-o cantitate mult mai mare decât la alte materiale folosite pentru execuţia sculelor;
- ◆ aşchiera aliajelor feroase cu conţinut scăzut de carbon, care au astfel o afinitate sporită faţă de carbonul din sculă.

Fenomenul acesta este pus în evidenţă detaliat şi de alţi autori cum ar fi *Militaru în [64]* şi *Dumitraş C. în [25]* care consideră că difuzia carbonului este responsabilă pentru uzura sculei pe faţa de degajare prin fenomenul de cavitaţie.

Pentru a beneficia totuşi de avantajele oferite de prelucrarea cu aceste scule, *Hummel şi Lahres în [39]<sup>13</sup>* au pus la punct o metodă foarte precisă de control activ al temperaturii din zona de aşchiere, la sculele diamantate, în scopul de a păstra temperatura din zona de aşchiere sub temperatura critică de difuzie a carbonului.

Temperatura este determinată fără contact direct şi în timp real, utilizând așa numita metodă a *ferestrei infraroșu* care face și obiectul patentului german P 4 233 035, 1992. În experimentul realizat de cei doi, s-a folosit o sculă la care în plăcuța amovibilă s-a montat o fereastră de diamant cu grosimea de 500 μm (figura 13), lustruită pe ambele fețe la o rugozitate de  $R_a=0.02 \mu\text{m}$ . Plăcuța astfel obținută s-a acoperit cu un strat de diamant de 5 μm folosind un cuptor de depunere. După lustruirea stratului depus peste ansamblu, grosimea acestuia a rămas de cca. 2 μm, cu o orien-

<sup>13</sup>Hummel, M., *Infrared temperature measurement on diamond-coated tools during machining*



tare în planul (111). Pentru acurateţea măsurătorii s-a determinat transmisivitatea optică şi coeficientul de reflexie al ferestrei de diamant astfel obţinute. Traseul optic se continuă printr-un canal conic în corpul sculei asigurând astfel transmiterea radiaţiei infraroşii către o cameră termografică.

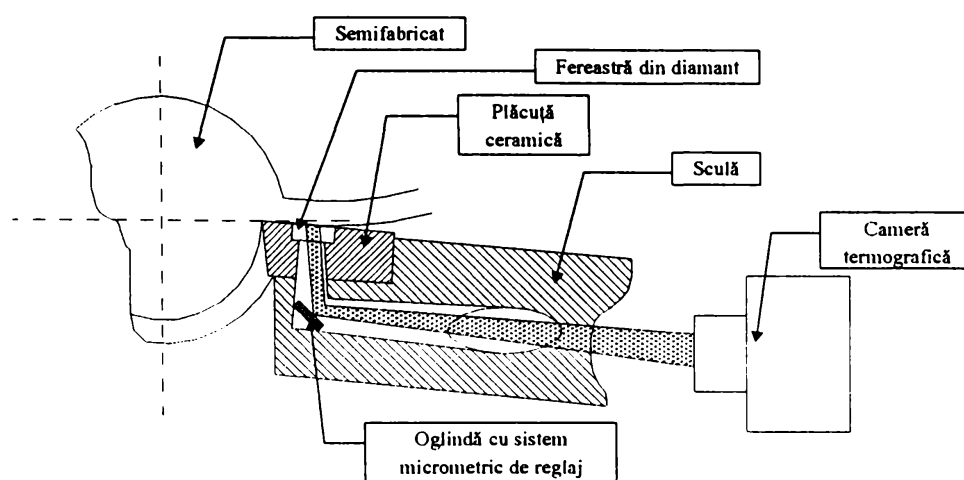


Fig. 13 Soluție pentru canal conducător al radiației infraroșii în corpul sculei

Forma conică a canalului este determinată de necesitatea ca fasciculul de fotoni să nu aibă interferență cu corpul sculei. Pentru devierea acestui fascicul către partea laterală a sculei, s-a folosit o oglindă orientabilă pe un sistem micrometric. Alinierea sistemului optic cu axa camerei termografice s-a făcut folosind un fascicul laser care emite pe direcția sursei termice. Astfel, pentru o sculă acoperită cu diamant, prin acest procedeu se poate determina temperatura de pe fața de degajare în zona de formare și desprindere a aşchiei.

Legat de acest procedeu, *Gould*, în [32], a prezentat posibilitatea utilizării unui sistem compus dintr-un mediu de transmisie (fibră optică, pentru ghidarea radiației infraroșii) și un amplificator optic care acoperă spectrul 2-40 $\mu$ m, deci face măsurabilă în condiții de siguranță și vârful de emisie corespunzătoare unor surse cu temperaturi mai mici de 30°C.

În [130], *Weck* a prezentat rezultate ale unor măsurători obținute cu termografie electronică pentru o mașină de frezat cu portal, punând în evidență faptul că la mașini de mari dimensiuni, echilibrul termic se atinge după un timp îndelungat (în acest caz 240 min), chiar în condițiile unei utilizări în regim constant al mașinii.

#### 1.4.3.3.2. Determinarea câmpului termic la aşchiera întreruptă

Probleme deosebite apar la procesele de aşchiere discontinuă, unde regimul termic nu este stabil (figura 2, pag. 5). Pentru a studia aceste, as-

pecte *Tonshoff* și *Denkena* au prezentat în [123]<sup>14</sup>, rezultatele cercetării lor în care au urmărit aspectele tribologice de la o aşchiere discontinuă. Cum măsurarea forțelor și temperaturilor la astfel de operații sunt foarte dificil de realizat, cei doi au simulat condițiile operației de frezare printr-o operație de strunjire la care piesa avea practicate canale longitudinale sub forma unor caneluri. Modificând dimensiunile acestor canale s-au putut obține condiții asemănătoare celor de la operația de frezare. Ei au stabilit astfel că la sculele de tip freză gradientul termic este foarte mare, înregistrându-se diferențe de până la 500°C între o ieșire și o nouă intrare în aşchiere a aceluiași dinte, la scule armate cu materiale ceramice (AL<sub>2</sub>O<sub>3</sub>TiC).

În [143]<sup>15</sup> sunt prezentate rezultatele măsurării temperaturii în cazul unui proces de aşchiere la viteze mari (200 m/min), experiment efectuat la strunjire pe o instalație balistică. Piesa a avut o configurație asemănătoare cu cea din experimentul efectuat de *Tonshoff* și *Denkena*, observându-se o temperatură de vârf la finalul unei curse active a sculei în material de aproape 1400°C, la strunjirea oțelului inoxidabil 08X18H10T cu o adâncime  $t=0,5$  mm. *Korneeva*, în [51], a efectuat un studiu pentru estimarea influenței mai multor factori, printre care și temperatura asupra uzurii sculei la un proces de aşchiere discontinuu. Ea a stabilit că la aşchiera discontinuă cu viteze ultrarapide, influența cea mai mare asupra uzurii o are factorul termic.

#### 1.4.3.4. Metoda cristalelor lichide

O metodă mai ieftină pentru determinarea unei distribuții termice o reprezintă acoperirea suprafeței cu o soluție organică ce este capabilă de a forma o stare intermediară între cea de solid și lichid. Aceste substanțe sunt de obicei foarte sensibile la schimbările de temperatură, lucru care poate fi evidențiat prin iluminare cu o sursă polarizată. Schimbările respective sunt măsurabile în spectrul vizibil sub forma diverselor culori în funcție de compoziția substanței active folosite, fiind astfel posibilă fotografierea lor pe un film color normal.

Sursa [157] recomandă acoperirea în prealabil a suprafeței cu un strat de vopsea neagră și utilizarea mai multor compoziții de soluții care să reacționeze la diverse intervale de temperatură. Datorită particularităților metodei, nu se poate aplica decât pentru studiul câmpului termic de la exte-

<sup>14</sup>Tönshoff, H.,K., *Tribological Aspects of Interrupted Cutting with Ceramic Tool Materials*

<sup>15</sup>\*\*\*, *Cercetări asupra temperaturii procesului de aşchiere discontinuu cu viteze foarte mari*

riorul maşinilor unelte, sau al suporturilor portscule, ceea ce explică restrânsa ei aplicare în activitatea de cercetare.

#### 1.4.3.5. Metoda interferometriei holografice

Este o metodă folosită pentru a determina deplasările de ordin micrometric ale unor subsansambluri statice, de obicei la intervale mari de timp. În figura 14.a, este expusă schema de principiu a metodei, care folosind o suprapunere pe o placă holografică a două imagini create de un fasciul laser tratat într-un sistem optic, permite, în cazul de faţă, estimarea deplasărilor păpuşii fixe.

Folosind interferenţele holografice, în [103], au fost determinate deplasările arborelui principal de la un strung într-un interval de timp, de 45 de minute. Aceste deplasări au avut valoarea de  $68 \mu\text{m}$  pe o direcţie oblică la  $50^\circ$  aşa cum se poate vedea din figura 14.b

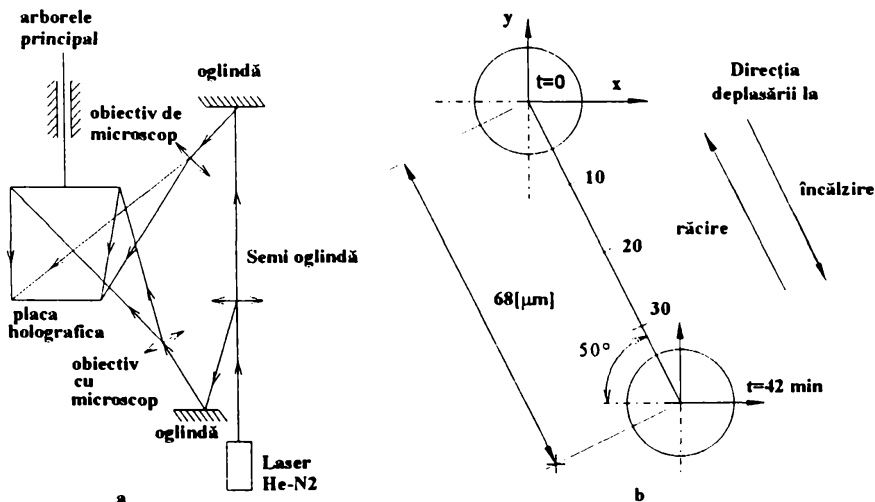


Fig. 14 Interferenţele holografice aplicată la un strung

#### 1.4.3.6. Metoda analogiei electrooptice

Metoda permite obţinerea unor soluţii la problemele de câmp guvernate de ecuaţii de tip Laplace, cum ar fi câmpul electrostatic, câmpul termic, câmpul magnetic. Ca principiu, dintr-un material transparent optic în spectrul vizibil, se creează un model la scară al corpului de analizat. Acest model se umple cu un electrolit ale cărui molecule se pot alinia după liniile unui câmp electrostatic creat de electrozi plasaţi astfel încât să simuleze condiţiile de contur din realitate. Schema de principiu a metodei este prezentată în figura 15.

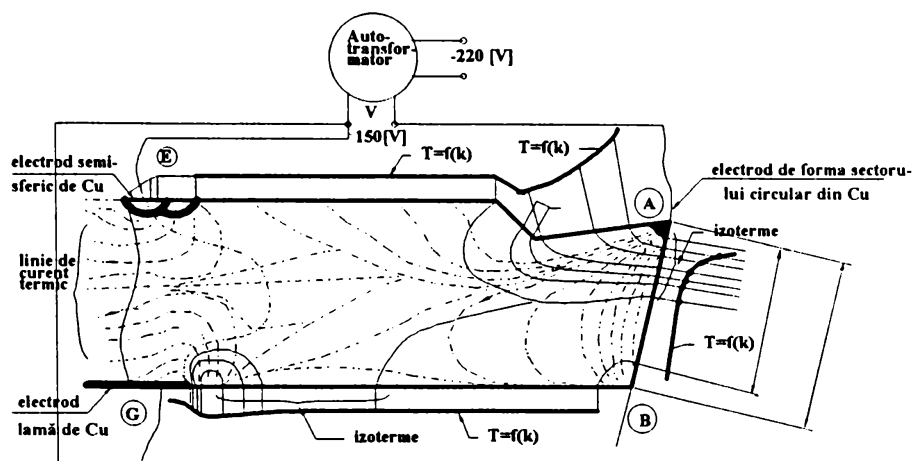


Fig. 15 Analogie electrooptică aplicată la un cuțit de strung

De fapt, se studiază câmpul termic necunoscut, prin intermediul unui câmp similar cunoscut; în acest caz câmpul electrostatic. Câmpul electrostatic devine observabil prin iluminarea cu lumină polarizată a cuvei, fiind astfel posibilă realizarea de fotografii. Problema dificilă o reprezintă etalonarea curbelor de izonivel, în [76] fiind dată soluția etalonării pe piese cu configurație regulată, la care se poate rezolva pe cale analitică ecuația lui Laplace.

#### 1.4.4. STUDIUL DEFORMAȚIILOR TERMICE

Efectul imediat al câmpului termic este producerea deformațiilor în sistemul de prelucrare, fiind astfel un factor care crează o buclă de reacție asupra preciziei de prelucrare. În multe din studiile analizate s-au făcut cercetări asupra mărimii acestor deformații și corelații ale lor cu factorii implicați în procesele de aşchiere.

Yokoyama, în [136], a făcut un studiu pentru operația de honuire, particularizat pentru piese cu pereți subțiri. El a stabilit că mărimea deformațiilor este sensibil egală cu cea de la piesele cu pereți mai groși, dar că ele progresează mult mai rapid. De asemenea, a stabilit că la pietrele de honuire tratate cu sulfuri temperatura este mai scăzută decât la cele cu parafină. Factorul cel mai pregnant în acest caz a fost gradul de uzură al pietrei, care are o influență considerabilă asupra câmpului termic.

Pethö, în [76], a realizat un cuțit autocompensator, care limitează influența factorului termic asupra alungirii acestuia. Soluția constructivă, care are o pastilă de cupru încorporată, este prezentată în figura 16

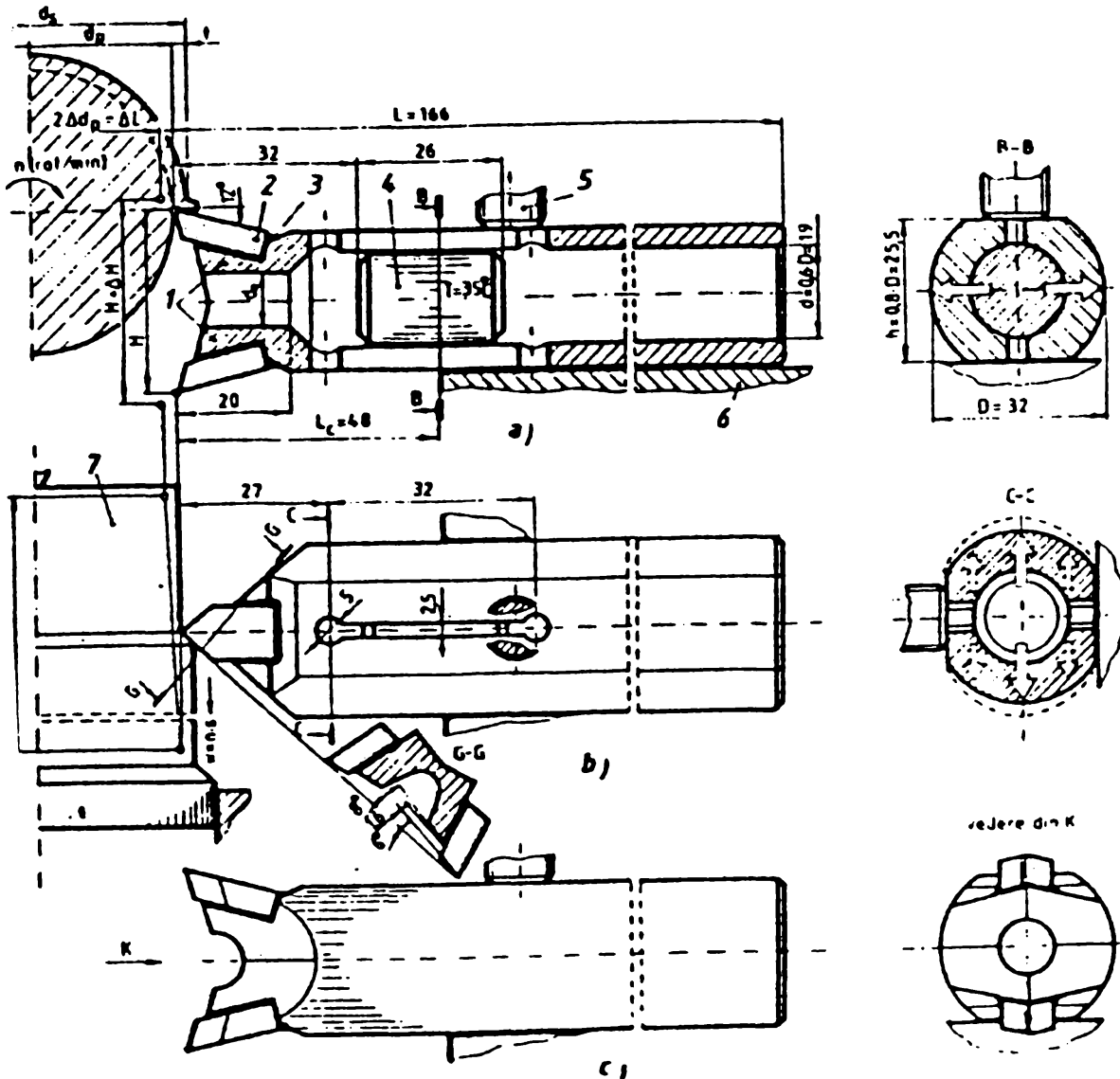


Fig. 16 Soluție autocompensatoare realizată de Pethö

Werner, în [131], a arătat că la operațiile de rectificare atât temperatura locală cât și deformările pot fi reduse, dacă se ridică viteza relativă dintre piesă și semifabricat. El a prezentat acest lucru sub forma unor grafice obținute experimental, unde temperatura este exprimată în funcție de volumul de material îndepărtat în unitatea de timp.

Pentru limitarea efectului de alungire a sculei, Drăghici în [24] propune mărirea înălțimii părții în consolă a sculei.

## 1.5. CONCLUZII

Din studiul lucrărilor de specialitate analizate, rezultă că:

- dintre sursele termice implicate într-un proces de aşchiere, cea mai importantă și totodată dificil de estimat, este cea produsă de detașarea așchiilor și frecările pe fețele de degajare și așezare ale sculei;
- dintre toate elementele sistemului de prelucrare, câmpul termic are cea mai mare influență asupra sculei;
- metodele analitice folosite la studiul câmpului termic, se bazează fie pe simplificări ale geometriei domeniului studiat, pentru a putea rezolva ecuația conducției căldurii prin metode clasice, fie pe utilizarea unor relații determinate experimental, ce permit calculul temperaturii în vârful sculei, sau a alungirilor rezultate;
- utilizarea metodelor analitice și numerice de studiu a câmpului termic, vizează numai obținerea distribuției acestuia, fără să se fi elaborat pe un fundament matematic, posibilități de apreciere a debitului sursei termice, care este de fapt *cauza* fenomenului; ba chiar multe simulări din materialele studiate, făcute prin metode numerice, îl ignoră<sup>16</sup>, deși este un termen esențial al ecuației conducției căldurii;
- dintre metodele experimentale existente, termografia electronică în infraroșu este cea mai precisă, oferind și cel mai mare volum de informații;
- câmpul termic are cea mai mare influență în mecanismul uzurii sculei.

---

<sup>16</sup> Din studiile făcute de subsemnatul, numai Tönshoff și Denkena, în [123], îl folosesc, însă fără să precizeze cum l-au determinat

## 1.6. OBIECTIVELE STUDIULUI

Analizând metodele de studiu și rezultatele cercetărilor în domeniul câmpului termic din procesele de aşchiere, s-a detașat următorul aspect, care nu a fost tratat în lucrările cercetate:

- determinarea termenului ce modelează aportul surselor interne de căldură din ecuația conducerii căldurii<sup>17</sup>, lucru care indiferent de metodele folosite pentru rezolvarea ei, nu poate fi ignorat; acest termen introduce, de fapt, în ecuația căldurii, sursa termică provenită din procesul de aşchiere.

Pentru estimarea acestui termen, s-au stabilit ca obiective următoarele:

- obținerea distribuției câmpului termic în diverse conjuncturi ale parametrilor de prelucrare prin utilizarea termografiei electronice în infraroșu;

- determinarea precisă a emisivității corpurilor implicate, ca factor esențial în etalonarea termogramelor;

- rezolvarea ecuației căldurii prin metoda elementului finit, după introducerea distribuțiilor de temperaturi, corectate cu valorile reale ale emisivităților, având ca necunoscută termenul ce modelează aportul surselor termice interne;

- determinarea forțelor de aşchiere pentru a servi ca element suplimentar de corelare cu ceilalți parametri;

- obținerea, prin regresie polinomială, a unor relații care să permită estimarea aportului surselor interne de căldură, în funcție de material, parametrii regimului de aşchiere etc.

### *Observație*

Pentru că folosirea termografiei în infraroșu a fost elementul experimental esențial și echipamentul de care s-a dispus impunea accesul direct la zona de aşchiere, s-au restrâns determinările la operația de strunjire exterioară, fără folosirea lichidului de răcire, care ar fi făcut imposibilă termografierea procesului.

---

<sup>17</sup> Elemente de detaliu despre ecuația conducerii căldurii sunt prezentate în capitolul 3.1.1.1

# CAPITOLUL 2

## METODA ELEMENTULUI FINIT

- *FORMULĂRI ALE METODEI*
- *ETAPE ALE ANALIZEI*
- *CONCLUZII PRIVIND POSIBILITĂȚILE DE UTILIZARE A METODEI ELEMENTULUI FINIT*



## 2. METODA ELEMENTULUI FINIT

Pentru rezolvarea problemelor ingineresti, există două moduri distincte de abordare a sistemelor fizice: tratarea unitară, aproximarea pe *subdomenii*.

În primul mod, sistemul este tratat ca un mediu continuu, pentru care sunt scrise una sau mai multe ecuații de echilibru, care rezolvate produc distribuția câmpului respectiv. În cel de-al doilea mod sistemul este idealizat ca un ansamblu de elemente, pentru fiecare dintre ele fiind scrisă câte o ecuație de echilibru, după care se rezolvă sistemul rezultat de ecuații pentru a obține o soluție aproximativă.

Pentru dezvoltarea ecuațiilor diferențiale care guvernează sistemul există două metode: formularea diferențială și formularea variațională. În prima, ecuațiile sunt obținute considerând echilibrul unui element diferențial infinitezimal, în cea de-a doua, ecuațiile de echilibru sunt exprimate considerând balanța energetică a sistemului (energia și lucrul mecanic din sistem). Ambele formulări sunt echivalente și se pot obține una din cealaltă. Ecuațiile de echilibru sunt însoțite de condițiile de contur și de condițiile inițiale.

Există mai multe metode de a rezolva problema cu condiții de contur și condiții inițiale pentru cazul liniar și neliniar, însă soluțiile exacte sunt de obicei disponibile numai pentru cazuri foarte simple (ex. domenii rectangulare). Aceste soluții pot fi obținute prin integrarea directă a ecuațiilor diferențiale prin tehnici cum ar fi: separarea variabilelor, transformatele Fourier sau Lapalce. Pentru sisteme la care nu se pot obține soluții exacte, se apelează la una din următoarele metode pentru a obține soluții aproximative: serii de puteri, scheme de probabilitate, metoda reziduurilor, metoda diferențelor finite, Rayleigh-Ritz și metoda elementului finit.

Metoda elementului finit se bazează pe următoarea premisă: în locul rezolvării directe a ecuațiilor diferențiale ce guvernează sistemul, se rezolvă o formă integrală a acestora. Această metodă permite rezolvarea unor probleme ingineresti de mare complexitate, prin tehnici specifice analizei numerice. Ea oferă soluții aproximative în puncte stabilite în prealabil ale sistemului studiat, pentru fenomene mecanice și electrice din cele mai diverse.

## 2.1. Formulări ale metodei

Formularea metodei elementului finit presupune transformarea ecuațiilor de echilibru de la domeniul continuu, la nivelul unei mulțimi discrete de subdomenii prestabilite conform celor prezentate la 2.2.1 și a cumulării acestora pentru a obține un sistem echivalent cu cel inițial. Patru metode sunt disponibile pentru obținerea ecuațiilor la nivel de subdomeniu, pentru cea variațională fiind prezentate în detaliu transformările generice ale formelor diferențiale în forme integrale.

### a) formularea directă

Această formulare derivă din metoda de calcul matricial a structurilor de bare. În prezent formularea nu este practic folosită în programele comerciale de analiză cu element finit, datorită faptului că reduce mult din generalitatea formulării problemei.

### b) formularea variațională

Această formulare se aplică în general pentru probleme de mecanica solidului, probleme de transfer termic. Ea pornește de la ecuațiile diferențiale care guvernează sistemul, manevrând conceptele de energie și lucru mecanic pentru obținerea ecuațiilor integrale. Dacă problema analizată poate fi descrisă de o ecuație diferențială, care în general are forma din relația (21), se urmărește transformarea ei într-o formă integrală, numită funcționala  $\Pi$ .

$$L(\Phi) - f = 0 \quad \text{pe } D \quad (21)$$

unde:  $D$  este domeniul sistemului, cu condițiile la limită;  $\Phi = \phi$  pe conturul  $B$  al domeniului  $D$ ;  $\Phi$  este variabila care reprezintă soluția exactă;  $f$  este o funcție cunoscută;  $L$  este un operator (liniar sau nelinier) ordinar sau parțial.

Pentru a obține funcționala  $\Pi$  din relația (21), cu condițiile la limită corespunzătoare, se caută o integrală  $I(\Phi)$  a cărei primă variație față de  $\Phi$  se anulează. Multiplicând relația (21) cu prima variație a lui  $\Phi$ ,  $\delta\Phi$  și integrând peste domeniul  $D$  se obține relația:

$$\int_D \delta\Phi [L(\Phi) - f] dD = 0 \quad (22)$$

Prin transformări matematice, folosind proprietățile operatorului  $\delta$ <sup>18</sup>, se urmărește scoaterea acestuia în afara integralei astfel încât noua expresie

<sup>18</sup> Proprietățile acestui operator și justificări ale folosirii lui în acest context, se află în lucrări de calcul variațional, cum ar fi [52], [54]

integrală să devină de fapt funcționala căutată. Generic acest lucru poate fi scris astfel:

$$\delta \int_D [L^*(\Phi) - f^*] dD = 0 \quad (23)$$

unde  $L^*(\Phi)$ ,  $f^*$  este un nou operator, respectiv funcție ce rezultă din cele inițiale

Analizând relația (23), putem face observația din relația (24),

$$\Pi = I(\Phi) = \int_D [L^*(\Phi) - f^*] dD \quad (24)$$

ceea ce în final va produce relația (25), care exprimă sintetic formularea variațională a unei probleme guvernate de ecuații diferențiale:

$$\delta [I(\Phi)] = \delta \Pi = 0 \quad (25)$$

Pentru aceste transformări ale formei diferențiale (21) în forma integrală (25), nu există un algoritm general pentru toate cazurile, de obicei folosindu-se integrarea prin părți și teorema lui Green. Dacă există însă principii variaționale pentru fenomenul analizat, folosind principiul energiei potențiale minime, sau principiul lui Hamilton, se poate scrie direct funcționala respectivă.

### c) formularea reziduală

Această metodă se aplică în general pentru probleme de mecanica fluidelor și probleme dependente de timp unde nu există, sau nu sunt cunoscute principii variaționale. Aplicarea metodei presupune parcurgerea a doi pași:

- alegerea unei soluții aproximative care satisface ecuația cu condițiile limită și condițiile inițiale;
- înlocuirea iterativă a ei în ecuația originală, pentru a evalua precizia aproximării alese.

De obicei, această soluție este aleasă sub forma unui polinom  $\Phi_a$ , ca în relația (26):

$$\Phi \approx \Phi_a = \sum_{i=1}^n c_i g_i \quad (26)$$

unde  $c_i$  sunt coeficienții necunoscuți, iar  $g_i$  sunt funcții de variabila independentă analizată

Înlocuirea sa în ecuația diferențială originală (21) nu va anula expresia, ci va produce un reziduu  $R$ :

$$R = L(\Phi_a) - f \neq 0 \quad (27)$$

Din acest moment problema care se pune este cea a minimizării rezidului, ceea ce echivalează cu obținerea unei soluții cât mai precise. Pentru acest lucru se formează o medie ponderată care trebuie să se anuleze pe întreg domeniul D:

$$\int_D RW, dD = \int_D [L(\Phi_a) - f] W_i dD = 0 \quad ; i = 1, 2, \dots, n \quad (28)$$

Pentru alegerea funcțiilor de pondere  $W_i$  sunt folosite următoarele metode:

- metoda cologației pe puncte;
- metoda cologației pe subdomenii;
- metoda Galerkin;
- metoda celor mai mici pătrate.

Metoda Galerkin folosește funcțiile  $g_i$  în locul funcțiilor de pondere  $W_i$ , ceea ce transformă ecuația (28) într-un sistem de  $n$  ecuații cu  $n$  necunoscute, coeficienții  $c_i$ , conform relației :

$$\int_D R g_i dD = \int_D [L(\Phi_a) - f] g_i dD = 0 \quad ; \quad i = 1, 2, \dots, n \quad (29)$$

Din punct de vedere matematic, ecuațiile (25) și (29) sunt echivalente, ele fiind baza formulărilor moderne ale metodei elementului finit.

#### *d) formularea energetică*

Este formularea cea mai generală, dar și cea mai dificil de implementat, fiind folosită la modelarea simultană a fenomenelor complexe, neliniare și nestaționare

În mod curent, cele mai folosite formulări sunt cea variațională și cea reziduală. Comun pentru ambele, este faptul că forma integrală (30),

$$\int_D F(\Phi) dD = 0 \quad (30)$$

la care se ajunge după transformarea ecuațiilor diferențiale originale, ce guvernează fenomenul, este similară și poate fi descompusă într-o sumă de integrale, relația (31):

$$\int_D F(\Phi) dD = \sum_{i=1}^m \int_{D^i} F^i(\Phi^i) dD^i = 0 \quad (31)$$

exprimate pe subdomenii geometrice care însumate reconstituie domeniul inițial conform relației (32):

$$D \approx \sum_{i=1}^m D^i \quad ; \quad \Phi \approx \sum_{i=1}^m \Phi^i \quad (32)$$

Această ultimă observație permite evaluarea integralelor din relația (31) pe domenii simple și transformarea sistemului rezultat într-un sistem algebric liniar, care poate fi transpus în formă matricială ușor de implementat într-un algoritm iterativ de calcul.

## 2.2. Etape ale analizei

Etapele generale ale acestei metode sunt prezentate în continuare, iar particularizările pentru ecuația căldurii, care au stat la baza programului, din anexa 1, fiind prezentate în capitolul 3.2.

### 2.2.1. Aproximarea geometrică a domeniului studiat

În această etapă se stabilesc punctele unde este aproximată soluția sistemului inițial, numite și nodurile rețelei. Ele se obțin prin divizarea sistemului într-un număr finit de subdomenii, numite elemente finite, cu formă geometrică simplă. Aceste subdomenii aproximează în primul rând din punct de vedere geometric sistemul analizat, fiind puține cazuri în care modelul geometric realizat se suprapune perfect peste cel real. Astfel s-a realizat înlocuirea unui sistem continuu cu un set de puncte, care conectate corect între ele determină elementele. În funcție de geometria sistemului, elementele finite folosite pot fi unidimensionale, bidimensionale și tridimensionale. Aspecte de sinteză legate de geometria celor mai uzuale elemente finite folosite, sunt prezentate în [96]<sup>19</sup> capitolul 3, precum și în multe alte lucrări de specialitate.

Descrierea elementelor trebuie făcută astfel încât să nu existe suprapuneri între acestea, sau goluri interne care nu există în realitate. Numeroși autori<sup>20</sup>, au evidențiat faptul că utilizarea unui număr mai mare de elemente la aproximarea geometrică mărește precizia soluției aproximative, însă aceasta tinde asimptotic către soluția exactă pe măsura măririi numărului de elemente folosite. Deci în funcție de precizia impusă de rezolvarea problemei, precum și de efortul de calcul estimat, se poate determina un număr optim de elemente folosite într-o anumită conjunctură.

<sup>19</sup> Roșca A.S., *Metode de analiză a câmpului termic în procesele de așchiere*, Referat nr.2, pentru stagiul de doctorat, U.P. Timișoara, 1995

<sup>20</sup> [21], [26], [36], [139]

### 2.2.2. Alegerea sau proiectarea elementelor finite

Un element finit este caracterizat de forma sa geometrică, numărul și tipul nodurilor (interioare sau exterioare), variabilele nodale pe care le poate manipula, tipul aproximării folosite.

Variabilele nodale, sau gradele de libertate ale elementului, sunt reprezentate de necunoscutele din ecuația fenomenului, care acum vor avea valori discrete în nodurile rețelei.

#### 2.2.2.1. Alegerea aproximării soluției pentru variabila de câmp în interiorul elementelor

După determinarea aspectului geometric al elementelor finite, în această etapă se stabilește tipul aproximării pentru variabila sau variabilele de câmp analizate, ceea ce va finaliza alegerea tipurilor de elemente finite folosite. În mod uzual se folosesc aproximări polinomiale, gradul polinomului fiind în concordanță cu numărul de noduri ale elementului și cu numărul de variabile nodale.

Acest lucru se stabilește la proiectarea prototipului elementului finit<sup>21</sup> și trebuie să asigure următoarele cerințe:

*Compatibilitate*, adică asigurarea continuității la hotarul elementelor a funcției de câmp și a derivatelor sale până la ordinul  $r-1$ , unde  $r$  este gradul maxim sub care apare funcția de câmp în formularea integrală a fenomenului respectiv. Elementele care respectă această condiție asigură, în plan imaginar, un proces de asamblare sistemului fără goluri virtuale în sistemul analizat.

*Completitudine*, respectiv asigurarea continuității până la ordinul  $r$  a funcției de câmp și a derivatelor sale în interiorul elementului. Aceasta presupune pentru polinoame, existența tuturor termenilor corespunzători cu gradul polinomului, inclusiv termenul liber. Astfel se asigură obținerea unei valori finite a variabilei de câmp, atunci când dimensiunea elementului tinde către zero, deci practic derivata funcției analizate este constantă pe subdomeniul considerat.

Pentru determinarea coeficienților polinomului de un anumit grad, se folosește triunghiul lui Pascal, prezentat în marea majoritate a lucrărilor de element finit. Astfel pentru elemente plane, polinomul de aproximare este cel din relația (33):

$$\Phi^r(x, y) = \sum_{k=1}^{n_r} \alpha_k x^i y^j \quad ; i + j \leq n; \quad n_r = (n+1)(n+2)/2 \quad (33)$$

<sup>21</sup> Pentru un program comercial acest aspect este predefinit de producător și precizat în documentație pentru toate tipurile de elemente finite din biblioteca programului respectiv

unde:  $\alpha$  sunt așa numitele coordonate generalizate ale elementului;  $n$  este gradul polinomului ( $n=1$  pentru elemente liniare,  $n=2$  pentru elemente cuadractice,  $n=3$  pentru elemente cubice etc);  $n_g$  este numărul de termeni din polinom.

Între numărul de noduri geometrice  $g$ , ale unui element, numărul de grade de libertate per nod  $n_l$  și numărul coordonatelor generalizate ale elementului  $e_g$ , trebuie să existe relația (34) care asigură un număr suficient de ecuații elementare pentru determinarea necunoscutelor:

$$g = e_g / n_l \quad (34)$$

Deși nu este obligatoriu, este de dorit ca un polinom de aproximare să aibă toți termenii, pentru că această simetrie va asigura invarianța geometrică, adică nu vor exista direcții preferențiale pentru elemente cu formă geometrică neregulată.

Îndeplinirea simultană a celor două condiții de mai sus asigură necondiționat îndeplinirea convergenței monotone, adică apropierea asimptotică de soluția exactă pe măsura rafinării rețelei de elemente finite<sup>22</sup>.

#### 2.2.2.2. Funcția de formă a elementului

Considerând elementul plan triunghiular cu variație liniară din figura 17 de la pagina 39, (care nu are noduri suplimentare pe laturi), variabila de câmp în interiorul lui va fi aproximată cu funcția din relația (35):

$$\Phi(x, y) = \alpha_1 + \alpha_2 x + \alpha_3 y \quad (35)$$

unde: variabila  $\Phi$  poate fi temperatura, o componentă a unei deplasări elastice, presiunea etc.;  $x, y$  sunt coordonatele geometrice;  $\alpha_i$  sunt așa numitele coordonate generalizate.

Pentru acest tip de element, funcția polinomială de gradul 1 nu are nevoie și de termenul  $\alpha_4 xy$ , așa cum rezultă din analiza triunghiului lui Pascal, fiind suficient sistemul de trei ecuații cu trei necunoscute din relația (36), pentru a putea evalua elementul.

În această situație valorile nodale ale variabilei de câmp vor fi  $\Phi_1, \Phi_2, \Phi_3$ , și pot fi exprimate din relația (35), în care se particularizează  $x$  și  $y$  pentru coordonatele nodurilor:

$$\begin{aligned} \Phi_1 &= \alpha_1 + \alpha_2 x_1 + \alpha_3 y_1 \\ \Phi_2 &= \alpha_1 + \alpha_2 x_2 + \alpha_3 y_2 \\ \Phi_3 &= \alpha_1 + \alpha_2 x_3 + \alpha_3 y_3 \end{aligned} \quad (36)$$

<sup>22</sup> [137] Zeid I. CAD/CAM Theory and practice, p.896; [21] Dhatt G., The finite element method displayed, p.33.

Relația (36) poate fi exprimată matricial conform relației (37):

$$[\Phi] = [G] \cdot [\alpha] \quad (37)$$

$$\text{unde } [\Phi] = \begin{bmatrix} \Phi_1 \\ \Phi_2 \\ \Phi_3 \end{bmatrix}; \quad [G] = \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}; \quad [\alpha] = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix}$$

Din (37) se poate determina expresia matricială pentru vectorul  $\alpha$ , care conține coordonatele generalizate și rezultă:

$$[\alpha] = [G]^{-1} [\Phi] \quad (38)$$

Din relația (35) rezultă o nouă formă pentru funcția de aproximare în interiorul elementului:

$$[\Phi]^r = [P] \cdot [\alpha] \quad (39)$$

$$\text{unde } P = [1 \ x \ y]$$

Înlocuind (38) în (39), rezultă o nouă expresie pentru funcția de aproximare a variabilei, în care apare așa numita funcție de formă a elementului:

$$[\Phi]^r = [N][\Phi] \quad (40)$$

unde vectorul  $[N] = [P] \cdot [G]^{-1} = [N_1 \ N_2 \ N_3]$ , conține funcțiile de formă  $N_i = N_i(x, y)$  ale elementului considerat

Ecuția (40) se poate dezvolta pentru elementul triunghiular liniar, conform relației (41):

$$\Phi^r = N_1 \Phi_1 + N_2 \Phi_2 + N_3 \Phi_3 \quad (41)$$

sau generaliza, pentru un element cu  $n$  noduri:

$$\Phi^r = N_1 \Phi_1 + N_2 \Phi_2 + \dots + N_n \Phi_n = \sum_{i=1}^n N_i \Phi_i \quad (42)$$

Din expresia de mai sus, punând condiția ca în nodul  $i$  funcția  $\Phi_i$  să aibă valoarea  $\Phi_i$ , se observă că o proprietate importantă a funcțiilor de formă  $N_i$  este aceea că au valoarea 1 în nodul  $i$  și valoarea zero în celelalte noduri, adică:

$$N_i = \begin{cases} 1; & k = i \\ 0; & k \neq i \end{cases} \quad (43)$$

Această observație, duce ulterior în spațiul parametric, la obținerea rapidă a expresiilor acestor funcții prin inspecție, pe baza unor considerente de natură geometrică.



După cum se vede, funcția de formă a elementului permite înlocuirea coordonatelor generalizate cu valorile nodale ale funcției analizate și coordonatele geometrice nodurilor.

Trebuie observat însă că obținerea funcțiilor de formă cu relația (40) poate fi imposibilă pentru anumite orientări ale axelor de coordonate, datorită anulării determinantului matricei  $G$ .

### 2.2.2.3. Dezvoltarea ecuațiilor la nivel de element

Pentru formularea variațională, metoda elementului finit presupune determinarea valorilor nodale  $\Phi_i$ , care fac funcționala  $\Pi$  staționară. Din relațiile (25) și (31), aplicând discretizarea în baza relației (32), rezultă:

$$\delta \Pi = \sum_{i=1}^m \delta \Pi^e = \sum_{i=1}^m \delta I^e(\Phi^e) = 0 \quad (44)$$

Din ecuația (44), dacă  $I^e$  este considerată în raport cu valorile nodale  $\Phi_i$ , condiția de staționaritate devine:

$$\begin{bmatrix} \frac{\partial I^e}{\partial \Phi_1} \\ \frac{\partial I^e}{\partial \Phi_2} \\ \vdots \\ \frac{\partial I^e}{\partial \Phi_n} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (45)$$

unde  $n$  este numărul de noduri al elementului

Relația (45) furnizează sistemul de ecuații necesar pentru a caracteriza comportamentul elementului  $e$ .

Ca procedură generală pentru obținerea matricelor elementare, se înlocuiește relația (42) în funcționala  $\Pi^e$ , la nivel de element, iar după diferențiere se obține relația (45) corespunzătoare elementului  $e$ . După transformări matematice, care sunt prezentate în detaliu pentru cazul transferului de căldură prin conducție în capitolul 3.2, ecuația generală la nivel de element este de forma:

$$[K]^e \cdot [X]^e = [F]^e \quad (46)$$

unde:  $[K]^e$  este matricea de rigiditate sau conductivitate a elementului;  $[X]^e$  este vectorul cu necunoscute;  $[F]^e$  este vectorul cu încărcările.

2.2.2.4. Evaluarea elementelor izoparametrice

Pentru evaluarea funcțiilor de formă ale elementului, este necesară și calcularea inversei matricei [G] din relația (37), ale cărei elemente depind de coordonatele nodale ale elementului. După cum s-a mai afirmat, este posibil ca pentru anumite orientări ale sistemului global de coordonate, inversa matricei [G] să nu existe, ceea ce duce la imposibilitatea evaluării matricelor elementare și deci a rezolvării problemei.

Pentru a evita aceste situații, se poate defini un sistem de coordonate local elementului, față de care acesta să aibă totdeauna o formă și orientare constantă. Acest sistem de coordonate definește așa numitul spațiu parametric, unde de exemplu un element plan quadrilateral va avea tot timpul formă pătrată cu latura de 2 unități, centrat în origine, un element triunghiular, ca cel din fig.17 va avea formă de triunghi dreptunghic cu catetele de o unitate, ș.a.m.d., elementul numindu-se element de referință.

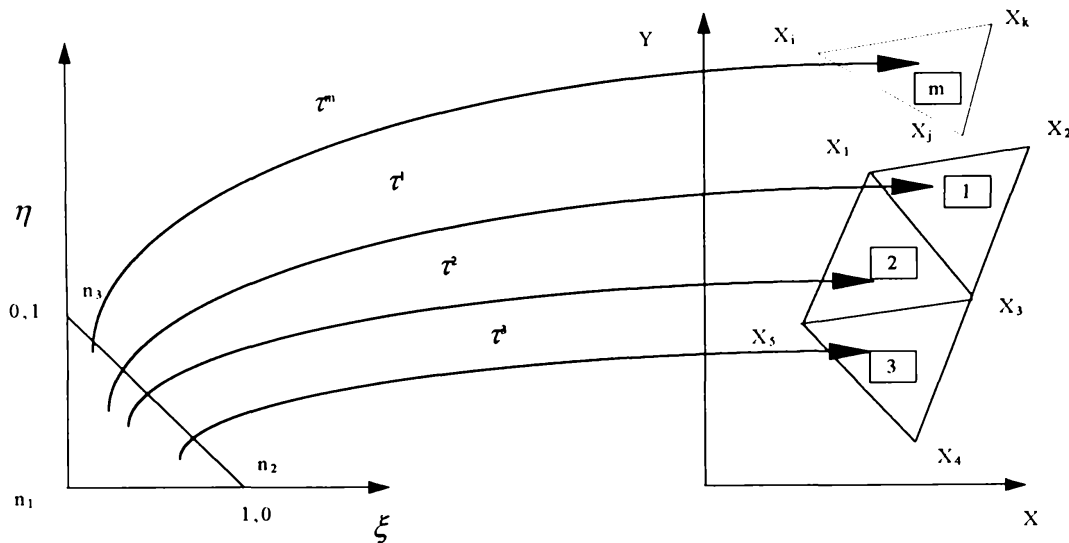


Fig. 17 - Relația spațiu parametric - spațiu cartezian pentru un element triunghiular

Pentru a raporta elementul de la sistemul parametric la sistemul global, se folosesc funcții de transformare care, evident, sunt diferite de la element la element (transformările  $\tau^j$  din figura 17). Aceste transformări permit exprimarea coordonatelor fiecărui punct din elementul real, în funcție de coordonatele punctului corespunzător din elementul de referință.

Transformările geometrice, pentru tipul de element considerat, pot fi exprimate prin relațiile (47), analoge cu relația (41), care aproximează variabila de câmp studiată.

$$\begin{aligned}
 x(\xi, \eta) &= N_1^*(\xi, \eta)x_i + N_2^*(\xi, \eta)x_j + N_3^*(\xi, \eta)x_k = [N^*] \cdot \begin{bmatrix} x_i \\ x_j \\ x_k \end{bmatrix} \\
 y(\xi, \eta) &= N_1^*(\xi, \eta)y_i + N_2^*(\xi, \eta)y_j + N_3^*(\xi, \eta)y_k = [N^*] \cdot \begin{bmatrix} y_i \\ y_j \\ y_k \end{bmatrix}
 \end{aligned}
 \tag{47}$$

În anumite situații funcțiile pentru transformările geometrice  $N^*$  sunt diferite de funcțiile  $N$ , pentru aproximarea soluției problemei în interiorul elementului. Totuși în multe cazuri funcțiile care aproximează geometria pot fi folosite și pentru aproximarea soluției, acest tip de elemente numindu-se izoparametrice.

Elementele izoparametrice sunt cele mai frecvent folosite în analizele ingineresti curente, însă în afara acestora mai există și elemente la care cele două funcții nu coincid. Astfel putem avea:

- *elemente pseudoparametrice* - funcțiile folosite pentru geometrie și aproximarea soluției sunt de natură diferită;

- *elemente subparametrice* - ambele funcții sunt polinomiale, gradul polinomului pentru transformarea geometrică fiind mai mic decât cel pentru aproximarea variabilei;

- *elemente superparametrice* - ambele funcții sunt polinomiale, gradul polinomului pentru transformarea geometrică fiind mai mare decât cel pentru aproximarea variabilei.

Pentru elementul de referință din figura 17 de la pagina 39, se observă că sunt valabile relațiile de natură geometrică (48), în spațiul parametric:

$$\xi + \eta \leq 1 \quad ; \quad \xi \geq 0 \quad ; \quad \eta \geq 0
 \tag{48}$$

În această situație, geometria unui element triunghiular, poate fi descrisă prin expresiile din relația (49):

$$x(\xi, \eta) = \begin{bmatrix} 1 - \xi - \eta & \xi & \eta \end{bmatrix} \cdot \begin{bmatrix} x_i \\ x_j \\ x_k \end{bmatrix} \quad ; \quad y(\xi, \eta) = \begin{bmatrix} 1 - \xi - \eta & \xi & \eta \end{bmatrix} \cdot \begin{bmatrix} y_i \\ y_j \\ y_k \end{bmatrix}
 \tag{49}$$

Pentru alte tipuri de elemente, cu altă formă și număr de noduri, se pot obține expresii analoge cu cea din relația (49), pornind de la considerații evidente în raport cu geometria elementului. Acest procedeu în literatura de specialitate este cunoscut ca fiind așa numita *obținere prin inspecție a funcțiilor de transformare*.

O condiție esențială la folosirea elementului de referință este unicitatea transformării, ceea ce presupune ca determinantul matricei iacobiene asociată acestei transformări (din relația (50)) să fie diferit de zero:

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} x_j - x_i & y_j - y_i \\ x_k - x_i & y_k - y_i \end{bmatrix} \quad (50)$$

Pentru elementul triunghiular liniar analizat aceasta înseamnă:

$$\det(J) = (x_j - x_i) \cdot (y_k - y_i) - (x_k - x_i) \cdot (y_j - y_i) \quad (51)$$

Din relația (51) se observă că iacobianul, care reprezintă dublul ariei triunghiului, se anulează în cazul elementului triunghiular când cele trei noduri care definesc elementul sunt coliniare.

Utilizarea spațiului parametric asigură și obținerea unor elemente cu laturile curbe în spațiul cartezian real, dacă funcțiile de aproximare sunt cel puțin de gradul doi. Acest lucru înseamnă o mai bună aproximare a domeniului geometric folosind elemente mai puține.

Deci pentru elementele izoparametrice, coordonatele unui punct din interiorul elementului pot fi exprimate în funcție de coordonatele nodale conform relației (52):

$$x = \sum_{i=1}^n N_i x_i \quad ; \quad y = \sum_{i=1}^n N_i y_i \quad (52)$$

Având în vedere aproximarea variabilei de câmp, pentru elementul triunghiular din relația (42), putem scrie:

$$\Phi^e = N_1 T_1 + N_2 T_2 + N_3 T_3 \quad (53)$$

După derivarea relației (53) rezultă:

$$\begin{aligned} \frac{\partial \Phi^e}{\partial x} &= \frac{\partial N_1}{\partial x} T_1 + \frac{\partial N_2}{\partial x} T_2 + \frac{\partial N_3}{\partial x} T_3 \\ \frac{\partial \Phi^e}{\partial y} &= \frac{\partial N_1}{\partial y} T_1 + \frac{\partial N_2}{\partial y} T_2 + \frac{\partial N_3}{\partial y} T_3 \end{aligned} \quad (54)$$

Ținând cont de transformarea de coordonate din spațiul  $(x, y)$  la  $(\xi, \eta)$ , derivatele funcțiilor  $N_i$ , (unde  $i = 1 \dots 3$ ) în raport cu  $\xi$  și  $\eta$  sunt:

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x} \cdot \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \cdot \frac{\partial y}{\partial \xi} \quad ; \quad \frac{\partial N_i}{\partial \eta} = \frac{\partial N_i}{\partial x} \cdot \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \cdot \frac{\partial y}{\partial \eta} \quad (55)$$

sau matricial:

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad (56)$$

Derivând relația (52) și introducând-o în (56), rezultă:

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^3 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^3 \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^3 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^3 \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad (57)$$

unde primul termen din membrul drept al ecuației (57) este iacobi-anul transformării  $(x, y) \rightarrow (\xi, \eta)$ , care poate fi exprimat matricial conform relației (58):

$$[J] = \begin{bmatrix} \sum_{i=1}^3 \frac{\partial N_i}{\partial \xi} x_i & \sum_{i=1}^3 \frac{\partial N_i}{\partial \xi} y_i \\ \sum_{i=1}^3 \frac{\partial N_i}{\partial \eta} x_i & \sum_{i=1}^3 \frac{\partial N_i}{\partial \eta} y_i \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \quad (58)$$

Din relația (57), rezultă

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = [J]^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} \quad (59)$$

Utilitatea acestei expresii este evidențiată în capitolul 3.2.2.1. la obținerea ecuațiilor la nivel de element pentru transferul căldurii prin conducție pentru elemente triunghiulare liniare izoparametrice.

### 2.2.3. Asamblarea matricelor elementare

În urma procesului de discretizare conform relației (31), apar ecuațiile la nivel de element conform relației (46). Pentru a obține soluția sistemului discretizat, ecuațiile matriciale (46) trebuie asamblate, rezultând astfel un sistem matricial global, după rezolvarea căruia se obțin valorile necunoscute. Asamblarea se bazează pe forma algebrică a ecuației (31):

$$\int_D F(\Phi) dD = \sum_{e=1}^n ([K]^e \cdot [X]^e - [F]^e) = 0 \quad (60)$$

sau

$$\sum_{e=1}^n [K]^e \cdot [X]^e = \sum_{e=1}^n [F]^e$$

ceea ce conduce la expresia finală:

$$[K] \cdot [X] = [F] \quad (61)$$

unde:  $[K]$  este matricea globală de tipul  $n \times n$ ;  $[X]$  este vectorul cu necunoscutele nodale;  $[F]$  este vectorul încărcărilor.

Procesul de asamblare se bazează pe acumularea efectelor din toate ecuațiile la nivel de element, pentru elementele care cuprind nodul considerat. Acest algoritm de asamblare se poate observa foarte bine în figura 18 precum și în programul din anexa 1.

Pentru a putea face asamblarea matricei globale, fiecare matrice elementară este expandată la o matrice de același ordin cu cea globală, prin inserarea de linii și coloane cu elemente zero. Astfel după expandare, elementele matricei inițiale ocupă pozițiile corespunzătoare numerotării globale a nodurilor ce compun elementul.

Matricea elementară astfel expandată este adunată cu matricea globală, încât după iterarea tuturor elementelor se obține matricea finală globală  $[K]$  din relația (61).

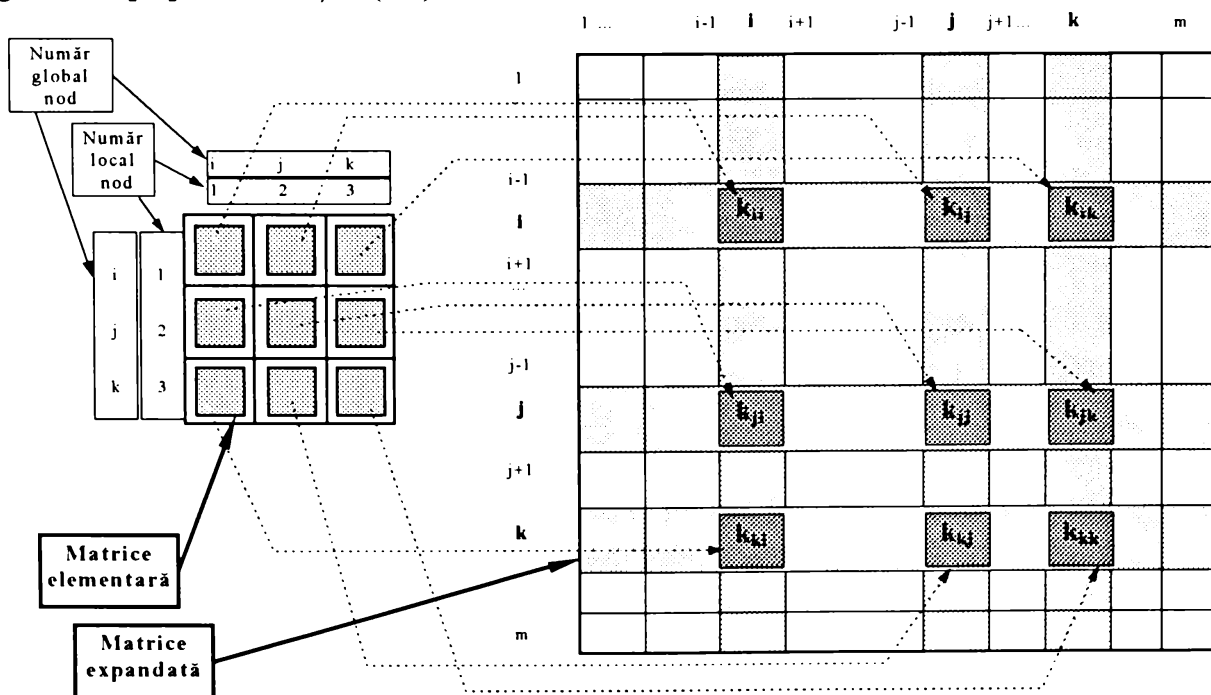


Fig. 18 - Expandarea matricei elementare

Așa cum este demonstrat în numeroase lucrări<sup>23</sup>, matricea globală pentru problemele statice este simetrică, ceea ce permite anumite simplificări la memorarea acesteia. Aspecte legate de aceste artificii de manevrare a datelor sunt prezentate în pragraful 3.2.3.

#### 2.2.4. Aplicarea încărcărilor

Metoda elementului finit reprezintă în esență înlocuirea unui sistem continuu cu o mulțime de puncte discrete în care sunt evaluate soluțiile problemei. În aceste condiții, problema încărcărilor ce sunt aplicate sistemului, trebuie și ea să fie adaptată pentru noua formă. Astfel, pentru încărcările ce sunt concentrate într-un punct, este suficient să se creeze un nod în care se aplică respectiva încărcare. Pentru încărcările distribuite, în unele situații este necesar să se evalueze contribuția încărcării distribuite pe mai multe noduri, reducându-se problema de fapt la mai multe încărcări concentrate.

După ce încărcările sunt evaluate în concordanță cu tipul elementului respectiv, sunt introduse în vectorul încărcărilor din ecuația (46) la nivel de element, lucru care se petrece de obicei simultan cu evaluarea matricei de rigiditate a elementului respectiv.

Ca și la matricea de rigiditate, în concordanță cu relația (60), vectorii de încărcare la nivel de element, sunt asamblați în vectorul global de încărcare, printr-un algoritm de expandare similar cu cel de la matricea de rigiditate.

Vectorul astfel rezultat împreună cu matricea de rigiditate constituie sistemul primar de ecuații pe care mai trebuie aplicate condițiile de contur pentru a-l putea rezolva.<sup>24</sup>

#### 2.2.5. Impunerea condițiilor de contur

Matricea globală a sistemului care rezultă după asamblarea matricelor elementare, este singulară, ceea ce duce la imposibilitatea rezolvării sistemului matricial din relația (61).

Pentru a putea rezolva acest sistem, este absolut necesară impunerea condițiilor de contur, ceea ce va duce la modificarea matricei globale și obținerea unor soluții realiste pentru sistemul analizat.

Aceste condiții permit de fapt determinarea soluției unice pentru ecuația sistemului (21), ceea ce ar echivala în cazul integrării directe a ecu-

<sup>23</sup> [9], [11], [13], [21], [29], [58], [86], [139]

<sup>24</sup> Acest aspect depinde de la un tip de problemă la altul; astfel pentru formularea variațională a ecuației căldurii numai condițiile Dirichlet trebuie aplicate, celelalte sunt cuprinse în funcționala discretizată și deci în ecuațiile la nivel de element.

ației diferențiale a sistemului, cu calcularea valorilor constantelor care apar după această integrare generică.

În mod concret, la rezolvarea ecuației prin metoda elementului finit, acest lucru presupune cunoașterea a priori, sau impunerea unor valori, pentru soluția problemei în câteva dintre nodurile rețelei. Aceste condiții de contur exprimă o realitate fizică incontestabilă. De exemplu pentru o problemă de analiză statică, ele pot fi impunerea unor deplasări nule în nodurile din încastrarea unei grinzi; pentru o problemă de analiză termică aceste condiții pot fi temperaturi cunoscute în anumite puncte ale structurii (temperatura mediului sau o valoare măsurată), sau precizarea fluxului termic cu exteriorul pe anumite zone de contur.

O remarcă importantă în această etapă, este că trebuie făcută distincție între condițiile de contur și încărcările care pot fi aplicate sistemului. Acest lucru este uneori mai greu de identificat la programele comerciale de analiză cu element finit, pentru că terminologia folosită este neunitară, chiar confuză. În această situație, singura soluție este urmărirea ecuațiilor fenomenului și a funcționalei care rezultă din acestea, pentru a aprecia la ce se referă un anumit termen.

După cum se observă din relația (98), care este expresia generală a funcționalei căldurii, condițiile de contur de tip Neumann, Cauchy și schimb convectiv de căldură pe frontiera domeniului<sup>25</sup>, sunt înglobate în ea, deci vor fi cuprinse și în ecuațiile la nivel de element în momentul evaluării formelor integrale discretizate la nivel elementar, fără a mai fi necesară o evaluare ulterioară. Pentru condițiile de tip Dirichlet, (temperatură cunoscută pe o zonă a frontierei domeniului), trebuie aplicat un tratament individual care este posibil numai după asamblarea matricei globale a sistemului, prin una din metodele menționate în literatura de specialitate<sup>26</sup>, care sunt prezentate în continuare.

#### *Eliminarea liniei și coloanei corespunzătoare nodului afectat*

Teoretic aceasta este metoda cea mai simplă de introducere a condițiilor de contur. Pentru a expune mecanismul metodei, vom urmări pe un sistem ipotetic cu trei noduri, pentru analiza unui transfer termic static, cazul introducerii unei condiții de contur de tip temperatură impusă în nodul 1. Expresia matricială a sistemului este cea din relația (62), în relațiile (63) fiind prezentate expresiile dezvoltate ale acestui sistem:

<sup>25</sup> Elemente despre aceste condiții limită sunt prezentate în paragraful 3.2.2

<sup>26</sup> [21], [139], [31], [36], [58], [61]



$$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \quad (62)$$

$$\begin{aligned} k_{11}T_1 + k_{12}T_2 + k_{13}T_3 &= F_1 \\ k_{21}T_1 + k_{22}T_2 + k_{23}T_3 &= F_2 \\ k_{31}T_1 + k_{32}T_2 + k_{33}T_3 &= F_3 \end{aligned} \quad (63)$$

Cum necunoscutele din sistem sunt temperaturile nodale  $T_i$ , iar coeficienții matricei de conductivitate  $k_{ij}$  sunt deja evaluați, în ipoteza impunerii temperaturii  $T_1$  prin mutarea produselor  $k_{i1}T_1$  în membrul drept, sistemul devine:

$$\begin{aligned} k_{12}T_2 + k_{13}T_3 &= F_1 - k_{11}T_1 \\ k_{22}T_2 + k_{23}T_3 &= F_2 - k_{21}T_1 \\ k_{32}T_2 + k_{33}T_3 &= F_3 - k_{31}T_1 \end{aligned} \quad (64)$$

În acest caz, expresia matricială a sistemului (64) este cea din relația (65):

$$\begin{bmatrix} 0 & k_{12} & k_{13} \\ 0 & k_{22} & k_{23} \\ 0 & k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} F_1 - k_{11}T_1 \\ F_2 - k_{21}T_1 \\ F_3 - k_{31}T_1 \end{bmatrix} \quad (65)$$

Pentru rezolvarea sistemului din relația (64) sunt suficiente două ecuații, mai comodă fiind renunțarea la prima ecuație, care corespunde nodului afectat, ceea ce corespunde sistemului

$$\begin{bmatrix} k_{22} & k_{23} \\ k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} F_2 - k_{21}T_1 \\ F_3 - k_{31}T_1 \end{bmatrix} \quad (66)$$

Practic această metodă presupune complicații ale codului programului care rezolvă sistemul și care imprimă rezultatele. Lucrul acesta este impus de operațiile de rearanjare a matricelor sistemului, precum și de păstrarea unei corespondențe între pozițiile corespunzătoare ale nodurilor înainte și după modificarea sistemului, pentru imprimarea corectă a rezultatelor. Aceste aspecte sunt incomode atunci când structura analizată este mare și sunt impuse mai multe condiții de contur pentru noduri care ocupă poziții oarecare în matricele sistemului.

#### *Curățirea liniei și coloanei corespunzătoare nodului afectat*

Metoda pornește de la forma (65) a sistemului, pe care o transformă conform relației (67), deci fără să elimine linia și coloana corespunzătoare nodului afectat. Pentru a simplifica algoritmi de rezolvare ai sistemului și imprimare a rezultatelor, ceea ce duce în anumite situații la scurtarea timpului de rulare, matricele sistemului sunt modificate astfel:

•elementele, din matricea globală, de pe linia și coloana corespunzătoare nodului primesc valoarea zero, mai puțin elementul de pe diagonala principală care primește valoarea 1;

•elementul din vectorul de încărcare, corespunzător nodului, primește valoarea impusă de condiția limită, ceea ce transformă ecuația respectivă într-o identitate de tipul  $T_1=T_1$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & k_{22} & k_{23} \\ 0 & k_{32} & k_{33} \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} = \begin{bmatrix} T_1 \\ F_2 - k_{21}T_1 \\ F_3 - k_{31}T_1 \end{bmatrix} \quad (67)$$

Această metodă păstrează pozițiile originale ale nodurilor în matricele sistemului, nemaifiind nevoie de manevrări și rearanjări ale datelor.

#### *Termen dominant pe diagonală*

Această metodă este cea mai simplă din punct de vedere tehnic și presupune următoarele modificări ale ecuației corespunzătoare nodului din condiția de contur:

•la termenul de pe diagonala principală, se adună un număr  $\alpha$  foarte mare, pentru reprezentare în dublă precizie a matricelor sistemului fiind recomnadabil  $\alpha \approx 10^{100}$ ;

•termenul liber din vectorul de încărcare se înmulțește și el cu același număr  $\alpha$ .

Deși este extrem de simplă, metoda introduce anumite erori pentru valori mari ale variabilei de câmp. Mărimea acestei erori este invers proporțională cu termenul  $\alpha$ , însă valoarea lui este limitată de posibilitatea apariției unei situații de excepție, când produsul între termenul liber din ecuație și  $\alpha$  poate fi în afara intervalului  $1.7 \cdot 10^{-308} \dots 1.7 \cdot 10^{308}$ , adică valorile extreme reprezentabile în dublă precizie.

### **2.2.6. Rezolvarea sistemului algebric echivalent**

Pentru rezolvarea sistemului final de ecuații algebrice, sunt disponibile mai multe metode. Pentru sistemele liniare cea mai comună metodă este metoda eliminării Gauss, combinată cu tehnici de pivotare adecvate, care să reducă din eroarea provocată de substituirea ultimei variabile în ecuațiile precedente. Detalii legate de aplicarea acestei metode sunt prezentate în paragraful 3.2.5 și anexa 1.

Pentru sisteme liniar dinamice, sau tranzitorii sunt necesare scheme de integrare în funcție de timp, situații în care apar și probleme de convergență a soluțiilor spre soluția exactă.

Pentru probleme statice neliniare, sunt folosite tehnici iterative cum ar fi Newton-Rapson. În astfel de situații este necesară aplicarea încărcărilor în incremente succesive, răspunsul sistemului fiind evaluat deseme-na incremental. Neliniaritățile sistemului pot fi de natură geometrică (deplasări mari), sau de material (când proprietățile acestuia depind de variabila studiată).

Cele mai complexe analize sunt acelea unde intervin probleme dinamice sau nestaționare combinate cu neliniarități.

### **2.2.7. Interpretarea rezultatelor**

Pentru ușurarea interpretării rezultatelor, majoritatea programelor comerciale au postprocesoare grafice care prezintă soluțiile problemei într-o manieră ușor de interpretat. Un program de acest gen este prezentat în anexa 2, ușurând interpretarea rezultatelor obținute cu programul de calcul prezentat în anexa 1.

De multe ori sunt necesare mai multe rulări făcute pe modele ce diferă între ele fie prin numărul și dimensiunea elementelor folosite la modelare, fie prin gradul polinomului de interpolare folosit. Ca o remarcă generală ultima soluție asigură o precizie mai bună la același efort de calcul.

### 2.3. Concluzii privind posibilitățile de utilizare a metodei elementului finit

Din cele studiate, rezultă că metoda elementului finit este un instrument puternic de analiză inginerescă, ce poate oferi soluții precise pentru o gamă largă de fenomene care apar în tehnică. Pentru utilizarea eficientă a acesteia, este necesară însă o bună cunoaștere a fundamentelor matematice care guvernează fenomenele studiate.

În cazul concret al studiului efectuat de noi, apreciem modul de soluționare al următoarelor aspecte ca fiind foarte bun în raport cu alte metode echivalente:

- *posibilitatea de rezolvare a ecuațiilor cu derivate parțiale de ordinul doi de tip eliptic, pentru diverse combinații de condiții de contur;*
- *precizia soluțiilor obținute;*
- *posibilitatea de modelare matematică a unor domenii geometrice cu forme complexe;*
- *multiple posibilități de acordare a metodei generale în raport cu cerințele unei probleme concrete, prin utilizarea unor funcții de interpolare diferite precum și a elementelor izoparametrice;*
- *posibilitatea combinării consecutive a studiului mai multor fenomene diferite, cum ar fi transfer termic și deformații elastice ca urmare a câmpului de temperaturi;*
- *ușurința utilizării metodei după stabilirea corectă a mecanismelor fenomenelor reale.*

# CAPITOLUL 3

## MODELAREA CÂMPULUI TERMIC

- *ECUAȚIILE TRANSFERULUI TERMIC*
- *MODELAREA CÂMPULUI TERMIC CU METODA ELEMENTULUI FINIT*
- *CONCLUZII PRIVIND MODELAREA CÂMPULUI TERMIC*

### 3. MODELAREA CÂMPULUI TERMIC

#### 3.1. ECUAȚIILE TRANSFERULUI TERMIC

##### 3.1.1. Transferul prin conducție

##### 3.1.1.1. Ecuația difuziei moleculare

La baza oricărui fenomen termic de conducție stă postulatul căldurii stabilit de Fourier, care exprimă cantitatea de căldură ce traversează o suprafață unitară (sau intensitatea fluxului termic), ca fiind proporțională cu gradientul de temperatură în punctul considerat:

$$q = \frac{1}{A} \frac{dQ}{dt} = -k \frac{\partial T}{\partial n} \quad (68)$$

*unde:  $Q$  este cantitatea de căldură;  $t$  este timpul;  $k$  este coeficientul de conducție termică;  $T$  este temperatura;  $n$  este direcția normalei la suprafața studiată; semnul negativ din relația (68) este o convenție pentru a exprima faptul că sensul pozitiv al fluxului termic este către exteriorul corpului pe direcția normalei.*

Dacă această cantitate de căldură este dependentă de timp, atunci pentru a exprima echilibrul termic la nivelul unui element infinitesimal, când cantitatea de căldură care intră pe o față a elementului nu mai este egală cu cea care părăsește elementul, trebuie luată în considerare variația energiei interne a acestuia:

$$\frac{dU}{dt} = \rho V C \frac{\partial T}{\partial t} \quad (69)$$

*unde:  $U$  este energia;  $\rho$  este densitatea;  $V$  este volumul;  $C$  este căldura specifică.*

În [6]<sup>27</sup>, Bayley, pornind de la postulatul căldurii din relația (68) și aplicând principiul conservării energiei din relația (69), pentru un element infinitesimal, prezintă expresia din relația (70) pentru modelarea transferului căldurii prin conducție:

<sup>27</sup> Bayley, F., J., *Op. cit.*, p.16

$$\frac{\partial}{\partial x} \left[ k(T) \frac{\partial T}{\partial x} \right] + \frac{\partial}{\partial y} \left[ k(T) \frac{\partial T}{\partial y} \right] + \frac{\partial}{\partial z} \left[ k(T) \frac{\partial T}{\partial z} \right] + q_s(T) = \rho C \frac{\partial T}{\partial t} \quad (70)$$

În ecuația de mai sus numai coeficientul de conducție  $k$  este considerat dependent de temperatură, densitatea  $\rho$  și căldura specifică  $C$  fiind apreciate drept constante, ceea ce corespunde celor mai multe situații din inginerie. Pentru formularea de maximă generalitate  $k$ ,  $\rho$ ,  $C$  pot fi exprimate ca funcții de  $x$ ,  $y$ ,  $z$ ,  $T$  iar sursele interne de căldură  $q_s$  ca funcție de  $x$ ,  $y$ ,  $z$ ,  $T$  și  $t$ .

Din ecuația de mai sus se obțin particularizări pentru cele mai cunoscute situații practice după cum urmează:

*Ecuația lui Fourier:*

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad (71)$$

unde  $\alpha = k / \rho C$  este coeficientul de difuzie termică

Ecuația modelează situația unui fenomen tranzitoriu de conducție, fără surse interne de căldură.

*Ecuația lui Poisson*

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{q_s}{k} = 0 \quad (72)$$

Ecuația modelează situația unui fenomen staționar de conducție cu surse interne de căldură.

*Ecuația lui Laplace*

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = 0 \quad (73)$$

Această ecuație modelează situația unui fenomen staționar de conducție fără surse interne de căldură, fiind de fapt cunoscută și sub numele de ecuația de potențial. Ea este comună, ca formulare, cu cea care guvernează alte fenomene (cum ar fi câmpul electrostatic, câmpul magnetic) și stă astfel la baza a numeroase analogii în studiul fenomenelor respective (metoda analogiei electrooptice prezentată în capitolul 1 al prezentei lucrări).

Pentru corpuri supuse la gradienti mari de temperatură, trebuie ținut cont și de modificarea coeficientului de conducție termică în raport cu temperatura. Astfel, în [6], se folosește o relație liniară de tipul  $k = BT + C$

pentru a aproxima variația coeficientului de conducție în funcție de temperatură.

O apreciere similară face și Gârbea în [31], considerând relația (74) ca satisfăcătoare pentru orice intervale de temperatură:

$$k = k_0(1 + b(t - t_0)) \quad (74)$$

unde:  $k_0$  este conductivitatea la temperatura  $t_0$ ;  $k$  este conductivitatea la temperatura  $t$ ;  $b$  este un coeficient care depinde de material.

### 3.1.1.2. Condiții de contur și condiții inițiale

Ecuatiile de mai sus stabilite pentru cazul general, sau particularizate în anumite condiții. Ele nu pot oferi soluții unice decât dacă sunt completate cu condiții de contur și eventual cu condiții inițiale.

În acest caz, condițiile de contur sunt aceleași ca la orice ecuație de tip eliptic și anume:

•de tip Dirichlet

$$T = T(x, y, z, t)|_{S_1}, \quad (75)$$

Ceea ce echivalează cu cunoașterea sau impunerea temperaturii în anumite puncte de pe porțiunea  $S_1$  a conturului corpului.

•de tip Cauchy

$$-k \frac{\partial T}{\partial n}|_{S_2} = q, \quad (76)$$

Ceea ce echivalează cu cunoașterea densității fluxului termic pe o anumită porțiune  $S_2$  a conturului corpului.

•de tip Newmann

$$\frac{\partial T}{\partial n}|_{S_3} = 0, \quad (77)$$

Care sunt de fapt o particularizare a condițiilor Cauchy, pentru cazul adiabatic, deci fără schimb de căldură pe porțiunea  $S_3$  a domeniului corpului.

•condiții de schimb prin convecție

$$k \left( \frac{\partial T}{\partial x} + \frac{\partial T}{\partial y} + \frac{\partial T}{\partial z} \right) + h \cdot (T - T_m) = 0 \text{ pe } S_4 \quad (78)$$

unde:  $h$  este un coeficient de convecție;  $T_m$  este temperatura mediului;  $S_4$  este porțiune din domeniul  $D$  pe care este definită condiția de contur din relația (78);



• condiții de schimb prin radiație

Aceste condiții se pot introduce printr-o relație similară cu (78), unde coeficientul de convecție  $h$  este înlocuit cu un termen de forma:

$$h = \varepsilon \cdot \sigma \cdot (T^2 + T_m^2) \cdot (T + T_m) \quad (79)$$

unde:  $\varepsilon$  este emisivitatea;  $\sigma$  este constanta lui Boltzmann.

### 3.1.2. Transferul prin convecție

Transferul termic prin convecție este propriu fluidelor, fiind cel mai complex mod de transmitere a căldurii. El este o combinație între transmiterea moleculară din fluidul propriuzis, (practic transmitere prin conducție) și transportul energetic creat de mișcarea particulelor de fluid. Dacă mișcarea particulelor este generată predominant de variațiile de densitate datorită gradientului termic, atunci este considerată a fi *convecție naturală*. Dacă predominantă în crearea mișcării este contribuția unui câmp de viteze impus fluidului, este considerată *convecție forțată*.

Din punct de vedere cinematic, al distribuției vitezelor, la acest mod de transmitere a căldurii, apare, în imediata apropiere a frontierei fluidului cu un solid, un așa numit strat limită, la care viteza variază asimptotic de la zero la o valoare oarecare corespunzătoare masei de fluid. În [6]<sup>28</sup> Bayley prezintă, prin analogie cu acest fenomen, existența unui *strat termic limită*, aflat la frontiera fluidului cu un corp solid, în care temperatura variază asimptotic de la temperatura solidului la o valoare oarecare, care este proprie masei de fluid (vezi figura 19).

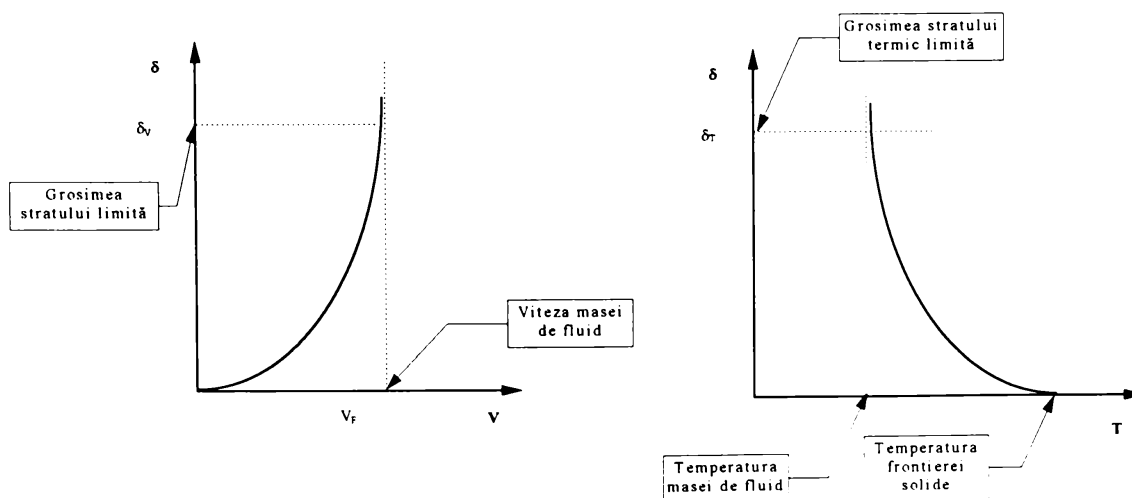


Fig. 19 Stratul limită la transferul convectiv

<sup>28</sup> Bayley, F.J., *Op.cit.*, p. 145

Pentru că transmiterea căldurii în acest strat termic limită se face prin conducție, pornind de la postulatul căldurii, se exprimă un coeficient de transmitere a căldurii  $h$  ca fiind:

$$h = -k \frac{\left(\frac{\partial T}{\partial n}\right)_s}{T_s - T_f} \quad (80)$$

unde:  $T_s$  și  $T_f$  sunt temperaturile frontierei solide, respectiv a fluidului iar ceilalți termeni sunt analogi celor din relația (68).

În cazul cel mai general al fenomenelor tridimensionale, unde curgerea și câmpul de temperatură interacționează, pentru descrierea completă a comportamentului sistemului trebuie rezolvat un sistem de opt ecuații cu opt necunoscute. Necunoscutele sunt: cele trei componente ale vitezelor:  $u$ ,  $v$ ,  $w$ , temperatura  $T$ , presiunea  $p$ , densitatea  $\rho$ , vâscozitatea  $\mu$  și conductivitatea termică a fluidului  $k$ . Cele opt ecuații se obțin în felul următor:

- o ecuație din legea conservării masei:

$$u \frac{\partial \rho}{\partial x} + v \frac{\partial \rho}{\partial y} + w \frac{\partial \rho}{\partial z} = -\rho \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \quad (81)$$

- o ecuație din expresia energiei:

$$\rho C_p \frac{dT}{dt} = T\beta \frac{dp}{dt} + \frac{\partial}{\partial x} \left( k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left( k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left( k \frac{\partial T}{\partial z} \right) + \mu \Phi \quad (82)$$

unde:  $C_p$  este căldura specifică;  $\beta$  este coeficientul de dilatare volumică;  $\mu$  este vâscozitatea;  $\Phi$  este o funcție de disipație ce ține cont de pierderile din frecările interne (de obicei se neglijează).

- trei ecuații din relațiile Navier-Stokes:

$$\begin{aligned} \rho \left( u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} + w \frac{\partial u}{\partial z} \right) &= Fx - \frac{\partial p}{\partial x} + \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \\ \rho \left( u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} + w \frac{\partial v}{\partial z} \right) &= Fy - \frac{\partial p}{\partial y} + \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \\ \rho \left( u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} \right) &= Fz - \frac{\partial p}{\partial z} + \mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right) \end{aligned} \quad (83)$$

- câte o ecuație din relațiile empirice pentru vâscozitate, densitate și conductivitate, care depind de proprietățile fluidului.

### 3.1.3. Transferul prin radiație

Transferul termic prin radiație reprezintă modalitatea de transmitere a căldurii sub formă de emisie electromagnetică, deci fără contact fizic în-

tre corpuri, folosind spectrul infraroșu cu o lungime de undă între 0,7 μm și circa 100 μm. Evoluția acestor fenomene a fost analizată pe baza legilor lui Planck, Wien și Boltzmann, care sunt amintite în continuare, pentru situația unui emițător de tip corp negru:

• *legea lui Planck*, reprezentată grafic în figura 20, exprimă distribuția energetică spectrală pentru corpul negru:

$$W_{\lambda b} = \frac{2\pi hc^2}{\lambda^5 (e^{hc/\lambda kT} - 1)} 10^{-6} \quad [W/m^2 \mu m] \quad (84)$$

unde:  $W_{\lambda b}$  este emitanța corpului negru la lungimea de undă  $\lambda$ ;  $c$  este viteza luminii;  $h$  este constanta lui Planck;  $k$  este constanta lui Boltzmann;  $T$  este temperatura absolută a corpului negru.

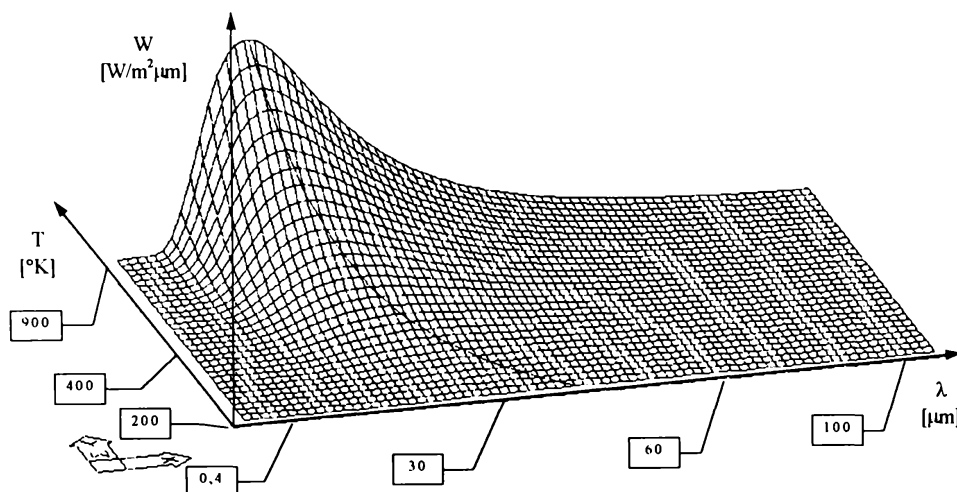


Fig. 20 Variația emitanței cu temperatura absolută și lungimea de undă

• *legea lui Wien*, ce exprimă lungimea de undă la care apare emisia maximă pentru corpul negru:

$$\lambda_{\max} = \frac{2898}{T} [\mu m] \quad (85)$$

• *legea lui Boltzmann*, care exprimă cantitatea de energia radiată de unitatea de suprafață a unui corp la temperatura  $T$ , în unitatea de timp:

$$Q_n = \sigma T^4 \quad (86)$$

sau, care pentru corpul gri devine:

$$Q = \sigma \varepsilon (T_e^4 - T_r^4) \quad (87)$$

unde:  $\sigma$  este constanta lui Boltzmann;  $\varepsilon$  este emisivitatea;  $T_e$ ,  $T_r$  sunt temperaturile absolute ale emițătorului respectiv receptorului.

Din relația (87), se poate observa că emisivitatea este o mărime importantă cu implicații asupra modului în care un corp emite radiații. După

cum arată literatura de specialitate<sup>29</sup>, emisivitatea depinde de temperatura corpului, de unghiul de făcut de direcția de măsurare cu normala la suprafață și de materialul respectiv.

Astfel în figura 21, este prezentată dependența emisivității totale față de temperatură pentru tungsten și platină, iar în figura 22 este prezentată dependența spectrală a emisivității față de  $\lambda$  pentru mai multe temperaturi la platină:

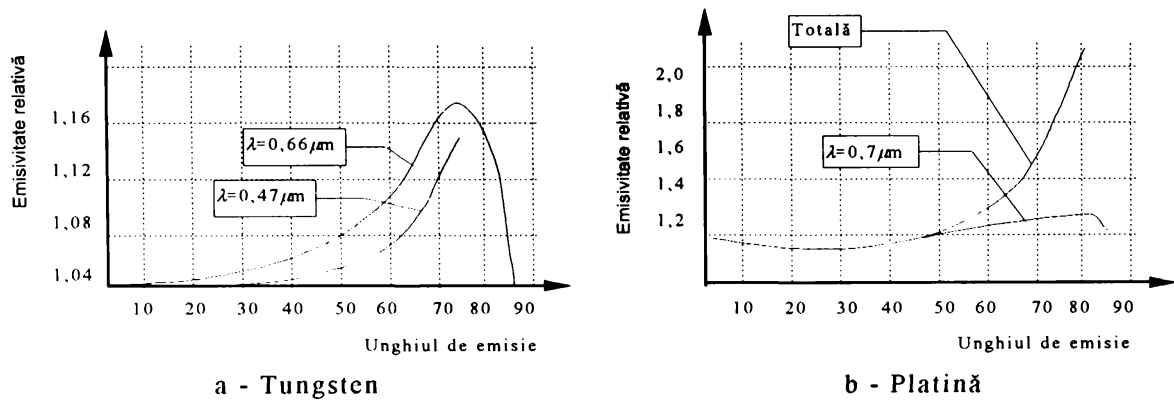


Fig. 21 Dependența emisivității de unghiul făcut cu normala la o suprafață plană

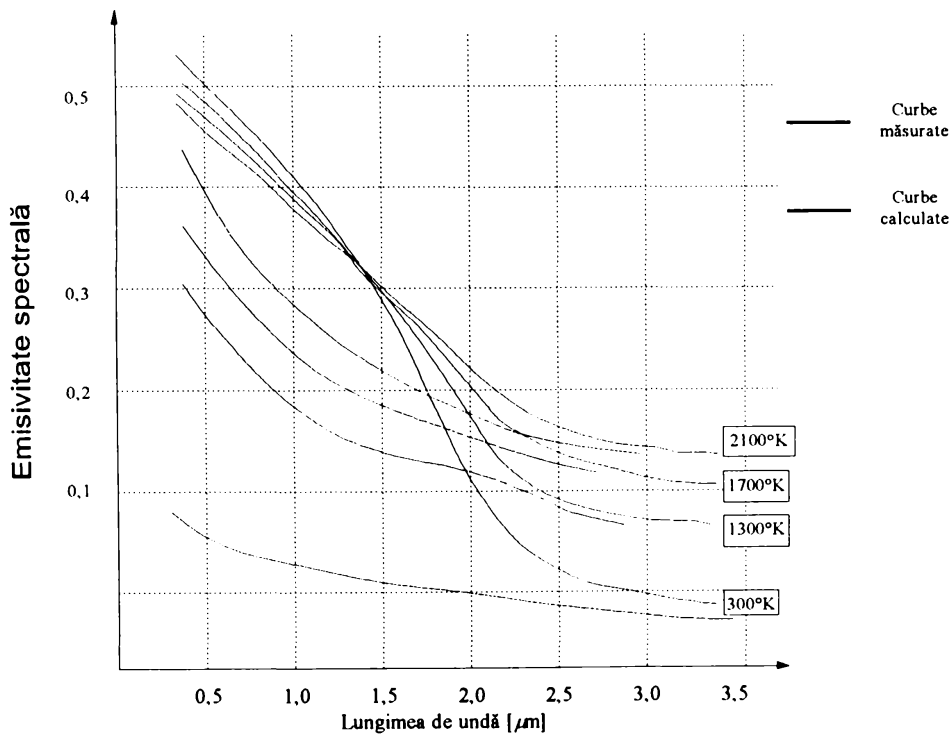


Fig. 22 Dependența emisivității de lungimea de undă pentru diverse temperaturi

<sup>29</sup> [87]Ross, J. *Experiments in heat transfer and thermodynamics*; [151] \*\*\*, *Metals Reference Book*.

## 3.2. MODELAREA CÂMPULUI TERMIC CU METODA ELEMENTULUI FINIT

### 3.2.1. Funcționala pentru transferul de căldură prin conducție

În cele ce urmează este prezentată obținerea funcționalei care stă la baza formulărilor variaționale din metoda elementului finit, pentru transferul prin conducție în cazul plan fără surse interne de căldură.

Pornind de la cazul particular al ecuației transferului prin conducție, prezentat în relația (73), la care se face simplificarea pentru domeniul plan, se obține relația (88)

$$K \frac{\partial^2 T}{\partial x^2} + K \frac{\partial^2 T}{\partial y^2} = 0 \quad (88)$$

Ținând cont de cele prezentate în relația (21), se pot identifica în (88) termenii:

$$\begin{aligned} L &= \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \\ f &= 0 \\ \Phi &= T \end{aligned} \quad (89)$$

Particularizând relația (22) pentru transferul de căldură, se obține:

$$\int_A \nabla^2 T \delta T dA = 0 \quad (90)$$

ținând cont că:

$$\begin{aligned} \nabla^2 T &= \nabla \cdot \nabla T \Rightarrow \\ \nabla \cdot (\delta T \nabla T) &= \nabla(\delta T) \cdot \nabla T + \delta T \nabla \cdot \nabla T \\ &= \nabla(\delta T) \cdot \nabla T + \delta T \nabla^2 T \Rightarrow \\ \delta T \nabla^2 T &= \nabla \cdot (\delta T \nabla T) - \nabla(\delta T) \cdot \nabla T \end{aligned} \quad (91)$$

Înlocuind (91) în (90) rezultă:

$$\int_A \nabla \cdot (\delta T \nabla T) dA - \int_A \nabla(\delta T) \cdot \nabla T dA = 0 \quad (92)$$

Aplicând teorema lui Gauss din relația (93), pentru primul termen,

$$\int_A \nabla u dA = \oint_B \vec{u} \cdot \vec{n} dS \quad (93)$$

unde:  $u$  este înlocuită cu funcția  $\delta T \nabla T$ , obținem:

$$\oint_B \delta T \nabla T \cdot \vec{n} dS - \int_A \nabla(\delta T) \cdot \nabla T dA = 0 \quad (94)$$

Datorită proprietăților operatorului  $\delta$  din relația (95), termenul al doilea se poate transforma conform relației (96):

$$\begin{aligned}\nabla(\delta T) &= \delta \nabla T \\ \nabla(\delta T) \cdot \nabla T &= \delta \nabla T \cdot \nabla T = \frac{1}{2} \delta (\nabla T \cdot \nabla T)\end{aligned}\quad (95)$$

$$\frac{1}{2} \delta \int_A (\nabla T \cdot \nabla T) - \oint_B \delta T \frac{\partial T}{\partial n} dS = 0 \quad (96)$$

Dacă toate condițiile de contur sunt nule, atunci funcționala  $\Pi$ , pentru conducția căldurii în cazul plan, devine:

$$\Pi = \frac{1}{2} \int_A (\nabla T \cdot \nabla T) dA = \frac{1}{2} \int_A \left[ \left( \frac{\partial T}{\partial x} \right)^2 + \left( \frac{\partial T}{\partial y} \right)^2 \right] dA \quad (97)$$

Dacă relația (97) se generalizează pentru cazul domeniului tridimensional și se introduc și termenii corespunzători surselor interne și condițiilor de contur, se obține expresia cea mai generală a funcționalei căldurii, pentru cazul staționar:

$$\begin{aligned}\Pi &= \frac{1}{2} \int_V \left( k_x \left( \frac{\partial T}{\partial x} \right)^2 + k_y \left( \frac{\partial T}{\partial y} \right)^2 + k_z \left( \frac{\partial T}{\partial z} \right)^2 \right) dV - \int_V T q_1 dV \\ &\quad - \int_{S_1} T q_2 dS + \int_{S_2} \alpha \left( \frac{T^2}{2} - T \cdot T_m \right) dS\end{aligned}\quad (98)$$

*unde: prima integrală modelează difuzia moleculară în corpul solid; a doua integrală este pentru sursele interne de căldură; a treia integrală este pentru condiții de contur de tip flux termic impus pe porțiunea  $S_1$  a conturului corpului; a patra integrală este pentru condiții de contur de tip schimb convectiv caracterizat de coeficientul  $\alpha$ , cu mediul a cărui temperatură este  $T_m$ .*

### 3.2.2. Proiectarea elementelor finite

Pentru modelarea transferului termic prin conducție, s-a scris un program care este prezentat în anexa 1. Programul folosește elemente finite triunghiulare liniare izotrope, pornind de la expresia (98) pentru funcționala căldurii. În continuare este prezentată metodologia obținerii ecuațiilor elementare pentru acest tip de element, precum și celelalte etape necesare pentru completarea algoritmilor de calcul.

### 3.2.2.1.Ecuatiile elementelor

Pornind de la relația (41), pentru elementul triunghiular liniar folosit în modelarea conducției căldurii, variația câmpului termic în interiorul elementului se poate aproxima polinomial conform relației (99):

$$T^e = N_1 T_1 + N_2 T_2 + N_3 T_3 = [N] \cdot [T^e] \quad (99)$$

$$\text{unde } [N] = [N_1 \quad N_2 \quad N_3] \quad ; \quad [T^e] = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix}$$

iar  $T^e$  este funcția de aproximare la nivel de element.

Derivând relația (99) în raport cu  $x$  și respectiv  $y$  se obține:

$$\begin{aligned} \frac{\partial T^e}{\partial x} &= \frac{\partial N_1}{\partial x} T_1 + \frac{\partial N_2}{\partial x} T_2 + \frac{\partial N_3}{\partial x} T_3 \\ \frac{\partial T^e}{\partial y} &= \frac{\partial N_1}{\partial y} T_1 + \frac{\partial N_2}{\partial y} T_2 + \frac{\partial N_3}{\partial y} T_3 \end{aligned} \quad (100)$$

Fie  $[T']$  și  $[K_T]$  vectorul și respectiv matricea din relația (101):

$$[T'] = \begin{bmatrix} \frac{\partial T^e}{\partial x} \\ \frac{\partial T^e}{\partial y} \end{bmatrix} \quad ; \quad [K_T] = \begin{bmatrix} K & 0 \\ 0 & K \end{bmatrix} \quad (101)$$

Înlocuind în relația (97)  $\Phi$  cu  $T$  și introducând termenul care modelează condițiile de contur, rezultă relația (102):

$$\Pi = \int_A \left[ \left( \frac{\partial T}{\partial x} \right)^2 + \left( \frac{\partial T}{\partial y} \right)^2 \right] dA + \int_A q T dA \quad (102)$$

În această ultimă relație, ținând cont de (99) și de (101), se obține în (103) o expresie matricială a ei:

$$\Pi^e = I^e = \int_A [T']^T \cdot [K_T] \cdot [T] dA + \int_A q \cdot [N] \cdot [T^e] dA \quad (103)$$

Dacă în relația (100) se fac notațiile din (104) :

$$N_{i,x} = \frac{\partial N_i}{\partial x} \quad ; \quad N_{i,y} = \frac{\partial N_i}{\partial y} \quad ; \quad i = 1..3 \quad (104)$$

se poate exprima  $T'$  din (101) conform (105):

$$[T] = \begin{bmatrix} \frac{\partial T^*}{\partial x} \\ \frac{\partial T^*}{\partial y} \end{bmatrix} = \begin{bmatrix} N_{1,x} & N_{2,x} & N_{3,x} \\ N_{1,y} & N_{2,y} & N_{3,y} \end{bmatrix} \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \quad (105)$$

sau  $[T] = [B] \cdot [T^*]$

În aceste condiții funcționala  $\Pi$  din relația (103) va căpăta expresia matricială:

$$\Pi^* = I^* = \int_A [T^*]^T \cdot [B]^T \cdot [K_T] \cdot [B] \cdot [T^*] dA + \int_A q \cdot [N] \cdot [T^*] dA \quad (106)$$

Aplicând principiul de staționaritate din relația (45) pentru funcționala din relația (106), ceea ce presupune derivarea ei în raport cu  $T_i$ , rezultă expresia din relația (107):

$$\frac{\partial I^*}{\partial T_i} = [T^*] \cdot \int_A [B]^T \cdot [K_T] \cdot [B] \cdot dA + \int_A q \cdot [N] dA = 0 \quad (107)$$

Din expresia (107) prima integrală reprezintă matricea de conductivitate, iar a doua ține cont de aportul surselor interne, ceea ce permite stabilirea expresiei matriciale a metodei elementului finit pentru transferul de căldură prin conducție în condiții staționare:

$$[K]^E \cdot [T]^E + [Q]^E = 0 \quad (108)$$

unde, în acest context, fluxul este pozitiv dacă este orientat spre elementul considerat.

Pentru a putea evalua efectiv termenii matricelor din ecuația elementară (108), se ține cont de izoparametria elementului și prin analogie cu relațiile (49), se scrie expresia vectorului cu funcțiile de formă  $N_i$ , ale căror derivate intră în expresia matricei B din relația (105):

$$[N] = \begin{bmatrix} 1 - \xi - \eta \\ \xi \\ \eta \end{bmatrix} \quad (109)$$

În continuare, pornind de la relația (58), se evaluează matricea iacobiană, derivând componentele  $N_i$  ale vectorului din (109) și rezultă:

$$[J] = \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} \quad (110)$$



Ținând cont că determinantul matricei este dublul ariei  $A$  a triunghiului elementar, inversa acestei matrice este:

$$[J]^{-1} = \frac{1}{2A} \begin{bmatrix} y_3 - y_1 & -(y_2 - y_1) \\ -(x_3 - x_1) & x_2 - x_1 \end{bmatrix} \quad (111)$$

Urmărind expresia lui  $[B]$  din relația (105) se poate modifica (58) ca mai jos:

$$[B] = [J]^{-1} [B]_{\xi} \quad (112)$$

unde:  $[B]_{\xi}$  reprezintă matricea derivatelor funcțiilor de formă raportată la spațiul parametric.

$$[B] = \frac{1}{2A} \begin{bmatrix} y_3 - y_1 & -(y_2 - y_1) \\ -(x_3 - x_1) & x_2 - x_1 \end{bmatrix} \cdot \begin{bmatrix} -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} y_2 - y_3 & y_3 - y_1 & y_1 - y_2 \\ x_3 - x_2 & x_1 - x_3 & x_2 - x_1 \end{bmatrix} \quad (113)$$

În aceste condiții, expresia matricei de conductivitate  $[K]$  devine

$$[K] = \int_A [B]^T \cdot [K_T] \cdot [B] dA = \int_0^1 \int_0^{1-\xi} [B]^T \cdot [K_T] \cdot [B] \det(J) d\eta d\xi \quad (114)$$

Dacă ținem cont și de existența altor fenomene conexe (schimb convectiv și prin radiație pe frontiera domeniului), care sunt cuprinse în expresia funcționalei (98), atunci matricea de conductivitate la nivel de element se obține prin însumarea matricei  $[K_1]^e$  din expresia (115) (provenită din relația (114)), cu matricele  $[K_2]^e$  și  $[K_3]^e$ , după cum urmează:

**• Pentru transfer prin conducție în element**

Cum în expresia (114) termenii sunt constanți în raport cu coordonatele din spațiul parametric, după integrare matricea de conductivitate pentru acest element va fi:

$$[K_1]^e = \frac{k}{4A} \cdot \begin{bmatrix} (y_3 - y_2)^2 & (y_3 - y_2)(y_1 - y_3) & (y_2 - y_1)(y_3 - y_2) \\ + (x_3 - x_2)^2 & + (x_3 - x_2)(x_1 - x_3) & + (x_2 - x_1)(x_3 - x_2) \\ \text{Simetric} & (y_3 - y_1)^2 & (y_1 - y_3)(y_2 - y_1) \\ & + (x_3 - x_1)^2 & + (x_1 - x_3)(x_2 - x_1) \\ \text{Simetric} & \text{Simetric} & (y_2 - y_1)^2 \\ & & + (x_2 - x_1)^2 \end{bmatrix} \quad (115)$$

**• Pentru transfer conectiv pe o latură:**

$$[K_2]^e = \frac{\alpha \cdot L}{6} \cdot [Z] \quad (116)$$

unde:  $\alpha$  este coeficientul de convecție;  $L$  este lungimea laturii elementului pe care are loc convecția; iar  $[Z]$  este o matrice definită ca în relația (117):

$$[Z] = \begin{bmatrix} 2 \cdot z_{11} & z_{12} & z_{13} \\ z_{21} & 2 \cdot z_{22} & z_{23} \\ z_{31} & z_{32} & 2 \cdot z_{33} \end{bmatrix} \quad (117)$$

unde:  $z_{ij}=1$  dacă nodurile  $i$  și  $j$  se află pe latura pe care are loc convecția și  $z_{ij}=0$  dacă nodurile  $i$  și  $j$  nu se află pe latura cu transfer conectiv.

•Pentru transfer prin radiație pe o latură:

$$[K_3]^e = \frac{\varepsilon \cdot \sigma \cdot (T^4 - T_{mediu}^4) \cdot L}{6 \cdot (T - T_{mediu})} \cdot [Z] \quad (118)$$

unde:  $\varepsilon$  este emisivitatea suprafeței;  $\sigma$  este constanta lui Boltzmann;  $T$  este temperatura medie absolută a laturii pe care are loc radiația;  $T_{mediu}$  este temperatura absolută a mediului;  $[Z]$  este matricea din relația (117).

În mod analog se obține și expresia vectorului încărcărilor  $[Q]^e$ , care în cazul cel mai general se compune prin însumarea a patru vectori<sup>30</sup>  $[Q_1]^e$ ,  $[Q_2]^e$ ,  $[Q_3]^e$ ,  $[Q_4]^e$ , corespunzători existenței următorilor factori:

•Aportul surselor interne de căldură:

$$[Q_1]^e = \frac{Aq}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (119)$$

unde:  $A$  este aria triunghiului elementar;  $q$  este încărcarea din sursă termică internă la nivel de element.

•Schimb de căldură prin flux termic impus (condiții Cauchy sau Newmann):

$$[Q_2]^e = \frac{qL}{2} \cdot [U] \quad (120)$$

unde:  $L$  este lungimea laturii pe care apare fluxul termic;  $q$  este valoarea fluxului;  $[U]$  este un vector definit ca în relația (121):

<sup>30</sup> Expresiile acestor vectori se deduc din analiza fenomenelor respective pe care le modelează și se regăsesc în forme asemănătoare în literatura de specialitate [21], [31], [29], [139]

$$[U] = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (121)$$

unde:  $u_i=1$  dacă nodul  $i$  aparține laturii pe care apare condiția limită și  $u_i=0$  dacă nodul nu aparține laturii considerate,  $i=1...3$ .

•Schimb de căldură prin convecție:

$$[Q_3]^e = \frac{\alpha \cdot T_{mediu} \cdot L}{2} [U] \quad (122)$$

unde:  $\alpha$  este coeficientul de convecție;  $T_{mediu}$  este temperatura mediului;  $L$  este lungimea laturii elementului pe care are loc convecția;  $[U]$  este un vector conform relației (121).

•Schimb de căldură prin radiație:

$$[Q_4]^e = \frac{\varepsilon \cdot \sigma \cdot (T^4 - T_{mediu}^4) \cdot L \cdot (T_{mediu} - 273)}{2 \cdot (T - T_{mediu})} \cdot [U] \quad (123)$$

unde:  $\varepsilon$  este emisivitatea suprafeței;  $\sigma$  este constanta lui Boltzmann;  $T$  este temperatura medie absolută a laturii pe care are loc radiația;  $T_{mediu}$  este temperatura absolută a mediului;  $[U]$  este vectorul din relația (121).

### 3.2.3.Asamblarea matricelor sistemului

#### 3.2.3.1.Considerații legate de manevrarea matricelor sistemului

Marea problemă a tuturor programelor de analiză cu element finit este manevrarea memoriei alocate pentru matricile sistemului. Aceste aspecte sunt impuse de volumele mari de date care trebuie manevrate. Astfel pentru o problemă de dimensiuni medii, care folosește 2000 de noduri, cu un singur grad de libertate, sunt necesari teoretic, la o reprezentare în dublă precizie, circa 31 Mocteți pentru stocarea datelor.

Din punct de vedere tehnic, programele folosesc două modalități de alocare a memoriei pentru zonele mari de date: statică și dinamică.

Cele două tehnici de alocare vor fi discutate în continuare pentru situații posibile pe un calculator compatibil IBM PC.

•Programe care rulează în modul real<sup>31</sup>

a)Alocarea statică

<sup>31</sup> Microprocesoarele din familia Intel superioare lui 80386, folosesc două moduri distincte de lucru: modul protejat și modul real, care au implicații asupra vitezei de lucru și asupra modului în care programele folosesc accesul la memorie.

Alocarea statică presupune declararea unei mărimi fixe a zonei de memorie la scrierea programului, care nu mai poate fi modificată ulterior după compilarea acestuia. Compilatorul alocă în zona de date a fișierului executabil această zona fixă și o inițializează cu valoarea zero. Aceasta presupune o limitare apriori a problemelor analizate, la un număr de grade de libertate care nu poate fi depășit, indiferent de resursele fizice ale calculatorului pe care rulează problema.

Dacă programul respectiv a fost compilat să ruleze în modul real al microprocesoarelor Intel, zona de memorie nu poate depăși pentru un vector sau o matrice limita de 64 Kocteți. Aceasta este situația celor mai multe programe ce rulează pe calculatoare personale în sistemul de operare DOS.

Un astfel de program se poate compila cu oricare dintre compilatoarele disponibile în momentul de față pe piață, pentru orice limbaj de programare, însă pentru manevrarea unor volume mari de date sunt necesare tehnici suplimentare. La compilarea programului folosind modele de memorie large sau huge<sup>32</sup>, se pot alocă static pentru date, mai multe segmente de 64 Ko, însă dimensiunea unei singure variabile vector nu poate depăși un segment. Acest lucru presupune distribuirea matricelor sistemului care depășesc valoarea respectivă, în mai multe variabile.

Zona maxim alocabilă static la programele compilate în modul real, este însă și ea limitată, dimensiunea tuturor vectorilor declarați neputând să depășească 1 Mb. La un calcul simplu se vede că folosind reprezentare în dublă precizie pentru elementele matricelor, se pot utiliza matrici globale cu dimensiune maximă 357 x 357, deci maxim 357 de grade de libertate, ceea ce este foarte puțin pentru un sistem de complexitate medie.

Pentru că la analizele statice liniare, matricele sunt simetrice și pentru că multe elemente sunt nule, s-au imaginat tehnici speciale de stocare a matricelor care memorează numai elementele nenule. Pentru aceasta, în prealabil, matricele sunt rearanjate astfel încât elementele nenule să fie concentrate în jurul diagonalei principale pe o zonă cât mai îngustă. Dimensiunea acestei zone se numește lățime de bandă și este determinată de diferența maximă între numerele nodurilor de la același element, calculată pentru toate elementele definite. Pentru a micșora această diferență, se practică o renumerotare a nodurilor, ceea ce introduce complicații suplimentare la partea de cod, pentru că trebuie păstrată și corespondența cu numerotarea inițială.

---

<sup>32</sup> Aceste modele de memorie sunt disponibile folosind limbajul de programare C și condiționează în mod direct dimensiunile zonelor de date și cod pe care le poate cuprinde un program, precum și lucrul cu variabile de tip pointer ce oferă accesul la orice zonă de memorie alocată static sau dinamic.

După rearanjarea matricei se determină linia de demarcație între elementele nule și cele nenule, așa numita sky-line și se rețin indecșii din matricea globală care o determină; în funcție de această linie se rețin numai elementele nenule.

Însă, cu toate aceste artificii, la o matrice simetrică, deci cazul cel mai favorabil, se poate urca maxim la circa 320% din spațiul normal disponibil, ceea ce înseamnă în jur de 1100 de grade de libertate. În această situație, pentru probleme mari, care presupun zeci de mii de grade de libertate, trebuie folosite tehnici de scriere a unor date intermediare în fișiere pe disc, ceea ce va reduce mult din viteza de rezolvare a sistemului și va introduce alte complicații în program.

#### *b)Alocarea dinamică*

Alocarea dinamică a unei zone de memorie presupune că în momentul scrierii și compilării programului dimensiunea zonei este necunoscută. Ea urmează să fie determinată în timpul execuției programului în funcție de valorile altor variabile, în cazul de față numărul de noduri declarate și numărul de grade de libertate pe fiecare nod. La programele compilate în modul real, memoria alocată dinamic este și ea limitată de dimensiunea zonei Heap, afectată de compilator în cadrul fișierului executabil și care nu poate depăși un segment de 64 Ko.

În acest caz avantajul față de alocarea statică este nesemnificativ, pentru că datele necesare depășesc oricum 64 Ko și deci sunt necesare operații de scriere pe disc a unor date intermediare, ca în cazul de mai sus.

#### **•Programe care rulează în modul protejat**

Pentru a realiza programe care să folosească microprocesorul în acest mod sub sistemul de operare DOS, trebuie folosit unul din compilatoarele care pot crea executabile de acest tip, cum ar fi în momentul actual cele ale firmelor Watcom, Zortech, Metaware.

#### *a)Alocarea statică*

Dacă programul a fost compilat să ruleze în modul protejat al microprocesoarelor Intel, se pot alocă vectori care să depășească limita de 64 Ko impusă de funcționarea în modul real al microprocesorului. Acest lucru simplifică numai anumite aspecte de programare, pentru că evită distribuirea datelor, care logic sunt într-o singură matrice, în mai mulți vectori.

Ca și la modul real al microprocesorului această tehnică limitează apriori dimensiunea maximă a unei probleme la cea declarată în momentul compilării. Totodată această metodă va produce un fișier executabil de dimensiuni foarte mari, pentru că zona de date alocată static este inclusă în

acesta de compilator, neputând fi modificată. Pentru probleme care ar depăși posibilitățile zonei declarate inițial, ar trebui folosite practic aceleași mecanisme de la funcționarea în modul real.

#### *b)Alocare dinamică*

La folosirea acestor mecanisme de alocare dinamică în modul protejat al microprocesorului, se poate alocă teoretic o zonă continuă de memorie până la circa 4,2 Gocteți (2 la puterea 32), care este adresa maximă de memorie pe care o poate folosi microprocesorul.

În mod normal, memoria fizică disponibilă la un moment dat este de ordinul a câteva zeci de mega bytes, însă programele compilate în modul protejat folosesc un așa numit DOS Extender, care rezolvă într-un mod transparent pentru utilizator și cererile de memorie peste cea fizic disponibilă. Acest lucru se face cu mecanismele de memorie virtuală, care eliberează memoria fizică, mutând pe disc o zonă de date din memorie, ce nu este folosită momentan.

Din aceste considerente practic, dimensiunea zonei alocabile este limitată de suma dintre memoria fizică disponibilă și spațiul liber rămas pe discul dur.

Deși mecanismele de memorie virtuală folosesc discul aparent ca și utilizatorul care stochează date intermediare, performanțele sunt net superioare pentru că:

- sunt copiate direct zone de memorie fără să mai fie nevoie de operații de acces controlate de utilizator pentru selectarea anumitor date;
- scrierea și citirea datelor pe disc se face direct folosind funcțiile sistem și nu funcții de intrare-ieșire din bibliotecile compilatoarelor, care sunt mult mai lente în special la scrierea datelor cu format;
- programatorul tratează matricea globală ca pe un tot, indiferent că la un moment dat o parte din ea, pe un anumit calculator, într-o anumită conjunctură de execuție, s-ar putea afla pe disc sub controlul mecanismelor de memorie virtuală;

Din analiza prezentată mai sus, rezultă clar că metoda cea mai avantajoasă în condițiile actuale pentru dezvoltarea unui program de analiză cu elemente finite, este folosirea unui compilator pentru modul protejat și a alocării dinamice a zonelor de date, combinație ce oferă performanțe maxim posibile pe un calculator, cu eforturi minime de programare.

#### **3.2.4.Încărcări și condiții de contur**

Pentru programul dezvoltat, având în vedere scopul cercetării de față, s-a considerat că sunt suficiente, ca tipuri de încărcări, sursele interne

distribuite în element. Acest lucru permite modelarea corectă a zonei de aşchiere şi determinarea distribuţiei de temperaturi, prin considerarea sursei termice existente, precizând valoarea debitului surselor interioare la nivel de element, în  $W/m^2$ .

Pentru a obţine rezultate conforme cu realitatea, care să fie reversibile la rularea problemei directe şi inverse<sup>33</sup>, a fost necesară implementarea algoritmilor pentru evaluarea tuturor condiţiilor limită care apar în realitate: condiţii Dirichlet, condiţii Neumann şi transfer convectiv şi prin radiaţie pe o latură a elementului. Detalii despre introducerea datelor pentru rulare sunt prezentate în anexele 1 şi 7.

### 3.2.5.Rezolvarea sistemului

Pentru rezolvarea sistemului algebric echivalent s-au folosit două variante ale metodei Gauss, selectarea variantei făcându-se printr-o linie de comandă din fişierul de intrare:

- *varianta simplificată*, care triangularizează matricea de conductivitate şi face eliminarea succesivă a variabilelor respectând ordinea ecuaţiilor ce rezultă după asamblarea sistemului;

- *varianta elaborată*, ce foloseşte tehnica pivotării totale şi care conform celor prezentate în [52]<sup>34</sup>, schimbă ordinea ecuaţiilor pentru a găsi cel mai avantajos pivot într-o anumită conjunctură.

Varianta simplificată rezolvă rapid sistemul rezultat, dacă problema a fost corect formulată (dacă au fost introduse condiţiile limită care particularizează soluţiile). Pentru configuraţii geometrice distorsionate şi cu multe elemente, este posibil ca folosind această metodă mai rapidă, să apară în procesul de eliminare un pivot nul, ceea ce duce la imposibilitatea rezolvării problemei în această formă.

Faţă de prima variantă, cea elaborată reduce din erorile inerente ce apar la utilizarea unor pivoţi cu valori absolute foarte mici (lucru observabil când sistemul are câteva sute de ecuaţii) şi de asemenea rezolvă problema în orice conjunctură de valori ale elementelor pivot, dacă ea a fost corect formulată.

În situaţia când proprietăţile de material depind de temperatură sau când apar condiţii de contur de tip radiaţie (unde intervine în expresia vectorului încărcărilor şi temperatura medie a laturii considerate, care este necunoscută), s-a utilizat un procedeu iterativ de rezolvare, care foloseşte în iteraţia curentă valorile temperaturilor nodale calculate în iteraţia ante-

<sup>33</sup> Detalii despre problema inversă sunt prezentate în paragraful 5.3.1

<sup>34</sup> Krcyszg.E., *Advanced Engineering Mathematics*, 7<sup>th</sup> Edition, p. 979

rioară. Condiția de încheiere a acestui proces iterativ<sup>35</sup> este ca diferența temperaturilor nodale calculate în două cicluri consecutive, pentru toate nodurile, să fie mai mică decât o valoare introdusă în fișierul de intrare, sau când este depășit un număr impus de iterații.

---

<sup>35</sup> Conform celor prezentate în [21] și [31], acest proces iterativ este convergent necondiționat



### 3.3. CONCLUZII PRIVIND MODELAREA CÂMPULUI TERMIC

Ca o sinteză a problematicii de modelare a câmpului termic, se desprind următoarele elemente esențiale:

- ecuația conducției căldurii cuprinde un termen pentru aportul surselor interne de căldură;

- rezolvarea ecuației conducției prin metoda elementului finit, folosind formularea variațională și elemente finite triunghiulare izoparametrice cu aproximare liniară, este simplificată datorită independenței expresiilor integrale la nivel de element, de coordonatele din spațiul parametric  $\xi$  și  $\eta$ .

- formularea matricială a metodei elementului finit, face posibilă implementarea unor algoritmi de calcul;

- expresia funcționalei care stă la baza formulării variaționale cuprinde toate tipurile de condiții limită posibile, mai puțin cele Dirichlet care trebuie aplicate separat;

- se pot simplifica algoritmi de manevrare a matricelor sistemului, dacă un astfel de program de calcul folosește microprocesorul în modul protejat și alocarea dinamică a memoriei;

- algoritmi implementați pot fi modificați pentru rezolvarea problemei conducției căldurii, când se cunoaște distribuția câmpului termic (funcția  $T(x,y,z)$ ) și necunoscuta este cauza care l-a produs (în cazul studiului de față sursele interne de căldură din procesul de așchiere);

- în urma acestor considerente s-a scris programul de calcul cu element finit, din anexa 1, bazat pe cele prezentate în capitolul 2 și 3;

- pentru programul respectiv s-a scris și un program de postprocesare a datelor, prezentat în anexa 2;

- emisivitatea unui corp afectează calibrarea imaginilor obținute prin termografiere și depinde de natura materialului, direcția de măsură, starea suprafeței și temperatură.

# CAPITOLUL 4

## CERCETĂRI EXPERIMENTALE

- *STRATEGIA DE EXPERIMENTARE*
- *PREZENTAREA STANDULUI*
- *ECHIPAMENTE FOLOSITE*
- *ETALONAREA MĂSURĂTORILOR TENSOMETRICE*
- *CONDIȚIILE DE MĂSURARE A TEMPERATURII PRIN TERMOGRAFIE ELECTRONICĂ*
- *CORECȚIA DE EMISIVITATE*
- *CONCLUZII DESPRE EXPERIMENT*

## 4. CERCETĂRI EXPERIMENTALE

### 4.1. STRATEGIA DE EXPERIMENTARE

După cum s-a arătat în capitolele anterioare, metoda elementului finit este calea cea mai confortabilă pentru a face predicții rapide, precise și necostisitoare asupra distribuției câmpului termic din procesele de așchiere și a mărimii componentelor câmpului deplasărilor ce rezultă din acest câmp termic.

Așa cum s-a prezentat în capitolul 1, sursa termică cea mai puternică și cel mai greu de controlat analitic, este cea din procesul de așchiere, care apare datorită fenomenelor de frecare intercristalină pe măsura formării și degajării așchiei și din frecările externe între sculă, piesă și așchie.

După analiza ecuației generale a transferului termic prin conducție și analiza funcționalei pe baza căreia se obțin ecuațiile la nivel de element, se poate spune că pentru a putea rezolva ecuația căldurii și obține astfel distribuția câmpului termic în regim staționar, sunt necesare cel puțin următoarele date:

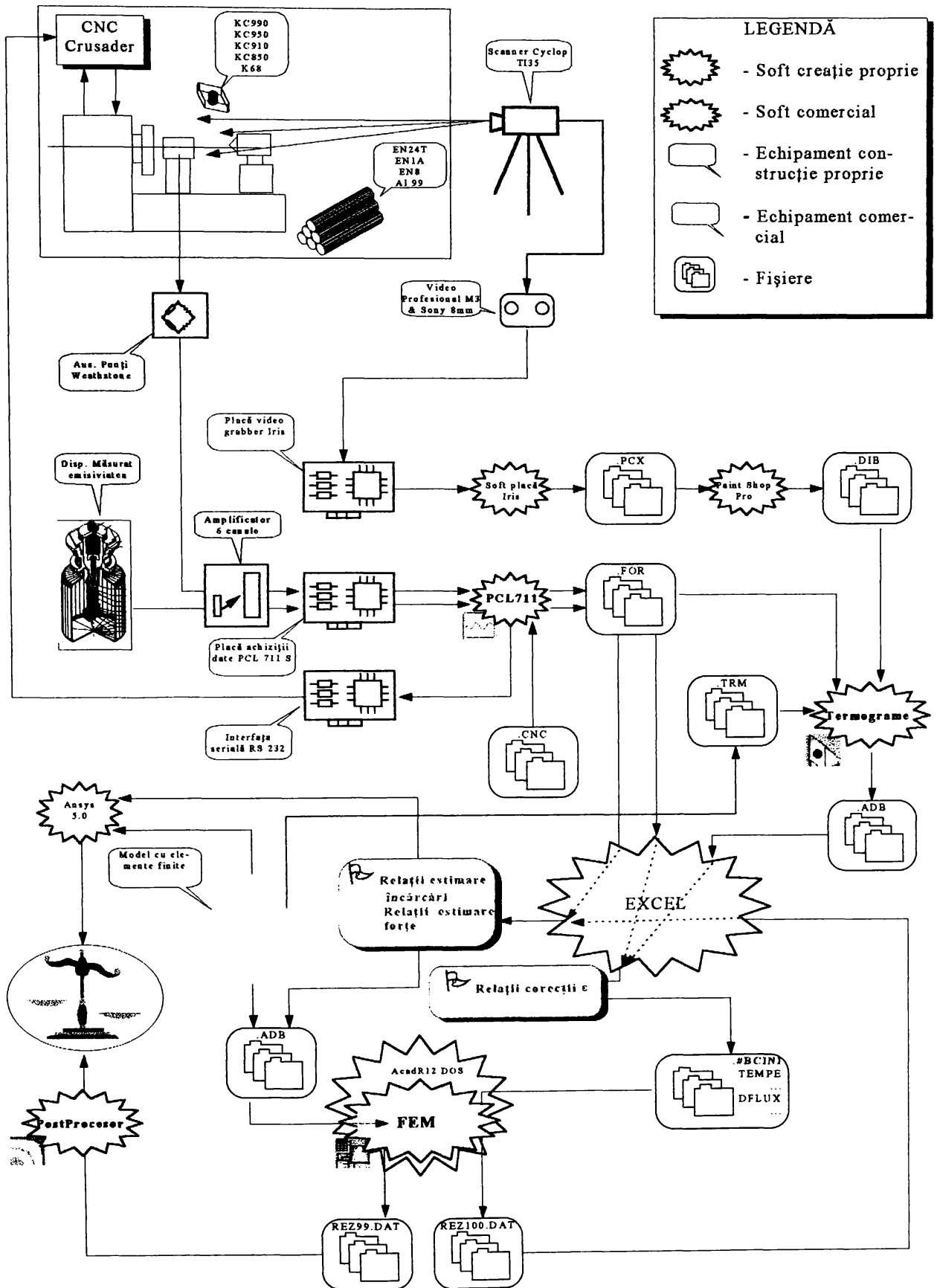
1. *geometria exactă a sistemului modelat;*
2. *coeficienții de conducție pentru materialele implicate în studiu;*
3. *temperatura în câteva din nodurile rețelei, pentru a putea fi folosite ca și condiții limită de tip Dirichlet.*

În condițiile de mai sus s-ar rezolva practic cazul particular corespunzător ecuației lui Laplace pentru conducția căldurii, lucru care este destul de departe de realitate în situația unui proces de așchiere, pentru că nu se ține cont și de factorul cel mai important:

4. *aportul surselor interne de căldură, care completează de fapt vectorul încărcărilor din ecuațiile elementare.*

Acest ultim factor apare, ca dată de intrare la majoritatea programelor de analiză cu element finit sub forma unei surse de căldură distribuită în element, sau eventual concentrată în nodurile din zona de contact a așchiei cu scula. Valoarea acestuia este însă aproape imposibil de estimat rapid, în lipsa unor măsurători termice greoaie și costisitoare, care oricum ar elimina practic necesitatea folosirii analizei cu element finit.

Din aceste motive, s-a considerat utilă determinarea unor relații, care să permită estimarea acestui factor, pentru cazul unor operații de așchiere. Strategia experimentului este prezentată în figura 23:



**LEGENDĂ**

- Soft creație proprie
- Soft comercial
- Echipament construcție proprie
- Echipament comercial
- Fișiere

Fig. 23 Strategia experimentului

Aceasta presupune realizarea următoarei succesiuni de acțiuni:

- obținerea întregii distribuții a câmpului termic, prin termografie electronică, pe o zonă cât mai întinsă în jurul punctului de presiune, pentru diverse combinații de regimuri de așchiere, materiale, plăcuțe; din punct de vedere matematic, aceasta echivalează cu cunoașterea soluțiilor ecuației conducției căldurii;

- realizarea unei rețele de elemente finite pentru problema studiată;

- determinarea temperaturilor nodale și a fluxului termic pe conturul domeniului studiat, din măsurătorile termografice;

- formularea variațională a problemei și construirea matricelor de conductivitate corespunzătoare elementelor rețelei;

- obținerea matricei globale prin asamblarea matricelor elementare de conductivitate;

- completarea vectorului  $[X]$  din ecuația elementului finit, cu temperaturile nodale corectate;

- rezolvarea sistemului de ecuații algebrice ce rezultă din ecuația globală a elementului finit, pentru cazul că necunoscutele sunt în vectorul încărcărilor și nu în cel al temperaturilor cum sunt în cazul normal și cum rezolvă orice program comercial de analiză cu element finit;

- obținerea vectorului cu încărcările care au produs câmpul termic determinat prin termografie;

- rezolvarea unui sistem de ecuații algebrice pentru a afla valorile numerice ale surselor de căldură la nivel de element, cunoscând vectorul încărcărilor care rezultă în mod normal după asamblarea încărcărilor elementare;

- determinarea prin regresie a relațiilor empirice pentru estimarea încărcărilor termice;

#### Observații:

1. rezolvarea sistemului global cu considerarea necunoscutelor în vectorul încărcărilor este posibilă pentru că matricea de conductivitate, obținută din relația funcționalei căldurii, nu depinde decât de coordonatele nodurilor de coeficientul de conducție și de emisivitate;
2. valorile temperaturilor determinate experimental sunt categoric soluții ale ecuației conducției căldurii și deci trebuie să verifice și ecuațiile ce derivă din aplicarea metodei elementului finit;
3. o soluție alternativă ar fi determinarea unei funcții polinomiale care să aproximeze datele experimentale și care să depindă de coordonatele nodurilor producând valoarea temperaturii în punctul considerat;

4. având în vedere că numărul de noduri din rețeaua folosită este destul de ridicat, (104) și gradul polinomului de aproximare va fi mare (în condițiile unor erori acceptabile), ideal fiind ca gradul acestuia să fie cât mai aproape de numărul punctelor de măsură;
5. acest lucru va duce automat la impunerea unor valori foarte mici pentru unii coeficienți ai polinomului (la gradul 6 deja apar termeni de ordinul  $10^{-20}$ ), iar polinomul va avea mulți termeni fiind dificil de manevrat.

Din aceste considerente s-a optat pentru soluția introducerii valorilor măsurate în ecuațiile elementului finit și rezolvarea acestora pentru cazul invers.

## 4.2. PREZENTAREA STANDULUI

Pentru partea experimentală am folosit un stand de determinări compus din echipamentele vizibile în figurile 24, 25, 26 și care sunt descrise în paragraful 4.3:

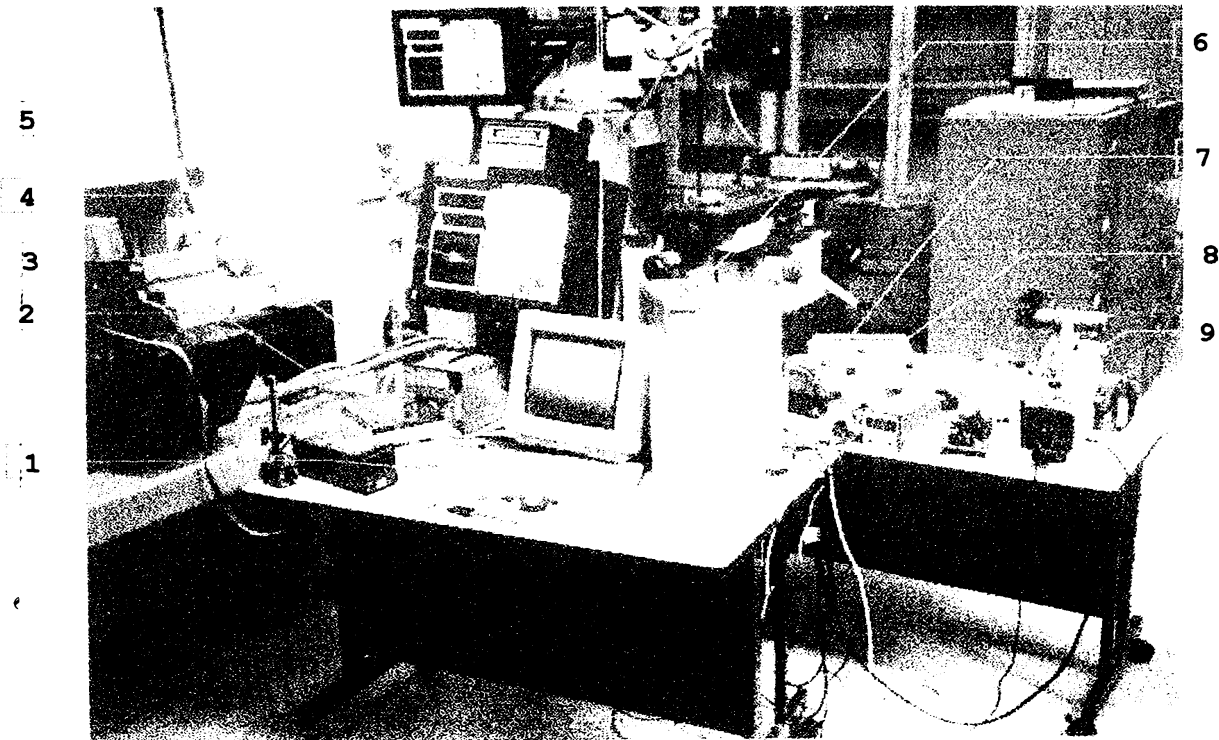


Fig. 24 - Stand experimental, zonă echipamente

- |  |                                   |
|--|-----------------------------------|
| 1. Calculator cu placă achiziții tip PCL711  | 6. Monitor portabil 12"           |
| 2. Amplificator tensometric cu 6 canale      | 7. Video recorder Sony 8 mm       |
| 3. Sursă alimentare amplificator tensometric | 8. Video recorder profesional M3  |
| 4. Comandă numerică Crusader 5L              | 9. Scanner infraroșu Cyclop Ti 35 |
| 5. Suport trepied scanner infraroșu          |                                   |





Fig. 25 Stand experimental, vedere generală zonă de lucru

- |                                       |                              |
|---------------------------------------|------------------------------|
| 1. Strung Harrison 1000 x 300         | 3. Zona echipamente          |
| 2. Sculă cu suport punți tensometrice | 4. Comanda numerică Crusader |

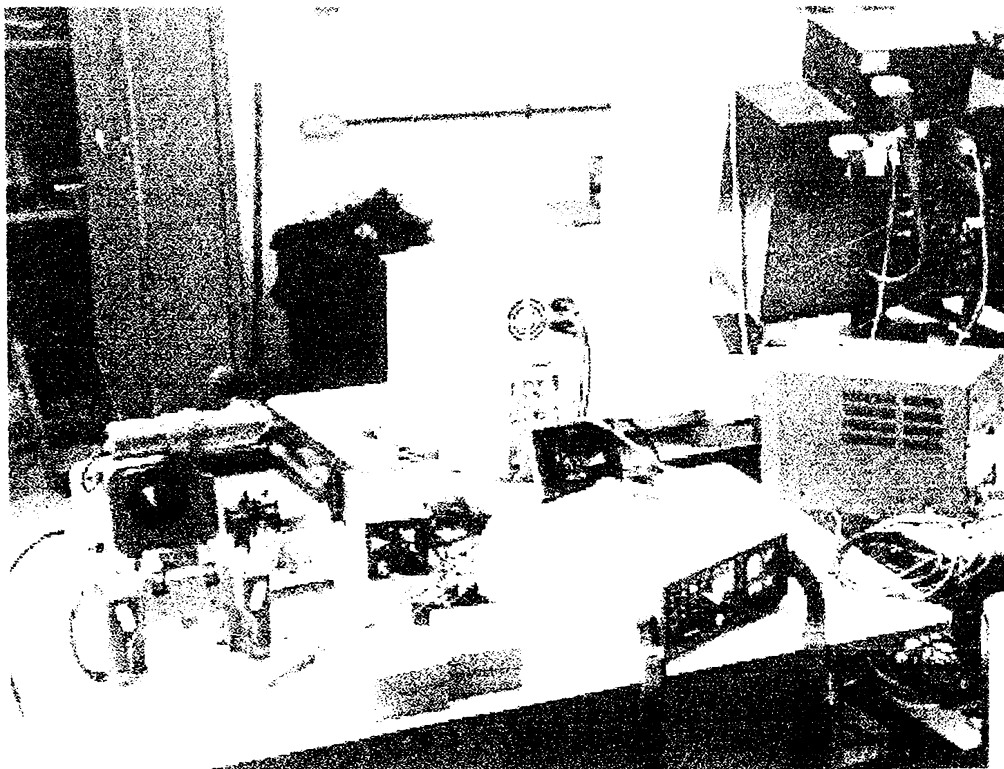


Fig. 26 - Stand experimental - zonă echipamente, (vedere spate)

- |                                  |                                    |
|----------------------------------|------------------------------------|
| 1. Video recorder profesional M3 | 4. Interfață RS232 la CNC Crusader |
| 2. Video recorder Sony 8mm       | 5. Scanner infraroșu Cyclop TI 35  |
| 3. Placă achiziții date PCL711   | 6. Modul sursă-amplificator        |

În cursul acestor experimente am urmărit determinarea forțelor de așchiere și a distribuției câmpului termic, precum și corelarea lor cu parametrii regimului de așchiere în vederea estimării debitului sursei termice în diverse condiții de prelucrare.

### 4.3. ECHIPAMENTE FOLOSITE

Determinările experimentale au fost făcute pe un stand compus din următoarele echipamente:

#### 4.3.1. Mașina unealtă

Strung Harrison 1000 x 300, echipat cu comandă numerică în 2 axe Crusader 5L, cu următoarele caracteristici care au legătură cu experimentul:

- CNC acceptă comunicația pe interfață serială RS232, cu următorii parametri: viteza de transmisie 9600 bauds/sec, 8 biți de date, fără control de paritate;

- CNC și partea de acționare acceptă introducerea avansului în mm/rot, prin funcția G29 folosind variabila LV21.

#### 4.3.2. Scule

Cuțit de strung Kennametal MDJNR-2525K 15K cu dimensiunile informative din figura 27.

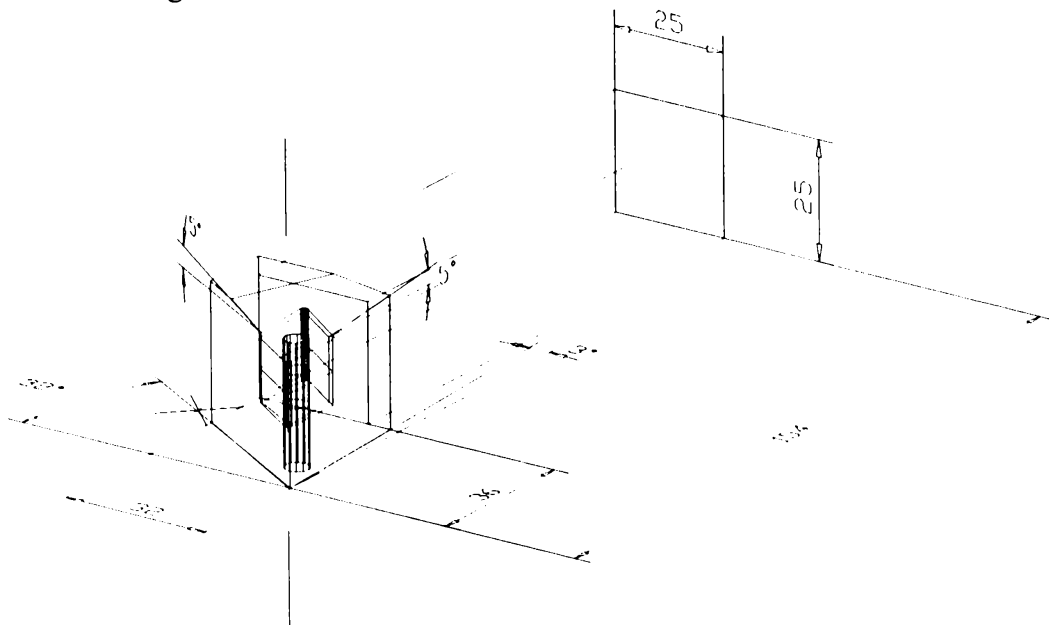


Fig. 27 - Corp sculă, (dimensiuni informative)

Corpul sculei a fost montat astfel încât să rămână o consolă de 55 mm de la vârful sculei până la suprafața suportului portsculă, condiție respectată și la calibrarea răspunsului punților tensometrice pe mașina de încercat la tracțiune Tensile Machine TM 1005.



### 4.3.3. Plăcuțe amovibile

În cursul experimentului au fost folosite următoarele tipuri de plăcuțe cu caracteristicile din tabelele 1 și 2 și cu dimensiunile informative din figura 28.

tab. 1 - Caracteristici plăcuțe folosite la determinări

Tip plăcuță	Compoziție	Domeniu recomandat	Echivalent ISO
KC990	Ceramică multistrat acoperită alternativ cu TiCN, Al <sub>2</sub> O <sub>3</sub> , TiN	Degroșări dure la oțeluri de orice tip, finisări pentru oțeluri inox	K05-K25 M10-M25 P05-P25
KC950	Ceramică multistrat cu substrat îmbogățit în Co, acoperită alternativ cu TiC, Al <sub>2</sub> O <sub>3</sub> , TiN	Degroșări și finisări la viteze mari pentru oțeluri feritice, martensitice și inox	K10-K20 M10-M25 P05-P25
KC910	Ceramică acoperită alternativ cu TiC, Al <sub>2</sub> O <sub>3</sub> , TiN	Oțeluri carbon, oțeluri aliate, oțeluri de scule, oțeluri feritice și martensitice, oțeluri turnate	K01-K15 M05-M20 P01-P20
KC850	Depunere trifazică TiC/TiCN/TiN pe un substrat extradur de Co	Așchiere întreruptă, degroșări la oțeluri carbon, oțeluri de scule, oțeluri inox	M25-M45 P25-P45
K68	Strat dur de WC/Co	Oțeluri inox, materiale neferoase	K05-K15 M10-M20

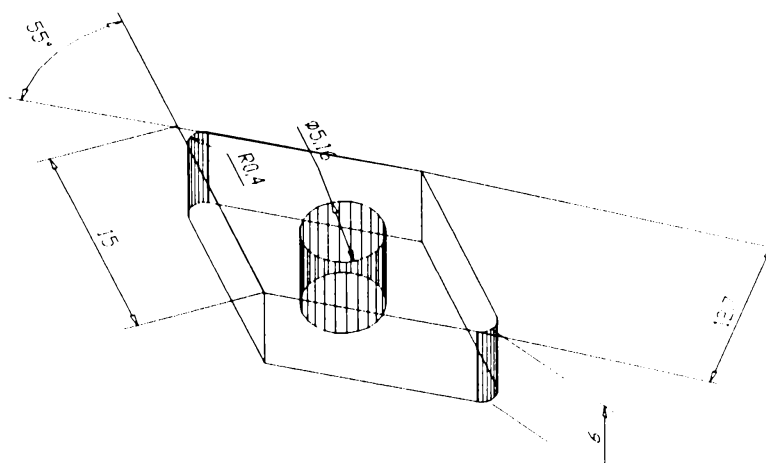


Fig. 28 - Plăcuță (dimensiuni informative)

tab. 2 Parametri geometriei aşchietoare

Plăcuță Parametri	K68	KC990	KC950	KC910	KC850
$\alpha$ [°]	6	6	6	6	6
$\gamma$ [°]	-5	-5	-5	-5	-5
$\chi$ [°]	90	90	90	90	90
$\chi^1$ [°]	35	35	35	35	35
$\lambda$ [°]	0	0	0	0	0
r [mm]	0.4	0.4	0.4	0.5	0.5
f [mm]	0.5	0.5	0.5	0.5	0.6
$\chi'$ [mm]	62.5	62.5	62.5	62.5	62.5
$\mu\alpha$	0.2	0.2	0.2	0.2	0.2
$\mu\gamma$	0.2	0.2	0.2	0.2	0.2

#### 4.3.4. Materiale

La determinările efectuate s-au folosit următoarele materiale cu proprietățile conform tabelului 3

tab. 3 - Caracteristicile materialelor folosite la determinări

		Echiv. DIN	$\sigma_r$ [N/mm <sup>2</sup> ] ]	HB	C %	Mn %	Ni	Cr	Alte elemente %
Al	Al 99.5	Al 99.5	75	22	-	-	-	-	-
Oțel	EN1A	96Mn28	380	103	< 0,15	<1,3	-	-	S < 0,3
Oțel	EN8M	45S20	495	255	0,32	1	-	-	S=0,12÷0,2
Oțel	EN24T	34CrNiMo6	940	375	0,44	0,7	1,7	1,4	Mo=0,35

#### 4.3.5. Scanner infraroșu

Pentru determinarea distribuției temperaturii, am folosit echipamentul Cyclop Ti 35, care poate fi observat în figura 24 poz. 9 și figura 26 poz. 5. Acesta este un scanner infraroșu, cu următoarele caracteristici:

tab. 4 Caracteristicile scanner-ului Cyclop TI35

Limitele temperaturii măsurate	-20°C ÷ 1500°C
Limitele de focalizare	0,5m ÷ ∞
Frecvența cadrelor	25 Hz
Tip detector	HgCdTe cu răcire termoelectrică
Calibrare măsurătoare	internă, cu corp negru
Transmitere imagine	PAL/NTSC/Video compozit, tonuri de gri

tab. 5 - Caracteristicile scanner-ului pentru ferestrele de temperatură

Fereastră de măsurare	Gama de temperaturi [°C]	Acuratețe	Compensare zgomot de fond atmosferic[°C]	Rezoluție [°C]
Low	-20 ÷ 120	±1,25%	-40 ÷ 120	1
Medium	100 ÷ 350	±1,5%	-40 ÷ 350	2
High	300 ÷ 1500	±3,5%	-40 ÷ 1500	5

#### 4.3.6. Placă achiziții date PCL711S

Pentru înregistrarea datelor din măsurătorile tensometrice și cele de profil, am folosit o placă de achiziții de date PCL711S, cuplată pe un conector ISA de 16 biți într-un calculator compatibil IBM PC cu procesor 486 100MHz, 16Mb Ram, 840 Mb HDD.

Ea poate citi date la o frecvență maximă de 10KHz, are 7 canale pentru semnal analog, cu tensiuni între  $\pm 5V$  și un canal pentru semnal digital, care pot fi cuplate pe o plachetă exterioară calculatorului la orice dispozitiv care produce un semnal analog sau digital.

Placa este cuplată logic pe portul 220h, adresă care este modificabilă hard și nu folosește sistemul de întreruperi (IRQ), sau adresarea directă a memoriei (DMA) ci un protocol simplu la nivel de port. Pentru aceasta are rezervate 16 adrese consecutive de la 220h unde se desfășoară dialogul dintre programul care citește datele și partea de hard a plăcii. Aplicația soft care a fost livrată cu placa citea și afișa numai 2 canale simultan, ceea ce era insuficient pentru experiment, unde era nevoie de 6 canale pentru tensometrie. Din acest motiv s-a scris programul din anexa 3, care rezolvă această limitare.

Programul PCL711 este o aplicație Windows, care poate afișa 8 canale simultan, numeric și grafic, datele fiind scrise într-un fișier de ieșire în format text. Frecvența de citire este ajustabilă, mecanismul fiind implementat cu un timer Windows, care este creat și setat pe evenimentul WM\_COMMAND, parametrul ID\_GO din bucla principală a funcției care tratează mesajele ferestrei. Partea de citire a datelor de pe placă este tratată pe evenimentul WM\_TIMER și constă în esență în:

- transmiterea la adresa portului plus valoarea 10 a numărului canalului care trebuie citit;
- setarea pe zero a declanșatorului soft de la adresa bază plus 12;
- citirea în buclă a cuvântului de la adresa bază plus 4 până când octetul superior are bitul 4 setat (atunci datele depuse de placă sunt actualizate);

- prelucrarea datelor citite.

Datorită convertorului analog-digital al plăcii, datele astfel transmise sunt valori întregi în intervalul 0...4095 (folosește transmiterea pe primii 12 octeți dintr-un cuvânt de memorie) și trebuie calibrate în funcție de echipamentul care a transmis datele.

Pentru prelucrarea datelor citite astfel de pe placă și scrise în fișiere text, am folosit programul Excel, care oferă facilități deosebite la lucrul cu date tabelare și reprezentările grafice ale acestora.

#### 4.3.7.Echipament tensometric

Pentru măsurarea forțelor de așchiere s-a folosit un echipament tensometric compus din: 2 Rozete tensometrice trielement la 45°, tip WK-06-240UZ-120-LE produse de Measurement Group și un amplificator adecvat.

S-au folosit două rozete active identice, cu dispunerea din figura 29, pentru că în zona de așchiere condițiile sunt dure, așchiile detașate provoacă destul de des secționarea firelor de legătură, îngreunând măsurătorile.

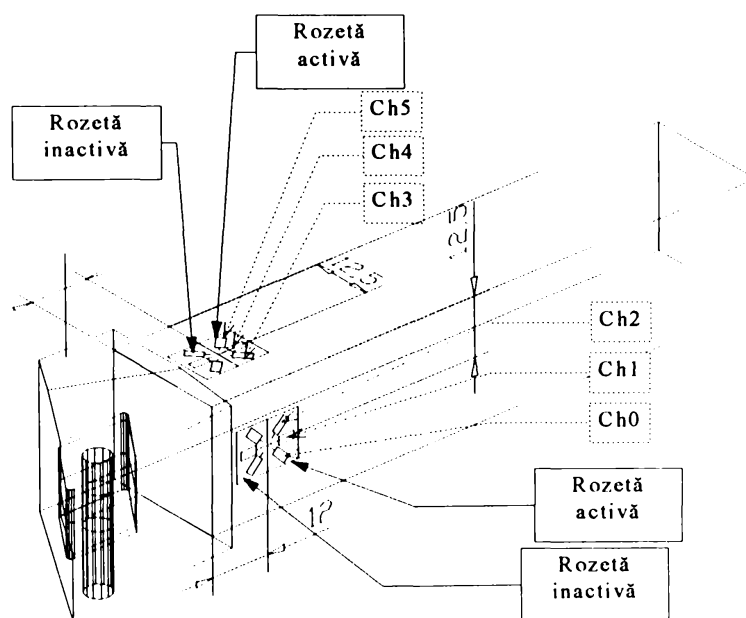


Fig. 29 - Plasamentul rozetelor tensometrice pe sculă

Rozetele au fost cuplate în 6 punți Wheatstone, iar pentru diminuarea efectului temperaturii, s-a folosit, conform recomandărilor din [149]<sup>36</sup> și [19]<sup>37</sup> metoda timbrului inactiv, care presupune realizarea unor conexiuni conform figurii 30:

<sup>36</sup> \*\*\*, Measurement Group, Technical Note TN-504-1 Strain Gage Thermal Output and Gage Factor Variation with Temperature, p.3

<sup>37</sup> Dally, J., Experimental Stress Analysis, p.227

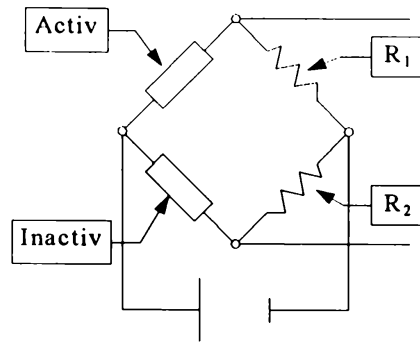


Fig. 30 - Conexiuni electrice în puntea Wheatstone

După cum se arată în [19], plasarea rezistențelor, alimentarea punții și culegerea semnalului conform celor din figura 30 este foarte importantă, pentru că altfel se poate diminua factorul de sensibilitate la jumătate, sau se poate amplifica efectul temperaturii. Condițiile de funcționare a schemei sunt:

- cele două timbre activ și inactiv să fie identice;
- să aibe punctul de măsură pe cât posibil coincident;
- rezistențele  $R_1$  și  $R_2$  să aibă aceeași valoare cu rezistența fiecărui timbru din rozetă (în acest caz  $120 \Omega$ )<sup>38</sup>;
- timbrul inactiv să fie izolat de starea de tensiuni dar în contact termic cu suprafața studiată.

Poziționarea rozetelor ca în figura 29 asigură situarea punctelor de măsură teoretice la o distanță de circa 1mm. Pentru a izola rozeta inactivă de starea de tensiuni, dar pentru a permite transferul termic foarte bun, aceasta din urmă a fost așezată pe un strat de vaselină Heat Sink Compound 55-431-1 pe bază de oxid de zinc și silicon, folosită la accelerarea transferului termic din componentele electronice. Ca adeziv pentru rozetele active am folosit M-Bond 600 pentru a asigura rezistența peliculei la temperatură până la  $260^\circ\text{C}$ .

Ca alte măsuri de reducere a semnalului parazit, provocat de modificarea rezistenței datorită variației temperaturii rozetelor, s-a ales special acest tip de rozete pentru că prin construcție sunt autocompensate termic. Totuși și în aceste condiții, influența temperaturii era vizibilă pe semnalul măsurat.

Stabilitatea și sensibilitatea punții față de câmpuri electrice parazite este foarte afectată de lungimea cablurilor dintre rezistoare. Din acest motiv singura variantă care a dat rezultate, a fost cea a plasării tuturor rezistoarelor cu valori fixe de  $120 \Omega \pm 0.1\%$ , din componența punților Wheat-

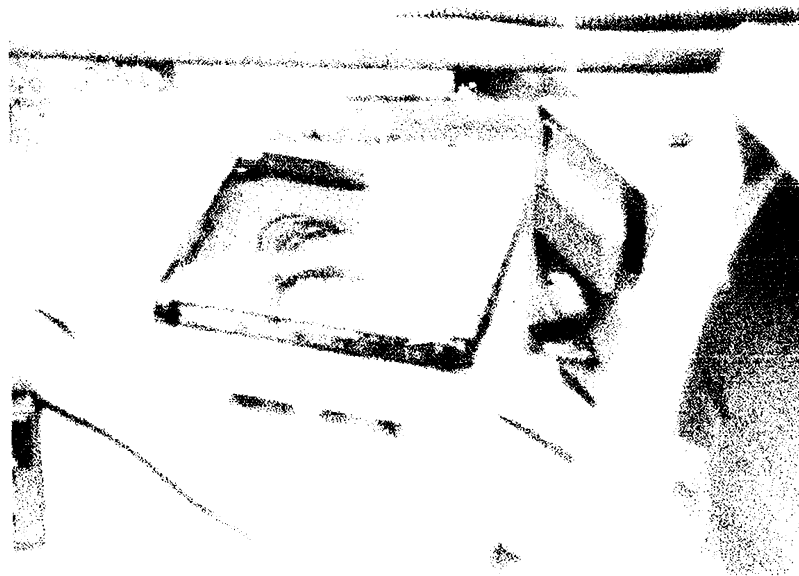
<sup>38</sup> S-au folosit rezistențe precis calibrate cu toleranță de  $\pm 0.1\%$

stone și a conexiunilor cu rozetele tensometrice, într-o cutie pe corpul sculei, așa cum se poate vedea în figura 44.

Valoarea tensiunii de excitare a punților a fost stabilită din diagrama 1 TN502<sup>39</sup> din [142], la 2V. Tensiunea de alimentare este adusă pe același conector cu semnalul util de pe placa amplificatorului, unde se obține cu o diodă Zenner și un tranzistor.

#### *Amplificator operațional cu 6 canale*

Pentru că amplificatorul profesional Tinsley TM08 de care s-a dispus pe parcursul experimentului înregistra datele pe bandă de hârtie și avea o ieșire analoagă de 2V pentru un singur canal, am construit un amplificator care să permită procesarea simultană a tuturor celor 6 canale tensometrice. Amplificatorul are 6 canale identice și a fost construit cu un circuit operațional care permite un factor de câștig de până la 100.000:1. Factorul de amplificare folosit a fost stabilit prin tatonări la o valoare de aproximativ 10,000:1 pentru montajul tensometric descris mai înainte.



Sursa alimentare

Plachetă conectare

Amplificator

Cablu conectare  
sculă

Cablu conectare  
placă

*Fig. 31 - Amplificator tensometric cu sursa de alimentare*

În cazul acestui amplificator, pentru stabilitatea semnalului în gol, s-a dovedit a fi extrem de importantă calitatea sursei de alimentare care asigură tensiunile de  $\pm 5V$ .

<sup>39</sup> Diagrama este prezentată în nota tehnică Optimizing Strain Gage Excitation Levels pag.4 și ține cont de aria timbrului tensometric, valoarea rezistenței și puterea disipată.

### 4.3.8. Organizarea experimentului

Experimentul a urmărit obținerea unor relații de estimare a puterii sursei termice în funcție de parametrii regimului de așchiere  $s$ ,  $v$ ,  $t$  și de forța măsurată. Pentru că plăcuțele de care s-a dispus aveau practic aceeași geometrie a părții așchietoare, nu s-au putut generaliza relațiile și pentru parametrii  $\alpha$ ,  $\gamma$ ,  $\kappa$ .

Programul a constat în aplicarea unui experiment factorial  $3^3$ , (cu variabile  $s$ ,  $v$ ,  $t$ ), pentru fiecare din cele patru materiale folosite, separat pentru câte două tipuri de plăcuțe de așchiere recomandate de producătorul Kennametal. În tabelul 6 sunt prezentate explicit combinațiile folosite la EN24T și plăcuța K68.

tab. 6 Combinații de parametri pentru EN24T

Combinații K68					
Nr.	Avans (s)	Adâncime (t)	v	D	Turația (n)
	[mm/rot]	[mm]	[m/min]	[mm]	[rot/min]
1	0.2	0.5	184.72565	73.5	800
2	0.15	0.5	184.72565	73.5	800
3	0.1	0.5	184.72565	73.5	800
4	0.2	0.5	152.6814	90	540
5	0.15	0.5	152.6814	90	540
6	0.1	0.5	152.6814	90	540
7	0.2	0.5	112.75176	97	370
8	0.15	0.5	112.75176	97	370
9	0.1	0.5	112.75176	97	370
10	0.2	0.6	184.72565	73.5	800
11	0.15	0.6	184.72565	73.5	800
12	0.1	0.6	184.72565	73.5	800
13	0.2	0.6	152.6814	90	540
14	0.15	0.6	152.6814	90	540
15	0.1	0.6	152.6814	90	540
16	0.2	0.6	112.75176	97	370
17	0.15	0.6	112.75176	97	370
18	0.1	0.6	112.75176	97	370
19	0.2	0.7	184.72565	73.5	800
20	0.15	0.7	184.72565	73.5	800
21	0.1	0.7	184.72565	73.5	800
22	0.2	0.7	152.6814	90	540
23	0.15	0.7	152.6814	90	540
24	0.1	0.7	152.6814	90	540
25	0.2	0.7	112.75176	97	370
26	0.15	0.7	112.75176	97	370
27	0.1	0.7	112.75176	97	370

În mod analog au fost realizate și combinațiile pentru celelalte materiale și plăcuțe, cu parametrii din tabelul 7.

tab. 7 - Parametri folosiți la determinări

Plăcuța	Avans	Viteza	Turația	Diametru	Adâncime
	[mm/rot]	[m/min]	[rot/min]	[mm]	[mm]
<b>AL</b>					
K68	0.35	184.73	1200	49	0.5
KC910	0.25	153.56	1040	47	0.7
	0.15	113.1	800	45	0.9
<b>EN1A</b>					
KC850	0.35	184.6	1040	56.5	0.5
KC910	0.3	153.31	800	61	0.7
	0.25	113.66	540	67	0.9
<b>EN8</b>					
KC950	0.25	184.6	1040	56.5	0.5
KC910	0.2	153.31	800	61	0.6
	0.15	113.66	540	67	0.7
<b>EN24T</b>					
K68	0.2	184.73	800	73.5	0.5
KC990	0.15	152.68	540	90	0.6
	0.1	112.75	370	97	0.7

Având în vedere numărul mare de determinări făcute, pentru operativitate, comanda numerică Crusader a fost conectată la interfața serială a calculatorului. Din programul de colectare a datelor de pe placa de achiziții PCL711, s-au operat și transmis direct modificările de parametri (avans, raza corespunzătoare cotei prelucrate și lungimea de prelucrare).

În timpul derulării celor 54 de serii de măsurători pentru fiecare material, au fost captate termogramele corespunzătoare transpuse pe bandă video (ulterior în fișiere bitmap) și fișierele text cu valorile forțelor măsurate de echipamentul tensometric.



#### 4.4. ETALONAREA MĂSURĂTORILOR TENSOMETRICE

Pentru acest lucru s-a folosit o mașină de laborator de încercat la tracțiune tip Tensile Machine TM 1005. Această mașină are comandă computerizată și poate dezvolta o forță maximă de 10 KN. S-a atașat la mașină un dispozitiv intermediar de solicitare la compresiune, pentru a putea aplica forțe similare componentelor radială, tangențială și principală din timpul așchierii și etalona semnalul digital produs de placa de achiziții de date PCL711S.

Etalonarea a vizat obținerea unei caracteristici de corespondență între valorile digitale măsurate pe placă și înscrise în fișier și valoarea unei forțe cunoscute aplicate consecutiv pe cele trei direcții ale componentelor forțelor de așchiere.

Pentru etalonare s-a înlocuit plăcuța de așchiere cu o piesă care să permită aplicarea componentelor forțelor, în poziția corespunzătoare vârfului sculei.



Fig. 32 - Poziționarea sculei pentru calibrare măsurări tensometrice

Forțele aplicate au fost menținute constante timp de circa 15 secunde, cu valorile care se pot observa din tabelele 8, 9 și 10 și au urmărit producerea unei situații apropiate de cele din procesele de așchiere studiate. Datele măsurate de 10 ori pe secundă au fost introduse pentru fiecare solicitare în câte un workbook Excel, unde pentru fiecare forță s-a creat câte un worksheet și câte un grafic.

Detaliat se prezintă în continuare în figurile de la 33 până la 38 graficele obținute pentru cele trei solicitări, cu componentă radială, tangențială și principală, la câteva valori ale forțelor. În figurile respective, pentru claritate, s-a ales cea mai potrivită scară de reprezentare în concordanță cu valorile măsurate.

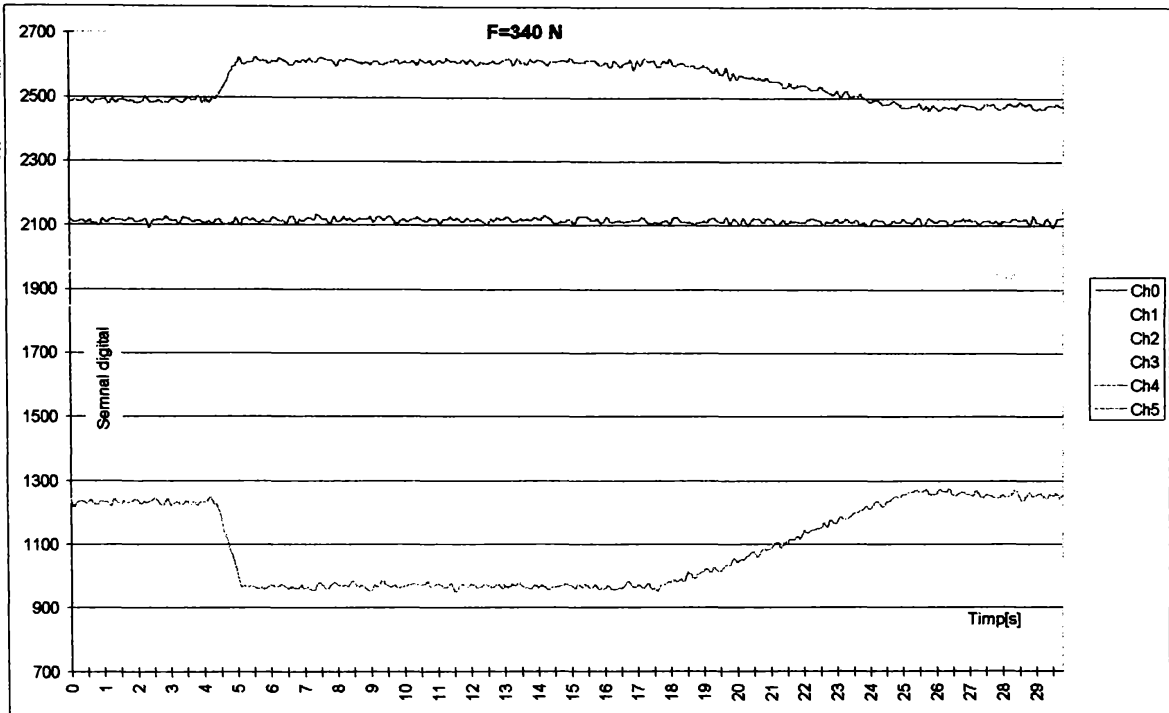


Fig. 33 - Grafice etalonare pentru componentă radială  $F=340\text{ N}$

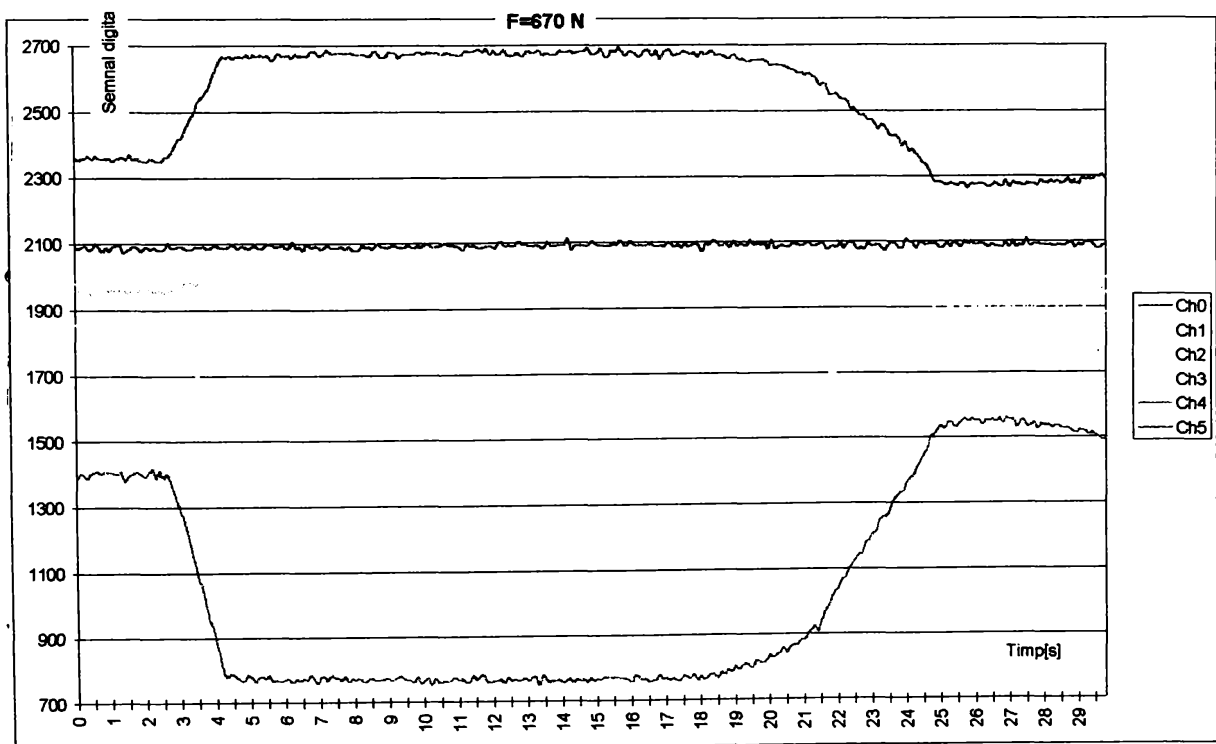


Fig. 34 - Grafice etalonare pentru componentă radială  $F=670\text{ N}$

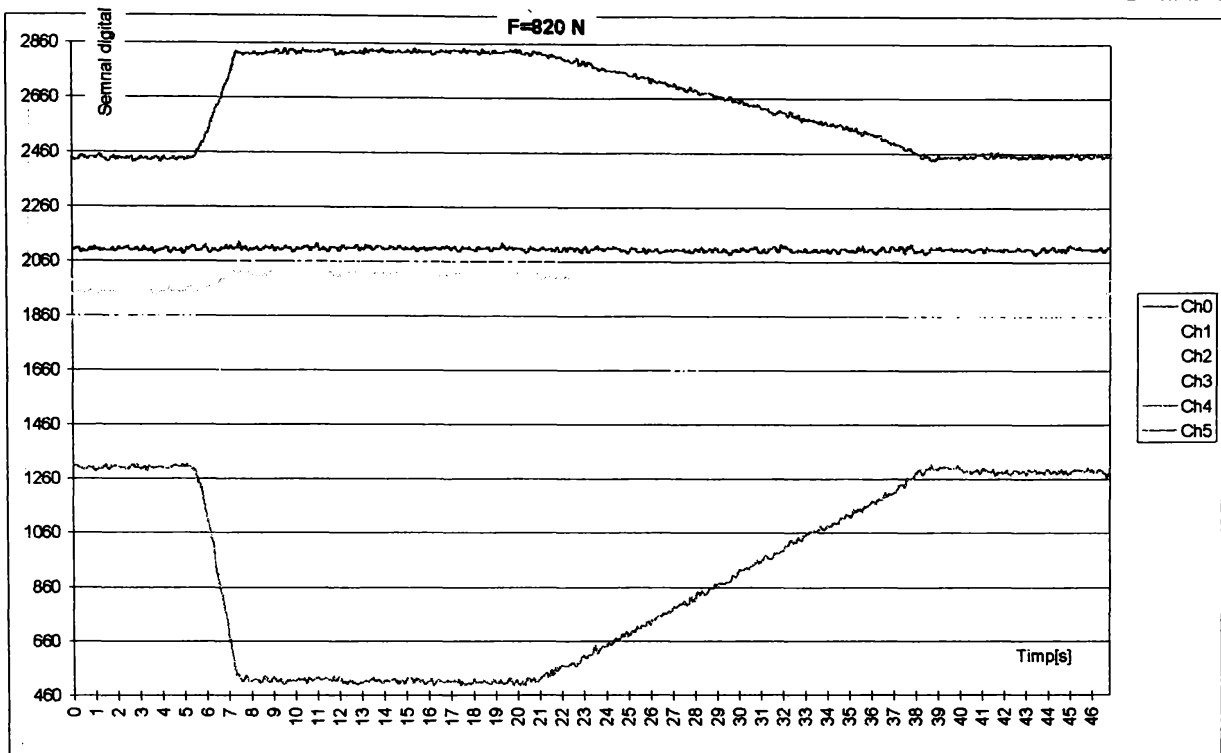


Fig. 35 - Grafice etalonare pentru componentă radială  $F=820N$

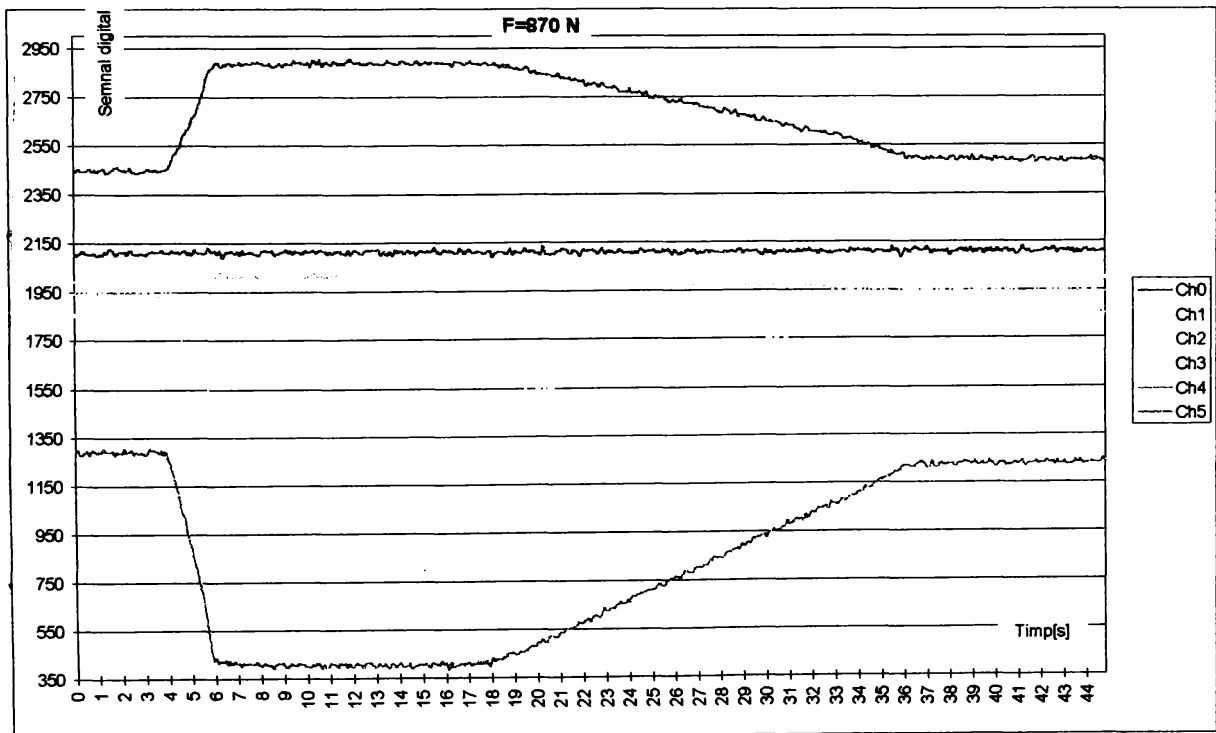


Fig. 36 - Grafic etalonare pentru componentă radială  $F=870N$

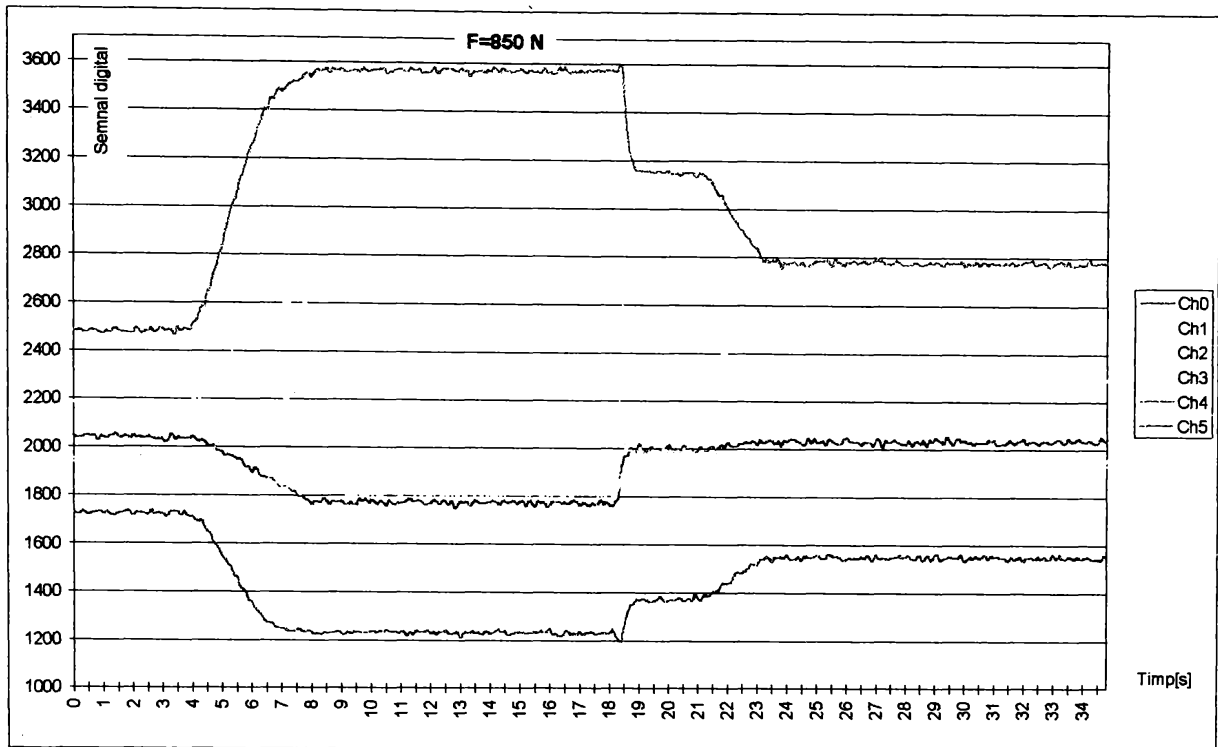


Fig. 37 - Grafic etalonare pentru componentă tangențială  $F=850N$

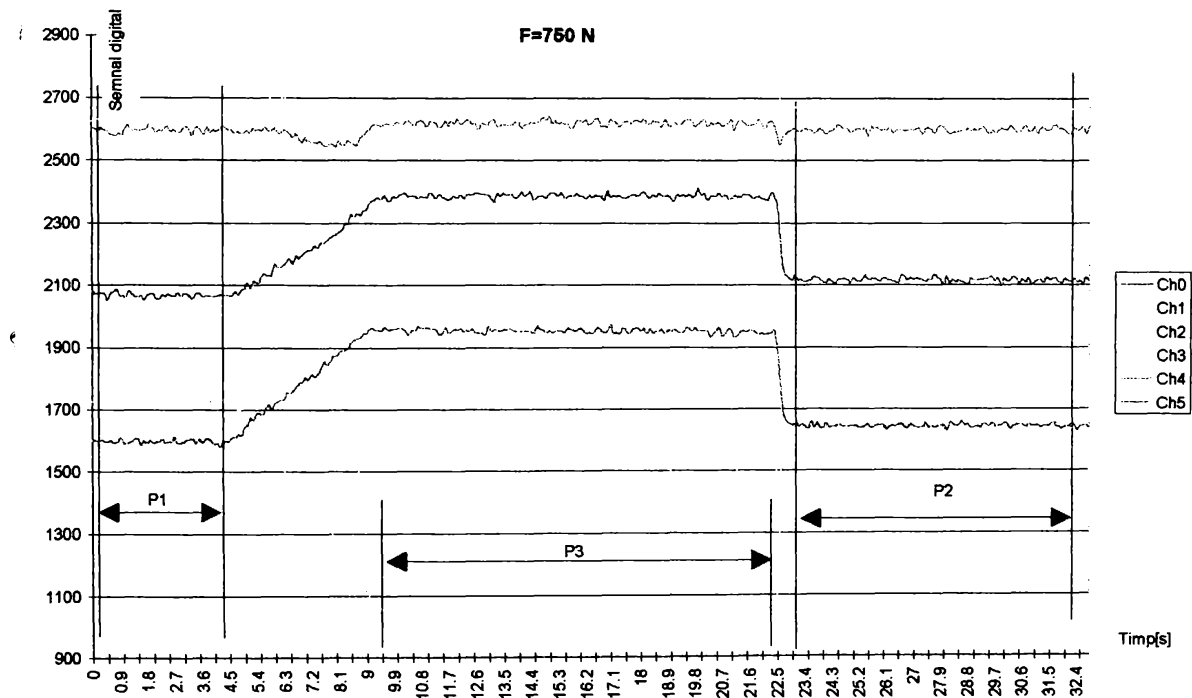


Fig. 38 - Grafic etalonare pentru componentă principală  $F=750N$

În worksheet-urile unde s-au colectat datele de etalonare, s-au făcut și calculele, determinând mediile valorilor pentru cele două paliere fără semnal, P1 și P2 și pentru palierul cu semnal P3. Diferențele valorilor medii corespunzătoare celor două paliere, pentru toate forțele de etalonare, au fost centralizate în alt worksheet unde s-au determinat caracteristicile

liniarizate pentru toate cele 6 canale. Este de menționat că în timpul experimentului, după fiecare prelucrare, datorită influenței temperaturii asupra stabilității punților Wheatstone, au fost necesare readuceri în zero ale fiecărui canal; în calculele de forțe fiind totdeauna introduse ca valori de semnal obținute prin scăderea valorii curente față de offsetul stabilit la începutul prelucrării. În tab 8, 9, 10 sunt prezentate, pe fiecare canal, valorile semnalelor după aplicarea consecutivă a celor trei forțe de etalonare.

tab. 8 - Media valorilor de corespondență semnal-forță radială

Forța [N]	Ch0	Ch1	Ch2	Ch3	Ch4	Ch5
340	18.44616	24.18272	43.38653	-118.271	-276.157	130.2742
360	19.42803	34.65771	52.11648	-151.073	-364.71	169.672
410	20.47477	34.90078	52.63953	-167.64	-403.219	188.6641
470	20.61297	37.38507	55.6717	-169.705	-440.548	207.7175
520	21.26683	40.23598	58.20225	-185.891	-468.714	247.0123
560	21.37179	46.43186	61.80009	-218.5	-514.724	254.9962
620	22.18294	47.98913	67.37012	-229.286	-561.184	268.3338
670	22.57718	57.92598	73.32377	-278.576	-688.439	350.8963
720	22.89632	58.90194	76.5192	-287.157	-717.6	354.1786
770	23.32222	59.93203	76.91909	-292.164	-725.224	368.6809
820	23.5936	65.31234	81.05056	-315.354	-781.315	386.0677
870	23.76683	70.30159	87.0678	-342.766	-853.25	418.5913
990	24.80788	79.91564	95.07096	-391.551	-968.089	469.4061
1080	25.45074	87.6468	102.4935	-431.89	-1066.97	513.5264
1190	26.23645	97.09599	111.5656	-481.194	-1187.82	567.4512
1280	26.87931	104.8272	118.9881	-521.533	-1286.7	611.5715
1370	27.52217	112.5583	126.4107	-561.873	-1385.59	655.6918
1490	28.37931	122.8665	136.3075	-615.659	-1517.43	714.5189
1600	29.16502	132.3157	145.3795	-664.963	-1638.28	768.4437
1710	29.95074	141.7649	154.4515	-714.267	-1759.13	822.3686
1790	30.52216	148.637	161.0493	-750.124	-1847.03	861.5866
1890	31.23645	157.2272	169.2966	-794.946	-1956.89	910.6092
2010	32.09359	167.5354	179.1934	-848.732	-2088.74	969.4363

În figura 39 este reprezentarea grafică a caracteristicilor celor 6 canale tensometrice la solicitare radială, care rezultă după centralizarea datelor în tabelul de mai sus. Pentru ușurința transpunerii datelor măsurate în timpul experimentelor în daN, s-au realizat regresii liniare pentru valorile măsurate și s-au obținut următoarele ecuații:

$$CH0: y = 0.0075x + 17.173$$

$$CH1: y = 0.0794x + 18.153$$

$$CH2: y = 0.0794x + 18.153$$

$$CH3: y = -0.4322x + 27.699$$

$$CH4: y = -1.0657x + 68.654$$

$$CH5: y = 0.4891x - 13.026$$

Unde  $x$  corespunde la forța în [N] și  $y$  corespunde semnalului digital măsurat pe placa PCL711.

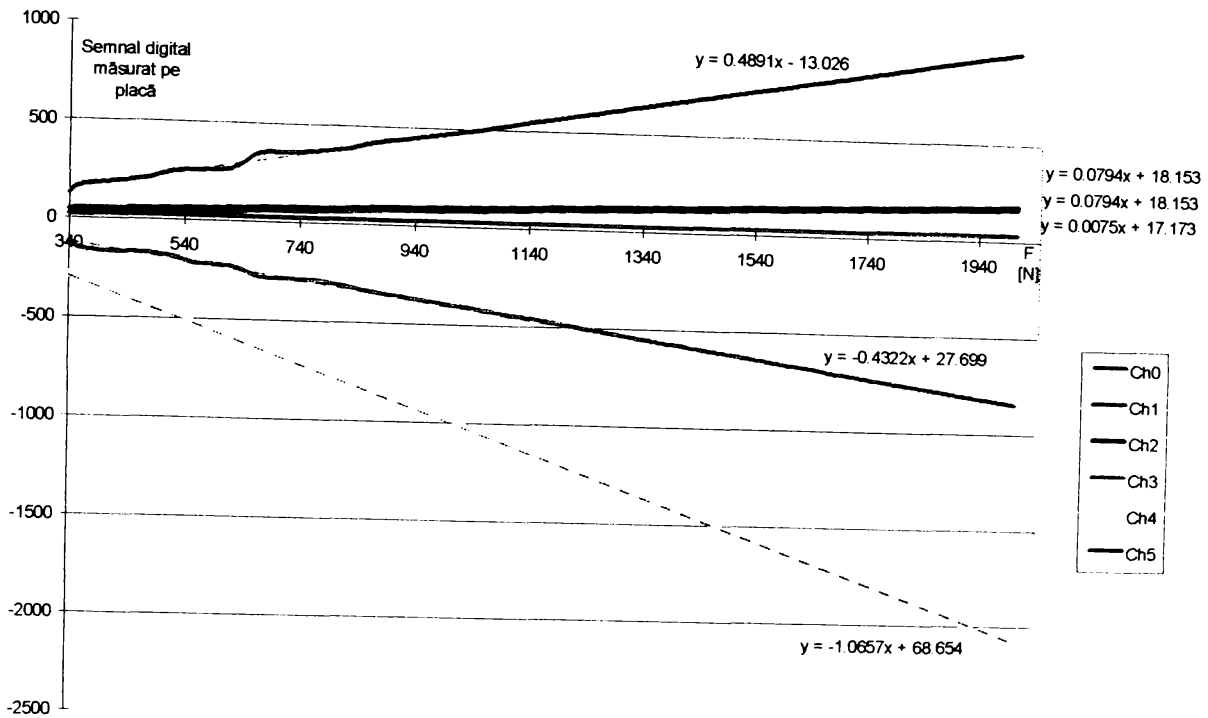


Fig. 39 - Caracteristici de răspuns pentru solicitare radială

În mod analog s-au stabilit caracteristicile de etalonare și pentru componenta tangențială și principală. În continuare sunt prezentate tabelele, reprezentările grafice și ecuațiile liniare pentru caracteristicile celor 6 canale.

Pentru componentă tangențială (din avans)

tab. 9 - Media valorilor de corespondență semnal-forță tangențială

Forța [N]	Ch0	Ch1	Ch2	Ch3	Ch4	Ch5
300	-53.9402	51.78557	-48.4899	254.0594	545.5133	-200.665
350	-64.7715	60.94153	-52.9983	264.8353	576.0981	-220.217
400	-81.724	79.35726	-62.6802	311.0739	686.1249	-271.171
440	-87.3831	113.6834	-56.6276	332.9663	811.4744	-315.444
500	-110.443	118.5996	-73.9913	334.56	813.9119	-342.454
550	-135.48	145.3658	-82.2567	338.9194	824.3213	-361.174
600	-147.106	163.7166	-86.4926	353.2835	828.4776	-372.128
650	-180.058	211.8451	-98.7322	359.2611	835.8639	-373.835
700	-196.474	227.8776	-105.383	371.3237	838.2177	-385.992
750	-216.888	252.2375	-110.476	389.7961	925.051	-395.954
800	-250.196	281.8667	-130.097	404.1695	936.4093	-407.554
850	-261.752	301.1608	-129.833	464.1187	1094.305	-470.144
990	-332.335	380.9714	-160.411	474.5014	1088.623	-476.872
1060	-362.597	417.4837	-171.58	500.4132	1144.701	-502.41

1170	-410.151	474.8601	-189.13	541.1316	1232.825	-542.541
1290	-462.029	537.4525	-208.276	585.5517	1328.96	-586.321
1380	-500.937	584.3968	-222.635	618.8668	1401.061	-619.155
1480	-544.168	636.5572	-238.59	655.8836	1481.173	-655.638
1610	-600.368	704.3657	-259.331	704.0054	1585.319	-703.066
1700	-639.276	751.31	-273.69	737.3205	1657.42	-735.901
1800	-682.507	803.4704	-289.645	774.3373	1737.532	-772.384
1890	-721.415	850.4147	-304.005	807.6524	1809.633	-805.219
1990	-764.646	902.5751	-319.959	844.6692	1889.746	-841.701

CH0:  $y = -0.4333x + 98.567$

CH1:  $y = 0.5162x - 128.19$

CH2:  $y = -0.1657x + 7.6816$

CH3:  $y = 0.3435x + 149.41$

CH4:  $y = 0.7545x + 370.85$

CH5:  $y = -0.3529x - 137.12$

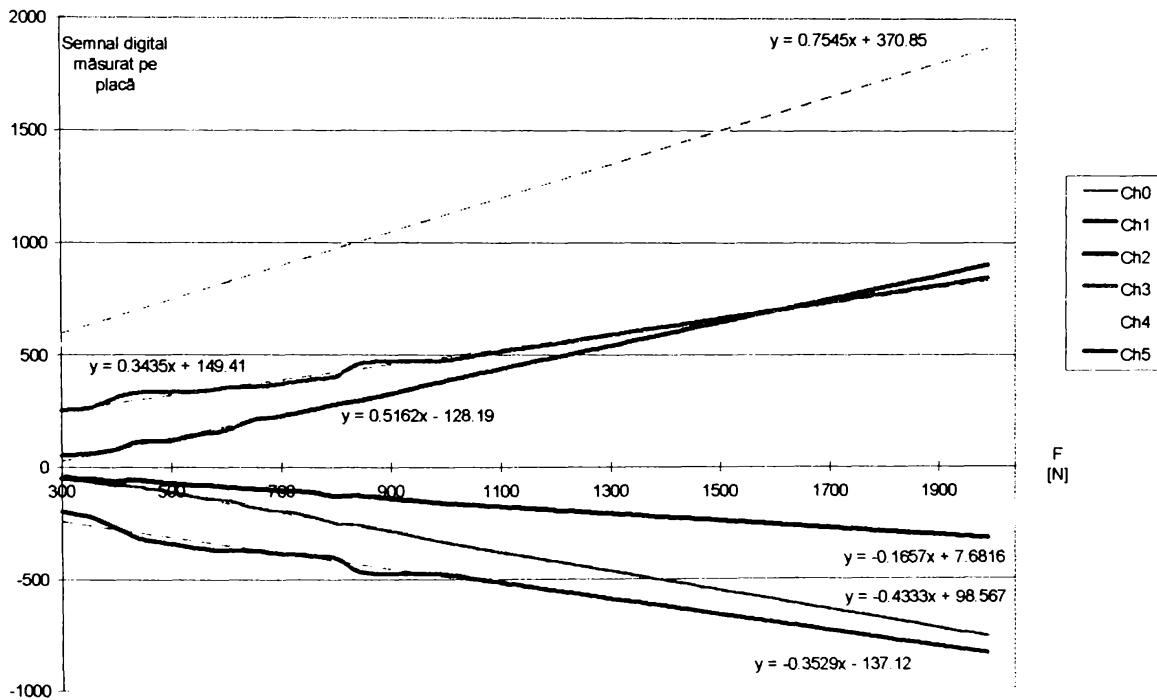


Fig. 40 Caracteristici de răspuns pentru solicitare tangențială

Pentru componentă principală

tab. 10 - Media valorilor de corespondență semnal-forță principală

Forța [N]	Ch0	Ch1	Ch2	Ch3	Ch4	Ch5
300	116.2424	-259.261	-37.173	62.73198	-9.29636	124.8009
340	123.1883	-262.89	-41.5454	66.46943	-2.61283	136.2931
400	140.6324	-312.104	-42.6579	76.12555	2.803146	150.0755
450	144.8708	-330.34	-47.1611	81.52426	3.798601	175.9651
490	165.1963	-371.791	-53.4734	93.3684	6.542283	201.7883
550	223.9497	-492.993	-75.1531	119.565	8.470385	254.0909

590	245.8761	-538.823	-83.9383	129.8435	9.768689	272.8261
650	266.5597	-591.006	-92.9882	145.436	12.53722	302.1311
700	291.9836	-633.774	-101.917	158.0952	12.86184	331.0877
750	294.9603	-635.689	-106.09	166.4523	22.34554	332.4769
810	322.3773	-700.584	-120.84	208.7437	75.49004	345.0866
850	335.4546	-732.929	-130.476	228.0658	88.3247	368.382
970	398.0477	-861.09	-155.063	268.6092	111.8378	419.1282
1100	459.5298	-991.501	-182.87	317.2499	141.3703	479.287
1210	511.5532	-1101.85	-206.398	358.4074	166.3594	530.1907
1290	549.3884	-1182.1	-223.51	388.3402	184.5333	567.2115
1370	587.2235	-1262.35	-240.621	418.2729	202.7071	604.2323
1480	639.2469	-1372.7	-264.15	459.4304	227.6962	655.1359
1610	700.7291	-1503.11	-291.956	508.0711	257.2287	715.2947
1690	738.5642	-1583.37	-309.068	538.0039	275.4026	752.3155
1800	790.5876	-1693.71	-332.596	579.1614	300.3917	803.2192
1890	833.1522	-1784	-351.847	612.8358	320.8373	844.8676
2000	885.1756	-1894.35	-375.375	653.9933	345.8264	895.7712

$$CH0: y = 0.4608x - 42.038$$

$$CH1: y = -0.9784x + 73.705$$

$$CH2: y = -0.2055x + 40.002$$

$$CH3: y = 0.3631x - 79.012$$

$$CH4: y = 0.224x - 106.6$$

$$CH5: y = 0.4519x - 11.484$$

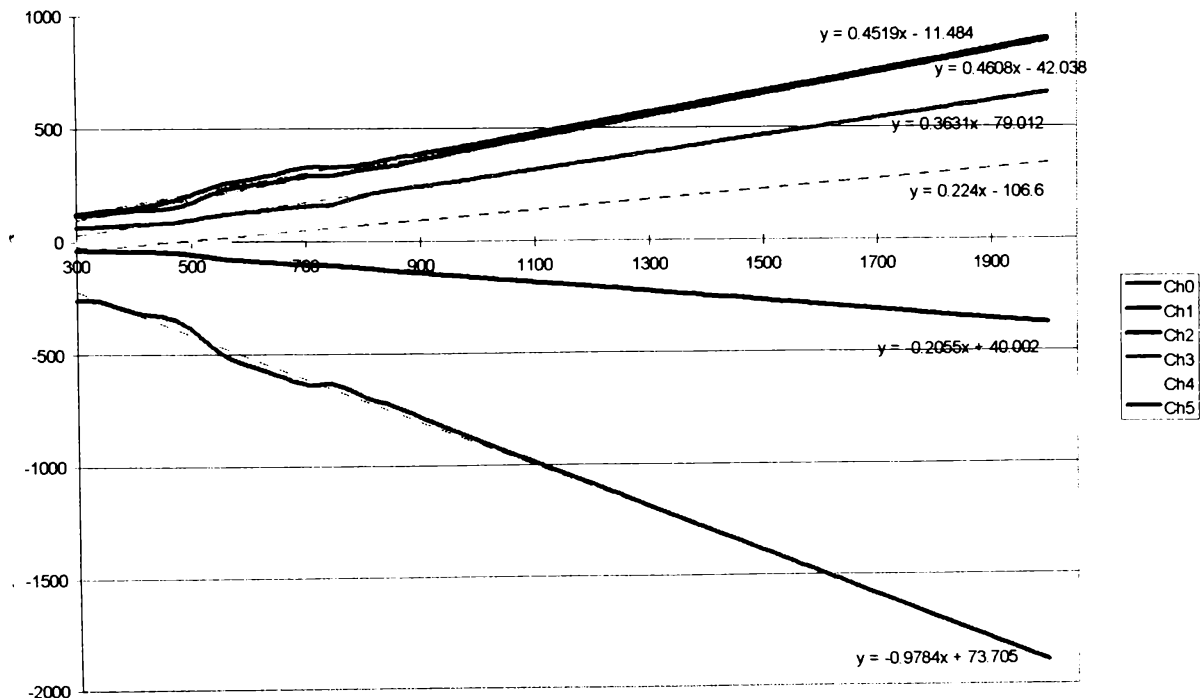


Fig. 41 - Caracteristici de răspuns pentru solicitare principală



#### 4.4.1. Stabilirea ecuațiilor de etalonare

Scopul acestor ecuații este determinarea componentelor forțelor de așchiere, din semnalul cumulat înregistrat care apare în timpul procesului de așchiere pe toate canalele. Acest lucru este necesar pentru că după mai multe tatonări, nu s-a reușit găsirea unei astfel de dispunerii a celor trei timbre tensometrice, încât fiecare dintre ele să ofere semnal la aplicarea unei singure componente iar pentru celelalte semnalul să fie neglijabil.

Analizând direcția de creștere a semnalului digital în raport cu forța, în figurile 39, 40 și 41, se observă că eforturile unitare  $\sigma$  și respectiv alungirile specifice  $\varepsilon$  produse de cele trei componente pe cele 6 canale, au sensuri variabile pe același canal, în funcție de direcția forței.

În figura 42 se pot observa pozițiile punctelor reale de măsură P1, P2, P3 (situate în centrul fiecărui timbru) și poziția punctului teoretic de măsură P, al rozetei. Având în vedere dimensiunile mici ale rozetei folosite, s-a făcut apriori aproximația că indicația fiecărui timbru va corespunde punctului teoretic P și nu punctului real P1, P2, P3.

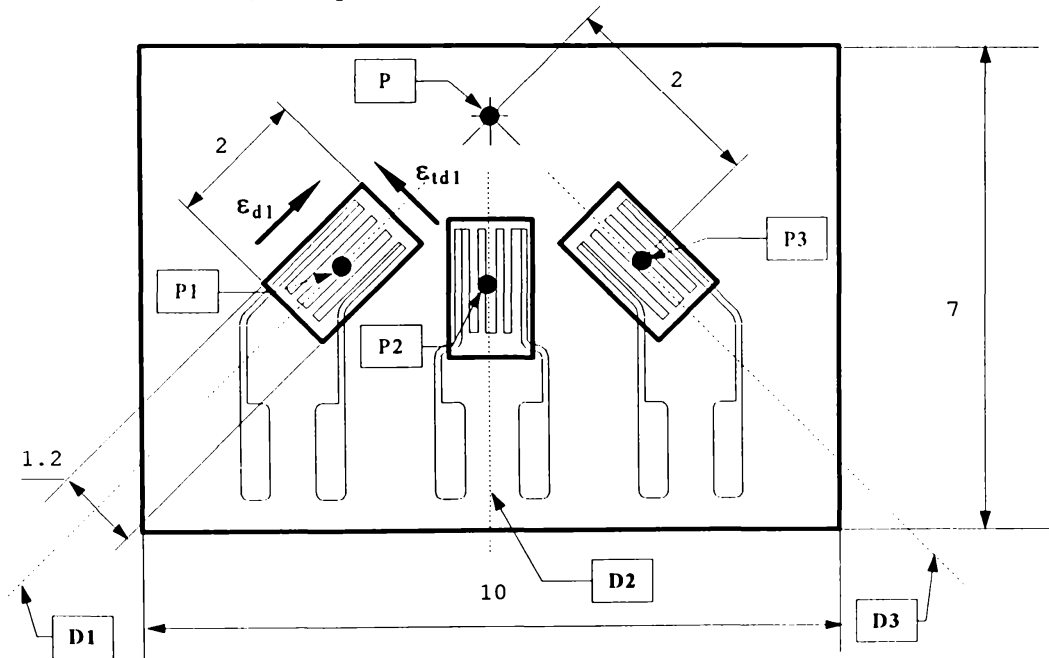


Fig. 42 - Puncte caracteristice la rozeta delta

O forță oarecare va produce o stare de tensiuni în corpul cuțitului, careia îi corespunde câte o stare plană de tensiuni pe cele două fețe unde sunt așezate rozetele tensometrice. Cu rozeta din figura 42, s-a studiat această stare plană de tensiuni, în punctul teoretic P, pe cele trei direcții D1, D2, D3, corespunzătoare axelor de simetrie ale timbrilor ce o compun. Această stare plană de tensiuni, este caracterizată de două eforturi unitare  $\sigma$  și un efort unitar tangențial  $\tau$ , între care există relații ce stau la baza con-

strucției grafice a cercului lui Mohr, (valorile celor trei eforturi unitare fiind dependente de orientarea axelor sistemului de coordonate).

Așa cum se arată în [121]<sup>40</sup>, fiecare timbru este sensibil numai la eforturile unitare  $\sigma$ , considerate de-a lungul axei sale de simetrie. Deci timbrul 1 va da un răspuns la starea plană de tensiuni, numai pentru efortul  $\sigma_{D1}$ , celelalte eforturi care caracterizează complet starea de tensiuni,  $\sigma_{iD1}$  și  $\tau_{D1}$  neafectând indicațiile sale<sup>41</sup>.

Aceasi stare plană de tensiuni, considerată pe direcțiile celorlalte două timbre D2 și D3, a fost urmărită prin indicațiile oferite de răspunsul la eforturile  $\sigma_{D2}$  și  $\sigma_{D3}$ , care apar pentru orientările respective ale sistemului de axe.

De fapt indicațiile oferite de aceste timbre, bazate pe variația rezistenței depuse în raport cu alungirile infinitesimale suferite, sunt direct proporționale cu alungirile specifice  $\epsilon$ , care însoțesc o stare de tensiuni, semnalul digital măsurat pe placa PCL711, pentru canalul  $i$ , fiind de forma:

$$V_i = K \cdot \epsilon_{Di} \quad (124)$$

*unde:  $V_i$  este semnalul digital măsurat;  $K$  este o constantă care depinde de: valoarea rezistenței timbrului, sensibilitatea lui, tensiunea de excitare a punții, factorul de amplificare al amplificatorului operațional, factorul de transformare din convertorul analog-digital al plăcii PCL711, iar  $\epsilon_{Di}$  este alungirea specifică apărută în punctul de măsură pe direcția  $D_i$ .*

În eventualitatea aplicării simultane a mai multor forțe, alungirea specifică într-un punct, pe o anumită direcție, va fi determinată de însumarea algebrică a alungirilor specifice rezultate din aplicarea individuală a fiecărei forțe; astfel semnalul digital din timpul aşchierii va fi o sumă algebrică a semnalelor corespunzătoare celor trei componente ale forței de aşchiere.

Deși la calculul efectiv al valorilor de forțe măsurate s-a folosit numai grupa de canale 0-1-2, canalele 3-4-5 fiind de rezervă, s-au stabilit ecuații de etalonare pentru toate cele șase, după metodologia prezentată în continuare.

După cum s-a determinat anterior, caracteristicile unui canal sunt de forma:

$$\begin{aligned} V_p &= a_i \cdot F_p + b_i \\ V_r &= c_i \cdot F_r + d_i \\ V_t &= e_i \cdot F_t + f_i \end{aligned} \quad (125)$$

<sup>40</sup> Timoshenko, S.,P., Goodier J.,N., *Theory of elasticity*, p.25

<sup>41</sup> Conform documentației [149], sensibilitatea transversală la aceste timbre folosite pe oțel, poate fi neglijată.

unde:  $V_p$  este semnalul digital la forța principală;  $V_r$  este semnalul digital la forța radială;  $V_t$  este semnalul digital la forța tangențială;  $F_p$  este forța principală;  $F_r$  este forța radială;  $F_t$  este forța tangențială;  $a_i, \dots, f_i$  sunt coeficienții liniari determinați în anterior pentru canalul  $i$  ( $i=0\dots5$ ).

Dacă semnalul măsurat la un moment dat pe un canal  $i$  este  $V_i$ , el se compune din însumarea algebrică a celor trei semnale produse de cele trei componente ale forței de așchiere:

$$V_i = V_{i_p} + V_{i_r} + V_{i_t} \quad (126)$$

Înlocuind relația (125) în relația (126) și grupând termenii rezultă:

$$a_i \cdot F_p + c_i \cdot F_r + e_i \cdot F_t = V_i - b_i - d_i - f_i \quad (127)$$

În relația (127) necunoscute sunt cele trei componente ale forței de așchiere, celelalte elemente fiind cunoscute. Pentru determinarea lor, este necesară rezolvarea sistemului alcătuit din trei ecuații de forma (127), compuse pe baza semnalului simultan produs de trei canale. Forma matricială a acestuia va fi:

$$\begin{bmatrix} a_1 & c_1 & e_1 \\ a_2 & c_2 & e_2 \\ a_3 & c_3 & e_3 \end{bmatrix} \cdot \begin{bmatrix} F_p \\ F_r \\ F_t \end{bmatrix} = \begin{bmatrix} V_1 - b_1 - d_1 - f_1 \\ V_2 - b_2 - d_2 - f_2 \\ V_3 - b_3 - d_3 - f_3 \end{bmatrix} \quad (128)$$

sau

$$[C] \cdot [F] = [V]$$

Rezolvarea acestui sistem s-a făcut direct în foile de calcul Excel, unde s-au colectat datele. Pentru aceasta s-a calculat inversa matricei  $[C]$  și s-a evaluat produsul  $[C]^{-1} \cdot [V]$ , folosind direct funcțiile predefinite, pentru toate datele colectate. Cu componentele forței de așchiere astfel determinate s-au calculat forțele folosite în calculele ulterioare.

Astfel, pentru oțelul EN8 cu parametrii din figura 43, după calculul de mai sus, pentru forța de așchiere s-a obținut valoarea de 54 daN.

După cum se observă, peste semnalul produs de forțele de așchiere s-a suprapus și semnalul parazit produs de creșterea temperaturii cuțitului. Pornind de la ipoteza că forța de așchiere este constantă, pentru calculul forțelor s-a luat în considerare porțiunea de diagramă de la începutul și sfârșitul așchierii, unde este vizibil semnalul indus.

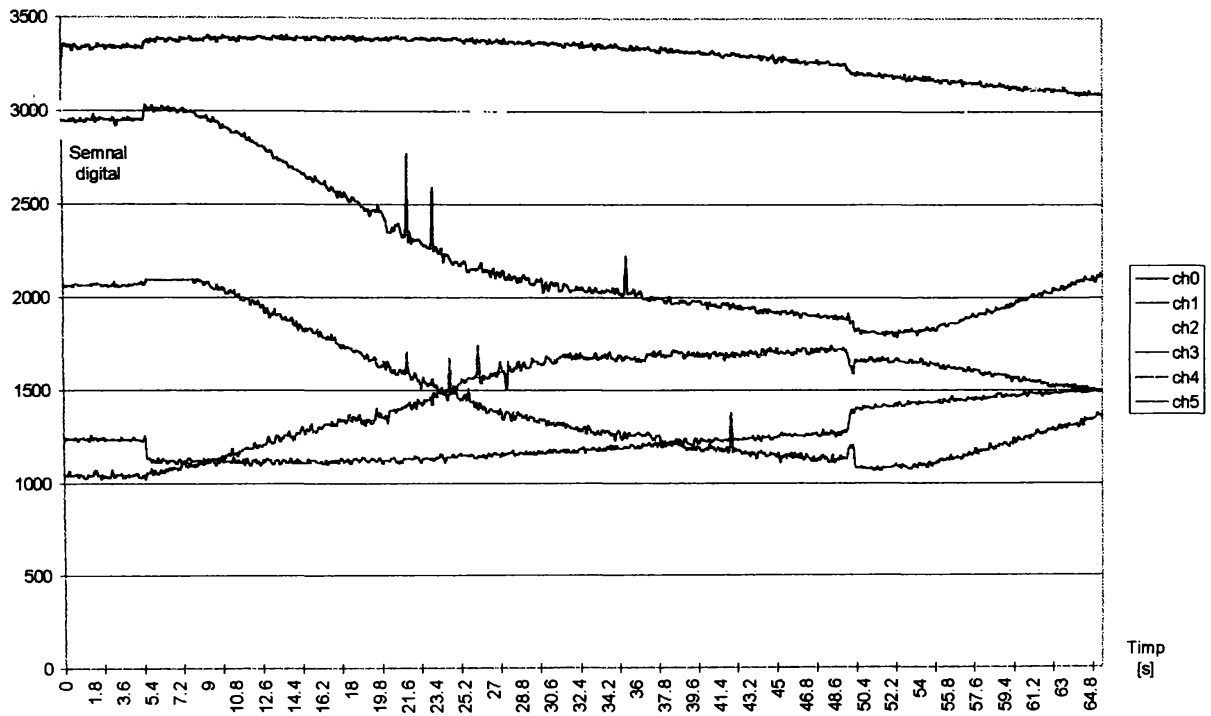


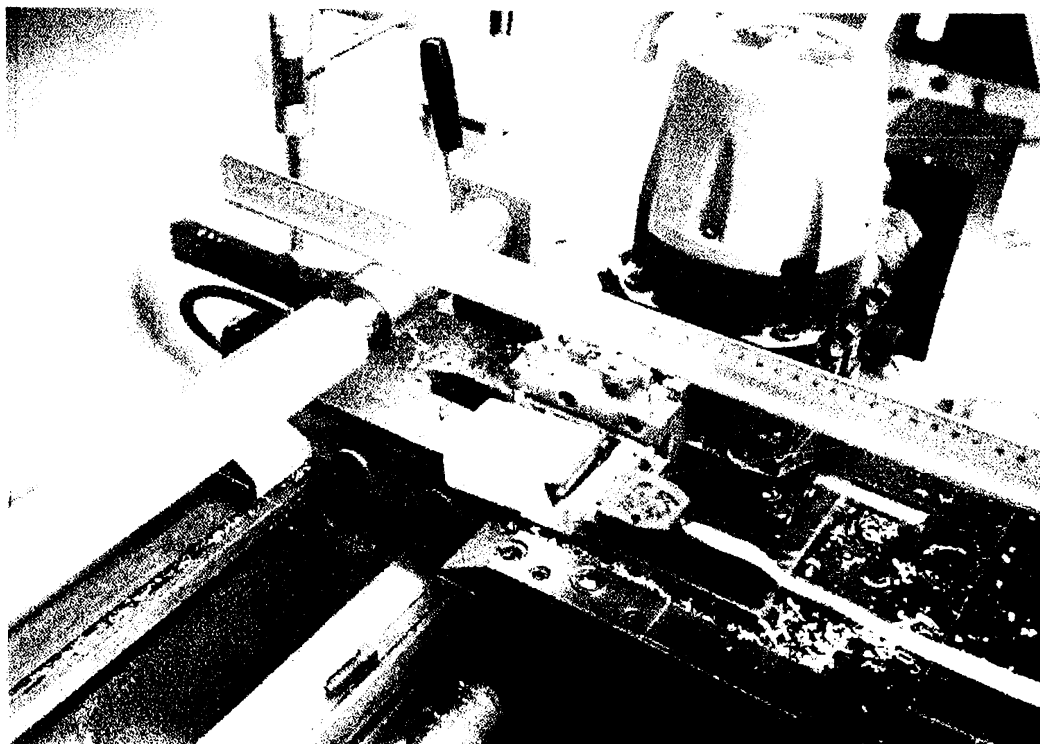
Fig. 43 - Forțe la așchiera EN8 \ 184 \ 0.25 \ 0.5<sup>42</sup>

<sup>42</sup> Unde codificarea are semnificația: v [m/min] \ s[mm/rot] \ t[mm]

#### 4.5. CONDIȚIILE DE MĂSURARE A TEMPERATURII PRIN TERMOGRAFIE ELECTRONICĂ

Pentru a stabili caracteristicile sursei termice, ne-am ocupat, în primul rând de măsurarea temperaturii de aşchiere prin termografie.

În cursul determinărilor s-au folosit trei poziții fixe ale scannerului plasat pe trepid, poziții care pot fi observate în figurile 44, 45, 46.



Cutie punți  
Wheatstone  
pentru  
tensometrie

Conector 25  
de pini  
placă  
PCL711

Cablu de  
legătura

Fig. 44 - Poziția de măsurare 1

Așa cum rezultă din cele prezentate la capitolul 1, paragraful 1.5.3.3, imaginea termică obținută prin măsurarea radiației emise de zona de aşchiere, trebuie corelată cu valorile emisivităților corpurilor implicate.

Conform celor prezentate în capitolul 3 paragraful 3.1.3, la transferul căldurii prin radiație, emisivitatea este însă dependentă de materialul corpului, de forma suprafețelor respectiv unghiul de măsurare făcut cu normala la suprafață și de temperatură.

În condițiile experimentului efectuat am încercat să determinăm cât mai precis posibil variația emisivității, pentru plăcuțele de aşchiere, în funcție de parametrii amintiți. Pentru celelalte corpuri din zona de aşchiere, (corpul cuțitului și piesă), s-au folosit valorile recomandate în tabele pentru materialele respective. S-a făcut această distincție pentru că:

- dintre toate corpurile implicate, plăcuțele au temperatura cea mai mare și ca urmare, variațiile emisivității pot fi destul de mari;

•în momentul efectuării experimentului nu existau informații despre caracteristicile de emisivitate pentru aceste materiale;

Din aceste motive, pentru cazul emisivității plăcuțelor, s-a conceput și construit special, o instalație de măsurat emisivitatea care este descrisă în continuare în paragraful 4.6.

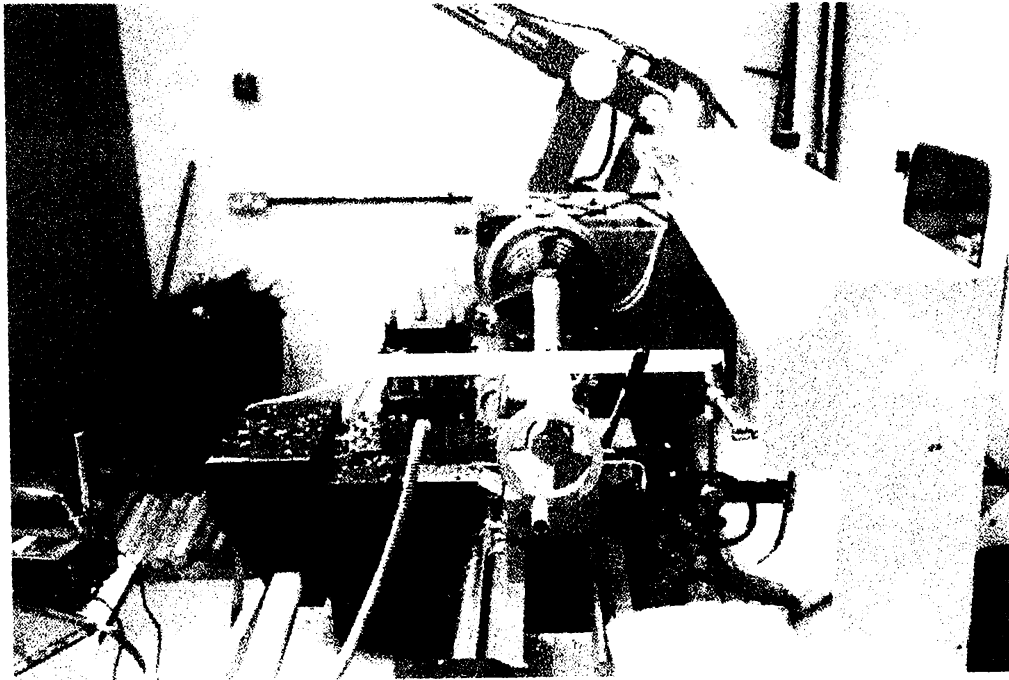


Fig. 45 - Poziția de măsurare 2

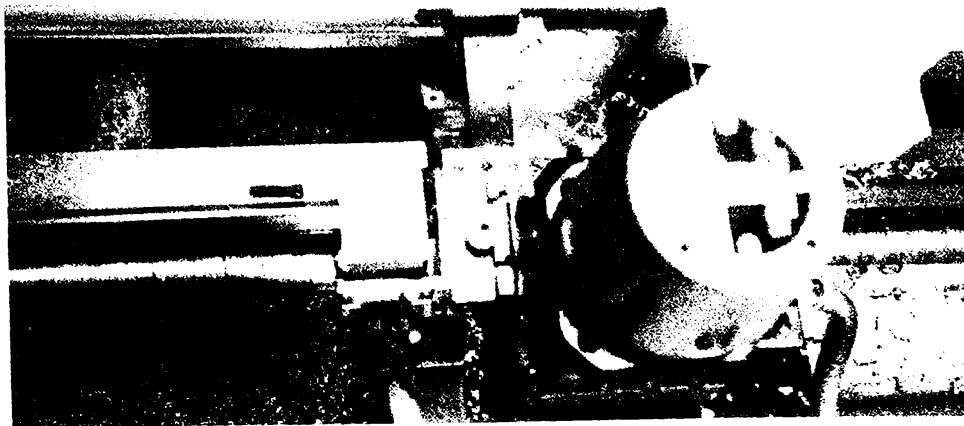


Fig. 46 - Poziția de măsurare 3

## 4.6. CORECȚIA DE EMISIVITATE

### 4.6.1.Principiul instalației de determinat emisivitatea

Această instalație a fost concepută și construită pentru măsurarea emisivității plăcuțelor pentru cuțitele de strung produse de firma Kennametal, în vederea calibrării exacte a termogramelor obținute din urmărirea proceselor de așchiere.

Principiul de măsurare al instalației se bazează pe legea lui Boltzmann într-o formă modificată:

$$q = \varepsilon \sigma (T_e^4 - T_a^4) + q_c \quad (129)$$

Unde termenii au semnificația din relația similară, (87) iar  $q_c$  este un factor de corecție pentru cantitatea de căldură transmisă într-o incintă prin convecție și conducție moleculară, ținând cont de presiunea existentă în momentul măsurării. Pentru estimarea acestui termen am folosit diagramele obținute experimental de către Ross, care sunt prezentate în figura 47.

Din relația (129) s-a obținut termenul  $\varepsilon$  care ține cont, în acest caz și de forma obiectului studiat<sup>43</sup>. Pentru aceasta s-au măsurat temperaturile  $T_e$  și  $T_a$  și s-a calculat termenul  $q$ , cunoscând consumul de energie folosit pentru ridicarea temperaturii (în ipoteza că întreaga cantitate de energie s-a transformat în căldură). Aceste determinări s-au făcut pentru mai multe valori ale temperaturilor, permițând construirea unor curbe de dependență a emisivității în funcție de temperatură și direcția de măsurare.

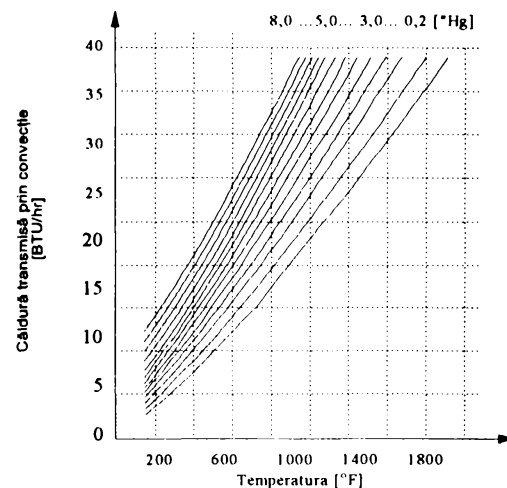


Fig. 47 Căldura transmisă prin convecție și conducție moleculară în funcție de temperatură, pentru diverse presiuni

### 4.6.2.Soluție constructivă pentru instalația de măsurat emisivitatea

În figurile 48, 49, 50 este prezentată soluția constructivă a instalației realizate.

Pentru determinarea curbelor de emisivitate, s-au creat condiții de transfer a căldurii numai prin radiație, în interiorul unei incinte din sticlă

<sup>43</sup> Pentru corpuri cu forme sferice sau cilindrice, Ross propune utilizarea unui coeficient cu care corectează emisivitatea stabilită pentru suprafețe plane



filtru 7-51 conform catalogului Galvoptics Ltd. cu grosimea de cca. 10mm, (poz. 12). Pentru aceasta s-a creat un vid între 12 și 20 mm Hg, prin racordarea incintei la o pompă de vid cu ulei pe racordul 18. Plăcuța 6 s-a încălzit la temperaturi controlate cu ajutorul unui cartuș de încălzire de 250 W (poz. 7), montat în suportul 4. Măsurarea temperaturii plăcuței (termenul  $T_e$ ) s-a făcut cu ajutorul a două traductoare rezistive montate pasiv pentru a elimina semnalul parazit produs de dilatarea suportului metalic al plăcuței. Concret traductoarele nu au fost lipite cu adeziv, ci s-au fixat pe un strat de vaselină aderentă Heat Sink Compound 55-431-1, care permite un transfer termic foarte bun de la suportul 4, dar le izolează din punct de vedere mecanic.

Pentru ca plăcuța să fie singura sursă radiantă în zona suprafeței absorbante 2, unde se măsoară temperatura  $T_a$ , s-a introdus un capac de protecție 8 cu o fantă prin care iese vârful plăcuței 6. Capacul are la interior o depunere reflectorizantă care îi asigură o absorbție neglijabilă. El se fixează pe arborele 11 prin trei șuruburi. Arborele de poliamidă 11 asigură etanșarea incintei pe un scaun conic în gura de etanșare 13 și prin garniturile de cauciuc 14, presate de presetupa 15. Gura de etanșare este fixată pe incintă prin intermediul unui adeziv siliconic.

Temperatura pe suprafața absorbantă s-a măsurat în trei puncte cu traductoare rezistive tip Balco B03 50 $\Omega$ , ce asigură o sensibilitate relativă de 0,0049  $\Omega/\Omega^\circ\text{C}$ . Pentru că suprafața absorbantă, în formă de emisferă centrată în vârful plăcuței, este realizată din rășină, cu coeficient de dilatare neglijabil, traductoarele au fost lipite direct pe această suprafață curbă, răspunsul lor nefiind afectat.

Pentru fiecare din cele 5 traductoare rezistive s-a folosit câte o punte Wheatstone cuplată la un amplificator operațional multicanal. Semnalul produs de amplificator a fost captat pe placa de achiziții de date PCL711S, datele fiind tratate în aceeași manieră ca și cele de la măsurătorile tensometrice.

Traductoarele din suprafața absorbantă au fost montate astfel încât să determine cu vârful plăcuței trei direcții analoage cu cele din figurile 44, 45, 46, unde a fost plasat scannerul în infraroșu când s-au captat termogramele.



- LEGENDA**
1. Traductor Ta
  2. Suprafață absorbantă
  3. Traductor Te
  4. Suport plăcuță
  5. Lamelă
  6. Plăcuță
  7. Cartuș încălzire
  8. Protecție
  9. Excentric
  10. Șurub M3
  11. Arbore
  12. Incintă
  13. Gură etanșare
  14. Garnituri
  15. Presetupă
  16. Dop
  17. Conectoare
  18. Racord

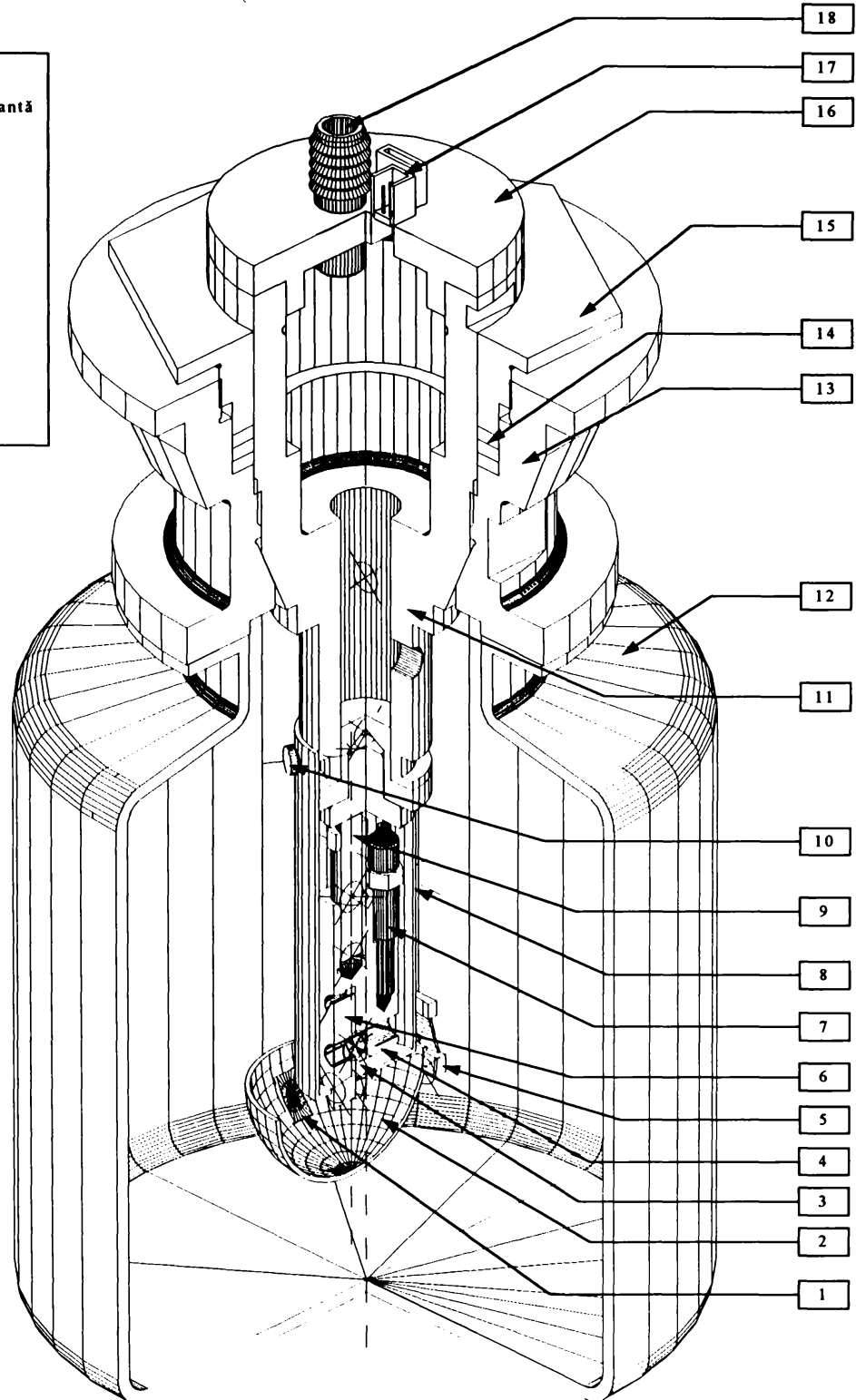


Fig. 48 Instalație de măsurat emisivitatea

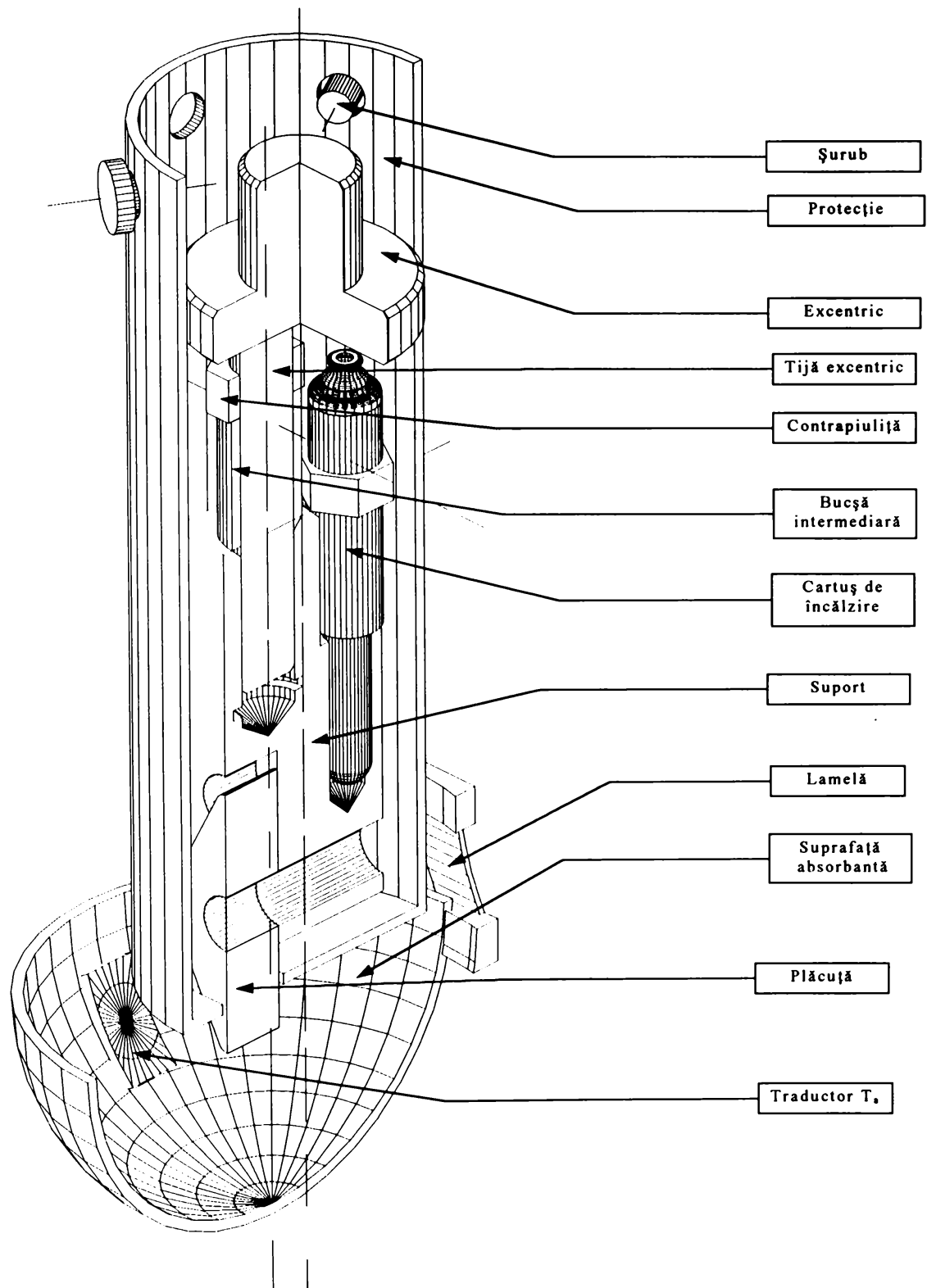


Fig. 49 - Instalație de măsurat emisivitatea, (detaliu suprafață absorbantă)

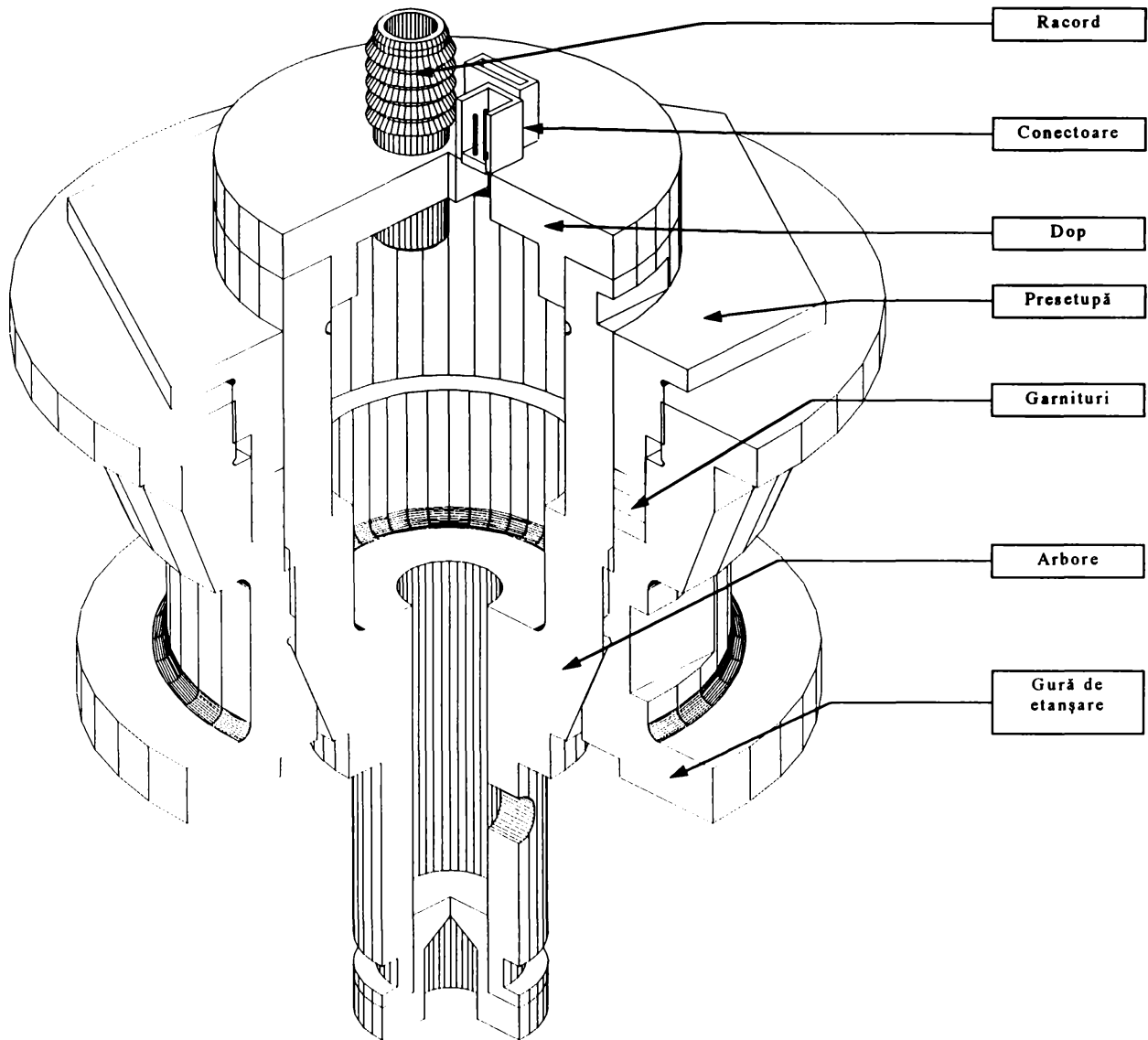


Fig. 50 - Instalație de măsurat emisivitatea, (detaliu gură de etanșare)

Pentru calibrarea informației oferite de cele cinci traductoare și transformarea valorilor în temperaturi pe scara Celsius, s-a introdus ulterior tot subansamblul detașabil al instalației, într-un cuptor cu temperatură controlabilă electronic și s-au construit curbele de etalonare ale instalației din figura 51 cu reprezentările polinomiale corespunzătoare celor din tabelul 1:

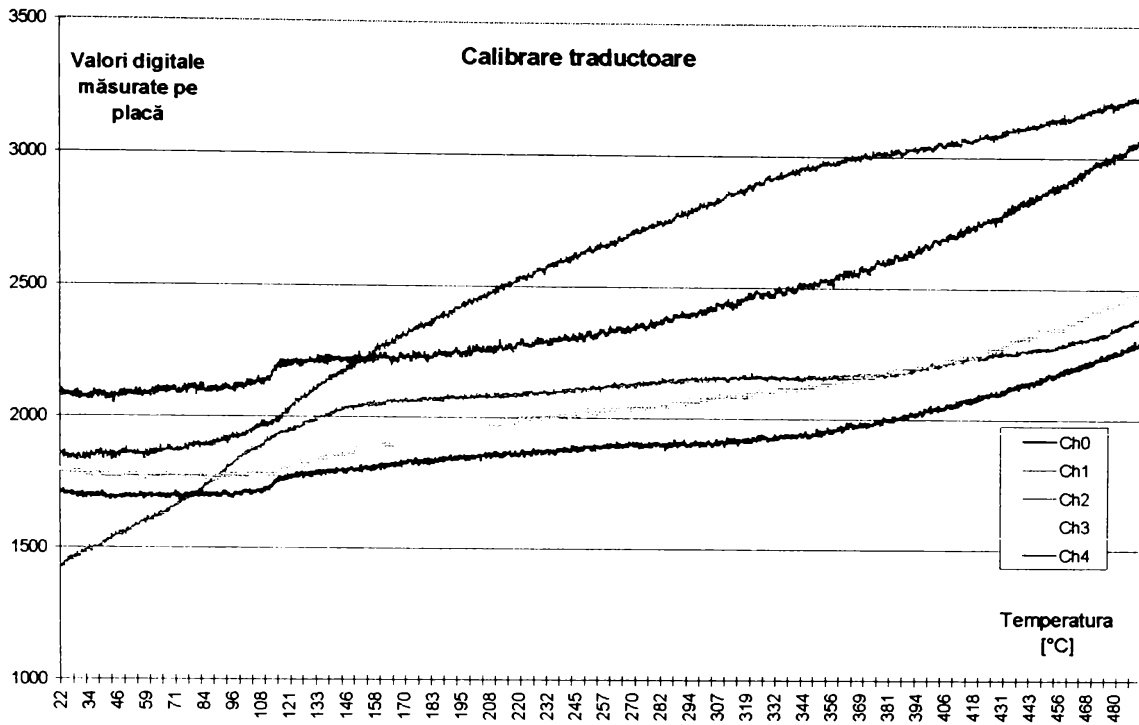


Fig. 51 - Calibrare traductoare temperatură

tab. 11 - Coeficienții polinomiali pentru calibrare temperatură

	$X^6$	$X^5$	$X^4$	$X^3$	$X^2$	$X^1$	$X^0$
<b>Ch0</b>	2E-17	3E-13	2E-09	- 3E-06	0.0023	- 0.5175	1722.5
<b>Ch1</b>	5E-16	- 3E-12	8E-09	- 9E-06	0.0038	0.5992	1438.3
<b>Ch2</b>	2E-16	- 1E-12	4E-09	- 7E-06	0.0054	- 1.0703	1895.5
<b>Ch3</b>	-3E-17	9E-14	3E-10	- 1E-06	0.0018	- 0.5416	1808.5
<b>Ch4</b>	2E-16	- 2E-12	4E-09	- 5E-06	0.0026	- 0.3409	2089.2

S-a folosit o viteză de încălzire de 2°C/min iar datele au fost citite pe placă la fiecare la 4 secunde. Temperatura a fost înregistrată pe placă direct din interfața digitală a cuptorului. Această soluție a permis și controlul influenței unor factori cum ar fi: autoîncălzirea traductoarelor datorită excitării punții, modificarea rezistenței firelor de legătură în raport cu temperatura.

#### 4.6.3. Determinarea emisivității plăcuțelor de așchiere

După prelucrarea datelor s-au obținut graficele din figurile 29, 30, 32, 33, 34 și polinoamele de gradul 6, cu coeficienții din tabelul 3, care le aproximează:

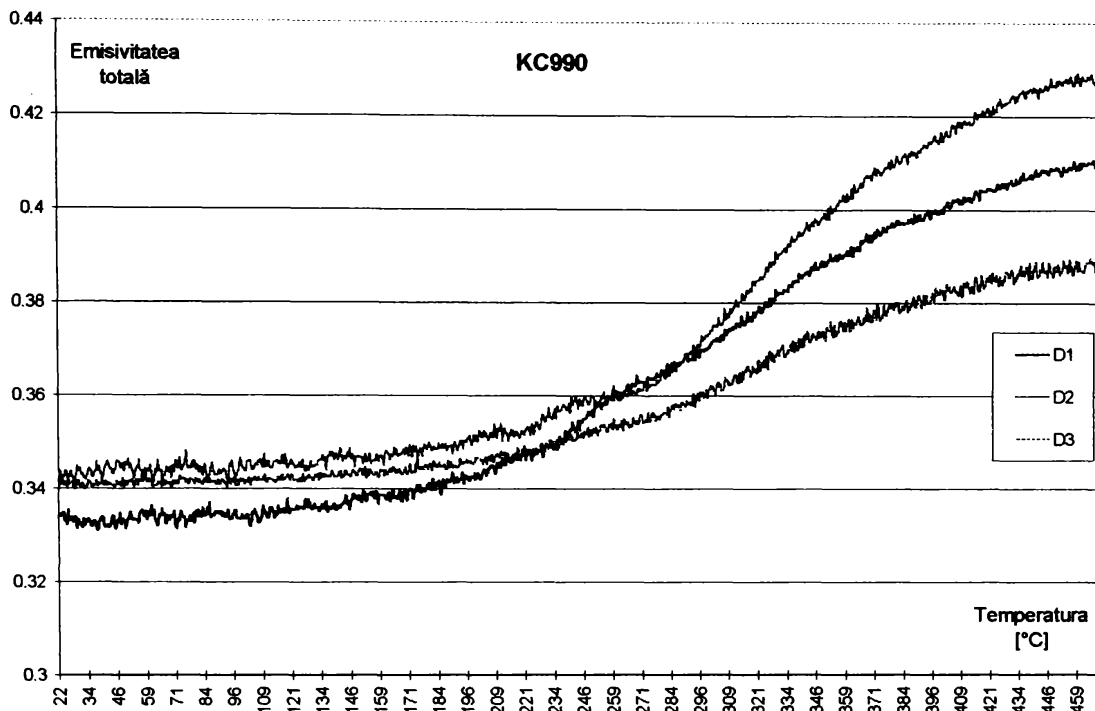


Fig. 52 Emisivitatea totală pentru KC990 pe trei direcții

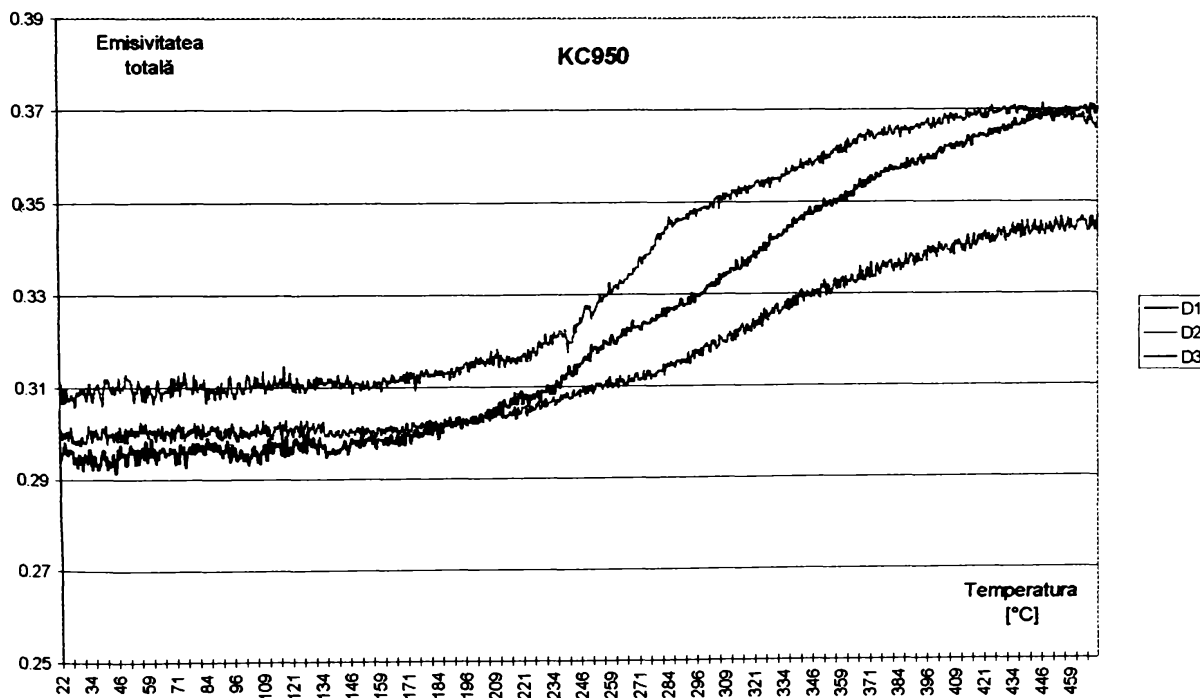


Fig. 53 - Emisivitatea totală pentru KC950 pe trei direcții

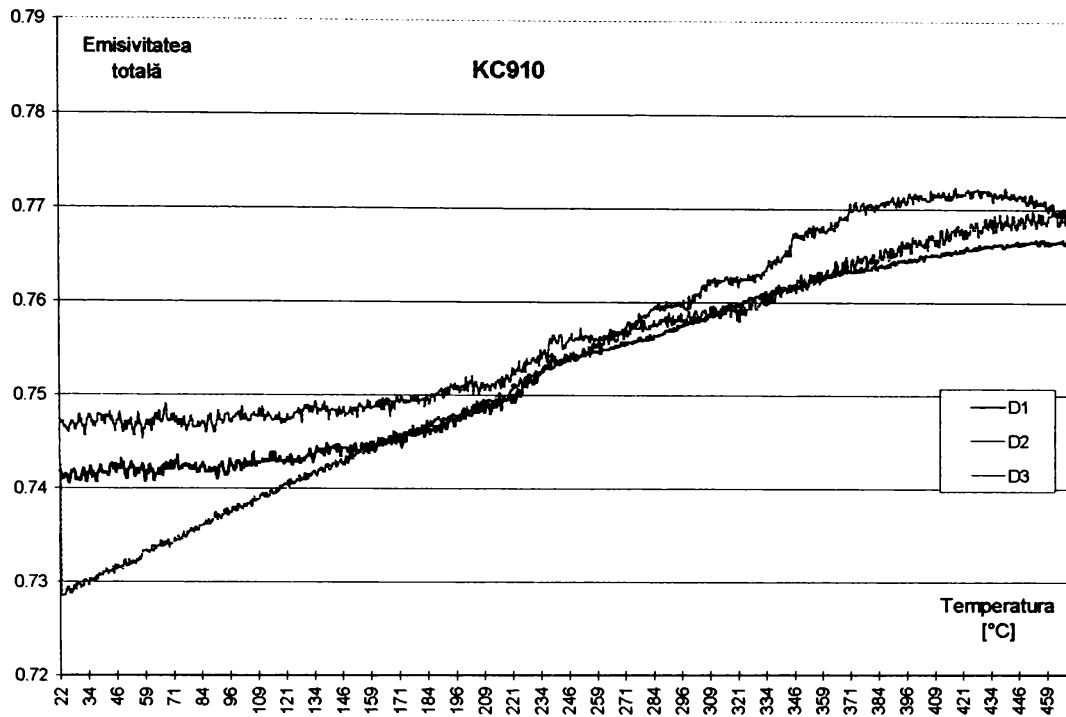


Fig. 54 - Emisivitatea totală pentru KC910 pe trei direcții

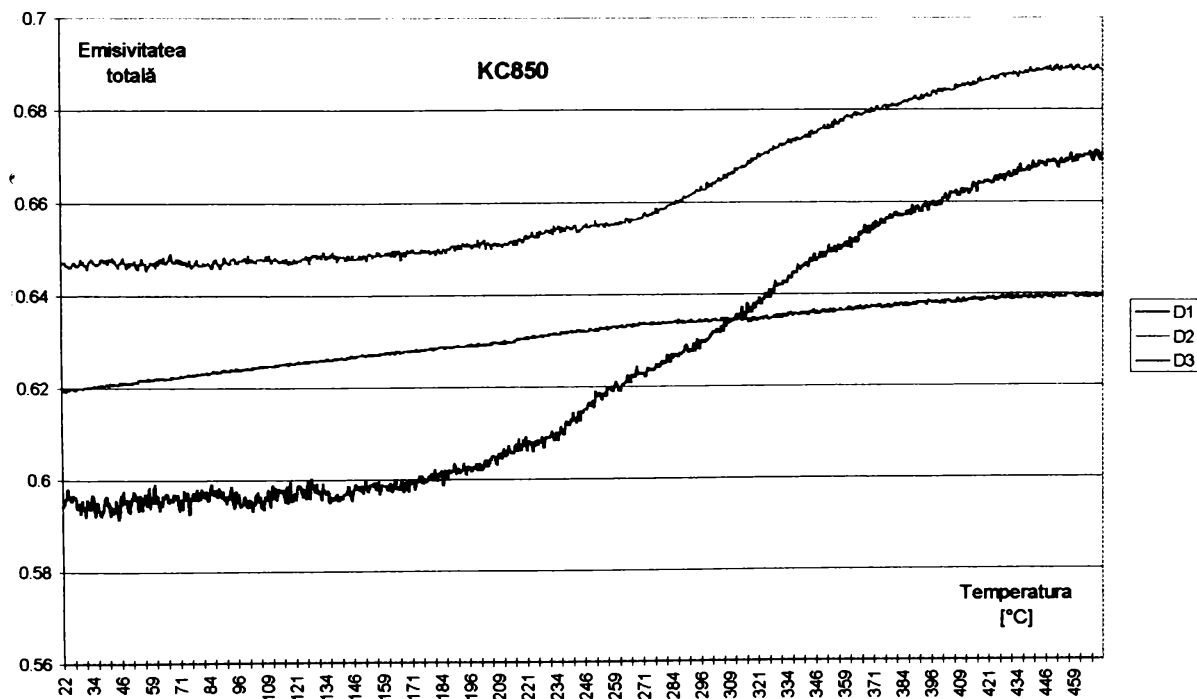


Fig. 55 - Emisivitatea totală pentru KC850 pe trei direcții

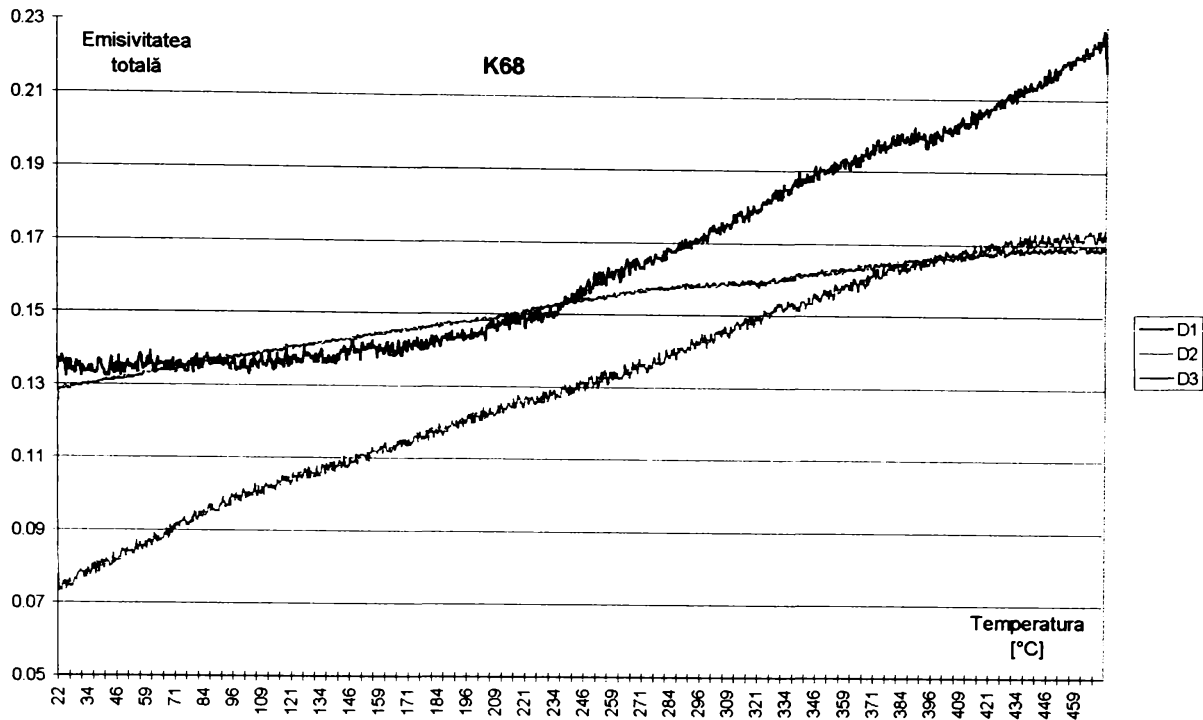


Fig. 56 - Emisivitatea totală pentru K68 pe trei direcții

tab. 12 Coeficienții polinomiali pentru curbele  $\varepsilon(T)$

	$T^6$	$T^5$	$T^4$	$T^3$	$T^2$	$T^1$	$T^0$
<b>KC990</b>							
<b>D1</b>	3.00E-18	-6.00E-15	5.00E-12	-1.00E-09	2.00E-07	-7.00E-07	0.3331
<b>D2</b>	6.00E-18	-2.00E-14	2.00E-11	-7.00E-09	1.00E-06	-0.0001	0.3454
<b>D3</b>	2.00E-18	-7.00E-15	6.00E-12	-3.00E-09	5.00E-07	-3.00E-05	0.3415
<b>KC950</b>							
<b>D1</b>	2.00E-18	-5.00E-15	3.00E-12	-4.00E-10	-1.00E-07	3.00E-05	0.2942
<b>D2</b>	4.00E-18	-1.00E-14	6.00E-12	-6.00E-10	-3.00E-07	7.00E-05	0.307
<b>D3</b>	2.00E-18	-5.00E-15	5.00E-12	-2.00E-09	2.00E-07	2.00E-06	0.2993
<b>KC910</b>							
<b>D1</b>	-1.00E-18	4.00E-15	-4.00E-12	2.00E-09	-5.00E-07	4.00E-05	0.7407
<b>D2</b>	-2.00E-19	3.00E-16	-4.00E-13	3.00E-10	-7.00E-08	7.00E-06	0.7468
<b>D3</b>	-1.00E-18	3.00E-15	-3.00E-12	2.00E-09	-3.00E-07	9.00E-05	0.728
<b>KC850</b>							
<b>D1</b>	2.00E-18	-5.00E-15	3.00E-12	-4.00E-10	-1.00E-07	3.00E-05	0.5942
<b>D2</b>	3.00E-18	-9.00E-15	9.00E-12	-4.00E-09	8.00E-07	-6.00E-05	0.6481
<b>D3</b>	-6.00E-19	2.00E-15	-2.00E-12	8.00E-10	-2.00E-07	4.00E-05	0.6192
<b>K68</b>							
<b>D1</b>	4.00E-18	-1.00E-14	8.00E-12	-3.00E-09	4.00E-07	-2.00E-05	0.135
<b>D2</b>	2.00E-18	-5.00E-15	5.00E-12	-2.00E-09	2.00E-07	2.00E-04	0.0743
<b>D3</b>	-1.00E-18	3.00E-15	-4.00E-12	2.00E-09	-4.00E-07	9.00E-05	0.1278

## 4.7. CONCLUZII

În urma acestei etape experimentale se pot trage următoarele concluzii:

- s-a scris programul PCL711 cu care s-au colectat datele tensometrice și cu care s-a comandat mașina unealtă;
- s-a construit un amplificator cu 6 canale pentru măsurătorile tensometrice;
- pentru experiment s-au efectuat un total de 216 prelucrări, organizate după un program factorial  $3^3$  (v, s, t);
- în timpul experimentului s-au ridicat cadre scanate în infraroșu, din diverse poziții, rezultând circa 500 de termograme;
- din acestea s-au reținut pentru tratare ulterioară, câte un cadru la fiecare prelucrare;
- s-a măsurat forța de așchiere, rezultând câte un fișier text pentru fiecare prelucrare;
- s-a făcut etalonarea măsurătorilor tensometrice, prin stabilirea unor ecuații de etalonare, în baza unor date măsurate pe o mașină de încercat la tracțiune;
- s-a conceput și construit o instalație pentru măsurat emisivitatea corpurilor mici, cu care s-au determinat experimental emisivitățile plăcuțelor de așchiere, în vederea calibrării termogramelor;
- pentru etapa de prelucrare volumului mare de date rezultate din experimentare, s-a procedat la scrierea unui program special pentru extragerea temperaturilor din termograme și a unui program pentru calculul prin metoda elementului finit a aportului surselor termice care au produs starea de temperaturi determinată.



# CAPITOLUL 5

## PRELUCRAREA DATELOR EXPERIMENTALE

- *TRATAREA TERMOGRAMELOR*
- *EXTRAGEREA TEMPERATURILOR DIN TERMOGRAME*
- *DETERMINAREA CARACTERISTICILOR SURSEI TERMICE*
- *ESTIMAREA PRECIZIEI RELAȚIILOR OBȚINUTE*
- *REPREZENTAREA GRAFICĂ A RELAȚIILOR*

## 5. PRELUCRAREA DATELOR EXPERIMENTALE

### 5.1. TRATAREA TERMOGRAMELOR

Folosind scannerul infraroșu prezentat anterior combinațiile de regimuri de așchiere, parametri și materialele prezentate în tabelul 7, s-au reținut imagini termografice din procesele de așchiere.

Pentru că scannerul infraroșu folosit poate emite semnal video, imaginile au fost înregistrate în paralel pe un echipament video profesional în format M3 și în format Sony 8mm. De pe banda video s-au făcut capturi de cadre pe calculator folosind o placă video grabber, în formatul bitmap PCX, 256 de nuanțe de gri.

O astfel de imagine primară neprelucrată, este prezentată în figura 57, din care s-a eliminat fondul negru, prin modificarea paletei de culori.

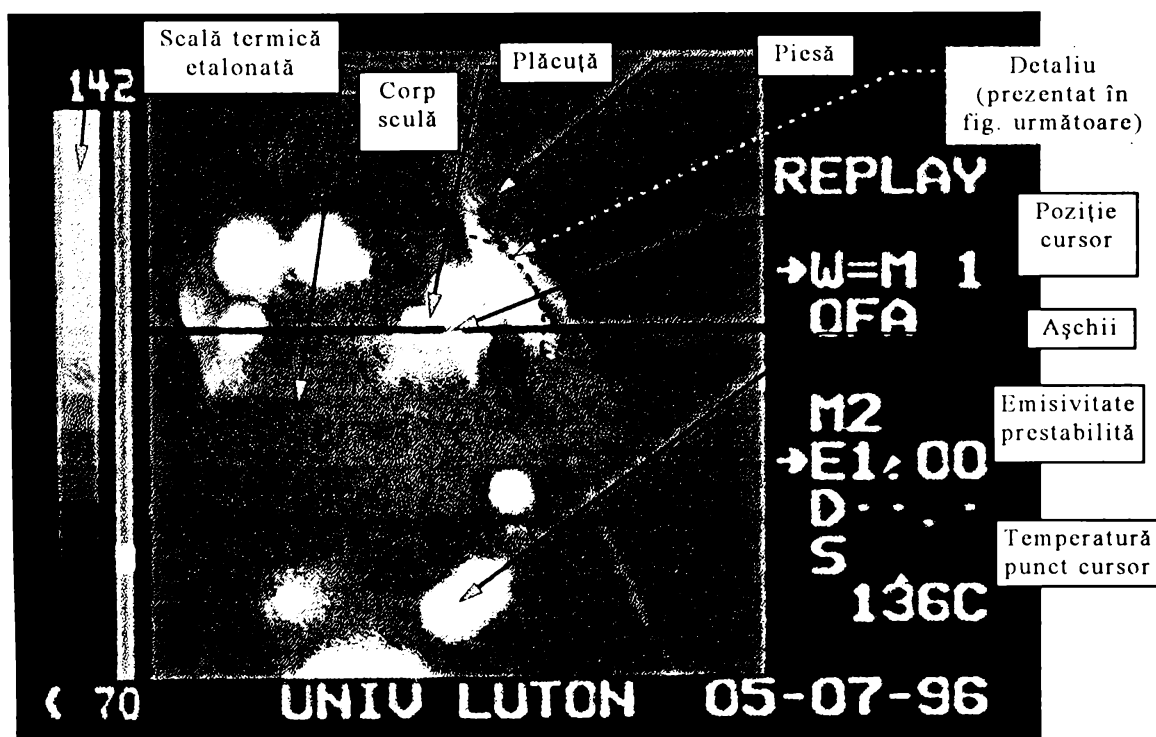
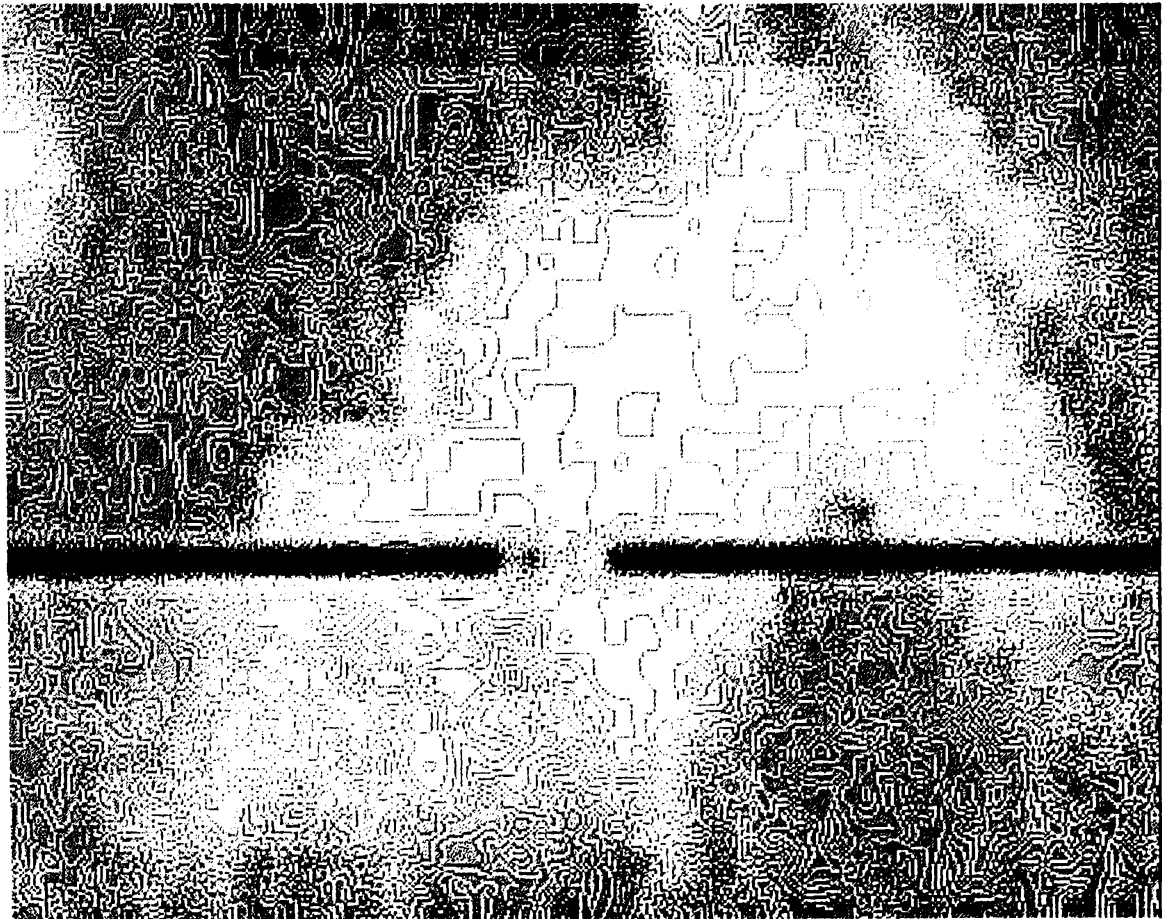


Fig. 57 Termogramă, (ENIA \ KC850 \ 113 \ 0,25 \ 0,5)<sup>44</sup>

<sup>44</sup> Figurile 57 și 58 au paleta de culori prestabilită de scanner, în format 256 nuanțe de gri. Cifrele din paranteză au semnificația: (material \ tip plăcuță \ viteza [m/min] \ avans [mm/rot] \ adâncime [mm])

Imaginea este calibrată automat de scanner și prezintă în partea din stânga o scară etalonată de temperaturi, iar în partea din dreapta valoarea temperaturii punctului unde se afla cursorul scannerului în momentul filmării.

Deși la o imprimare a termogramei respective este greu de observat cu ochiul liber, o astfel de imagine conține izotermele reale din momentul realizării instantaneului. Pentru a evidenția acest lucru, în figura 58 este prezentat un detaliu, după aplicarea unui filtru de căutare a muchiiilor fictive, determinate de diferențele de gri care evidențiază izotermele (tratament cu programul comercial Paint Shop Pro).



*Fig. 58 Termogramă detaliu mărit, cu evidențiere izoterme*

Pentru obținerea unei termograme utilizabile, ca cea din figura 59, s-a recurs la aplicarea unor filtre de imagine și schimbarea nuanțelor de gri într-o paletă de culori adecvată, care facilitează într-o măsură considerabilă interpretarea vizuală.

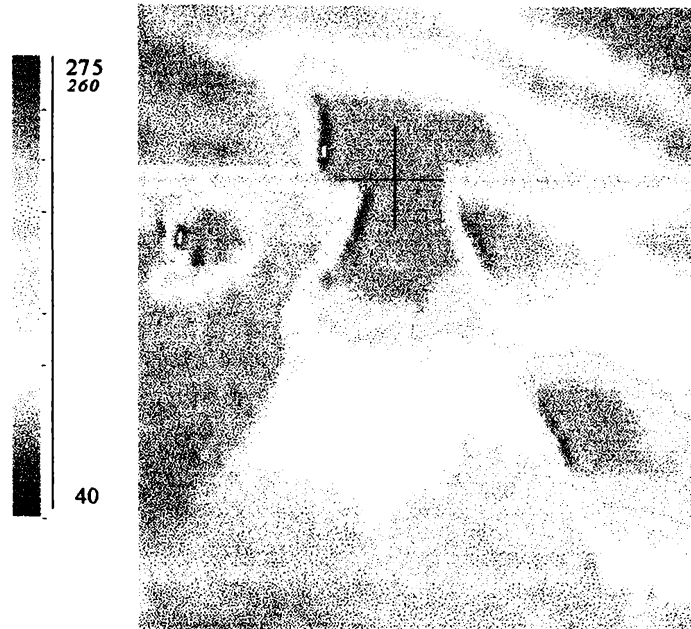


Fig. 59 Termogramă după filtrare și schimbarea paletii de culori (EN24T \ K68 \ 112 \ 0,15 \ 0,7)

Faptul că scannerul stabilește automat, pentru fiecare cadru în parte, corespondența între nuanțele de gri și temperaturile din zona vizibilă, a impus elaborarea manuală a paletii de culori pentru fiecare termogramă. Din motive de spațiu, sunt prezentate în continuare numai câteva din cele mai elocvente imagini, corespunzătoare regimului termic staționar.

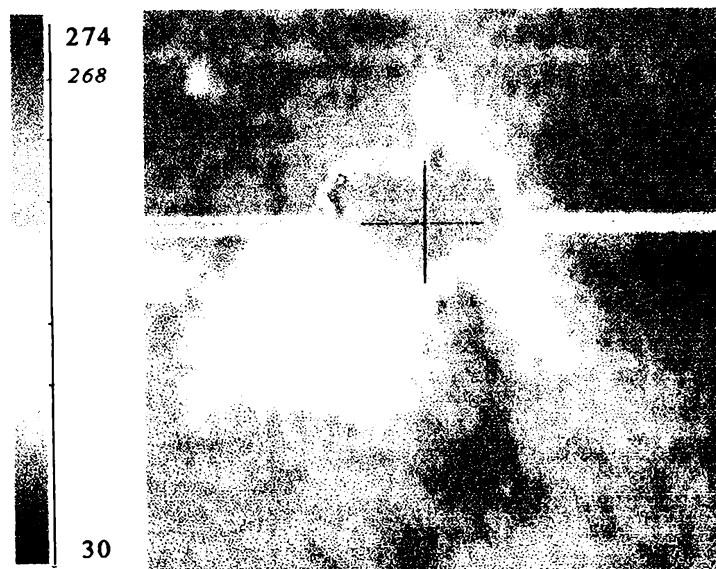


Fig. 60 Termogramă (EN24T \ K68 \ 112 \ 0,2 \ 0,7)



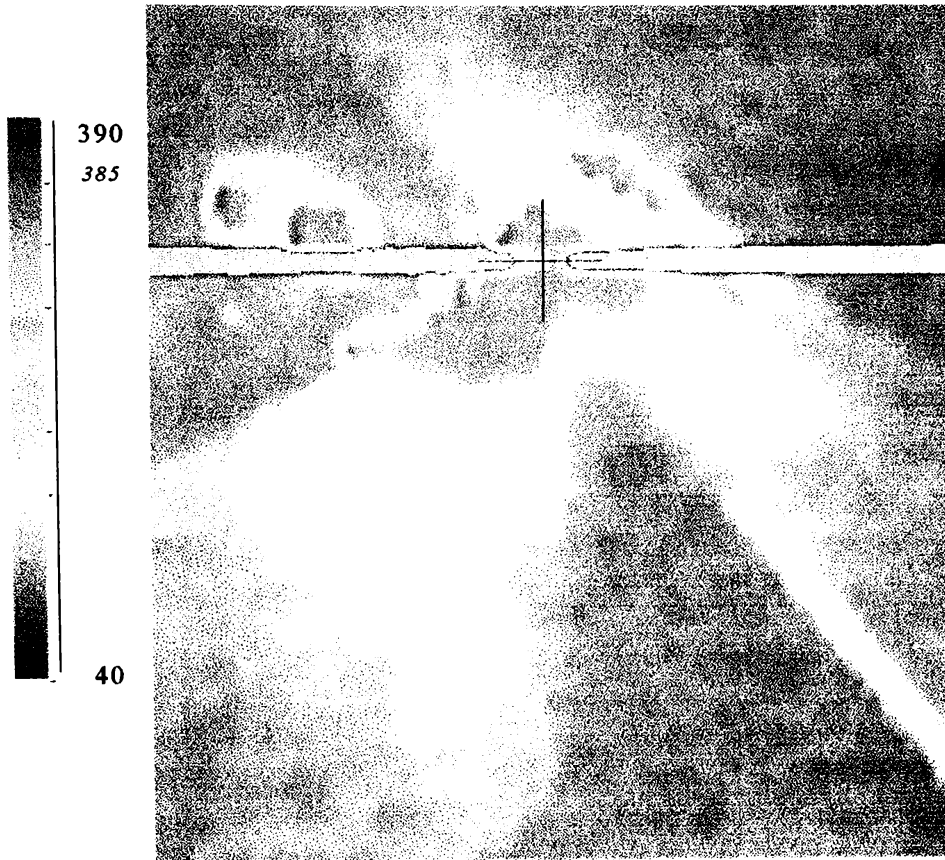


Fig. 61 Termogramă (EN24T \ KC990 \ 152 \ 0,15 \ 0,6)

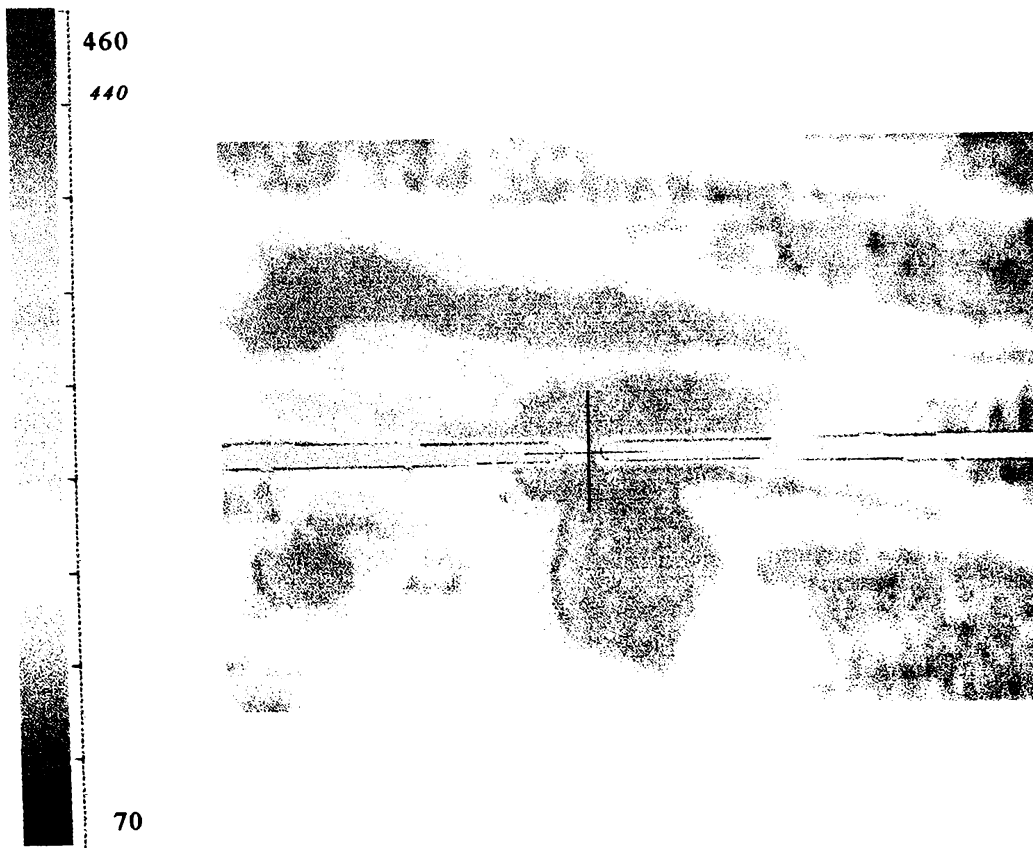


Fig. 62 Termogramă (EN24T \ KC990 \ 184 \ 0,15 \ 0,6)

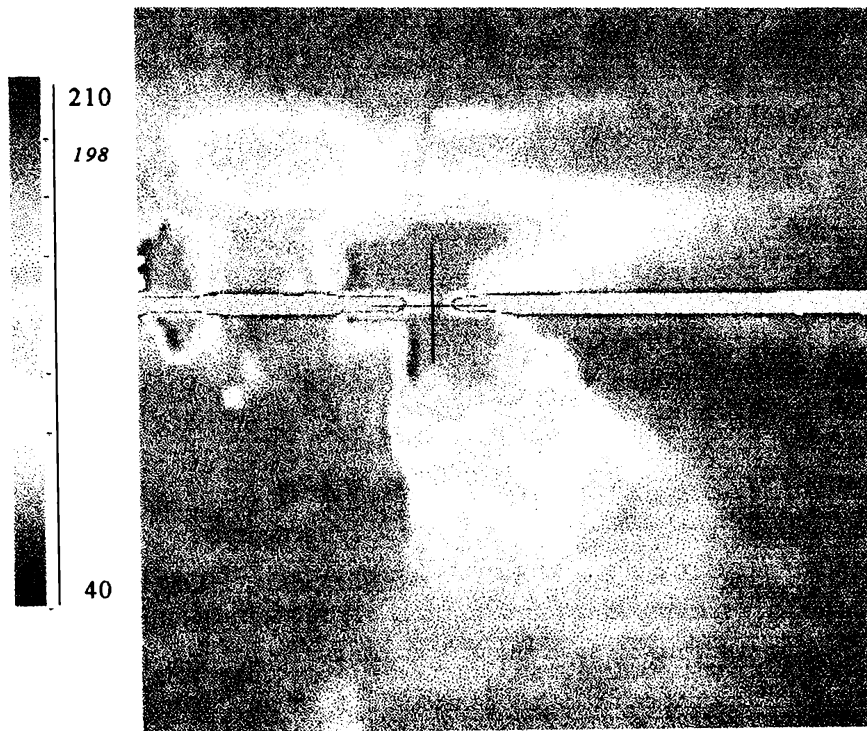


Fig. 63 Termogramă (EN24T \ K68 \ 112 \ 0,15 \ 0,5)

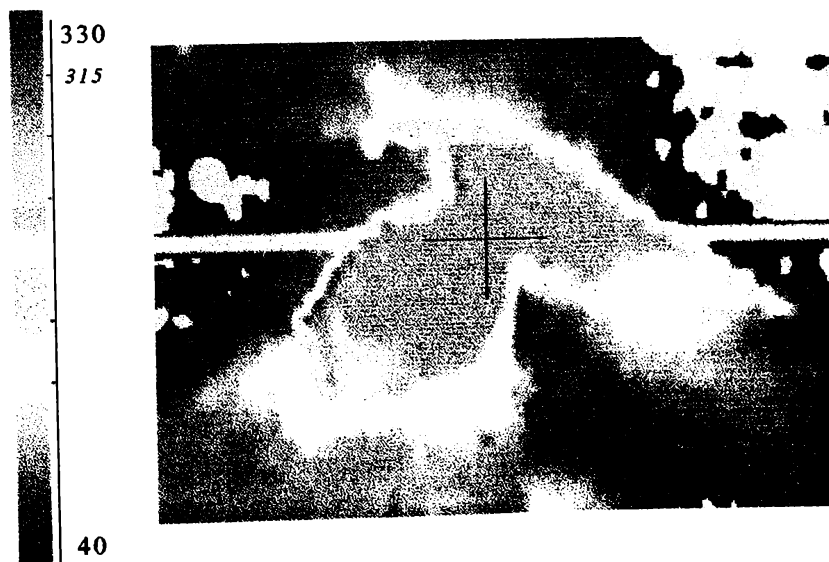


Fig. 64 Termogramă (EN24T \ KC990 \ 152 \ 0,1 \ 0,5)

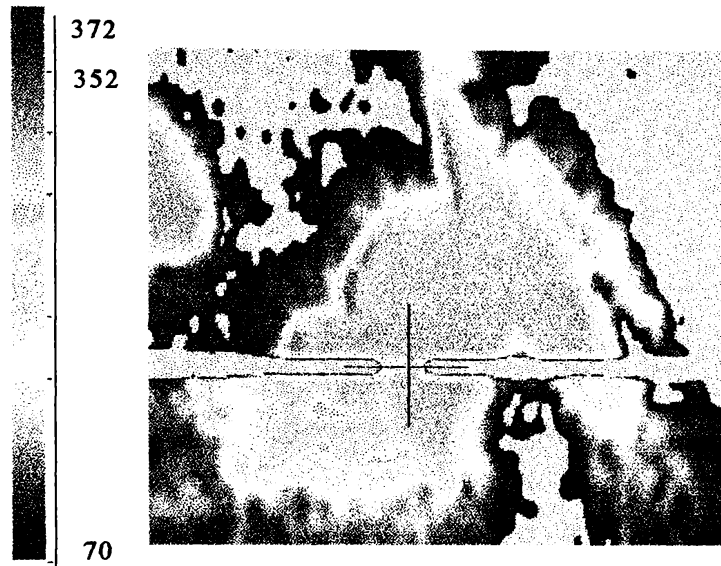


Fig. 65 Termogramă (EN24T \ K68 \ 152 \ 0,2 \ 0,7)

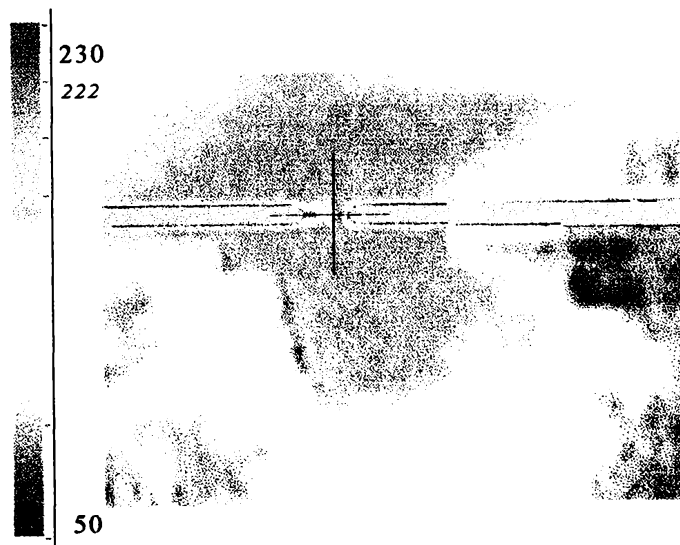


Fig. 66 Termogramă (EN24T \ K68 \ 112 \ 0,1 \ 0,6)



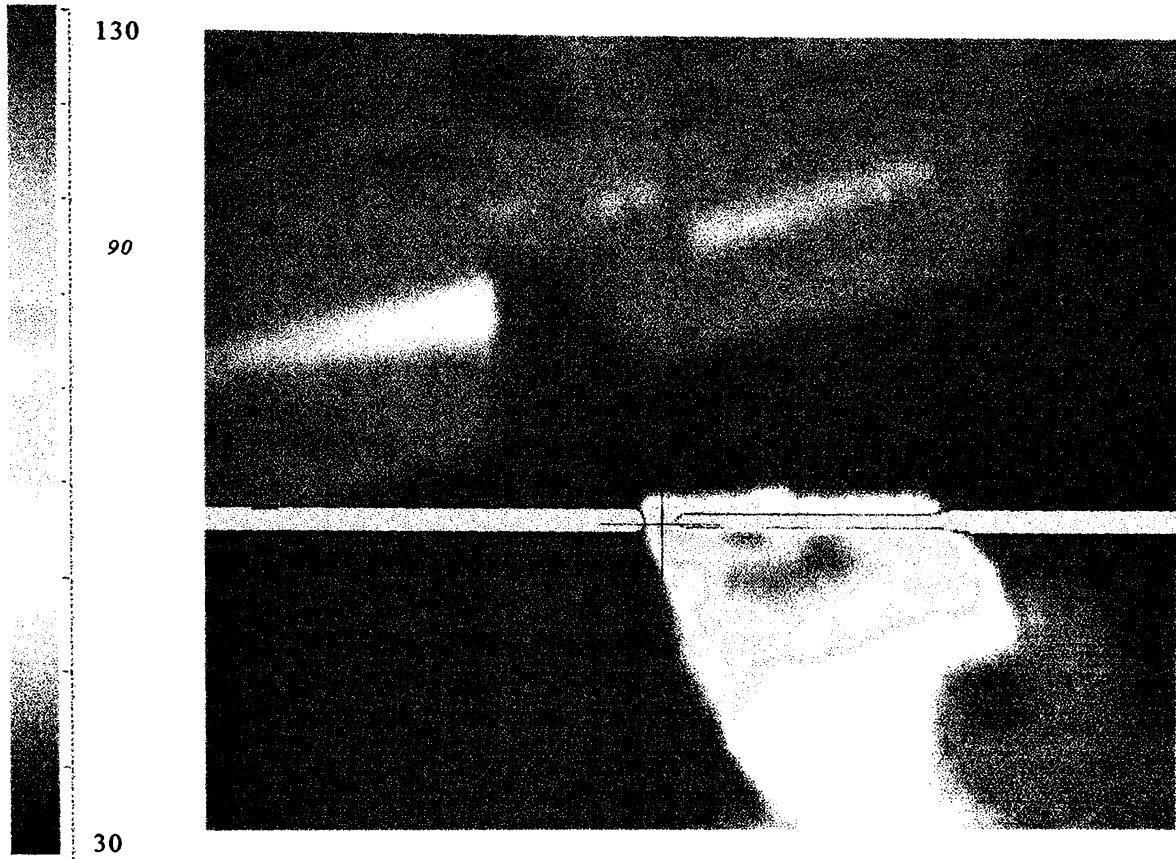


Fig. 67 Termogramă sculă în repaos<sup>45</sup>

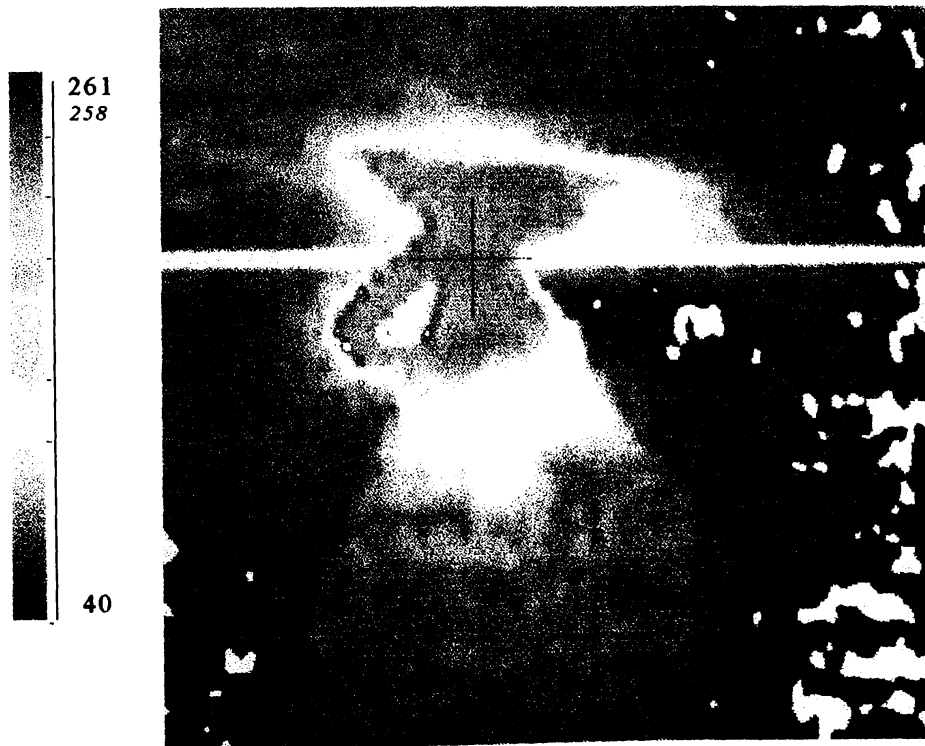


Fig. 68 Termogramă (EN8 \ KC910 \ 153 \ 0,2 \ 0,7)

<sup>45</sup> La 3 secunde după încheierea prelucrării EN24T \ K68 \ 152 \ 0,1 \ 0,6





Fig. 69 Termogramă (EN8 \ KC950 \ 184 \ 0,2 \ 0,6)

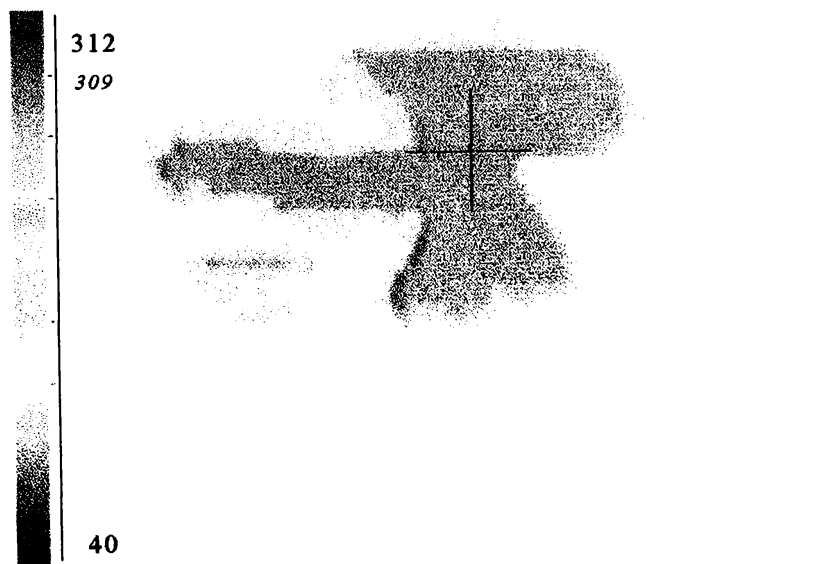


Fig. 70 Termogramă (EN8 \ KC910 \ 184 \ 0,25 \ 0,7)

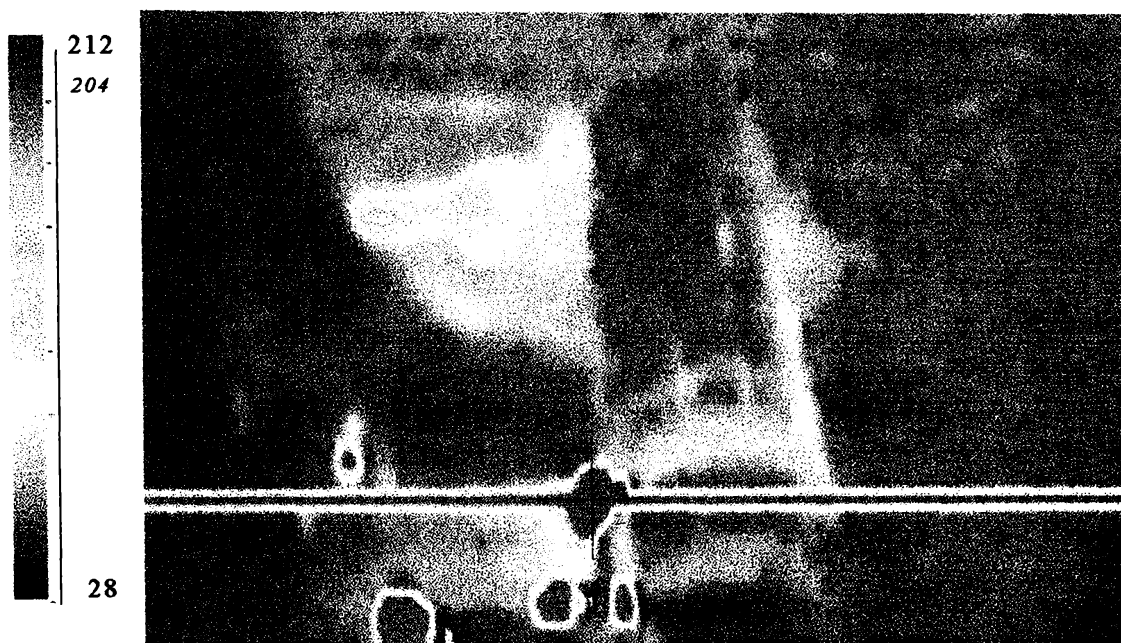


Fig. 71 Termogramă (EN8 \ KC910 \ 113 \ 0,2 \ 0,7)

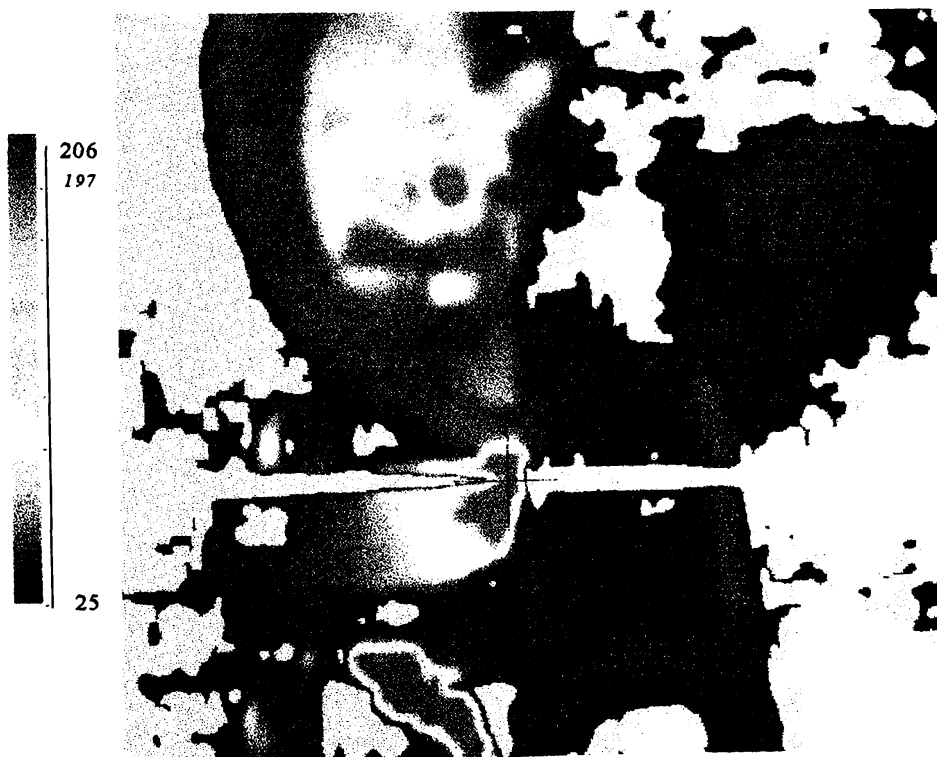


Fig. 72 Termogramă (EN8 \ KC950 \ 153 \ 0,2 \ 0,7)

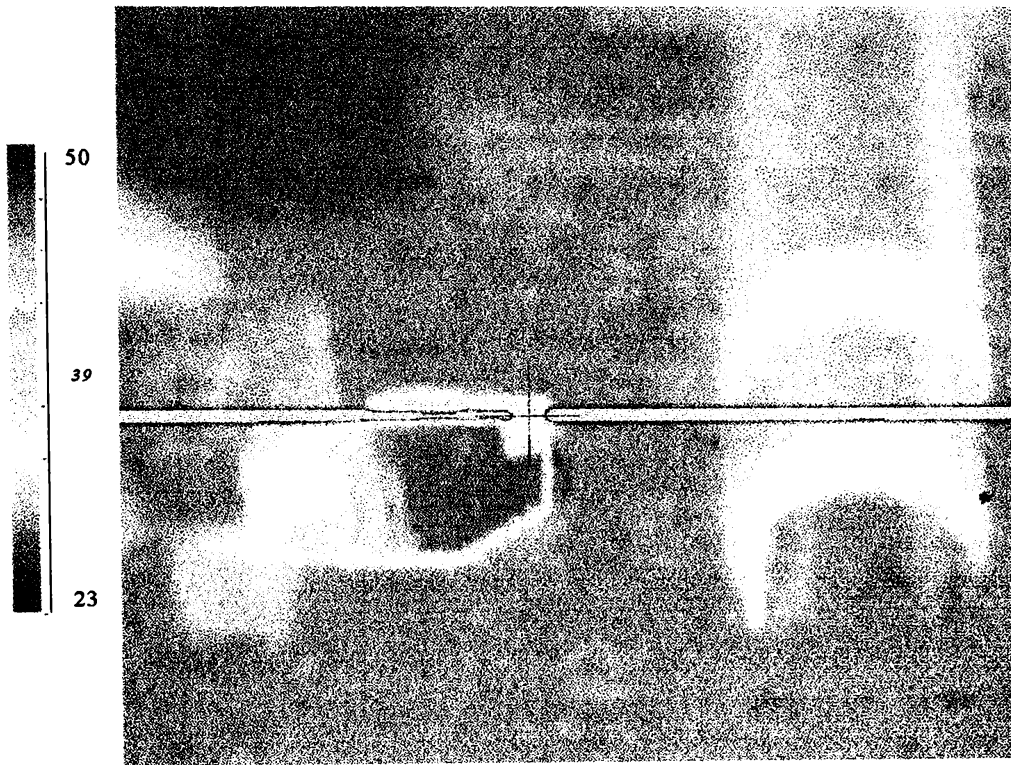


Fig. 73 Sculă în repaos<sup>46</sup>

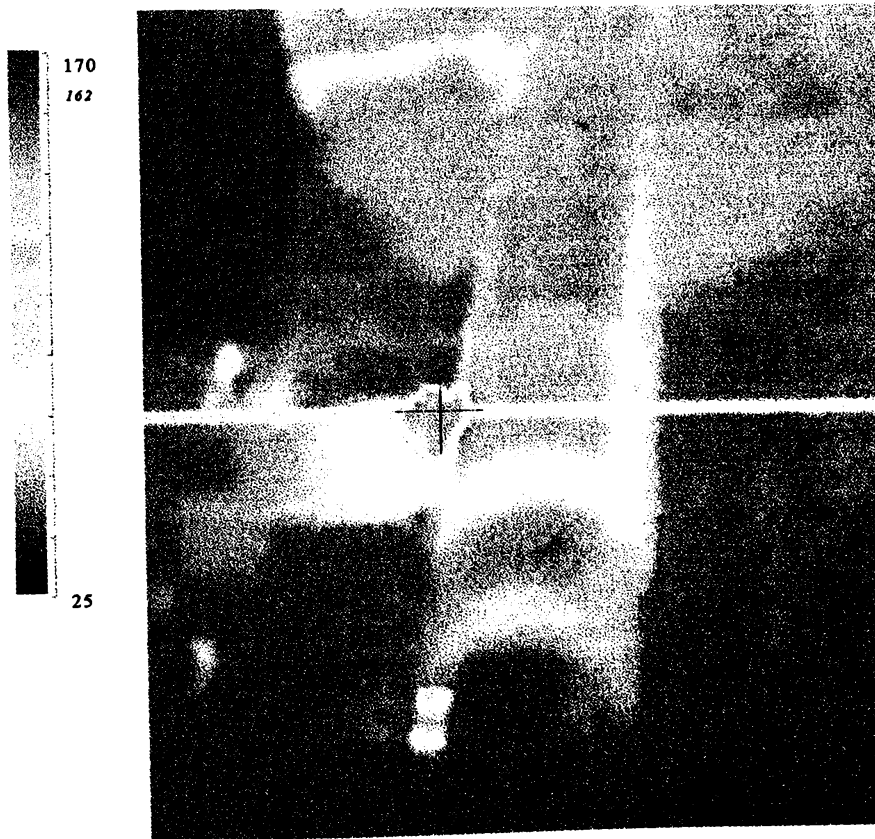


Fig. 74 Termogramă (EN8 \ KC950 \ 113 \ 0,25 \ 0,7)

<sup>46</sup> La 3 secunde după prelucrarea EN8 \ KC910 \ 113 \ 0,2 \ 0,7



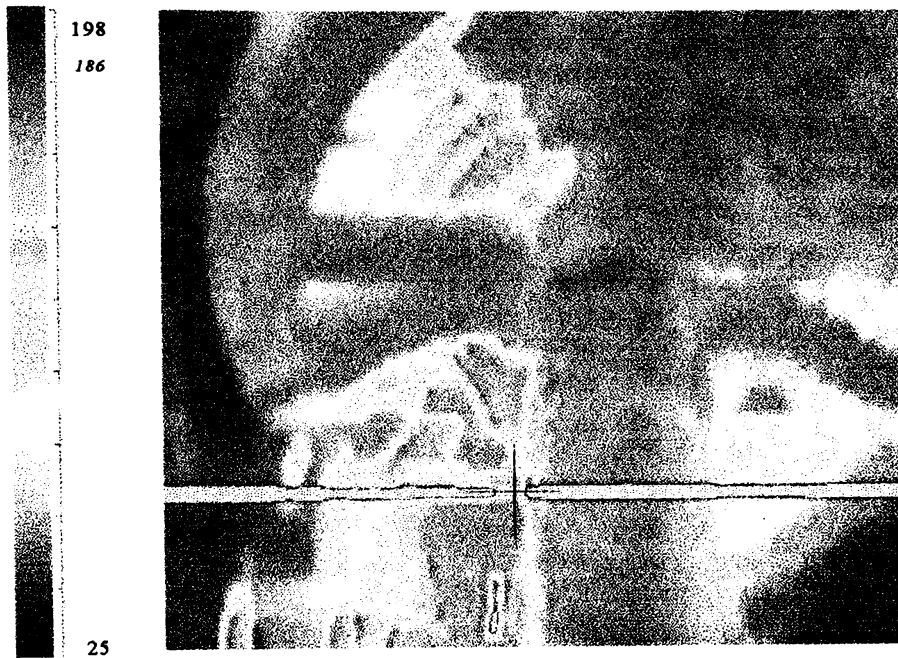


Fig. 75 Termogramă (EN8 \ KC910 \ 153 \ 0,15 \ 0,5)

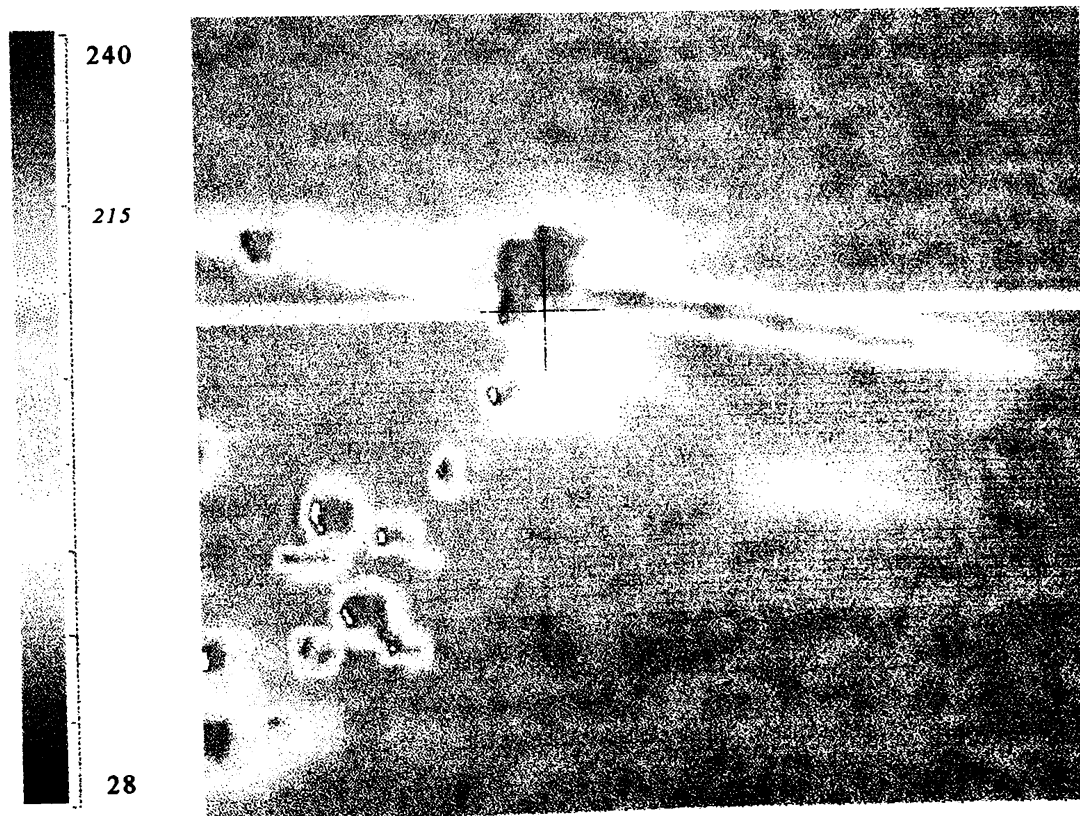


Fig. 76 Termogramă (EN1A \ KC850 \ 184 \ 0,3 \ 0,7)

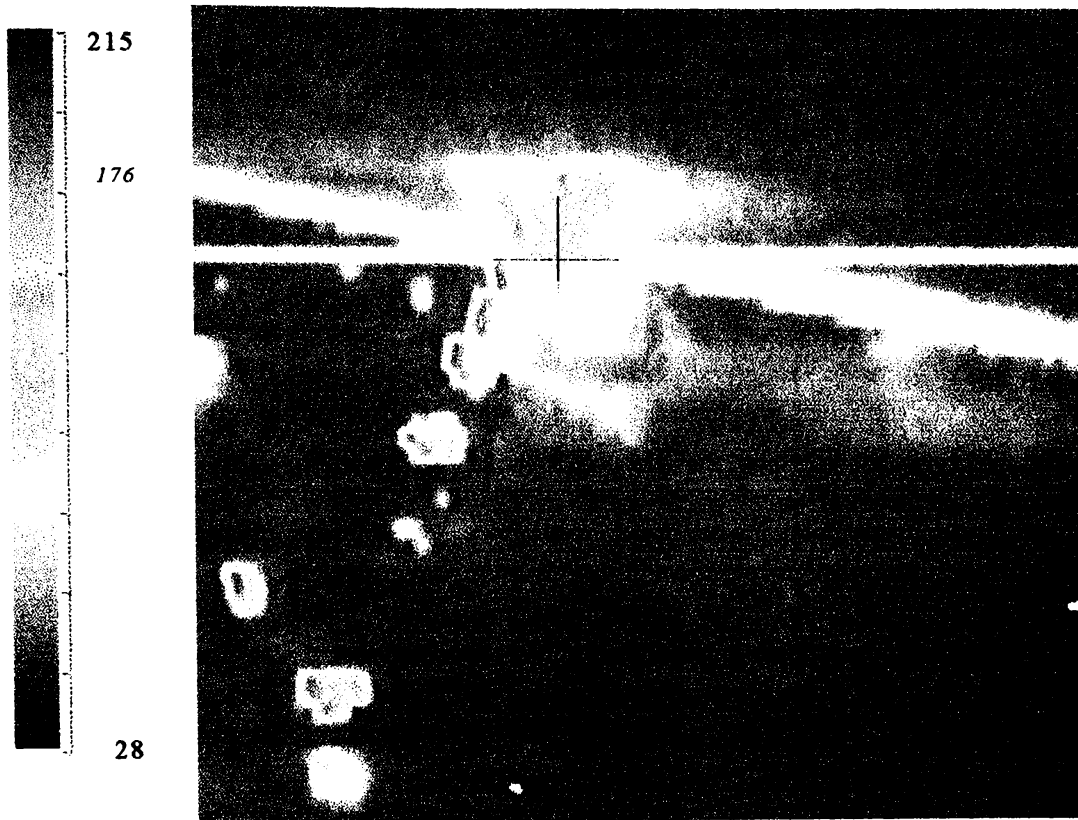


Fig. 77 Termogramă (ENIA \ KC850 \ 153 \ 0,3 \ 0,7)

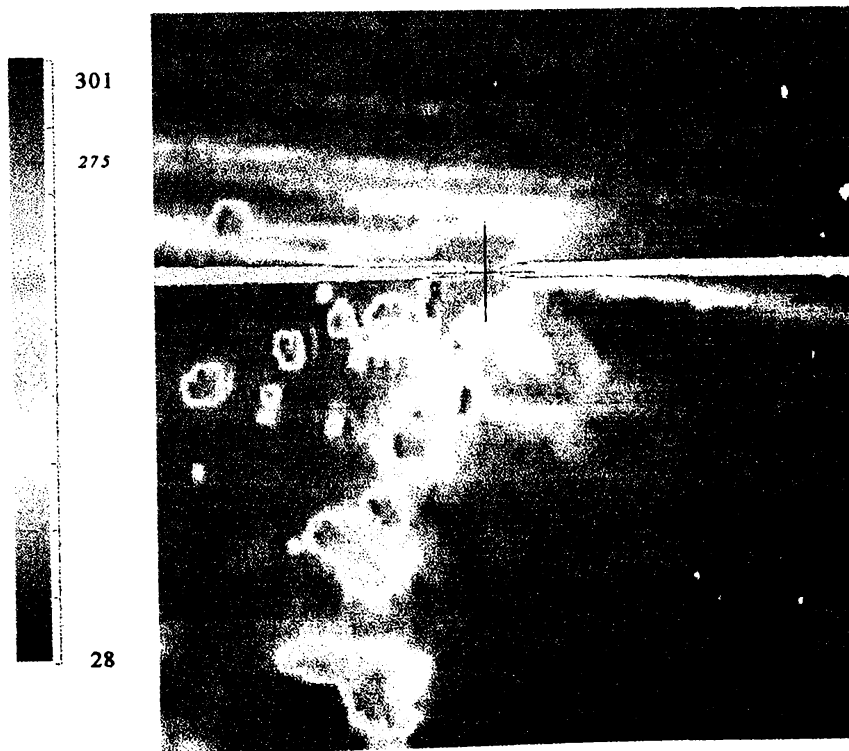


Fig. 78 Termogramă (ENIA \ KC850 \ 184 \ 0,35 \ 0,9)

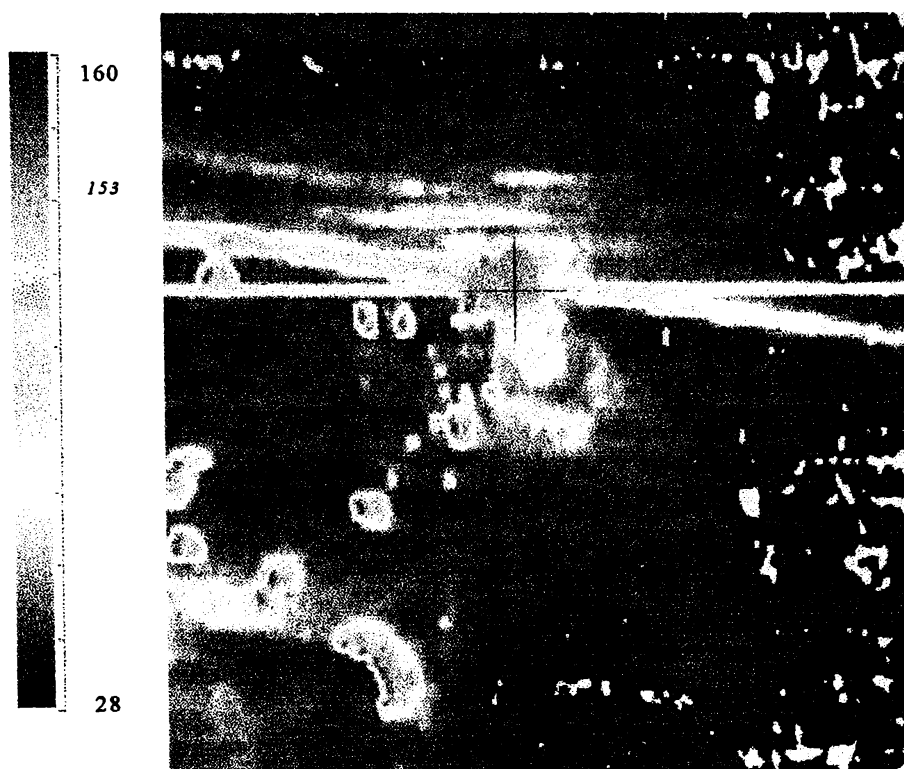


Fig. 79 Termogramă (ENIA \ KC850 \ 153 \ 0,25 \ 0,5)

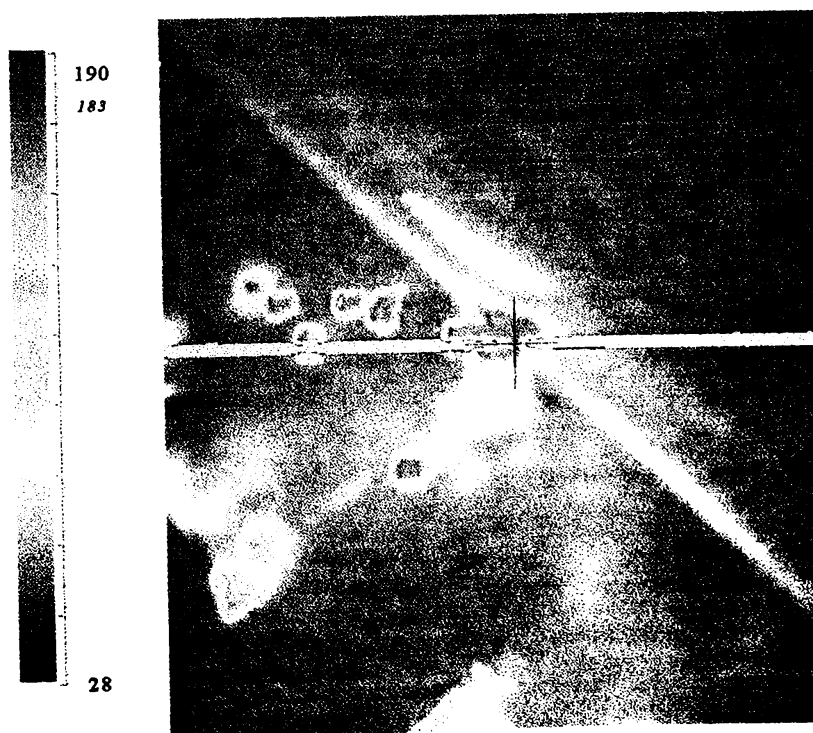


Fig. 80 Termogramă (ENIA \ KC850 \ 113 \ 0,35 \ 0,9)



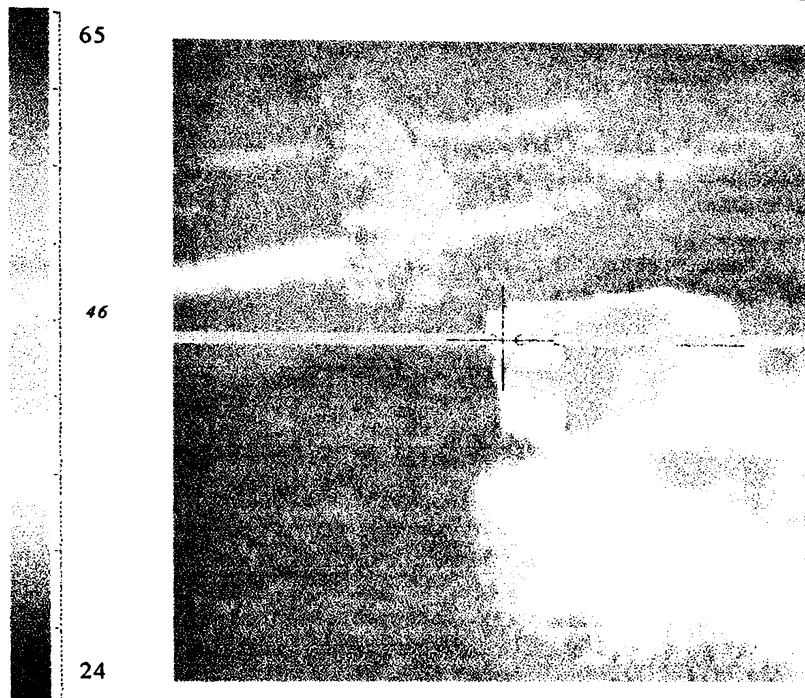


Fig. 81 Sculă în repaos<sup>47</sup>

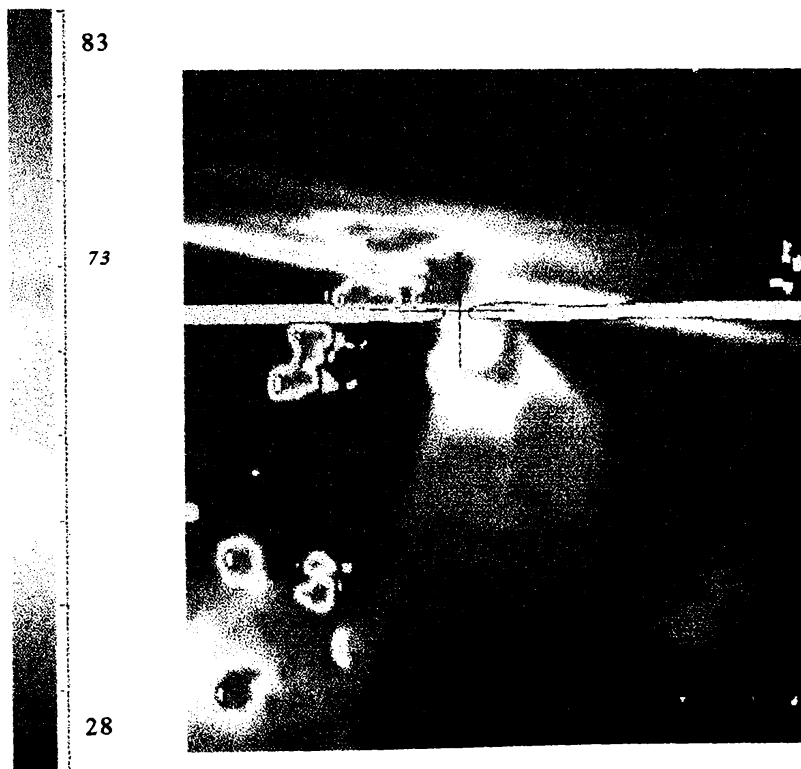


Fig. 82 Termogramă (Al \ K68 \ 184 \ 0,35 \ 0,9)

<sup>47</sup> La 3 secunde după prelucrarea EN1A \ KC910 \ 153 \ 0,35 \ 0,9

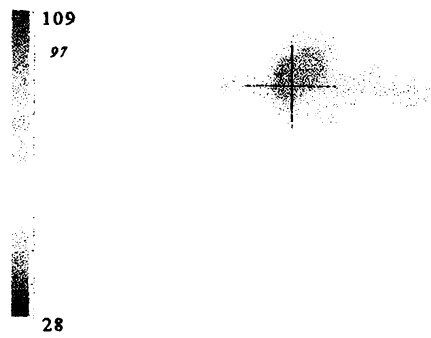


Fig. 83 Termogramă (Al \ KC910 \ 153 \ 0,35 \ 0,9)

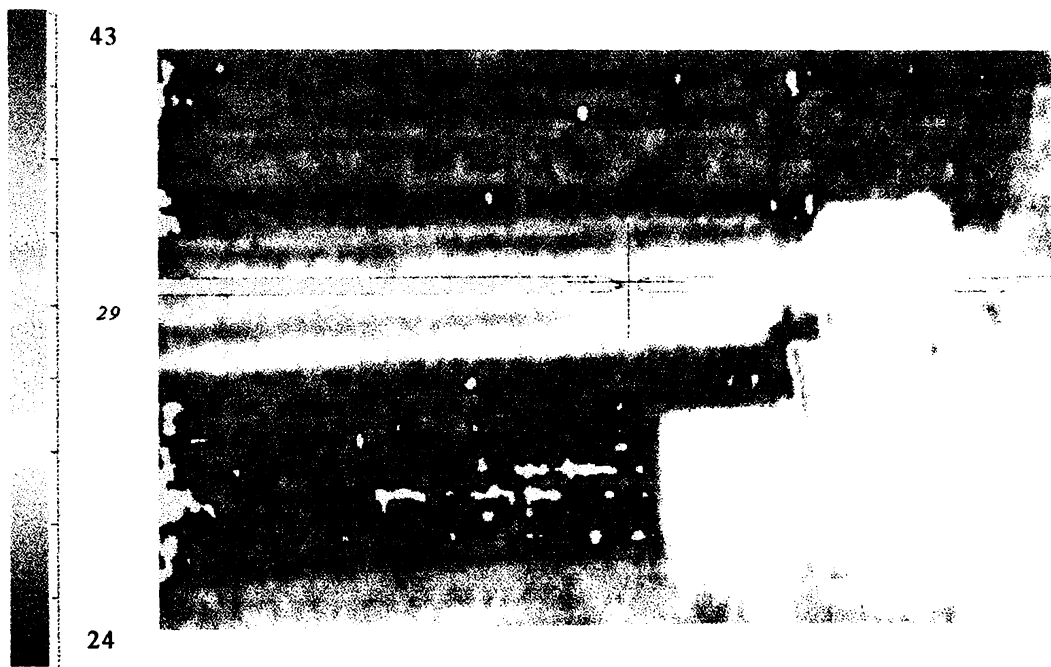
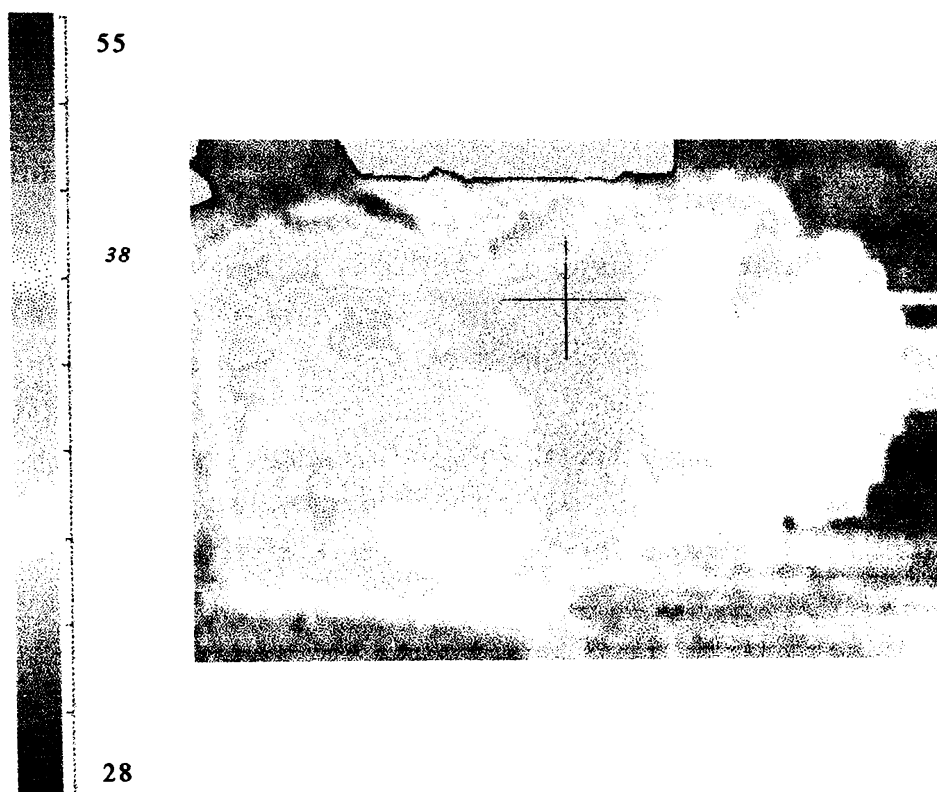


Fig. 84 Cadru repaos - piesă, păpușă mobilă, suport portsculă<sup>48</sup>

<sup>48</sup> La 3 secunde după prelucrarea Al \ K68 \ 153 \ 0,25 \ 0,7





*Fig. 85 Maşina unealtă, zona păpuşa fixă<sup>49</sup>*

<sup>49</sup> La 200 min de la pornire

## 5.2. EXTRAGEREA TEMPERATURILOR DIN TERMOGRAME

Imaginile termografice prezentate, reflectă valoarea temperaturii corespunzătoare emisivității  $\epsilon=1$ , așa cum a fost prestabilită înainte de începerea experimentului. Pentru obținerea valorii reale a temperaturii în punctele studiate<sup>50</sup>, precum și a fluxului termic pe conturul domeniului stabilit, s-a aplicat următoarea strategie:

- transformarea imaginilor termice în format DIB (Device Independent Bitmap);

- prelucrarea acestor imagini transformate cu programul *Termograme*, conceput de autor special în acest scop și care permite suprapunerea rețelei de elemente finite peste o termogramă în vederea obținerii temperaturilor nodale;

- tratarea fișierelor ASCII create de acest program, în worksheet-uri Excel, pentru corectarea valorii temperaturii cu emisivitatea corpului urmărit;

- completarea automată a datelor din secțiunea #BC a fișierului de intrare pentru programul Fem; acesta a rulat problema inversă cu elemente finite și a calculat puterea surselor la nivel de element, care corelată cu condițiile limită, a produs câmpul termic scanat în infraroșu.

### 5.2.1. OBȚINEREA TEMPERATURILOR NECORECTATE

Programul *Termograme* acceptă ca date de intrare<sup>51</sup> un fișier în format DIB sau BMP, (care conține termograma) și un fișier în format ASCII, cu extensia TRM, care conține descrierea rețelei de elemente finite și informații despre conturul domeniului, pentru a putea calcula fluxul termic pe hotar. În figura 86 este prezentată fereastra controlată de program, cu termograma din figura 63, peste care este suprapusă rețeaua completă de elemente finite (prezentată în figura 90). În figura 87 este prezentată termograma din figura 70, peste care s-a suprapus modelul cu elemente finite al plăcuței (prezentat în figura 91). Suprapunerea se face indicând poziția punctelor  $P_1$ ,  $P_2$  și  $P_3$  din figurile 86, respectiv 87, programul calculând automat coordonatele ecran pentru punctele nodale. Acest lucru presupune determinarea parametrilor transformării inverse (cosinușii directori), indicată în [137]<sup>52</sup>. Pentru că în transformarea inversă respectivă intervin

<sup>50</sup> Aceste puncte sunt în concordanță cu rețeaua nodală de elemente finite folosită la rezolvarea problemei inverse cu metoda elementului finit, ceea ce duce la obținerea încărcărilor termice la nivel de element, care au produs distribuția studiată a câmpului termic.

<sup>51</sup> Detalii descriptive sunt în anexa 4

<sup>52</sup> Zeid, I. *CAD/CAM Theory and Practice*, p. 503

funcții transcendente, algoritmul implementat este simplificat, beneficiind de faptul că este vorba de o rețea plană, rotită în spațiul tridimensional.

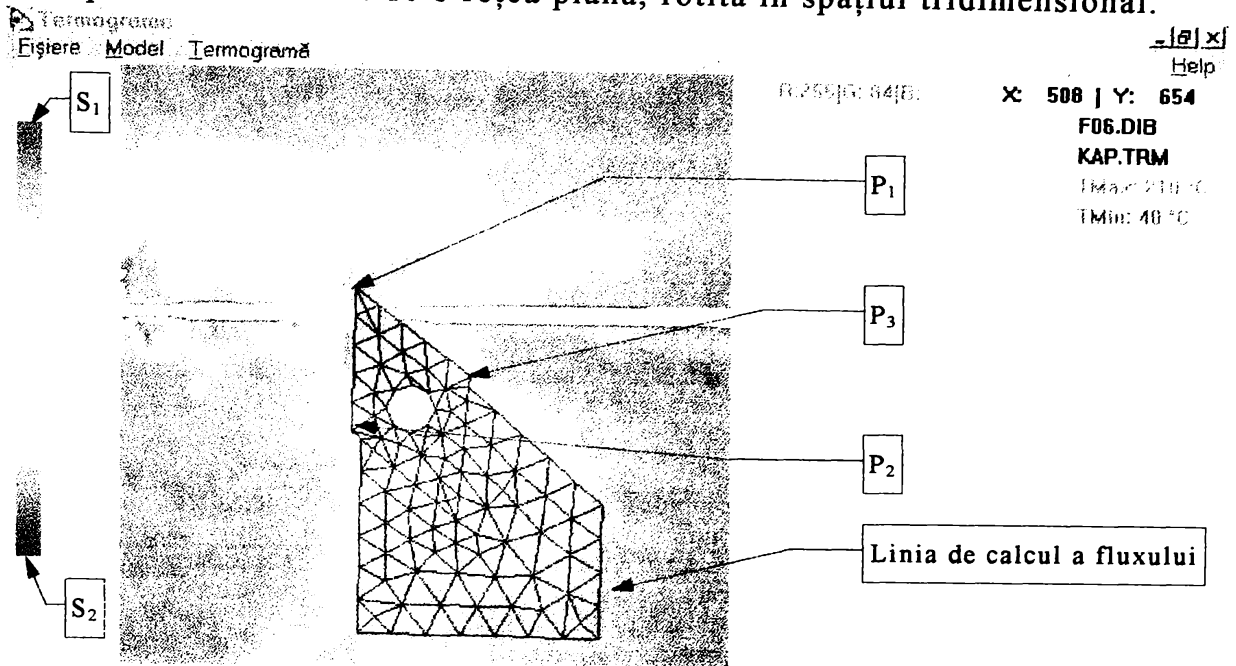


Fig. 86 Fereastra programului Termograme cu model MEF complet

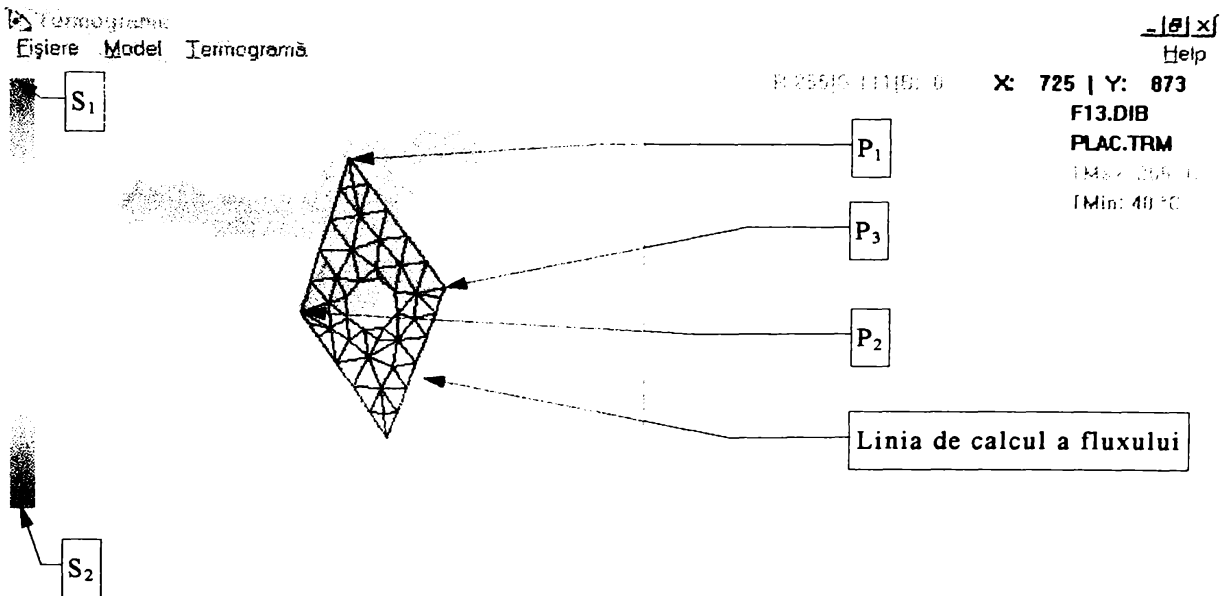


Fig. 87 Fereastra programului Termograme cu model MEF plăcuță

După cum se poate vedea în anexa 5 și pentru unghiuri nefavorabile de ridicare a termogramelor rețeaua este aplicată corect peste imaginea bit-map. După ce s-au indicat temperaturile extreme, distanța de calcul a fluxului pe contur<sup>53</sup> și s-au punctat cele două extreme ale scalei gradate

<sup>53</sup> Pentru claritate, în figurile 86 și 87 linia de estimare a fluxului pe contur a fost scoasă în afara modelului, dar în calculele făcute ea a fost plasată în interior

(punctele  $S_1$  și  $S_2$ ), programul scrie într-un fișier text temperaturile nodale și cele corespunzătoare unui contur paralel pentru calcularea ulterioară a fluxului. Acest lucru s-a făcut realizând o corespondență între culoarea pixelilor corespunzători nodurilor rețelei și valoarea temperaturii raportată la scala gradată din stânga imaginii.

### 5.2.2. APLICAREA CORECȚIEI DE EMISIVITATE

Aplicarea corecției de emisivitate presupune rezolvarea ecuației (131) pentru temperatura fiecărui nod, ecuație care derivă din legea lui Boltzmann scrisă în forma:

$$\varepsilon_r(T) \cdot \sigma \cdot (T_N^4 - T_a^4) = \varepsilon_1 \cdot \sigma \cdot (T_{M\acute{a}s}^4 - T_a^4) \quad (130)$$

unde:  $\varepsilon_r(T)$  este emisivitatea reală a corpului;  $\sigma$  este constanta lui Boltzmann;  $T_N$  este temperatura reală a nodului;  $T_a$  este temperatura mediului ambiant;  $\varepsilon_1$  este emisivitatea fictivă cu valoarea 1;  $T_{M\acute{a}s}$  este temperatura indicată din termogramă.

Dacă se ține cont de faptul că anterior s-au obținut expresii polinomiale de ordinul 6 pentru curbele de emisivitate, atunci din relația (130) rezultă următoarea ecuație în  $T_N$ :

$$(a_6 T_N^6 + a_5 T_N^5 + a_4 T_N^4 + a_3 T_N^3 + a_2 T_N^2 + a_1 T_N + a_0) \cdot (T_N^4 - T_a^4) - (T_{M\acute{a}s}^4 - T_a^4) = 0 \quad (131)$$

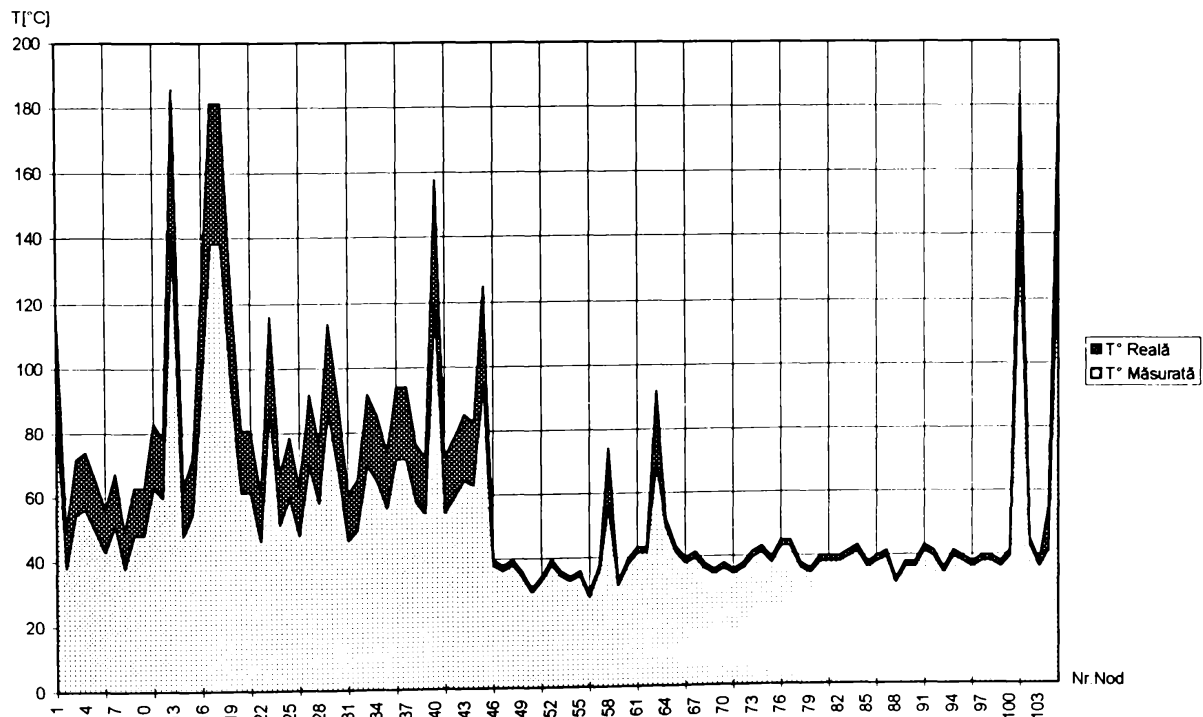


Fig. 88 Grafic comparativ între temperatura nodală corectată și necorectată

Rezolvarea cea mai comodă a ecuației de corecție, pentru toate termogramele analizate, se face cu ajutorul programării facilității *Goal Seek*

Scenario oferită de Excel. Funcția care completează automat acest lucru este prezentată în anexa 6.

După aplicarea corecției de emisivitate, așa cum se observă în figura 88, s-au obținut diferențe apreciabile de temperatură, care pentru valorile numerice din anexa 7 au o valoare maximă de peste 40°C.

Cu valorile temperaturii reale astfel calculate, se completează automat secțiunea #BC din fișierul de intrare pentru rularea problemei inverse. Această secțiune cuprinde temperaturile reale din noduri și fluxul unitar pe laturile elementelor de contur, calculat conform relației :

$$Q = K \cdot \frac{T_E - T_C}{d \cdot l} \quad (132)$$

unde:  $K$  este conductivitatea materialului din zona elementului;  $T_E$  este temperatura medie a laturii de contur a elementului;  $T_C$  este temperatura medie pe un contur paralel cu latura la distanța<sup>54</sup>  $d=1\text{mm}$  (în interiorul domeniului);  $l$  este lungimea laturii elementului.

Pentru valorile numerice din anexa 7, temperaturile pe contur și pe linia de calcul a fluxului, au următoarea variație:

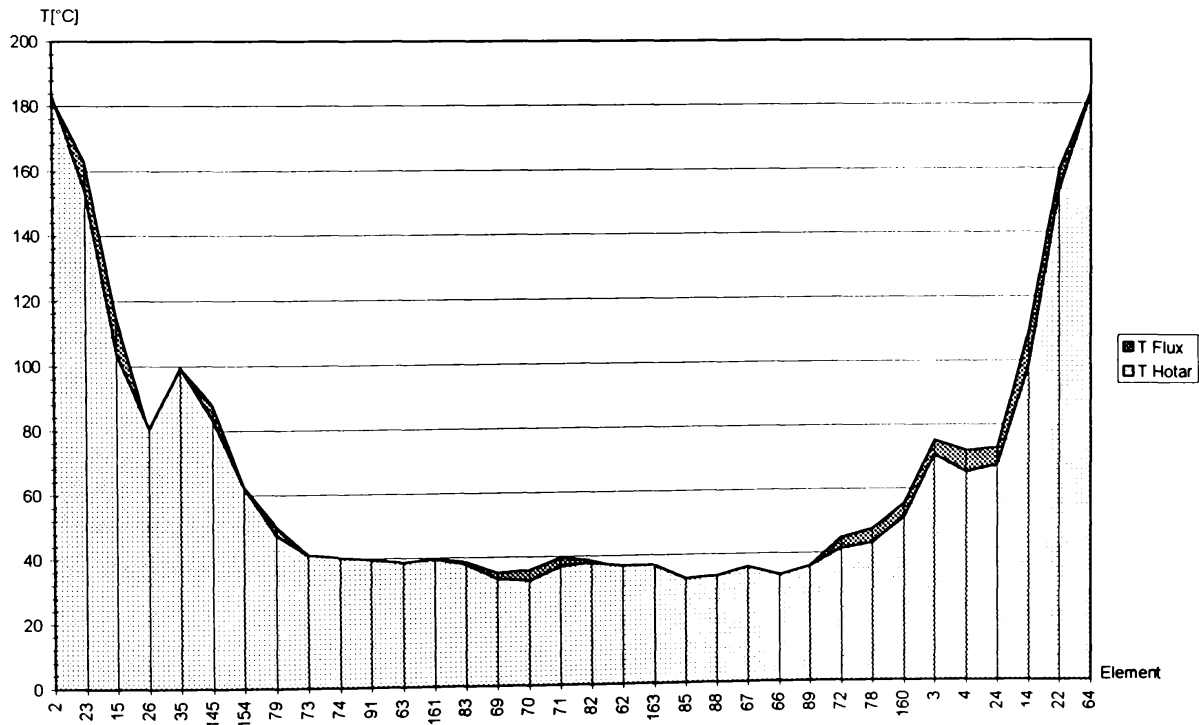


Fig. 89 Variația temperaturii pe conturul modelului și pe linia de calcul a fluxului<sup>55</sup>

<sup>54</sup> Distanța de 1mm a fost adoptată în funcție de dimensiunile termogramei, pentru a avea o rezoluție geometrică minimă în scopul determinării temperaturii punctului respectiv.

<sup>55</sup> Sunt luate în considerare numai elementele cu o latură pe conturul modelului; ordinea lor de pe abscisă este cea de pe modelul cu elemente finite.

### 5.3. DETERMINAREA CARACTERISTICILOR SURSEI TERMICE

#### 5.3.1. ALGORITMUL DE OBTINERE A ÎNCĂRCĂRILOR LA NIVEL DE ELEMENT

După completarea fișierului de intrare pentru programul *Fem*, se face rularea pentru problema inversă în vederea obținerii valorilor pentru puterea surselor termice din zona de așchiere. În continuare este descris algoritmul realizat pentru rezolvarea acestei probleme.

În cazul problemei inverse a transferului static de căldură prin conducție, pornind de la forma matricială din relația (133) a metodei elementului finit pentru formularea variațională, se cunoaște vectorul  $[T]$ , cel cu valorile temperaturilor nodale (obținut prin termografie în infraroșu) și este necesară determinarea vectorului  $[Q]$  al încărcărilor la nivel de element.

$$[K] \cdot [T] = [Q] \quad (133)$$

Pentru relația de mai sus, matricea globală de conductivitate  $[K]$  se poate forma exact ca în cazul problemei directe a metodei elementului finit, când necunoscutele sunt în vectorul  $[T]$  și se poate construi vectorul  $[Q]$  pornind de la valorile încărcărilor termice la nivel de element.

În situația inversă, vectorul  $[Q]$  se poate afla direct prin înmulțirea matricială respectivă, însă după modificarea matricei  $[K]$  prin aplicarea condițiilor la limită<sup>56</sup> de tip Dirichlet, Newmann și a condițiilor de transfer prin convecție și radiație pe frontiera domeniului, în funcție de situația concretă. Acest aspect este esențial, pentru că temperaturile nodale din vectorul  $[T]$  sunt soluții ale sistemului algebric echivalent obținut după aplicarea condițiilor limită și nu ale celui inițial care de fapt este nedeterminat, pentru că matricea  $[K]$  este singulară.

Cu această operație nu se rezolvă însă problema, deoarece componentele vectorului  $[Q]$  astfel determinate, trebuie descompuse în încărcări la nivel de element, după un proces invers celui prezentat în 2.2.3, 2.2.4, 2.2.5. Pentru aceasta se ține cont că fiecare componentă se obține conform relației (134), care provine din relația (119), cu precizarea că  $J=2 \cdot A$ , iar suma apare din asamblarea vectorului încărcărilor conform celor de la 2.2.3, 2.2.4:

$$Q_i = \sum_{k=1}^n \frac{J_k}{6} L_k \quad (134)$$

<sup>56</sup> Considerații despre metodele de aplicare ale acestor condiții limită au fost prezentate în paragraful 2.2.5.

unde:  $Q_i$  este componenta corespunzătoare nodului  $i$ ;  $n$  este numărul de elemente conectate la nodul respectiv și care au surse interne de căldură;  $J_k$  este iacobianul transformării de coordonate spațiu cartezian-spațiu parametric corespunzător elementului  $k$ ;  $L_k$  este valoarea puterii sursei termice interne din elementul respectiv.

Deci, pentru a rezolva complet problema, trebuie determinate de fapt valorile  $L_k$ <sup>57</sup>, pentru toate elementele considerate a avea sursă internă de căldură, care produc distribuția câmpului termic corespunzătoare vectorului [T].

Pornind de la relația (134), pentru elementele care au astfel de sursă internă de căldură, s-a scris sistemul de ecuații cu expresia matricială din relația (135):

$$\begin{array}{c}
 \leftarrow n \text{ elemente } \dots \rightarrow \\
 \uparrow \\
 m_{\text{noduri}} \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & \dots & v_{mn} \end{bmatrix} \cdot \begin{bmatrix} L_1 \\ L_2 \\ \vdots \\ L_n \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_m \end{bmatrix} \\
 \downarrow
 \end{array} \quad (135)$$

sau, mai concis :

$$[V] \cdot [L] = [Q] \quad (136)$$

unde: matricea [V] este de tipul  $m \times n$  cu semnificațiile:  $m$  reprezintă numărul nodurilor implicate în definirea elementelor cu surse interne de căldură;  $n$  reprezintă numărul elementelor cu sursă internă de căldură; [L] este vectorul ce conține puterile surselor termice  $L_i$  la nivel de element;  $[Q_1]$  este vectorul constituit cu componentele din vectorul [Q] al sistemului matricial general, corespunzătoare nodurilor implicate în elementele cu sursă internă.

Pentru matricea V componentele  $v_{ik}$  se pot defini ca în relația (137):

$$v_{ik} = \begin{cases} \frac{J_k}{6} & \text{dacă nodul } i \text{ aparține elementului } k \\ 0 & \text{dacă nodul } i \text{ nu aparține elementului } k \end{cases} \quad (137)$$

În sistemul de ecuații (135), respectiv (136), se poate determina complet matricea [V] și vectorul  $[Q_1]$ , urmând ca sistemul să fie rezolvat în raport cu necunoscutele din vectorul [L]. În sistemele de ecuații de mai sus, dacă numărul de noduri  $n$  este mai mare decât numărul elementelor  $m$  pe care le definesc<sup>58</sup> (cazul cel mai frecvent), se poate renunța la câteva ecuații pentru a aduce matricea [V] la forma  $m \times m$  și vectorul  $[Q_1]$  la tipul  $m$ ,

<sup>57</sup> Unde termenul  $L_k$  este echivalentul lui  $q$  din relația (119)

<sup>58</sup> Această discuție se face numai pentru zona de rețea în care apar elementele cu sursele de căldură și nu pentru toată rețeaua, unde raportul număr total de noduri per număr total de elemente, poate fi arbitrar



sistemul fiind astfel compatibil determinat și putând fi rezolvat prin metoda eliminării Gauss.

Alegerea ecuațiilor trebuie făcută astfel încât matricea  $[V]$  să nu fie singulară. Având în vedere particularitățile matricei  $[V]$ <sup>59</sup>, această situație poate fi generată de:

- selectarea unor ecuații pentru construirea matricei  $[V]$ , astfel încât pe două coloane diferite din  $[V]$ , elementele nule și nenule să corespundă ca poziție;

- selectarea unor ecuații pentru construirea matricei  $[V]$  astfel încât o linie din matricea  $[V]$  să fie nulă.

Aceste două situații au fost luate în considerare în algoritmul elaborat de autor și sunt eliminate a priori într-un proces iterativ.

Deși în modelările făcute nu a fost cazul, orice altă situație care ar anula determinantul matricei  $[V]$ , fără să fie depistată în algoritmul pentru cazurile de mai sus, este tratată la rezolvarea sistemului. Pentru eliminarea singularității în această situație se impune reanalizarea modului de distribuire a surselor termice pe elementele definite, sau modificarea rețelei de elemente finite.

Același tratament este necesar și dacă numărul  $n$  al nodurilor implicate este mai mic decât  $m$ , numărul elementelor cu sursă internă de căldură, ceea ce nu oferă suficiente ecuații pentru a rezolva sistemul.

În figura 13 este prezentat modelul general folosit pentru analiza efectuată, cu precizarea că pentru nodurile ce definesc elementele cu sursă internă, coordonatele au fost modificate în funcție de adâncimea de așchiere, tipul plăcuței și materialul folosit. În anexa 8 sunt prezentate rezultatele unui test de la procesul formulării normale și inverse cu metoda elementului finit, folosit pentru testarea algoritmilor implementați. În acest exemplu, ca și pe structura folosită la modelarea procesului de așchiere, diferențele între datele inițiale și cele rulate într-un ciclu direct-invers sunt sub 1%, ceea ce dovedește corectitudinea raționamentului.

Pentru rețeaua de elemente finite plane triunghiulare ortotrope, care modelează vârful sculei ca în figura 60, în cazul analizării încărcării termice la vârful sculei, sistemul de ecuații (135) devine cel din relația (138), considerând că elementele cu sursă internă de căldură sunt cele din tabelul 13:

<sup>59</sup> Se are în vedere faptul că elementele nenule din aceeași coloană sunt egale



Structura generală folosită la  
modelarea inversă

Noduri:	104
Elemente :	166
Elemente cu sursă internă:	6
Noduri implicate:	8
Materiale:	2

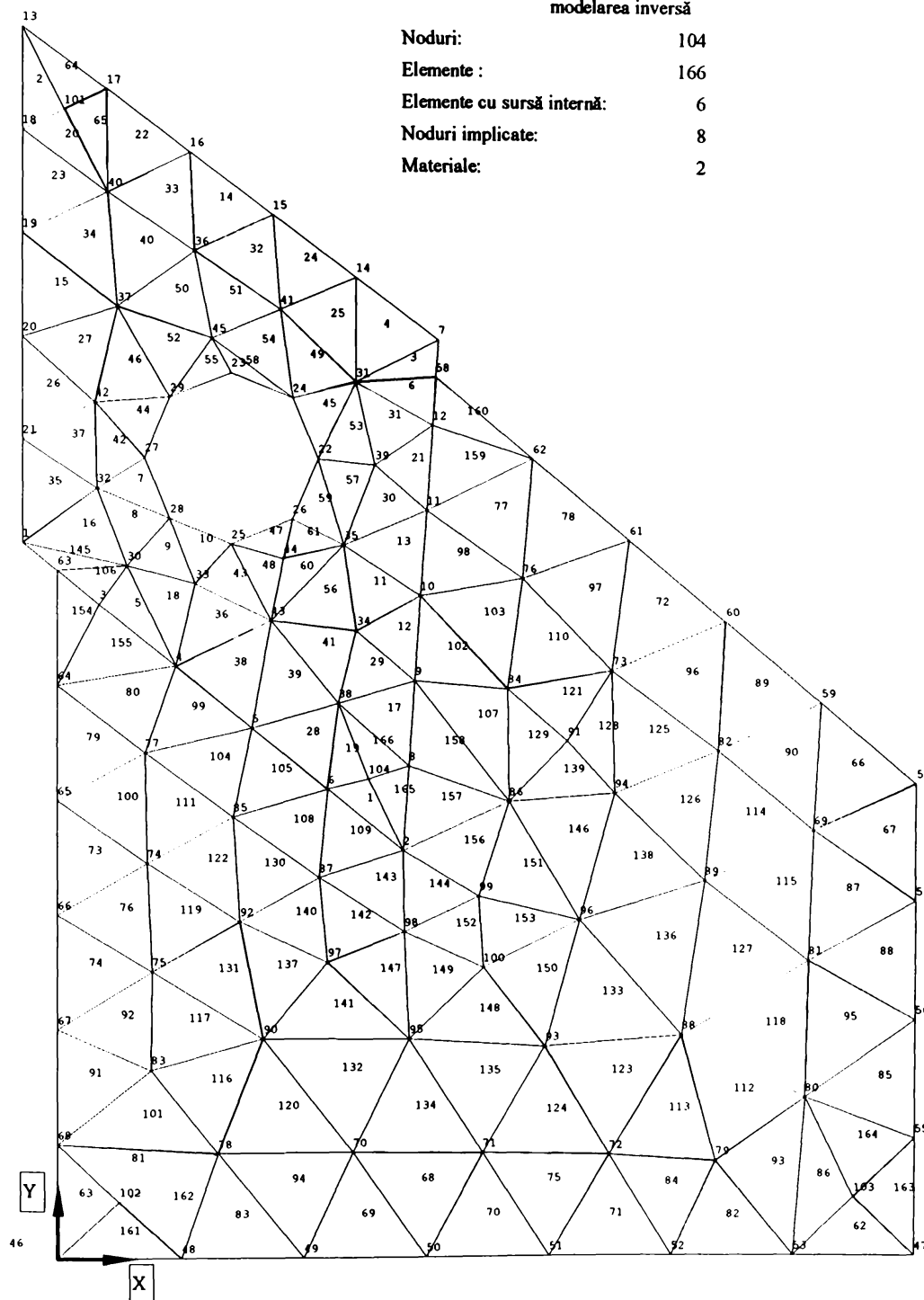


Fig. 90 Structura generală folosită pentru modelarea inversă<sup>60</sup>

<sup>60</sup> Coordonatele nodurilor din zona sursei termice sunt variabile în funcție de regimurile de așchiere, material și tipul plăcuței; în anexa 7 este prezentat un fișier de intrare cu datele corespunzătoare.

tab. 13 Definiere elemente cu sursă termică

Element	Nod 1	Nod 2	Nod 3
2	13	18	19
15	19	20	37
20	17	18	40
23	18	19	40
34	19	37	40
40	36	37	40

În figura 91 este prezentată structura redusă ce s-a folosit pentru termogramele care nu cuprind toată suprafața modelului anterior cu elemente finite

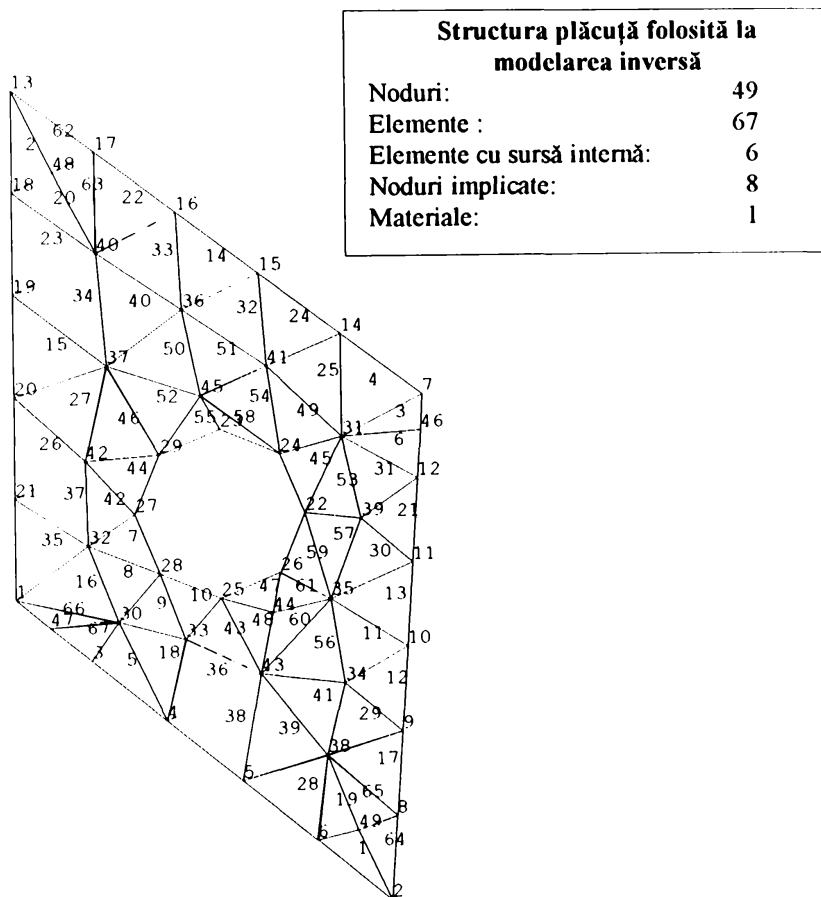


Fig. 91 Structură redusă pentru model cu elemente finite

$$\begin{array}{c}
 \text{Elemente} \\
 \begin{array}{cccccc}
 & 2 & 15 & 20 & 23 & 34 & 40 \\
 13 & \left[ \begin{array}{c} \frac{J_2}{6} \\ \frac{J_2}{6} \\ \frac{J_2}{6} \end{array} \right. & 0 & 0 & 0 & 0 & 0 \\
 17 & & 0 & \frac{J_{20}}{6} & 0 & 0 & 0 \\
 18 & & 0 & \frac{J_{20}}{6} & \frac{J_{23}}{6} & 0 & 0 \\
 19 & 0 & \frac{J_{15}}{6} & 0 & \frac{J_{23}}{6} & \frac{J_{34}}{6} & 0 \\
 20 & 0 & \frac{J_{15}}{6} & 0 & 0 & 0 & 0 \\
 36 & 0 & 0 & 0 & 0 & 0 & \frac{J_{40}}{6} \\
 37 & 0 & \frac{J_{15}}{6} & 0 & 0 & \frac{J_{34}}{6} & \frac{J_{40}}{6} \\
 40 & 0 & 0 & \frac{J_{20}}{6} & \frac{J_{23}}{6} & \frac{J_{34}}{6} & \frac{J_{40}}{6}
 \end{array}
 \right]
 \begin{array}{c}
 \left[ \begin{array}{c} L_2 \\ L_{15} \\ L_{20} \\ L_{23} \\ L_{34} \\ L_{40} \end{array} \right]
 \end{array}
 =
 \begin{array}{c}
 \left[ \begin{array}{c} q_{13} \\ q_{17} \\ q_{18} \\ q_{19} \\ q_{20} \\ q_{36} \\ q_{37} \\ q_{40} \end{array} \right]
 \end{array}
 \quad (138)
 \end{array}$$

Una din variantele care permit construirea unei matrici [V] nesingulare este prin păstrarea ecuațiilor corespunzătoare nodurilor 13, 18, 19, 20, 36, 40, ceea ce produce sistemul din relația (139), care este compatibil determinat și permite calcularea tuturor încărcărilor  $L_i$  pentru elementele considerate:

$$\begin{array}{rcl}
 \frac{J_2}{6} L_2 + 0 + 0 + 0 + 0 + 0 & = & q_{13} \\
 \frac{J_2}{6} L_2 + 0 + \frac{J_{20}}{6} L_{20} + \frac{J_{23}}{6} L_{23} + 0 + 0 & = & q_{18} \\
 \frac{J_2}{6} L_2 + \frac{J_{15}}{6} L_{15} + 0 + \frac{J_{23}}{6} L_{23} + 0 + 0 & = & q_{19} \\
 0 + \frac{J_{15}}{6} L_{15} + 0 + 0 + 0 + \frac{J_{15}}{6} L_{40} & = & q_{20} \\
 0 + \frac{J_{15}}{6} L_{15} + 0 + 0 + \frac{J_{15}}{6} L_{34} + \frac{J_{15}}{6} L_{40} & = & q_{36} \\
 0 + 0 + \frac{J_{15}}{6} L_{20} + \frac{J_{15}}{6} L_{23} + \frac{J_{15}}{6} L_{34} + \frac{J_{15}}{6} L_{40} & = & q_{40}
 \end{array} \quad (139)$$

Deci rezumând, pașii importanți pentru determinarea încărcărilor  $L_i$  prin metoda elaborată de autor sunt:

- citirea datelor de intrare din fișierul ASCII cu extensia adb;
- construirea matricelor de conductivitate la nivel de element, cu impunerea condițiilor limită de transmitere prin convecție, radiație și flux uniform distribuit;
- asamblarea matricii globale de conductivitate;
- salvarea coloanelor din matricea [K], ce vor fi afectate de aplicarea condițiilor limită;

- transformarea matricei [K] prin aplicarea condițiilor de contur inițiale de tip Dirichlet, introduse în secțiunea BCINI;
- calcularea vectorului [Q] din relația (133)
- reconstituirea vectorului [Q] inițial înainte de aplicarea condițiilor limită, folosind valorile din coloanele salvate anterior;
- completarea unei liste simplu înlănțuite, bazată pe structura de definire a elementelor, cu elementele care au sursă termică;
- completarea unui vector cu numărul nodurilor care apar în componența elementelor din lista de mai sus;
- ordonarea crescătoare și eliminarea aparițiilor multiple în vectorul respectiv;
- alcătuirea unei matrici intermediare pentru a selecta ecuațiile potrivite; matricea este analoagă cu cea din relația (135), cu diferența că pentru rapiditatea selectării ecuațiilor potrivite, elementele nenule conțin numărul nodului atașat; pentru situația din figura 90, matricea este cea din relația (140)

$$\begin{array}{cccccccc}
 13 & 17 & 18 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 19 & 20 & 0 & 37 & 0 \\
 0 & 17 & 18 & 0 & 0 & 0 & 0 & 40 \\
 0 & 0 & 18 & 19 & 0 & 0 & 0 & 40 \\
 0 & 0 & 0 & 19 & 0 & 0 & 37 & 40 \\
 0 & 0 & 0 & 0 & 0 & 36 & 37 & 40
 \end{array} \tag{140}$$

- alegerea unei combinații de coloane pentru completarea matricii [V], care să producă un determinant nenul; acest lucru este asigurat printr-un algoritm recursiv foarte compact și eficient implementat de funcțiile *Scoate*, *Singular* și *Identice*;
- completarea matricii [V] și a matricii [Q<sub>1</sub>];
- alcătuirea sistemului final și rezolvarea lui prin metoda Gauss.

Pentru a păstra unitatea datelor de intrare în programul realizat și compatibilitatea cu problema normală, temperaturile pentru toate nodurile sunt introduse ca și condiții limită, în secțiunea corespunzătoare a fișierului de intrare, delimitată de linia #BC. Valorile numerice au fost obținute după prelucrarea termogramelor obținute pentru diverse condiții de prelucrare.

Condițiile limită inițiale s-au folosit pentru modificarea sistemului primar de ecuații algebrice și s-au introdus prin secțiunea de date #BCINI, în aceeași sintaxă ca și condițiile limită normale. Ele pot fi oricare din temperaturile nodale din secțiunea #BC, mai puțin cele implicate în elementele care se consideră a avea sursă de căldură.

Indicarea elementelor care au sursă internă de căldură și care au produs câmpul termic de mai sus, se face în secțiunea încărcărilor, delimitată

tată de linia #LOADS în fișierul de intrare. Valoarea numerică precizată în situația problemei inverse nu este luată în considerare, ea fiind calculată în final după parcurgerea pașilor de mai sus.

După rularea în acest program a tuturor datelor corespunzătoare prelucrărilor, s-au obținut pentru puterea sursei termice valorile din tabelele 21-tab.24.

În aceleași tabele se află valorile forțelor determinate experimental prin tensometrie, valorile forțelor calculate analitic prin relațiile din [25]<sup>61</sup> și puterea de așchiere calculată. Tabelele prezintă cele trei variabile  $t$ ,  $s$ ,  $v$  grupate împreună cu valorile parametrilor urmăriți, pe plăcuțe și materiale așa cum s-a realizat experimentul. În tabelul 14 sunt prezentate valorile extreme ale forțelor și puterii sursei așa cum rezultă din combinațiile folosite de parametri.

tab. 14 Valori extreme pentru puterea sursei și forță

Material	$F_{min}$ [daN]	$F_{max}$ [daN]	$Q_{min}$ [W/mm <sup>2</sup> ]	$Q_{max}$ [W/mm <sup>2</sup> ]
EN24T	120.0879	182.3079	150.5696	355.9053
EN8	50.0279	81.4588	67.2260	157.2529
EN1A	35.9327	76.2019	53.1575	157.5364
Al99	7.1824	16.7606	11.66	35.9152

După cum se observă, rezistența admisibilă la rupere,  $\sigma_r$  pentru materialul prelucrat, a jucat un rol determinant în valorile forțelor și puterilor termice calculate. Se remarcă faptul că puterile termice maxime pentru EN8 și EN1A nu se încadrează în această observație, explicația fiind în faptul că deși EN8 are  $\sigma_r$  cu circa 30% mai mare, EN1A fiind un oțel moale, formează așchii mai lungi, care produc o frecare mai mare pe plăcuța de așchiere, iar suprafața mai mare a așchiei este o sursă radiantă suplimentară care afectează termogramele.

În figura 92 este prezentat un grafic de variație a puterii sursei termice obținut cu toate valorile experimentate, grupate pe materiale. Acest grafic subliniază influența pe care o are materialul semifabricatului asupra puterii sursei.

<sup>61</sup> Dumitraș, C., Militaru, C., *Op.cit.* p. 107-110

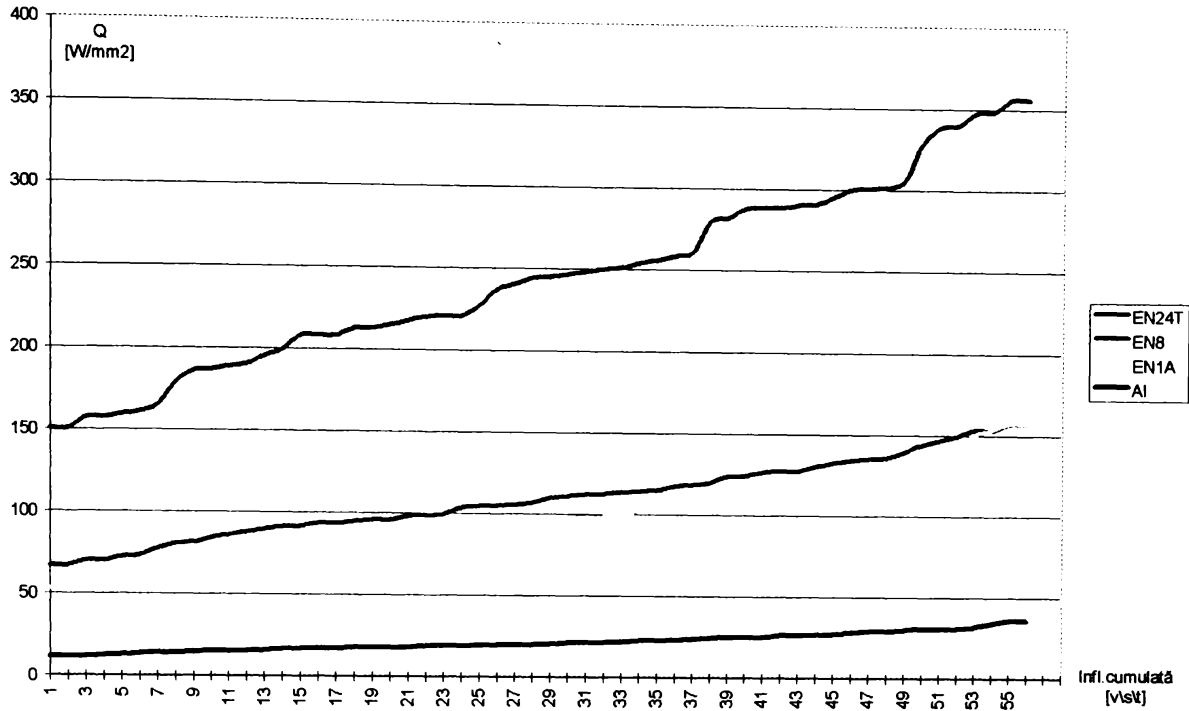


Fig. 92 Variația puterii sursei la influențe cumulate  $v, s, t$

În figura 93 este prezentată influența pe care o are creșterea avansului asupra puterii sursei termice, pentru oțelul EN24T și plăcuța K68, la o adâncime de așchiere de 0,5mm.

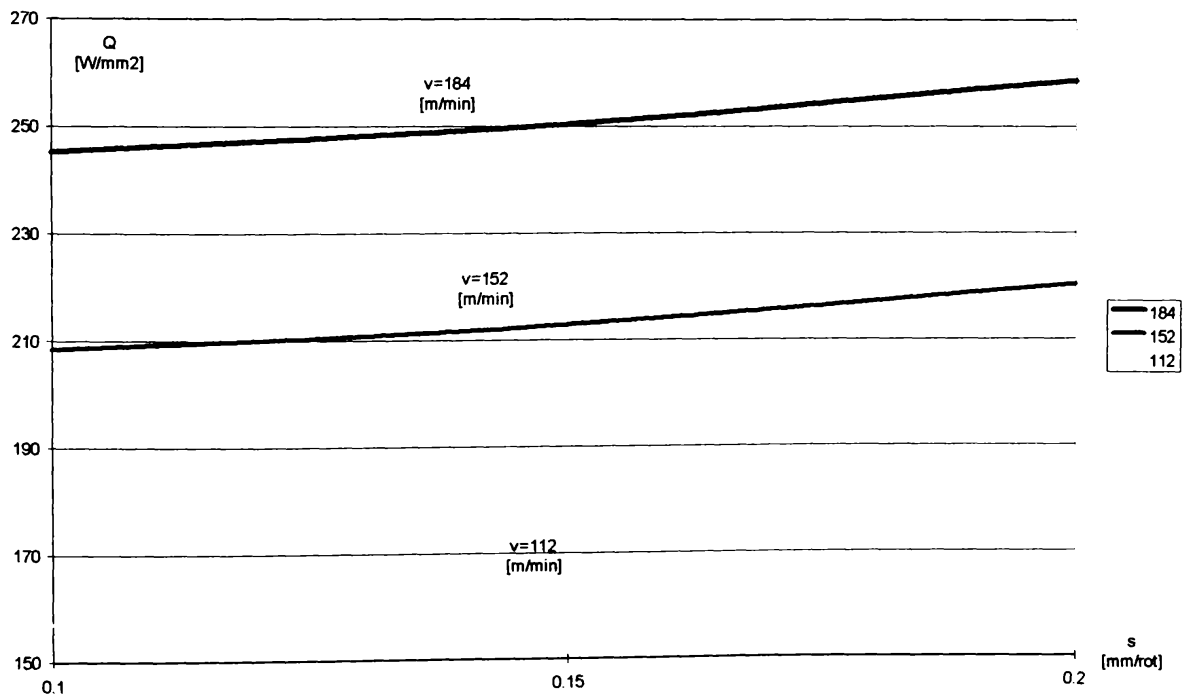


Fig. 93 Variația puterii sursei termice față de avans la EN24T \ K68 \ 0,5

În figura 94 este reprezentată grafic influența adâncimii de așchiere asupra puterii sursei termice, pentru EN24T, la un avans constant de 0,2 [mm/rot].

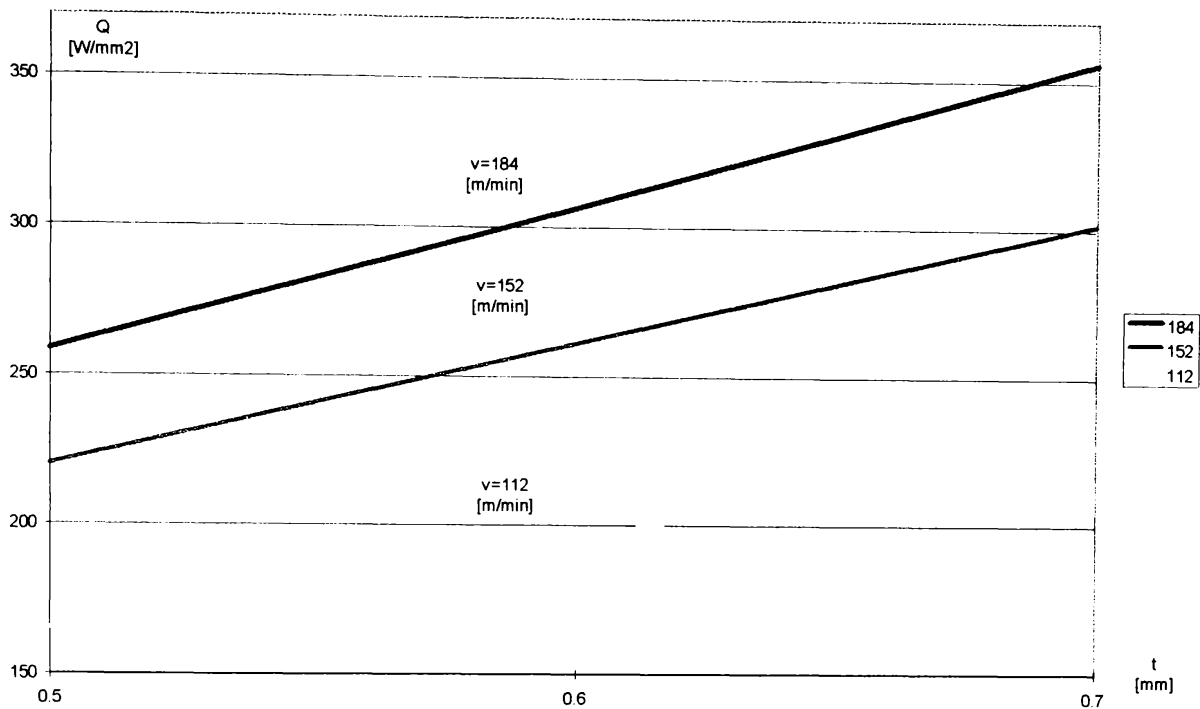


Fig. 94 Variația puterii sursei termice față de adâncimea de așchiere la EN24T \ K68 \ 0,2

Întrucât evoluția factorilor de influență pentru celelalte materiale și combinații experimentate, este asemănătoare cu cele de mai sus, s-a apreciat că nu mai este oportună reprezentarea grafică a acestora.

În concluzie se poate spune că influența cea mai mare asupra puterii sursei termice este exercitată în ordine de: materialul semifabricatului, viteza de așchiere, adâncimea de așchiere și avansul lucru.

### 5.3.2. Obținerea relațiilor de estimare a puterii sursei

Pentru estimarea puterii sursei termice, s-au preluat datele din experimentul factorial  $3^3$ , cu variabile t, v, s și s-au prelucrat prin regresie liniară, folosind metoda celor mai mici pătrate. Prin acest tratament s-a urmărit obținerea unor relații de forma:

$$Q = a_0 \cdot t + a_1 \cdot v + a_2 \cdot s + a_3 \quad (141)$$

Aceste aspecte se pot urmări în detaliu în tabelele 25-32, care sunt chiar foile de calcul Excel<sup>62</sup>, unde s-au făcut analizele.

Pentru conciziune, în tabelele 15 și 16 sunt prezentați coeficienții polinomului de forma (141), pentru puterea sursei termice și forță.

<sup>62</sup> În formatul curent al foilor de calcul Excel, datele sunt afișate prin rotunjire în funcție lățimea coloanei, însă precizia cu care s-a făcut calculul este de circa 16 zecimale

tab. 15 Coeficienții polinomiali pentru puterea sursei termice

$a_0$	$a_1$	$a_2$	$a_3$	$a_0$	$a_1$	$a_2$	$a_3$
$t$	$v$	$s$	*	$t$	$v$	$s$	*
EN24T							
K68				KC990			
392	1.51	131	-2.34	386	1.47	127	-229
ENS							
KC950				KC910			
166	0.64	84.7	-106	164	0.68	89.4	-108
EN1A							
KC850				KC910			
113	0.57	88.9	-96	124	0.61	90.8	-105
Al							
K68				KC910			
26.9	0.11	16.8	-19	28	0.14	10.3	-20

tab. 16 Coeficienții polinomiali pentru forță

$a_0$	$a_1$	$a_2$	$a_3$	$a_0$	$a_1$	$a_2$	$a_3$
$t$	$v$	$s$	*	$t$	$v$	$s$	*
EN24T							
K68				KC990			
239	-0.008	82.3	7.27	241	-0.08	82.6	6.05
ENS							
KC950				KC910			
97.23	-0.04	47.94	0.106	100.1	-0.04	57.93	0.22
EN1A							
KC850				KC910			
62.5	-0.03	52.3	-3.59	71.2	-0.03	50.9	-3.62
Al							
K68				KC910			
15.64	-0.02	9.384	0.894	16.08	-0.01	10.05	-0.44

Verificarea relațiilor pentru gama de prelucrări experimentată, este prezentată în tabelele 37-43, iar în tabelul 17 este prezentată sinteza acestor verificări:



tab. 17 Tabel de apreciere a preciziei relațiilor de forță și putere [%]

EN24T							
F				Q			
K68		KC990		K68		KC990	
Min	Max	Min	Max	Min	Max	Min	Max
0.74	0.99	-0.94	1.11	-5.04	6.51	-5.01	6.12
EN8							
KC950		KC910		KC950		KC910	
Min	Max	Min	Max	Min	Max	Min	Max
-1.7	1.05	-1.12	1.38	-4.82	6.96	-5.41	7.6
EN1A							
KC850		KC910		KC850		KC910	
Min	Max	Min	Max	Min	Max	Min	Max
-2.35	2.1	-2.2	2.6	-8.48	14.83	-7.25	12.57
Al							
K68		KC910		K68		KC910	
Min	Max	Min	Max	Min	Max	Min	Max
-13.25	11.79	-4.75	7.85	-11.54	21.81	-21.02	14.39

După cum se observă, valorile calculate cu relațiile obținute sunt în limite acceptabile la oțeluri pentru putere  $\pm (5\div 10) \%$  și pentru forțe  $\pm(0.7\div 2)\%$ . La aluminiu însă, aceste valori ajung la  $\pm 20 \%$  pentru putere și  $\pm(4\div 12) \%$  pentru forțe. Diferențele se explică prin faptul că valorile foarte mici măsurate la prelucrarea aluminiului au fost afectate destul de serios de erorile introduse de echipamente și de prelucrarea repetată în lanț a datelor măsurate.

Pornind de la observația că în relația teoretică a forței din [25] nu apare viteza de așchiere, s-au mai determinat o serie de relații de forma (142), pentru puterea sursei termice, folosind regresia polinomială de ordinul doi, având ca variabile forța și viteza de așchiere.

$$Q = a_0 \cdot v^2 + a_1 \cdot F^2 + a_2 \cdot F \cdot v + a_3 \cdot v + a_4 \cdot F + a_5, \quad (142)$$

unde:  $F$  este forța de așchiere și  $v$  este viteza.

Partea aferentă calculului pentru obținerea acestor relații, poate fi urmărită în tabelele 37-44, în tabelul 18 fiind prezentați numai coeficienții polinomiali ai funcției  $Q$ .

tab. 18 Coeficienți polinomiali pentru relațiile  $Q(f, v)$

Material	Plăcuță	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
		$v^2$	$F^2$	$F \cdot v$	$v$	$F$	*
EN24T	K68	1.42E-5	0.000123	0.010412	0.076549	0.042675	-7.36256
	KC990	-0.00051	-0.00155	0.010513	0.163533	0.495469	-47.0619
EN8	KC950	-0.00025	0.00835	0.01241	0.00457	-1.18883	37.2195
	KC910	0.00048	-0.00534	0.01152	-0.16922	0.61197	-6.0814
EN1A	KC850	-0.00017	-0.00664	0.012918	0.005211	0.55996	-11.9426
	KC910	0.000204	-0.00295	0.011672	-0.07007	0.334405	-2.17425
Al	K68	-0.00024	0.009594	0.01874	0.075495	-0.15119	-3.96245
	KC910	-0.00016	-0.02795	0.01686	0.064879	0.751596	-7.84037

Rezultatele obținute cu aceste relații, comparate cu valorile măsurate, sunt prezentate în tabelele 45-48.

Pentru valorile calculate s-au obținut diferențe față de valorile experimentale. Acestea sunt sintetizate în tabelul 19.

tab. 19 Tabel pentru estimarea preciziei relațiilor  $Q=f(F,v)$

Material	Plăcuță	Min[%]	Max [%]
EN24T	K68	-1.35	0.73
	KC990	-1.36	0.87
EN8	KC950	-1.36	1.63
	KC910	-2.47	1.92
EN1A	KC850	-4.7	2.55
	KC910	-1.92	1.95
Al	K68	-2.3	2.56
	KC910	-13.13	10.88

Analizând datele, se observă și aici același aspect ca la grupa anterioară de relații și anume faptul că pentru oțeluri, relațiile au o precizie foarte bună:  $\approx \pm 2.5\%$ , iar pentru aluminiu valorile sunt mai mari: ( $\approx \pm 12\%$ ). Explicația este aceeași ca în cazul precedent.

Relativ la calitatea aproximării cu cele două grupe de relații, după cum era de așteptat, regresia polinomială de gradul 2 oferă o precizie mai bună, lucru care se poate vedea în graficul din figura 95:

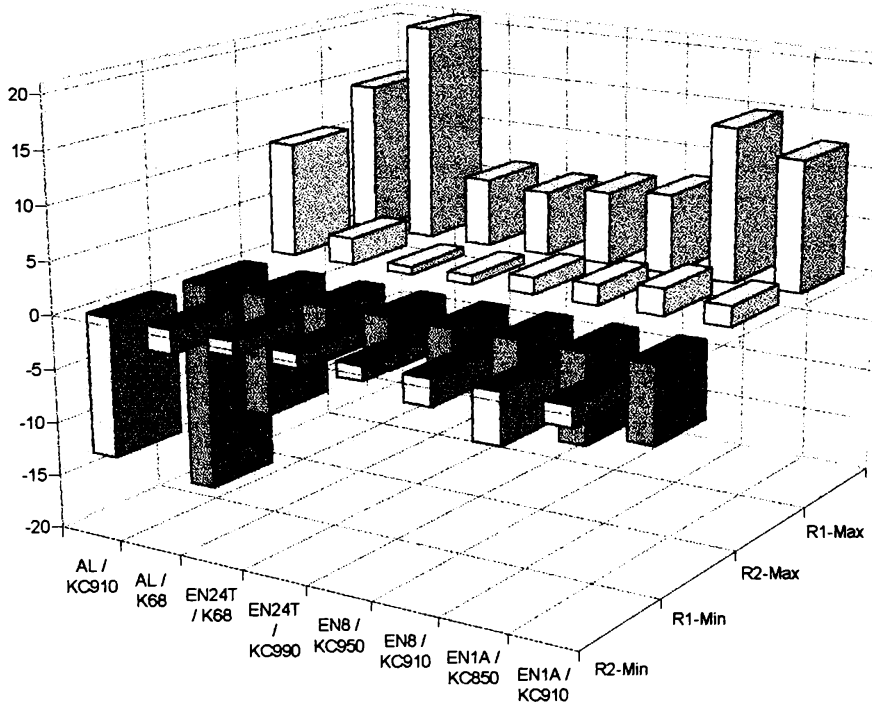


Fig. 95 Grafic comparativ pentru precizia relațiilor polinomiale liniare și de ordinul 2

Reprezentările grafice ale funcțiilor  $Q=f(F, V)$  sunt prezentate în figurile 96-104.

#### 5.4. ESTIMAREA PRECIZIEI RELAȚIILOR OBȚINUTE

Pentru estimarea preciziei relațiilor de forma (142) din tabelul 15, s-au făcut mai multe rulări ale problemei directe a metodei elementului finit. Valorile parametrilor au fost cele din tabelul 20.

Tab. 20 Estimarea preciziei relațiilor

Material	Plăcuță	F [daN]	V [m/min]	Q [W/mm <sup>2</sup> ]	s [mm/rot]	t [mm]	Tmax Calc [°C]	Tmax Măs [°C]	Dif [%]
EN24T	KC990	130	152	211.9	0.2	0.5	429	432	0.69
EN8	KC950	72.5	184	149.3	0.25	0.7	299	290	3.01
EN1A	KC850	72.5	184	92.8	0.35	0.7	219	207	5.47
Al	KC910	12	153	23.2	0.15	0.7	82	98	16.32

După cum se observă, valorile pentru oțeluri prezintă diferențe mici, însă cele pentru aluminiu sunt destul de mari, cauzele pentru acest comportament fiind cele care au provocat și abaterile pentru relațiile de calcul a puterii termice și forțelor și anume: mărimile măsurate la aluminiu au avut valori mici, care au fost în măsură mai mare afectate de factorii perturbatori cu valoare constantă (zgomotul termic de fond al atmosferei, surse radiante parazite, zgomotul de fond al echipamentelor electronice).

5.4.1. Date experimentale pentru forță și debit

tab. 21

EN24T

Combinatii K63								
Nr.	Avans (s)	Turația (n)	Adâncime (t)	Diametrul	Viteza	F. măsurată	F. calculată	Debit sursă
	[mm/rot]	[rot/min]	[mm]	[mm]	[m/min]	[daN]	[daN]	[W/mm2]
1	0.2	800	0.5	73.5	184.726	127.46971	103.32528	258.58741
2	0.15	800	0.5	73.5	184.726	123.62883	100.24771	250.35507
3	0.1	800	0.5	73.5	184.726	120.08794	97.297794	245.26702
4	0.2	540	0.5	90	152.681	130.30991	103.32528	220.27569
5	0.15	540	0.5	90	152.681	126.34322	100.24771	212.83653
6	0.1	540	0.5	90	152.681	123.4042	97.297794	208.43012
7	0.2	370	0.5	97	112.752	133.18833	103.32528	165.19257
8	0.15	370	0.5	97	112.752	129.4506	100.24771	161.26731
9	0.1	370	0.5	97	112.752	125.02602	97.297794	157.03677
10	0.2	800	0.6	73.5	184.726	151.36443	122.7819	306.42387
11	0.15	800	0.6	73.5	184.726	147.42758	119.42969	301.08202
12	0.1	800	0.6	73.5	184.726	143.51642	116.20977	290.66007
13	0.2	540	0.6	90	152.681	155.33136	122.7819	261.16456
14	0.15	540	0.6	90	152.681	151.4664	119.42969	256.07015
15	0.1	540	0.6	90	152.681	147.16251	116.20977	248.51997
16	0.2	370	0.6	97	112.752	157.25452	122.7819	195.42403
17	0.15	370	0.6	97	112.752	152.87564	119.42969	188.97584
18	0.1	370	0.6	97	112.752	149.26077	116.20977	186.06169
19	0.2	800	0.7	73.5	184.726	175.74941	142.24943	355.9053
20	0.15	800	0.7	73.5	184.726	171.67476	138.61835	347.52939
21	0.1	800	0.7	73.5	184.726	166.91396	135.12496	340.28671
22	0.2	540	0.7	90	152.681	179.61855	142.24943	301.65263
23	0.15	540	0.7	90	152.681	175.10477	138.61835	295.60677
24	0.1	540	0.7	90	152.681	171.29265	135.12496	288.74602
25	0.2	370	0.7	97	112.752	182.118	142.24943	227.31071
26	0.15	370	0.7	97	112.752	177.86472	138.61835	220.99082
27	0.1	370	0.7	97	112.752	174.195	135.12496	217.85401
Combinatii KC990								
1	0.2	800	0.5	73.5	184.726	127.38915	103.32528	251.40677
2	0.15	800	0.5	73.5	184.726	123.41029	100.24771	244.60409
3	0.1	800	0.5	73.5	184.726	120.21211	97.297794	237.44343
4	0.2	540	0.5	90	152.681	130.82537	103.32528	212.63973
5	0.15	540	0.5	90	152.681	127.27202	100.24771	208.05347
6	0.1	540	0.5	90	152.681	122.80654	97.297794	199.35558
7	0.2	370	0.5	97	112.752	132.63307	103.32528	159.35156
8	0.15	370	0.5	97	112.752	129.09352	100.24771	157.46856
9	0.1	370	0.5	97	112.752	125.42244	97.297794	150.56964
10	0.2	800	0.6	73.5	184.726	152.07526	122.7819	300.28032
11	0.15	800	0.6	73.5	184.726	146.99363	119.42969	288.85976
12	0.1	800	0.6	73.5	184.726	143.61847	116.20977	282.42258
13	0.2	540	0.6	90	152.681	155.15762	122.7819	254.00331
14	0.15	540	0.6	90	152.681	151.22023	119.42969	246.64716
15	0.1	540	0.6	90	152.681	147.5534	116.20977	240.7642
16	0.2	370	0.6	97	112.752	158.30771	122.7819	190.47521
17	0.15	370	0.6	97	112.752	153.921	119.42969	186.79311
18	0.1	370	0.6	97	112.752	148.87554	116.20977	178.79865
19	0.2	800	0.7	73.5	184.726	176.09091	142.24943	348.5154
20	0.15	800	0.7	73.5	184.726	171.33777	138.61835	338.70869
21	0.1	800	0.7	73.5	184.726	167.30093	135.12496	328.93346
22	0.2	540	0.7	90	152.681	179.32057	142.24943	291.21734
23	0.15	540	0.7	90	152.681	175.54836	138.61835	287.91262
24	0.1	540	0.7	90	152.681	170.66967	135.12496	279.94567
25	0.2	370	0.7	97	112.752	182.3079	142.24943	220.89977
26	0.15	370	0.7	97	112.752	177.89676	138.61835	215.0822
27	0.1	370	0.7	97	112.752	173.40271	135.12496	207.99225



tab. 22

## EN8

Combinatii KC950								
Nr.	Avans (s)	Turația (n)	Adâncime (t)	Diametrul	Viteza	F măsurată	F calculată	Debit sursă
	[mm/rot]	[rot/min]	[mm]	[mm]	[m/min]	[daN]	[daN]	[W/mm2]
1	0.25	1040	0.5	56.5	184.6	54.187783	43.341383	113.64464
2	0.2	1040	0.5	56.5	184.6	51.999809	41.569151	109.81069
3	0.15	1040	0.5	56.5	184.6	50.027934	39.870414	105.02085
4	0.25	800	0.5	61	153.31	55.303898	43.341383	98.650013
5	0.2	800	0.5	61	153.31	53.189894	41.569151	93.781644
6	0.15	800	0.5	61	153.31	50.552112	39.870414	89.733632
7	0.25	540	0.5	67	113.663	56.113039	43.341383	72.432966
8	0.2	540	0.5	67	113.663	53.910118	41.569151	70.145478
9	0.15	540	0.5	67	113.663	51.518625	39.870414	67.226068
10	0.25	1040	0.6	56.5	184.6	63.600922	51.136637	134.23221
11	0.2	1040	0.6	56.5	184.6	60.804607	49.212115	127.52455
12	0.15	1040	0.6	56.5	184.6	59.033218	47.365196	124.04367
13	0.25	800	0.6	61	153.31	65.024707	51.136637	115.12129
14	0.2	800	0.6	61	153.31	62.97212	49.212115	111.91586
15	0.15	800	0.6	61	153.31	60.593923	47.365196	105.43347
16	0.25	540	0.6	67	113.663	66.211444	51.136637	87.86933
17	0.2	540	0.6	67	113.663	64.202951	49.212115	84.009505
18	0.15	540	0.6	67	113.663	60.896561	47.365196	80.650029
19	0.25	1040	0.7	56.5	184.6	73.186514	58.94155	153.2413
20	0.2	1040	0.7	56.5	184.6	70.970897	56.862019	149.55615
21	0.15	1040	0.7	56.5	184.6	67.731536	54.864361	144.02371
22	0.25	800	0.7	61	153.31	75.084279	58.94155	132.54411
23	0.2	800	0.7	61	153.31	71.869821	56.862019	126.46587
24	0.15	800	0.7	61	153.31	69.166425	54.864361	119.85414
25	0.25	540	0.7	67	113.663	75.892835	58.94155	98.304406
26	0.2	540	0.7	67	113.663	73.915442	56.862019	95.852919
27	0.15	540	0.7	67	113.663	70.35553	54.864361	91.673645
Combinatii KC910								
1	0.25	1040	0.5	56.5	184.6	57.502853	46.559507	118.53086
2	0.2	1040	0.5	56.5	184.6	55.14378	44.421184	114.20821
3	0.15	1040	0.5	56.5	184.6	52.2476	42.355313	106.84135
4	0.25	800	0.5	61	153.31	59.857806	46.559507	103.53965
5	0.2	800	0.5	61	153.31	56.017742	44.421184	95.814594
6	0.15	800	0.5	61	153.31	54.454856	42.355313	93.726845
7	0.25	540	0.5	67	113.663	60.617707	46.559507	77.489377
8	0.2	540	0.5	67	113.663	58.035838	44.421184	73.658678
9	0.15	540	0.5	67	113.663	55.027518	42.355313	70.259104
10	0.25	1040	0.6	56.5	184.6	67.697879	54.702527	139.63202
11	0.2	1040	0.6	56.5	184.6	65.463291	52.40067	135.30072
12	0.15	1040	0.6	56.5	184.6	61.860965	50.174578	127.35274
13	0.25	800	0.6	61	153.31	69.482326	54.702527	117.51758
14	0.2	800	0.6	61	153.31	66.888951	52.40067	112.48253
15	0.15	800	0.6	61	153.31	64.457321	50.174578	111.04551
16	0.25	540	0.6	67	113.663	70.801019	54.702527	91.130214
17	0.2	540	0.6	67	113.663	67.687169	52.40067	86.246628
18	0.15	540	0.6	67	113.663	65.276064	50.174578	81.630286
19	0.25	1040	0.7	56.5	184.6	77.729954	62.857343	157.25291
20	0.2	1040	0.7	56.5	184.6	74.477064	60.388751	153.21921
21	0.15	1040	0.7	56.5	184.6	72.043436	57.999355	146.7767
22	0.25	800	0.7	61	153.31	79.342381	62.857343	135.88423
23	0.2	800	0.7	61	153.31	76.974931	60.388751	130.29066
24	0.15	800	0.7	61	153.31	73.292223	57.999355	123.84703
25	0.25	540	0.7	67	113.663	81.458866	62.857343	104.41725
26	0.2	540	0.7	67	113.663	78.134674	60.388751	99.425814
27	0.15	540	0.7	67	113.663	74.2421	57.999355	94.767211

EN1A

Nr.	Combinatii KC850			Diametrul [mm]	Viteza [m/min]	F măsurată [daN]	F calculată [daN]	Debit sursă [W/mm2]
	Avans (s) [mm/rot]	Turația (n) [rot/min]	Adâncime (l) [mm]					
1	0.35	1040	0.5	56.5	184.6	40.034779	32.357043	90.495053
2	0.3	1040	0.5	56.5	184.6	38.190741	30.549513	88.198401
3	0.25	1040	0.5	56.5	184.6	35.93277	28.7849	80.39729
4	0.35	800	0.5	61	153.31	41.787748	32.357043	80.268175
5	0.3	800	0.5	61	153.31	39.709898	30.549513	74.461125
6	0.25	800	0.5	61	153.31	36.311874	28.7849	69.002191
7	0.35	540	0.5	67	113.663	42.179119	32.357043	60.000966
8	0.3	540	0.5	67	113.663	39.836826	30.549513	57.269189
9	0.25	540	0.5	67	113.663	37.760389	28.7849	53.157519
10	0.35	1040	0.7	56.5	184.6	52.570738	42.552165	118.64921
11	0.3	1040	0.7	56.5	184.6	50.067109	40.466662	113.5048
12	0.25	1040	0.7	56.5	184.6	48.015001	38.431857	108.39004
13	0.35	800	0.7	61	153.31	53.935405	42.552165	102.32308
14	0.3	800	0.7	61	153.31	52.148158	40.466662	97.724188
15	0.25	800	0.7	61	153.31	48.483086	38.431857	91.382798
16	0.35	540	0.7	67	113.663	55.316207	42.552165	78.727075
17	0.3	540	0.7	67	113.663	52.623166	40.466662	72.452511
18	0.25	540	0.7	67	113.663	49.951531	38.431857	69.589647
19	0.35	1040	0.9	56.5	184.6	66.002517	52.78663	150.24914
20	0.3	1040	0.9	56.5	184.6	62.387409	50.417474	140.36962
21	0.25	1040	0.9	56.5	184.6	60.110847	48.106163	135.70125
22	0.35	800	0.9	61	153.31	67.536504	52.78663	127.91649
23	0.3	800	0.9	61	153.31	63.838655	50.417474	120.04009
24	0.25	800	0.9	61	153.31	60.682748	48.106163	115.33121
25	0.35	540	0.9	67	113.663	68.70598	52.78663	96.781667
26	0.3	540	0.9	67	113.663	64.652227	50.417474	91.726923
27	0.25	540	0.9	67	113.663	62.025665	48.106163	86.678392
Combinatii KC910								
1	0.35	1040	0.5	56.5	184.6	44.470422	35.641099	98.189645
2	0.3	1040	0.5	56.5	184.6	42.176943	33.873072	93.147929
3	0.25	1040	0.5	56.5	184.6	40.532133	32.147865	89.215184
4	0.35	800	0.5	61	153.31	46.145485	35.641099	84.948172
5	0.3	800	0.5	61	153.31	43.178495	33.873072	77.504754
6	0.25	800	0.5	61	153.31	40.697612	32.147865	75.119508
7	0.35	540	0.5	67	113.663	46.008201	35.641099	61.461459
8	0.3	540	0.5	67	113.663	43.469559	33.873072	59.300462
9	0.25	540	0.5	67	113.663	42.113891	32.147865	57.197051
10	0.35	1040	0.7	56.5	184.6	58.600227	47.211105	126.84037
11	0.3	1040	0.7	56.5	184.6	56.246929	45.172323	121.33657
12	0.25	1040	0.7	56.5	184.6	53.60657	43.183122	115.49301
13	0.35	800	0.7	61	153.31	59.810795	47.211105	107.56396
14	0.3	800	0.7	61	153.31	58.003001	45.172323	105.72295
15	0.25	800	0.7	61	153.31	55.661425	43.183122	99.098887
16	0.35	540	0.7	67	113.663	61.270784	47.211105	83.814432
17	0.3	540	0.7	67	113.663	58.600812	45.172323	78.473754
18	0.25	540	0.7	67	113.663	56.188336	43.183122	75.718337
19	0.35	1040	0.9	56.5	184.6	73.032607	58.820386	157.53638
20	0.3	1040	0.9	56.5	184.6	70.619904	56.504575	153.82671
21	0.25	1040	0.9	56.5	184.6	66.891073	54.244669	143.43144
22	0.35	800	0.9	61	153.31	74.902204	58.820386	136.01745
23	0.3	800	0.9	61	153.31	71.532245	56.504575	129.31302
24	0.25	800	0.9	61	153.31	69.473764	54.244669	123.9694
25	0.35	540	0.9	67	113.663	76.201901	58.820386	103.35301
26	0.3	540	0.9	67	113.663	72.575652	56.504575	98.686685
27	0.25	540	0.9	67	113.663	70.470374	54.244669	95.145491



Aluminiu

Combinatii K68								
Nr.	Avans (s)	Turația (n)	Adâncime (f)	Diametrul	Viteza	F măsurată	F calculată	Debit sursă
	[mm/rot]	[rot/min]	[mm]	[mm]	[m/min]	[daN]	[daN]	[W/mm2]
1	0.35	1200	0.5	49	184.726	9.3463	7.0813091	20.27981
2	0.25	1200	0.5	49	184.726	7.9645	6.5149017	17.08325
3	0.15	1200	0.5	49	184.726	7.1824	5.988108	15.57247
4	0.35	1040	0.5	47	153.561	9.9363	7.0813091	18.08871
5	0.25	1040	0.5	47	153.561	8.9832	6.5149017	16.77058
6	0.15	1040	0.5	47	153.561	8.6842	5.988108	15.39327
7	0.35	800	0.5	45	113.097	10.6498	7.0813091	14.00574
8	0.25	800	0.5	45	113.097	9.4314	6.5149017	12.46669
9	0.15	800	0.5	45	113.097	8.8934	5.988108	11.65997
10	0.35	1200	0.7	49	184.726	11.4598	9.520931	25.06748
11	0.25	1200	0.7	49	184.726	10.5456	8.857289	22.84811
12	0.15	1200	0.7	49	184.726	9.9874	8.2385409	21.58655
13	0.35	1040	0.7	47	153.561	11.9863	9.520931	21.81178
14	0.25	1040	0.7	47	153.561	10.9843	8.857289	19.5835
15	0.15	1040	0.7	47	153.561	10.0834	8.2385409	17.86929
16	0.35	800	0.7	45	113.097	12.4546	9.520931	16.29255
17	0.25	800	0.7	45	113.097	11.9047	8.857289	15.4413
18	0.15	800	0.7	45	113.097	10.0982	8.2385409	13.85512
19	0.35	1200	0.9	49	184.726	15.9824	11.966893	34.22002
20	0.25	1200	0.9	49	184.726	14.6451	11.20371	31.07062
21	0.15	1200	0.9	49	184.726	13.7098	10.490787	29.20538
22	0.35	1040	0.9	47	153.561	16.0345	11.966893	28.53344
23	0.25	1040	0.9	47	153.561	15.2366	11.20371	26.86381
24	0.15	1040	0.9	47	153.561	13.9894	10.490787	25.0667
25	0.35	800	0.9	45	113.097	16.7532	11.966893	22.29322
26	0.25	800	0.9	45	113.097	15.9345	11.20371	21.57936
27	0.15	800	0.9	45	113.097	15.0834	10.490787	19.77832
Combinatii KC910								
1	0.35	1200	0.5	49	184.726	9.6087342	7.6767263	20.61336
2	0.25	1200	0.5	49	184.726	9.2600088	7.0003869	19.89556
3	0.15	1200	0.5	49	184.726	8.3683092	6.3625121	19.1371
4	0.35	1040	0.5	47	153.561	9.9085701	7.6767263	18.17607
5	0.25	1040	0.5	47	153.561	9.9307546	7.0003869	17.48203
6	0.15	1040	0.5	47	153.561	8.1764998	6.3625121	17.07475
7	0.35	800	0.5	45	113.097	10.150414	7.6767263	14.20341
8	0.25	800	0.5	45	113.097	9.8848125	7.0003869	11.88522
9	0.15	800	0.5	45	113.097	8.3741171	6.3625121	13.00279
10	0.35	1200	0.7	49	184.726	13.443308	10.227516	30.25083
11	0.25	1200	0.7	49	184.726	12.333178	9.4475929	27.09246
12	0.15	1200	0.7	49	184.726	11.096785	8.7105754	27.33155
13	0.35	1040	0.7	47	153.561	13.535685	10.227516	25.50082
14	0.25	1040	0.7	47	153.561	13.057576	9.4475929	23.79458
15	0.15	1040	0.7	47	153.561	12.171392	8.7105754	22.8761
16	0.35	800	0.7	45	113.097	14.224735	10.227516	19.08506
17	0.25	800	0.7	45	113.097	12.448213	9.4475929	17.84592
18	0.15	800	0.7	45	113.097	11.485044	8.7105754	14.91627
19	0.35	1200	0.9	49	184.726	16.27533	12.785936	35.91518
20	0.25	1200	0.9	49	184.726	15.575224	11.899758	32.05371
21	0.15	1200	0.9	49	184.726	14.698091	11.060929	30.99985
22	0.35	1040	0.9	47	153.561	16.760672	12.785936	30.79495
23	0.25	1040	0.9	47	153.561	15.826906	11.899758	29.11833
24	0.15	1040	0.9	47	153.561	14.726553	11.060929	26.54514
25	0.35	800	0.9	45	113.097	16.530245	12.785936	24.84711
26	0.25	800	0.9	45	113.097	15.772392	11.899758	23.50916
27	0.15	800	0.9	45	113.097	15.161152	11.060929	19.05677

5.4.2. Regresii liniare pentru forța de așchiere și debitul sursei termice

tab. 25

EN24T

K68		EN24T													D																		
s	v	t	Debit	Forța	s <sup>v</sup>	s <sup>t</sup>	v <sup>t</sup>	s <sup>2</sup>	v <sup>2</sup>	t <sup>2</sup>	Dx	s	v	t	Fx	s	v	t	coef.	Inv	F x s.v.t	0	1	2	3	coef.	Inv	F x s.v.t	0	1	2	3	coef.
0.2	184.726	0.5	260.32453	127.783	36.94513	0.1	92.363	0.04	34124	0.25	52.065	48.089	130.1623	25.557	23605	63.892	25.557	23605	63.892	-233.6	27	4.05	4051.4	16.2	4067.5	3.4991	-3.333	-0.006	-3.3333	-	7.26646		
0.15	184.726	0.5	252.16134	123.584	27.70885	0.075	92.363	0.0225	34124	0.25	37.824	46581	126.0807	18.538	22829	61.792	18.538	22829	61.792	131.34	4.05	0.6525	607.71	2.43	613.83	-3.333	22.222	-1E-16	-4E-14	s	82.2638		
0.1	184.726	0.5	246.93264	120.339	18.47256	0.05	92.363	0.01	34124	0.25	24.693	45615	123.4663	12.034	22230	60.17	12.034	22230	60.17	1.5115	4.051	607.71	631333	2430.858	608375	-0.006	-1E-16	4E-05	-5E-16	v	-0.0839		
0.2	152.681	0.5	218.4483	130.419	30.53628	0.1	76.341	0.04	23312	0.25	43.69	33353	109.2241	26.084	19913	65.209	26.084	19913	65.209	392.32	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.15	152.681	0.5	214.95004	127.046	22.90221	0.075	76.341	0.0225	23312	0.25	32.243	32819	107.475	19.057	19398	63.523	19.057	19398	63.523	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.1	152.681	0.5	207.14564	123.595	15.26814	0.05	76.341	0.01	23312	0.25	20.715	31627	103.5728	12.359	18871	61.797	12.359	18871	61.797	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.2	112.752	0.5	163.8866	133.534	22.55035	0.1	56.376	0.04	12713	0.25	32.777	18479	81.9433	26.707	15056	66.767	26.707	15056	66.767	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.15	112.752	0.5	160.99719	128.881	16.91276	0.075	56.376	0.0225	12713	0.25	24.15	18153	80.49859	19.332	14532	64.44	19.332	14532	64.44	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.1	112.752	0.5	156.33363	125.251	11.27518	0.05	56.376	0.01	12713	0.25	15.633	17627	78.16682	12.525	14122	62.625	12.525	14122	62.625	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.2	184.726	0.6	307.03374	152.174	36.94513	0.12	110.84	0.04	34124	0.36	61.407	56717	184.2202	30.435	28111	91.305	30.435	28111	91.305	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.15	184.726	0.6	300.4626	147.109	27.70885	0.09	110.84	0.0225	34124	0.36	45.069	55503	180.2776	22.066	27175	88.265	22.066	27175	88.265	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.1	184.726	0.6	290.17933	142.953	18.47256	0.06	110.84	0.01	34124	0.36	29.018	53604	174.1076	14.295	26407	85.772	14.295	26407	85.772	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.2	152.681	0.6	261.11377	155.007	30.53628	0.12	91.609	0.04	23312	0.36	52.223	39867	156.6683	31.001	23667	93.004	31.001	23667	93.004	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.15	152.681	0.6	254.41249	150.976	22.90221	0.09	91.609	0.0225	23312	0.36	38.162	38844	152.6475	22.646	23051	90.586	22.646	23051	90.586	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.1	152.681	0.6	248.17898	147.426	15.26814	0.06	91.609	0.01	23312	0.36	24.818	37892	148.9074	14.743	22509	88.456	14.743	22509	88.456	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.2	112.752	0.6	197.19445	157.214	22.55035	0.12	67.651	0.04	34124	0.36	39.439	22234	118.3167	31.443	17726	94.328	31.443	17726	94.328	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.15	112.752	0.6	192.71046	154.117	16.91276	0.09	67.651	0.0225	12713	0.36	28.907	21728	115.6263	23.118	17377	92.47	23.118	17377	92.47	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.1	112.752	0.6	185.05101	148.761	11.27518	0.06	67.651	0.01	12713	0.36	18.505	20865	111.0306	14.876	16773	89.257	14.876	16773	89.257	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.2	184.726	0.7	356.92946	175.214	36.94513	0.14	129.31	0.04	34124	0.49	71.386	65934	249.8506	35.043	32367	122.65	35.043	32367	122.65	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.15	184.726	0.7	345.54975	170.563	27.70885	0.105	129.31	0.0225	34124	0.49	51.832	63832	241.8848	25.584	31507	119.39	25.584	31507	119.39	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.1	184.726	0.7	337.81553	166.611	18.47256	0.07	129.31	0.01	34124	0.49	33.782	62403	236.4709	16.661	30777	116.63	16.661	30777	116.63	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.2	152.681	0.7	302.44743	179.33	30.53628	0.14	106.88	0.04	23312	0.49	60.489	46178	211.7132	35.866	27380	125.53	35.866	27380	125.53	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.15	152.681	0.7	292.58848	174.838	22.90221	0.105	106.88	0.0225	23312	0.49	43.888	44673	204.8119	26.226	26694	122.39	26.226	26694	122.39	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.1	152.681	0.7	288.06257	171.192	15.26814	0.07	106.88	0.01	23312	0.49	28.806	43982	201.6438	17.119	26138	119.83	17.119	26138	119.83	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.2	112.752	0.7	226.34767	182.473	22.55035	0.14	78.926	0.04	12713	0.49	45.27	25521	156.4434	36.495	20574	127.73	36.495	20574	127.73	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.15	112.752	0.7	221.78952	178.119	16.91276	0.105	78.926	0.0225	12713	0.49	33.268	25007	155.2527	26.718	20083	124.68	26.718	20083	124.68	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
0.1	112.752	0.7	215.82131	172.983	11.27518	0.07	78.926	0.01	12713	0.49	21.582	24334	151.0749	17.298	19504	121.09	17.298	19504	121.09	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			
4.05	4051.4	16.2	6704.87	4067.49	607.7144	2.43	2430.9	0.6525	631333	9.9	1011.6	1E+06	4093.538	613.83	608375	2483.6	613.83	608375	2483.6	0.6525	16.2	2.43	2430.9	9.9	2483.6	-3.333	-1E-14	-5E-16	t	239.386			



EN8

KC950

s	v	t	Debit	Forța	s*v	s*t	v*t	s^2	v^2	t^2	Dx	s	v	t	Fx	s	v	t	coef.
0.25	184.6	0.5	111.58323	53.6562	46.15	0.125	92.3	0.0625	34077	0.25	27.896	20598	55.79161	13.414	9904.9	26.828			
0.2	184.6	0.5	107.73752	51.3964	36.92	0.1	92.3	0.04	34077	0.25	21.548	19888	53.86876	10.279	9487.8	25.698			
0.15	184.6	0.5	103.36167	49.219	27.69	0.075	92.3	0.0225	34077	0.25	15.504	19081	51.68083	7.3829	9085.8	24.61			
0.25	153.31	0.5	97.442418	55.4292	38.32743	0.125	76.655	0.0625	23504	0.25	24.361	14939	48.72121	13.857	8497.8	27.715			
0.2	153.31	0.5	94.632066	52.8239	30.66194	0.1	76.655	0.04	23504	0.25	18.926	14508	47.31603	10.565	8098.4	26.412			
0.15	153.31	0.5	90.312121	50.7965	22.99646	0.075	76.655	0.0225	23504	0.25	13.547	13846	45.15606	7.6195	7787.6	25.398			
0.25	113.663	0.5	72.588802	56.1006	28.41571	0.125	56.831	0.0625	12919	0.25	18.147	8250.6	36.2944	14.025	6376.6	28.05			
0.2	113.663	0.5	70.658463	53.22	22.73256	0.1	56.831	0.04	12919	0.25	14.132	8031.2	35.32923	10.644	6049.1	26.61			
0.15	113.663	0.5	67.406327	51.284	17.04942	0.075	56.831	0.0225	12919	0.25	10.111	7661.6	33.70316	7.6926	5829.1	25.642			
0.25	184.6	0.6	135.31219	63.7507	46.15	0.15	110.76	0.0625	34077	0.36	33.828	24979	81.18731	15.938	11768	38.25			
0.2	184.6	0.6	128.74334	60.6945	36.92	0.12	110.76	0.04	34077	0.36	25.749	23766	77.246	12.139	11204	36.417			
0.15	184.6	0.6	124.69137	59.2508	27.69	0.09	110.76	0.0225	34077	0.36	18.704	23018	74.81482	8.8876	10938	35.55			
0.25	153.31	0.6	113.76815	65.3202	38.32743	0.15	91.986	0.0625	23504	0.36	28.442	17442	68.26089	16.33	10014	39.192			
0.2	153.31	0.6	108.69001	62.8071	30.66194	0.12	91.986	0.04	23504	0.36	21.738	16663	65.214	12.561	9628.9	37.684			
0.15	153.31	0.6	106.75387	60.5519	22.99646	0.09	91.986	0.0225	23504	0.36	16.013	16366	64.05232	9.0828	9283.2	36.331			
0.25	113.663	0.6	85.223212	66.3533	28.41571	0.15	68.198	0.0625	12919	0.36	21.306	9686.7	51.13393	16.588	7541.9	39.812			
0.2	113.663	0.6	83.604991	63.9412	22.73256	0.12	68.198	0.04	12919	0.36	16.721	9502.8	50.16299	12.788	7267.7	38.365			
0.15	113.663	0.6	79.503486	61.6254	17.04942	0.09	68.198	0.0225	12919	0.36	11.926	9036.6	47.70209	9.2438	7004.5	36.975			
0.25	184.6	0.7	154.32235	72.5529	46.15	0.175	129.22	0.0625	34077	0.49	38.581	28488	108.0256	18.138	13393	50.787			
0.2	184.6	0.7	147.43005	70.7783	36.92	0.14	129.22	0.04	34077	0.49	29.486	27216	103.201	14.156	13066	49.545			
0.15	184.6	0.7	142.64115	67.6624	27.69	0.105	129.22	0.0225	34077	0.49	21.396	26332	99.8488	10.149	12490	47.364			
0.25	153.31	0.7	133.17779	74.8073	38.32743	0.175	107.32	0.0625	23504	0.49	33.294	20417	93.22445	18.702	11469	52.365			
0.2	153.31	0.7	127.6286	72.756	30.66194	0.14	107.32	0.04	23504	0.49	25.525	19566	89.33862	14.551	11154	50.929			
0.15	153.31	0.7	121.3429	69.9807	22.99646	0.105	107.32	0.0225	23504	0.49	18.201	18603	84.94003	10.497	10729	48.986			
0.25	113.663	0.7	100.95924	76.3525	28.41571	0.175	79.564	0.0625	12919	0.49	25.24	11475	70.67146	19.088	8678.4	53.447			
0.2	113.663	0.7	95.586029	73.2399	22.73256	0.14	79.564	0.04	12919	0.49	19.117	10865	66.91022	14.648	8324.6	51.268			
0.15	113.663	0.7	92.104518	70.8069	17.04942	0.105	79.564	0.0225	12919	0.49	13.816	10469	64.47316	10.621	8048.1	49.565			
5.4	4064.15	16.2	2897.2038	1687.16	812.8306	3.24	2438.5	1.125	634502	9.9	583.25	450694	1768.269	339.59	253120	1029.8			
0	1	2	3	D x s.v.t	Inv	coef.	0	1	2	3	F x s.v.t	Inv	coef.						
27	5.4	4064.15	16.2	2897.2	3.9219	-4.444	-0.007	-3.33333	-	16.2	1687.2	-4.444	-0.007	-3.3333	-	0.106344			
5.4	1.125	812.831	3.24	583.254	-4.44444	22.222	3E-16	-6.6E-14	s	3.24	339.59	-4.444	22.222	3E-16	s	47.93922			
4064.2	812.831	634502	2438.4917	450694	-0.00662	1E-16	4E-05	-2.6E-16	v	9.9	2438.492	1E-16	4E-05	-3E-16	v	-0.03683			
16.2	3.24	2438.49	9.9	1768.27	-3.33333	-4E-14	-2E-16	5.55556	t	9.9	1029.8	-3.333	-4E-14	-2E-16	t	97.22827			

EN8

KC910

s	v	t	Debit	Forța	s*v	s*t	v*t	s^2	v^2	t^2	Dx	s	v	t	Fx	s	v	t	coef.
0.25	184.6	0.5	120.39048	58.0161	46.15	0.125	92.3	0.0625	34077	0.25		30.098	22224	60.19524	14.504	10710	29.008		
0.2	184.6	0.5	113.91984	55.7815	36.92	0.1	92.3	0.04	34077	0.25		22.784	21030	56.95992	11.156	10297	27.891		
0.15	184.6	0.5	107.91172	52.5749	27.69	0.075	92.3	0.0225	34077	0.25		16.187	19921	53.95586	7.8662	9705.3	26.287		
0.25	153.31	0.5	101.77061	59.5667	38.32743	0.125	76.655	0.0625	23504	0.25		25.443	15602	50.88531	14.892	9132.2	29.783		
0.2	153.31	0.5	96.050244	56.5846	30.66194	0.1	76.655	0.04	23504	0.25		19.21	14725	48.02512	11.317	8675	28.292		
0.15	153.31	0.5	91.423367	53.6532	22.99646	0.075	76.655	0.0225	23504	0.25		13.714	14016	45.71168	8.048	8225.6	26.827		
0.25	113.663	0.5	76.997144	60.3189	28.41571	0.125	56.831	0.0625	12919	0.25		19.249	8751.7	38.49857	15.08	6856	30.159		
0.2	113.663	0.5	74.157423	57.451	22.73256	0.1	56.831	0.04	12919	0.25		14.831	8428.9	37.07871	11.49	6530	28.725		
0.15	113.663	0.5	70.658486	54.6831	17.04942	0.075	56.831	0.0225	12919	0.25		10.599	8031.2	35.32924	8.2025	6215.4	27.342		
0.25	184.6	0.6	140.87015	67.6562	46.15	0.15	110.76	0.0625	34077	0.36		35.218	26005	84.52209	16.914	12489	40.594		
0.2	184.6	0.6	134.8473	64.5713	36.92	0.12	110.76	0.04	34077	0.36		26.969	24893	80.90838	12.914	11920	38.743		
0.15	184.6	0.6	128.16252	62.1629	27.69	0.09	110.76	0.0225	34077	0.36		19.224	23659	76.89751	9.3244	11475	37.298		
0.25	153.31	0.6	118.20275	69.319	38.32743	0.15	91.986	0.0625	23504	0.36		29.551	18122	70.92165	17.33	10627	41.591		
0.2	153.31	0.6	113.26139	66.7032	30.66194	0.12	91.986	0.04	23504	0.36		22.652	17364	67.96683	13.341	10226	40.022		
0.15	153.31	0.6	109.87988	64.4004	22.99646	0.09	91.986	0.0225	23504	0.36		16.482	16846	65.92793	9.6601	9873.2	38.64		
0.25	113.663	0.6	87.573377	70.5221	28.41571	0.15	68.198	0.0625	12919	0.36		21.893	9953.8	52.54403	17.631	8015.7	42.313		
0.2	113.663	0.6	85.578171	68.1638	22.73256	0.12	68.198	0.04	12919	0.36		17.116	9727.1	51.3469	13.633	7747.7	40.898		
0.15	113.663	0.6	83.550524	64.751	17.04942	0.09	68.198	0.0225	12919	0.36		12.533	9496.6	50.13031	9.7127	7359.8	38.851		
0.25	184.6	0.7	159.60014	78.4892	46.15	0.175	129.22	0.0625	34077	0.49		39.9	29462	111.7201	19.622	14489	54.942		
0.2	184.6	0.7	154.57135	74.9814	36.92	0.14	129.22	0.04	34077	0.49		30.914	28534	108.1999	14.996	13842	52.487		
0.15	184.6	0.7	147.60823	71.9988	27.69	0.105	129.22	0.0225	34077	0.49		22.141	27248	103.3258	10.8	13291	50.399		
0.25	153.31	0.7	135.3964	80.2481	38.32743	0.175	107.32	0.0625	23504	0.49		33.849	20758	94.77748	20.062	12303	56.174		
0.2	153.31	0.7	129.21281	76.5472	30.66194	0.14	107.32	0.04	23504	0.49		25.843	19810	90.44897	15.309	11735	53.583		
0.15	153.31	0.7	126.99773	73.1909	22.99646	0.105	107.32	0.0225	23504	0.49		19.05	19470	88.89841	10.979	11221	51.234		
0.25	113.663	0.7	101.5652	80.6995	28.41571	0.175	79.564	0.0625	12919	0.49		25.391	11544	71.09564	20.175	9172.5	56.49		
0.2	113.663	0.7	98.066777	77.3022	22.73256	0.14	79.564	0.04	12919	0.49		19.613	11147	68.64674	15.46	8786.4	54.112		
0.15	113.663	0.7	95.721992	75.285	17.04942	0.105	79.564	0.0225	12919	0.49		14.358	10680	67.00539	11.293	8557.1	52.699		
5.4	4064.15	16.2	3003.946	1795.62	812.8306	3.24	2438.5	1.125	634502	9.9		604.81	467648	1831.914	361.73	269478	1095.4		

D	0	1	2	3	Dx s.v.t	Inv	coef.	0	1	2	3	F x s.v.t	Inv	coef.
27	5.4	4064.15	16.2	3003.95	3.9219	-4.444	-107.54	27	5.4	4064.2	16.2	1795.6	3.9219	-0.21982
5.4	1.125	812.831	3.24	604.812	-4.44444	22.222	89.3909	5.4	1.125	812.83	3.24	361.73	-4.444	57.92839
4064.2	812.831	634502	2438.4917	467648	-0.00662	1E-16	0.68051	4064	812.83	634502	2438.492	269478	-0.007	-0.035466
16.2	3.24	2438.49	9.9	1831.91	-3.33333	-4E-14	164.145	16.2	3.24	2438.5	9.9	1095.4	-3.333	100.0625



EN1A

KC850

s	v	t	Debit	Forța	s*v	s*t	v*t	s^2	v^2	t^2	Dx	s	v	t	Fx	s	v	t
0.35	184.6	0.5	93.620739	40.5235	64.60999	0.175	92.3	0.1225	34077	0.25	32.767	17282	46.81037	14.183	7480.6	20.262		
0.3	184.6	0.5	84.53868	37.9891	55.38	0.15	92.3	0.09	34077	0.25	25.362	15606	42.26934	11.397	7012.8	18.995		
0.25	184.6	0.5	80.422961	36.0663	46.15	0.125	92.3	0.0625	34077	0.25	20.106	14846	40.21148	9.0166	6657.8	18.033		
0.35	153.31	0.5	76.710015	41.1583	53.6584	0.175	76.655	0.1225	23504	0.25	26.849	11760	38.35501	14.405	6310	20.579		
0.3	153.31	0.5	72.768127	38.9747	45.99292	0.15	76.655	0.09	23504	0.25	21.83	11156	36.38406	11.692	5975.2	19.487		
0.25	153.31	0.5	69.97828	36.7562	38.32743	0.125	76.655	0.0625	23504	0.25	17.495	10728	34.98914	9.189	5635.1	18.378		
0.35	113.663	0.5	57.625155	42.3237	39.78199	0.175	56.831	0.1225	12919	0.25	20.169	6549.8	28.81258	14.813	4810.6	21.162		
0.3	113.663	0.5	56.697317	39.1048	34.09885	0.15	56.831	0.09	12919	0.25	17.009	6444.4	28.34866	11.731	4444.8	19.552		
0.25	113.663	0.5	55.144179	37.9021	28.41571	0.125	56.831	0.0625	12919	0.25	13.786	6267.8	27.57209	9.4755	4308.1	18.951		
0.35	184.6	0.7	121.00035	52.6918	64.60999	0.245	129.22	0.1225	34077	0.49	42.35	22337	84.70025	18.442	9726.9	36.884		
0.3	184.6	0.7	113.97535	49.9918	55.38	0.21	129.22	0.09	34077	0.49	34.193	21040	79.78274	14.998	9228.5	34.994		
0.25	184.6	0.7	109.36894	48.0685	46.15	0.175	129.22	0.0625	34077	0.49	27.342	20190	76.55825	12.017	8873.5	33.648		
0.35	153.31	0.7	101.93232	54.6252	53.6584	0.245	107.32	0.1225	23504	0.49	35.676	15627	71.35262	19.119	8374.6	38.238		
0.3	153.31	0.7	99.148328	51.3936	45.99292	0.21	107.32	0.09	23504	0.49	29.744	15200	69.40383	15.418	7879.1	35.975		
0.25	153.31	0.7	93.891338	48.8836	38.32743	0.175	107.32	0.0625	23504	0.49	23.473	14394	65.72394	12.221	7494.3	34.218		
0.35	113.663	0.7	77.391867	54.5969	39.78199	0.245	79.564	0.1225	12919	0.49	27.087	8796.6	54.17431	19.109	6205.6	38.218		
0.3	113.663	0.7	75.229585	52.4651	34.09885	0.21	79.564	0.09	12919	0.49	22.569	8550.8	52.66071	15.74	5963.3	36.726		
0.25	113.663	0.7	70.232142	49.4309	28.41571	0.175	79.564	0.0625	12919	0.49	17.558	7982.8	49.1625	12.358	5618.5	34.602		
0.35	184.6	0.9	145.91988	65.3579	64.60999	0.315	166.14	0.1225	34077	0.81	51.072	26937	131.3279	22.875	12065	58.822		
0.3	184.6	0.9	141.62625	62.0803	55.38	0.27	166.14	0.09	34077	0.81	42.488	26144	127.4636	18.624	11460	55.872		
0.25	184.6	0.9	133.91498	59.7152	46.15	0.225	166.14	0.0625	34077	0.81	33.479	24721	120.5235	14.929	11023	53.744		
0.35	153.31	0.9	126.90426	67.2491	53.6584	0.315	137.98	0.1225	23504	0.81	44.416	19456	114.2138	23.537	10310	60.524		
0.3	153.31	0.9	120.84927	64.5229	45.99292	0.27	137.98	0.09	23504	0.81	36.255	18527	108.7643	19.357	9892	58.071		
0.25	153.31	0.9	115.72769	61.1676	38.32743	0.225	137.98	0.0625	23504	0.81	28.932	17742	104.1549	15.292	9377.6	55.051		
0.35	113.663	0.9	94.272178	68.6822	39.78199	0.315	102.3	0.1225	12919	0.81	32.995	10715	84.84496	24.039	7806.6	61.814		
0.3	113.663	0.9	89.680729	65.0441	34.09885	0.27	102.3	0.09	12919	0.81	26.904	10193	80.71266	19.513	7393.1	58.54		
0.25	113.663	0.9	86.659763	62.1047	28.41571	0.225	102.3	0.0625	12919	0.81	21.665	9850	77.99379	15.526	7059	55.894		
8.1	4064.15	18.9	2565.2307	1388.87	1219.246	5.67	2844.9	2.475	634502	13.95	773.57	399045	1877.271	419.02	208386	1017.2		

F

D

0	1	2	3	Dx s.v.t	Inv	coef.	0	1	2	3	F x s.v.t	Inv	coef.
27	8.1	4064.15	18.9	2565.23	3.713567	-0.97222	-96.47	27	8.1	4064.2	18.9	1388.9	-3.58932
8.1	2.475	1219.25	5.67	773.571	-6.66667	-7.1E-14	88.929	8.1	2.475	1219.2	5.67	419.02	52.3482
4064.2	1219.25	634502	2844.9069	399045	-0.00662	-9.8E-17	0.5677	4064	1219.2	634502	2844.907	208386	-0.02956
18.9	5.67	2844.91	13.95	1877.27	-0.97222	-8E-14	113.36	18.9	5.67	2844.9	13.95	1017.2	62.5348

EN1A

KC910

s	v	t	Debit	Forța	s*v	s*t	v*t	s^2	v^2	t^2	Dx	s	v	t	Fx	s	v	t
0.35	184.6	0.5	97.586251	44.9211	64.60999	0.175	92.3	0.1225	34077	0.25	0.25	34.155	18014	48.79313	15.722	8292.4	22.461	
0.3	184.6	0.5	91.869439	42.7397	55.38	0.15	92.3	0.09	34077	0.25	0.25	27.561	16959	45.93472	12.822	7889.8	21.37	
0.25	184.6	0.5	88.177315	40.1269	46.15	0.125	92.3	0.0625	34077	0.25	0.25	22.044	16278	44.08866	10.032	7407.4	20.063	
0.35	153.31	0.5	81.796692	44.9389	53.6584	0.175	76.655	0.1225	23504	0.25	0.25	28.629	12540	40.89835	15.729	6889.6	22.469	
0.3	153.31	0.5	77.135278	43.3119	45.99292	0.15	76.655	0.09	23504	0.25	0.25	23.141	11826	38.56764	12.994	6640.1	21.656	
0.25	153.31	0.5	74.606495	41.2691	38.32743	0.125	76.655	0.0625	23504	0.25	0.25	18.652	11438	37.30325	10.317	6327	20.635	
0.35	113.663	0.5	61.268955	45.6216	39.78199	0.175	56.831	0.1225	12919	0.25	0.25	21.444	6964	30.63448	15.968	5185.5	22.811	
0.3	113.663	0.5	59.939114	43.4922	34.09885	0.15	56.831	0.09	12919	0.25	0.25	17.982	6812.8	29.96956	13.048	4943.4	21.746	
0.25	113.663	0.5	56.749545	41.5386	28.41571	0.125	56.831	0.0625	12919	0.25	0.25	14.187	6450.3	28.37477	10.385	4721.4	20.769	
0.35	184.6	0.7	127.16098	58.6493	64.60999	0.245	129.22	0.1225	34077	0.49	0.49	44.506	23474	89.01268	20.527	10827	41.055	
0.3	184.6	0.7	121.6585	55.9569	55.38	0.21	129.22	0.09	34077	0.49	0.49	36.498	22458	85.16095	16.787	10330	39.17	
0.25	184.6	0.7	116.71507	53.7237	46.15	0.175	129.22	0.0625	34077	0.49	0.49	29.179	21546	81.70055	13.431	9917.4	37.607	
0.35	153.31	0.7	110.55637	59.8344	53.6584	0.245	107.32	0.1225	23504	0.49	0.49	38.695	16949	77.38946	20.942	9173.2	41.884	
0.3	153.31	0.7	102.62007	57.0214	45.99292	0.21	107.32	0.09	23504	0.49	0.49	30.786	15733	71.83405	17.106	8741.9	39.915	
0.25	153.31	0.7	99.984444	54.6396	38.32743	0.175	107.32	0.0625	23504	0.49	0.49	24.996	15329	69.98911	13.66	8376.8	38.248	
0.35	113.663	0.7	83.587288	60.804	39.78199	0.245	79.564	0.1225	12919	0.49	0.49	29.256	9500.8	58.5111	21.281	6911.2	42.563	
0.3	113.663	0.7	78.742116	58.3337	34.09885	0.21	79.564	0.09	12919	0.49	0.49	23.623	8950.1	55.11948	17.5	6630.4	40.834	
0.25	113.663	0.7	74.859675	55.4632	28.41571	0.175	79.564	0.0625	12919	0.49	0.49	18.715	8508.8	52.40177	13.866	6304.1	38.824	
0.35	184.6	0.9	158.06866	73.038	64.60999	0.315	166.14	0.1225	34077	0.81	0.81	55.324	29179	142.2618	25.563	13483	65.734	
0.3	184.6	0.9	151.9918	69.7405	55.38	0.27	166.14	0.09	34077	0.81	0.81	45.598	28058	136.7926	20.922	12874	62.766	
0.25	184.6	0.9	146.24018	67.4513	46.15	0.225	166.14	0.0625	34077	0.81	0.81	36.56	26996	131.6162	16.863	12452	60.706	
0.35	153.31	0.9	132.49904	75.1335	53.6584	0.315	137.98	0.1225	23504	0.81	0.81	46.375	20313	119.2491	26.297	11519	67.62	
0.3	153.31	0.9	129.38747	71.3037	45.99292	0.27	137.98	0.09	23504	0.81	0.81	38.816	19836	116.4487	21.391	10932	64.173	
0.25	153.31	0.9	123.27845	68.5737	38.32743	0.225	137.98	0.0625	23504	0.81	0.81	30.82	18900	110.9506	17.143	10513	61.716	
0.35	113.663	0.9	102.34112	75.6734	39.78199	0.315	102.3	0.1225	12919	0.81	0.81	35.819	11632	92.107	26.486	8601.2	68.106	
0.3	113.663	0.9	99.393088	73.4896	34.09885	0.27	102.3	0.09	12919	0.81	0.81	29.818	11297	89.45378	22.047	8353	66.141	
0.25	113.663	0.9	92.535931	70.0013	28.41571	0.225	102.3	0.0625	12919	0.81	0.81	23.134	10518	83.28234	17.5	7956.5	63.001	
8.1	4064.15	18.9	2740.7493	1546.79	1219.246	5.67	2844.9	2.475	634502	13.95	13.95	826.31	426460	2007.846	466.33	232190	1134	

D	0	1	2	3	D x s.v.t	Inv	coef.	0	1	2	3	F x s.v.t	Inv	coef.
27	8.1	4064.15	18.9	2740.75	3.713567	-6.667	-104.6	27	8.1	4064.2	18.9	1546.8	-6.667	-3.62244
8.1	2.475	1219.25	5.67	826.311	-6.66667	22.222	90.798	8.1	2.475	1219.2	5.67	466.33	-6.667	50.9188
4064.2	1219.25	634502	2844.9069	426460	-0.00662	6E-30	0.6115	4064	1219.2	634502	2844.907	232190	-0.007	-0.02809
18.9	5.67	2844.91	13.95	2007.85	-0.97222	-8E-14	124.06	18.9	5.67	2844.9	13.95	1134	-0.972	71.2347



Aluminiu 99

K68		F										D							
s	v	t	Debit	Forța	s*v	s*t	v*t	s^2	v^2	t^2	Dx	s	v	t	Fx	s	v	t	coef.
0.35	184.726	0.5	20.664806	9.3463	64.65398	0.175	92.363	0.1225	34124	0.25	7.2327	3817.3	10.3324	3.2712	1726.5	4.6732	0.853983		
0.25	184.726	0.5	17.104442	7.9645	46.18141	0.125	92.363	0.0625	34124	0.25	4.2761	3159.6	8.552221	1.9911	1471.2	3.9823	0.853983		
0.15	184.726	0.5	15.748344	7.1824	27.70885	0.075	92.363	0.0225	34124	0.25	2.3623	2909.1	7.874172	1.0774	1326.8	3.5912	0.853983		
0.35	153.561	0.5	18.365944	9.9363	53.74637	0.175	76.781	0.1225	23581	0.25	6.4281	2820.3	9.182972	3.4777	1525.8	4.9682	0.853983		
0.25	153.561	0.5	16.468699	8.9832	38.39026	0.125	76.781	0.0625	23581	0.25	4.1172	2529	8.234349	2.2458	1379.5	4.4916	0.853983		
0.15	153.561	0.5	15.643337	8.6842	23.03416	0.075	76.781	0.0225	23581	0.25	2.3465	2402.2	7.821669	1.3026	1333.6	4.3421	0.853983		
0.35	113.097	0.5	13.925104	10.6498	39.58407	0.175	56.549	0.1225	12791	0.25	4.8738	1574.9	6.962552	3.7274	1204.5	5.3249	0.853983		
0.25	113.097	0.5	12.496786	9.4314	28.27433	0.125	56.549	0.0625	12791	0.25	3.1242	1413.4	6.248393	2.3579	1066.7	4.7157	0.853983		
0.15	113.097	0.5	12.204186	8.8934	16.9646	0.075	56.549	0.0225	12791	0.25	1.8306	1380.3	6.102093	1.334	1005.8	4.4467	0.853983		
0.35	184.726	0.7	24.839265	11.4598	64.65398	0.245	129.31	0.1225	34124	0.49	8.6937	4588.4	17.38749	4.0109	2116.9	8.0219	0.853983		
0.25	184.726	0.7	22.54698	10.5456	46.18141	0.175	129.31	0.0625	34124	0.49	5.6367	4165	15.78289	2.6364	1948	7.3819	0.853983		
0.15	184.726	0.7	21.190751	9.9874	27.70885	0.105	129.31	0.0225	34124	0.49	3.1786	3914.5	14.83353	1.4981	1844.9	6.9912	0.853983		
0.35	153.561	0.7	21.694377	11.9863	53.74637	0.245	107.49	0.1225	23581	0.49	7.593	3331.4	15.18606	4.1952	1840.6	8.3904	0.853983		
0.25	153.561	0.7	19.54731	10.9843	38.39026	0.175	107.49	0.0625	23581	0.49	4.8868	3001.7	13.88312	2.7461	1686.8	7.689	0.853983		
0.15	153.561	0.7	18.487975	10.0834	23.03416	0.105	107.49	0.0225	23581	0.49	2.7732	2839	12.94158	1.5125	1548.4	7.0584	0.853983		
0.35	113.097	0.7	16.393384	12.4546	39.58407	0.245	79.168	0.1225	12791	0.49	5.7377	1854	11.47537	4.3591	1408.6	8.7182	0.853983		
0.25	113.097	0.7	15.736564	11.9047	28.27433	0.175	79.168	0.0625	12791	0.49	3.9341	1779.8	11.0156	2.9762	1346.4	8.3333	0.853983		
0.15	113.097	0.7	13.633631	10.0982	16.9646	0.105	79.168	0.0225	12791	0.49	2.045	1541.9	9.543542	1.5147	1142.1	7.0687	0.853983		
0.35	184.726	0.9	34.00976	15.9824	64.65398	0.315	166.25	0.1225	34124	0.81	11.903	6282.5	30.60878	5.5938	2952.4	14.384	0.853983		
0.25	184.726	0.9	31.131999	14.6451	46.18141	0.225	166.25	0.0625	34124	0.81	7.783	5750.9	28.0188	3.6613	2705.3	13.181	0.853983		
0.15	184.726	0.9	29.01	13.7098	27.70885	0.135	166.25	0.0225	34124	0.81	4.3515	5358.9	26.109	2.0565	2532.6	12.339	0.853983		
0.35	153.561	0.9	28.66773	16.0345	53.74637	0.315	138.2	0.1225	23581	0.81	10.034	4402.2	25.80096	5.6121	2462.3	14.431	0.853983		
0.25	153.561	0.9	27.756215	15.2366	38.39026	0.225	138.2	0.0625	23581	0.81	6.9391	4262.3	24.98059	3.8092	2339.7	13.713	0.853983		
0.15	153.561	0.9	25.375149	13.9894	23.03416	0.135	138.2	0.0225	23581	0.81	3.8063	3896.6	22.83763	2.0984	2148.2	12.59	0.853983		
0.35	113.097	0.9	22.600486	16.7532	39.58407	0.315	101.79	0.1225	12791	0.81	7.9102	2556.1	20.34044	5.8636	1894.7	15.078	0.853983		
0.25	113.097	0.9	21.268982	15.9345	28.27433	0.225	101.79	0.0625	12791	0.81	5.3172	2405.5	19.14208	3.9836	1802.1	14.341	0.853983		
0.15	113.097	0.9	19.56064	15.0834	16.9646	0.135	101.79	0.0225	12791	0.81	2.9341	2212.3	17.60458	2.2625	1705.9	13.575	0.853983		
6.75	4062.46	18.9	556.07285	317.945	1015.614	4.725	2843.7	1.8675	634460	13.95	142.05	86149	408.6029	81.175	47466	233.82	15.63817		
0	1	2	3	Dx s.v.t	Inv	coef.	0	1	2	3	Fx s.v.t	Inv	coef.						
27	6.75	4062.46	18.9	556.073	2.039883	-1.389	-0.006	-0.97222	-	-18.51	27	6.75	4062.5	18.9	317.94	2.0399	-1.389	-0.006	-0.9722
6.75	1.8675	1015.61	4.725	142.049	-1.38889	5.5556	6E-32	-6.9E-15	s	16.8371	6.75	1.8675	1015.6	4.725	81.175	-1.389	5.5556	6E-32	-7E-15
4062.5	1015.61	634460	2843.7194	86149	-0.00648	0	4E-05	0	v	0.10688	4062	1015.6	634460	2843.719	47466	-0.006	0	4E-05	0
18.9	4.725	2843.72	13.95	408.603	-0.97222	-7E-15	-1E-17	1.388889	t	26.8776	18.9	4.725	2843.7	13.95	233.82	-0.972	-7E-15	-1E-17	1.3889



5.4.3. Estimarea preciziei relațiilor de forță și debit

tab. 33

K68				EN24T			
D	măș.	calc.	diff%	F	măș.	calc.	diff%
1	260.325	268.071	-2.98%	1	127.78	127.91	-0.10%
2	252.161	261.504	-3.70%	2	123.58	123.8	-0.18%
3	246.933	254.937	-3.24%	3	120.34	119.69	0.54%
4	218.448	219.637	-0.54%	4	130.42	130.6	-0.14%
5	214.95	213.07	0.87%	5	127.05	126.49	0.44%
6	207.146	206.503	0.31%	6	123.59	122.38	0.99%
7	163.887	159.284	2.81%	7	133.53	133.95	-0.31%
8	160.997	152.717	5.14%	8	128.88	129.84	-0.74%
9	156.334	146.15	6.51%	9	125.25	125.73	-0.38%
10	307.034	307.302	-0.09%	10	152.17	151.85	0.21%
11	300.463	300.735	-0.09%	11	147.11	147.74	-0.43%
12	290.179	294.168	-1.37%	12	142.95	143.63	-0.47%
13	261.114	258.868	0.86%	13	155.01	154.54	0.30%
14	254.412	252.301	0.83%	14	150.98	150.43	0.36%
15	248.179	245.734	0.99%	15	147.43	146.31	0.75%
16	197.194	198.516	-0.67%	16	157.21	157.89	-0.43%
17	192.71	191.949	0.40%	17	154.12	153.78	0.22%
18	185.051	185.382	-0.18%	18	148.76	149.66	-0.61%
19	356.929	346.534	2.91%	19	175.21	175.79	-0.33%
20	345.55	339.967	1.62%	20	170.56	171.68	-0.65%
21	337.816	333.4	1.31%	21	166.61	167.56	-0.57%
22	302.447	298.1	1.44%	22	179.33	178.48	0.47%
23	292.588	291.533	0.36%	23	174.84	174.37	0.27%
24	288.063	284.966	1.07%	24	171.19	170.25	0.55%
25	226.348	237.747	-5.04%	25	182.47	181.83	0.35%
26	221.79	231.18	-4.23%	26	178.12	177.72	0.23%
27	215.821	224.614	-4.07%	27	172.98	173.6	-0.36%
	Max		6.51%		Max		0.99%
	Min		-5.04%		Min		-0.74%

KC990				F			
D	măș.	calc.	diff%	F	măș.	calc.	diff%
1	252.36	260.27	-3.13%	1	128.08	128.09	-0.01%
2	243.01	253.93	-4.49%	2	124.03	123.96	0.06%
3	238.03	247.59	-4.02%	3	120.58	119.83	0.62%
4	215.49	213.2	1.06%	4	130.97	130.68	0.22%
5	207.39	206.86	0.25%	5	126.96	126.56	0.32%
6	201.39	200.53	0.43%	6	123.81	122.43	1.11%
7	161.3	154.55	4.18%	7	132.67	133.92	-0.94%
8	155.28	148.22	4.55%	8	128.71	129.79	-0.85%
9	151.13	141.88	6.12%	9	125.6	125.66	-0.06%
10	299.18	298.86	0.11%	10	152.24	152.19	0.04%
11	292.99	292.52	0.16%	11	146.98	148.06	-0.73%
12	283.41	286.19	-0.98%	12	143.88	143.93	-0.03%
13	252.7	251.79	0.36%	13	155.14	154.78	0.23%
14	248.4	245.46	1.18%	14	151.12	150.66	0.31%
15	242.1	239.12	1.23%	15	147.15	146.53	0.42%
16	189.49	193.15	-1.93%	16	158.26	158.02	0.15%
17	187.18	186.81	0.20%	17	152.95	153.89	-0.62%
18	182.22	180.48	0.96%	18	149.12	149.77	-0.43%
19	344.92	337.45	2.16%	19	176.12	176.29	-0.09%
20	338.3	331.12	2.12%	20	171.14	172.16	-0.59%
21	330.01	324.78	1.58%	21	166.63	168.03	-0.84%
22	295.39	290.39	1.69%	22	179.56	178.88	0.37%
23	285.19	284.05	0.40%	23	175.49	174.76	0.42%
24	280.01	277.72	0.82%	24	170.8	170.63	0.10%
25	220.68	231.74	-5.01%	25	182.89	182.12	0.42%
26	216.42	225.4	-4.15%	26	178.52	177.99	0.30%
27	209.16	219.07	-4.74%	27	174.04	173.87	0.10%
	Max		6.12%		Max		1.11%
	Min		-5.01%		Min		-0.94%

tab. 34

KC950				EN8			
D	măș.	calc.	diff%	F	măș.	calc.	diff%
1	111.583	116.765	-4.64%	1	53.656	53.906	-0.47%
2	107.738	112.528	-4.45%	2	51.396	51.509	-0.22%
3	103.362	108.292	-4.77%	3	49.219	49.113	0.22%
4	97.4424	96.6905	0.77%	4	55.429	55.059	0.67%
5	94.6321	92.4538	2.30%	5	52.824	52.662	0.31%
6	90.3121	88.2172	2.32%	6	50.797	50.265	1.05%
7	72.5888	71.2547	1.84%	7	56.101	56.519	-0.75%
8	70.6585	67.018	5.15%	8	53.22	54.122	-1.70%
9	67.4063	62.7814	6.86%	9	51.284	51.725	-0.86%
10	135.312	133.402	1.41%	10	63.751	63.629	0.19%
11	128.743	129.165	-0.33%	11	60.695	61.232	-0.89%
12	124.691	124.929	-0.19%	12	59.251	58.835	0.70%
13	113.768	113.328	0.39%	13	65.32	64.782	0.82%
14	108.69	109.091	-0.37%	14	62.807	62.385	0.67%
15	106.754	104.854	1.78%	15	60.552	59.988	0.93%
16	85.2232	87.8918	-3.13%	16	66.353	66.242	0.17%
17	83.605	83.6551	-0.06%	17	63.941	63.845	0.15%
18	79.5035	79.4185	0.11%	18	61.625	61.448	0.29%
19	154.322	150.039	2.78%	19	72.553	73.352	-1.10%
20	147.43	145.803	1.10%	20	70.778	70.955	-0.25%
21	142.641	141.566	0.75%	21	67.662	68.558	-1.32%
22	133.178	129.965	2.41%	22	74.807	74.505	0.40%
23	127.627	125.728	1.49%	23	72.756	72.108	0.89%
24	121.343	121.491	-0.12%	24	69.981	69.711	0.39%
25	100.959	104.529	-3.54%	25	76.352	75.965	0.51%
26	95.586	100.292	-4.92%	26	73.24	73.568	-0.45%
27	92.1045	96.0556	-4.29%	27	70.807	71.171	-0.51%
	Max		6.86%		Max		1.05%
	Min		-4.92%		Min		-1.70%

KC910				F			
D	măș.	calc.	diff%	F	măș.	calc.	diff%
1	120.39	122.5	-1.75%	1	58.016	58.186	-0.29%
2	113.92	118.03	-3.61%	2	55.782	55.29	0.88%
3	107.91	113.56	-5.24%	3	52.575	52.393	0.35%
4	101.77	101.21	0.55%	4	59.567	59.296	0.45%
5	96.05	96.738	-0.72%	5	56.585	56.399	0.33%
6	91.423	92.269	-0.92%	6	53.653	53.503	0.28%
7	76.997	74.228	3.60%	7	60.319	60.702	-0.64%
8	74.157	69.758	5.93%	8	57.451	57.806	-0.62%
9	70.658	65.289	7.60%	9	54.683	54.909	-0.41%
10	140.87	138.92	1.39%	10	67.656	68.192	-0.79%
11	134.85	134.45	0.30%	11	64.571	65.296	-1.12%
12	128.16	129.98	-1.42%	12	62.163	62.4	-0.38%
13	118.2	117.62	0.49%	13	69.319	69.302	0.02%
14	113.26	113.15	0.10%	14	66.703	66.406	0.45%
15	109.88	108.68	1.09%	15	64.4	63.509	1.38%
16	87.573	90.642	-3.50%	16	70.522	70.708	-0.26%
17	85.578	86.173	-0.69%	17	68.164	67.812	0.52%
18	83.551	81.703	2.21%	18	64.751	64.915	-0.25%
19	159.6	155.33	2.68%	19	78.489	78.199	0.37%
20	154.57	150.86	2.40%	20	74.981	75.302	-0.43%
21	147.61	146.39	0.82%	21	71.999	72.406	-0.57%
22	135.4	134.04	1.00%	22	80.248	79.308	1.17%
23	129.21	129.57	-0.27%	23	76.547	76.412	0.18%
24	127	125.1	1.50%	24	73.191	73.516	-0.44%
25	101.57	107.06	-5.41%	25	80.7	80.715	-0.02%
26	98.067	102.59	-4.61%	26	77.302	77.818	-0.67%
27	95.722	98.118	-2.50%	27	75.285	74.922	0.48%
	Max		7.60%		Max		1.38%
	Min		-5.41%		Min		-1.12%

tab. 35

EN1A							
KC850							
D	măș.	calc.	dif%	F	măș.	calc.	dif%
1	93.6207	96.132	-2.68%	1	40.523	40.543	-0.05%
2	84.5387	91.6855	-8.45%	2	37.989	37.925	0.17%
3	80.423	87.239	-8.48%	3	36.066	35.308	2.10%
4	76.71	78.3671	-2.16%	4	41.158	41.468	-0.75%
5	72.7681	73.9206	-1.58%	5	38.975	38.85	0.32%
6	69.9783	69.4741	0.72%	6	36.756	36.233	1.42%
7	57.6252	55.8577	3.07%	7	42.324	42.64	-0.75%
8	56.6973	51.4113	9.32%	8	39.105	40.022	-2.35%
9	55.1442	46.9648	14.83%	9	37.902	37.405	1.31%
10	121	118.801	1.82%	10	52.692	53.05	-0.68%
11	113.975	114.355	-0.33%	11	49.992	50.432	-0.88%
12	109.369	109.908	-0.49%	12	48.069	47.815	0.53%
13	101.932	101.036	0.88%	13	54.625	53.975	1.19%
14	99.1483	96.59	2.58%	14	51.394	51.367	0.07%
15	93.8913	92.1436	1.86%	15	48.884	48.74	0.29%
16	77.3919	78.5271	-1.47%	16	54.597	55.147	-1.01%
17	75.2296	74.0807	1.53%	17	52.465	52.529	-0.12%
18	70.2321	69.6342	0.85%	18	49.431	49.912	-0.97%
19	145.92	141.471	3.05%	19	65.358	65.557	-0.30%
20	141.626	137.024	3.25%	20	62.08	62.939	-1.38%
21	133.915	132.578	1.00%	21	59.715	60.322	-1.02%
22	126.904	123.706	2.52%	22	67.249	66.482	1.14%
23	120.849	119.259	1.32%	23	64.523	63.864	1.02%
24	115.728	114.813	0.79%	24	61.168	61.247	-0.13%
25	94.2722	101.197	-7.35%	25	68.682	67.654	1.50%
26	89.6807	96.7501	-7.88%	26	65.044	65.036	0.01%
27	86.6598	92.3036	-6.51%	27	62.105	62.419	-0.51%
			14.83%				2.10%
			Max				Max
			Min				Min
			-8.48%				-2.35%

KC910							
D	măș.	calc.	dif%	F	măș.	calc.	dif%
1	97.586	102.07	-4.60%	1	44.921	44.63	0.65%
2	91.869	97.535	-6.17%	2	42.74	42.084	1.53%
3	88.177	92.995	-5.46%	3	40.127	39.538	1.47%
4	81.797	82.941	-1.40%	4	44.939	45.509	-1.27%
5	77.135	78.401	-1.64%	5	43.312	42.963	0.80%
6	74.606	73.861	1.00%	6	41.269	40.417	2.06%
7	61.269	58.697	4.20%	7	45.622	46.623	-2.20%
8	59.939	54.157	9.65%	8	43.492	44.077	-1.35%
9	56.75	49.617	12.57%	9	41.539	41.531	0.02%
10	127.16	126.89	0.22%	10	58.649	58.877	-0.39%
11	121.66	122.35	-0.57%	11	55.957	56.331	-0.67%
12	116.72	117.81	-0.94%	12	53.724	53.785	-0.11%
13	110.56	107.75	2.54%	13	59.834	59.756	0.13%
14	102.62	103.21	-0.58%	14	57.021	57.21	-0.33%
15	99.984	98.673	1.31%	15	54.64	54.664	-0.05%
16	83.587	83.509	0.09%	16	60.804	60.87	-0.11%
17	78.742	78.969	-0.29%	17	58.334	58.324	0.02%
18	74.86	74.429	0.58%	18	55.463	55.778	-0.57%
19	158.07	151.7	4.03%	19	73.038	73.124	-0.12%
20	151.99	147.16	3.18%	20	69.741	70.578	-1.20%
21	146.24	142.62	2.48%	21	67.451	68.032	-0.86%
22	132.5	132.56	-0.05%	22	75.133	74.003	1.50%
23	129.39	128.02	1.05%	23	71.304	71.457	-0.22%
24	123.28	123.48	-0.17%	24	68.574	68.911	-0.49%
25	102.34	108.32	-5.84%	25	75.673	75.117	0.74%
26	99.393	103.78	-4.41%	26	73.49	72.571	1.25%
27	92.536	99.24	-7.25%	27	70.001	70.025	-0.03%
			12.57%				2.06%
			Max				Max
			Min				Min
			-7.25%				-2.20%

tab. 36

Aluminiu 99							
K68							
D	măș.	calc.	dif%	F	măș.	calc.	dif%
1	20.6648	20.5658	0.48%	1	9.3463	9.0375	3.30%
2	17.1044	18.882	-10.39%	2	7.9645	8.099	-1.69%
3	15.7483	17.1983	-9.21%	3	7.1824	7.1606	0.30%
4	18.3659	17.2348	6.16%	4	9.9363	9.5368	4.02%
5	16.4687	15.5511	5.57%	5	8.9832	8.5984	4.28%
6	15.6433	13.8674	11.35%	6	8.6842	7.66	11.79%
7	13.9251	12.9099	7.29%	7	10.65	10.185	4.36%
8	12.4968	11.2262	10.17%	8	9.4314	9.2468	1.96%
9	12.2042	9.54248	21.81%	9	8.8934	8.3084	6.58%
10	24.8393	25.9413	-4.44%	10	11.46	12.165	-6.15%
11	22.547	24.2576	-7.59%	11	10.546	11.227	-6.46%
12	21.1908	22.5739	-6.53%	12	9.9874	10.288	-3.01%
13	21.6944	22.6103	-4.22%	13	11.986	12.664	-5.66%
14	19.5473	20.9266	-7.06%	14	10.984	11.726	-6.75%
15	18.488	19.2429	-4.08%	15	10.083	10.788	-6.98%
16	16.3934	18.2854	-11.54%	16	12.455	13.313	-6.89%
17	15.7366	16.6017	-5.50%	17	11.905	12.374	-3.95%
18	13.6336	14.918	-9.42%	18	10.098	11.436	-13.25%
19	34.0098	31.3168	7.92%	19	15.982	15.293	4.32%
20	31.132	29.6331	4.81%	20	14.645	14.354	1.99%
21	29.01	27.9494	3.66%	21	13.71	13.416	2.14%
22	28.6677	27.9858	2.38%	22	16.035	15.792	1.51%
23	27.7562	26.3021	5.24%	23	15.237	14.854	2.51%
24	25.3751	24.6184	2.98%	24	13.989	13.915	0.53%
25	22.6005	23.6609	-4.69%	25	16.753	16.441	1.87%
26	21.269	21.9772	-3.33%	26	15.935	15.502	2.71%
27	19.5606	20.2935	-3.75%	27	15.083	14.564	3.45%
			21.81%				11.79%
			Max				Max
			Min				Min
			-11.54%				-13.25%

KC910							
D	măș.	calc.	dif%	F	măș.	calc.	dif%
1	21.368	23.196	-8.56%	1	9.5545	10.008	-4.75%
2	18.312	22.161	-21.02%	2	8.8522	9.0028	-1.70%
3	19.114	21.127	-10.53%	3	7.8558	7.9975	-1.80%
4	19.493	18.822	3.44%	4	9.9795	10.195	-2.16%
5	18.847	17.788	5.62%	5	9.9722	9.1897	7.85%
6	18.109	16.753	7.49%	6	8.1929	8.1844	0.10%
7	12.907	13.144	-1.83%	7	10.144	10.438	-2.90%
8	13.392	12.109	9.58%	8	9.4405	9.4323	0.09%
9	12.936	11.075	14.39%	9	8.4492	8.427	0.26%
10	31.161	28.797	7.59%	10	12.764	13.225	-3.61%
11	28.271	27.762	1.80%	11	12.786	12.219	4.44%
12	27.388	26.728	2.41%	12	11.329	11.214	1.01%
13	25.905	24.423	5.72%	13	13.718	13.412	2.24%
14	24.545	23.389	4.71%	14	13.09	12.406	5.23%
15	21.598	22.354	-3.50%	15	11.058	11.401	-3.10%
16	17.652	18.744	-6.19%	16	13.632	13.654	-0.16%
17	16.912	17.71	-4.72%	17	12.425	12.649	-1.80%
18	16.543	16.675	-0.80%	18	11.889	11.644	2.06%
19	35.584	34.397	3.33%	19	16.178	16.441	-1.63%
20	33.944	33.363	1.71%	20	15.35	15.436	-0.56%
21	31.985	32.328	-1.07%	21	14.25	14.431	-1.27%
22	29.061	30.024	-3.31%	22	16.783	16.628	0.93%
23	29.442	28.989	1.54%	23	15.791	15.623	1.07%
24	28.333	27.955	1.34%	24	14.936	14.617	2.14%
25	23.256	24.345	-4.68%	25	17.538	16.871	3.80%
26	21.927	23.311	-6.31%	26	15.275	15.865	-3.86%
27	21.758	22.276	-2.38%	27	14.235	14.86	-4.39%
			14.39%				7.85%
			Max				Max
			Min				Min
			-21.02%				-4.75%





EN24T

KC990

Forța	V	Debit	F*v	F^2	v^2	F^2*v	F*v^2	F^3	v^3	F^3*v	F*v^3	F^4	v^4	(F*v)^2	D^F	D^v	D^F*v	D^F^2	D^v^2
128.077	184.726	252.358	23659.2	16403.8	34123.6	3030207	4370458	2100960	6303498	3.9E+08	8.1E+08	2.7E+08	1.2E+09	5.6E+08	32321.414	46617.1	5970594	4139643	8611368
124.029	184.726	243.013	22911.3	15383.2	34123.6	2841666	4232308	1907957	6303498	3.5E+08	7.8E+08	2.4E+08	1.2E+09	5.2E+08	30140.616	44890.7	5567745	3738307	8292465
120.583	184.726	238.028	22274.7	14540.2	34123.6	2685946	4114712	1753296	6303498	3.2E+08	7.6E+08	2.1E+08	1.2E+09	5E+08	28702.08	43969.9	5302010	3460975	8122369
130.972	152.681	215.493	19997	17153.8	23311.6	2619060	3053177	2246669	3559249	3.4E+08	4.7E+08	2.9E+08	5.4E+08	4E+08	28223.604	32901.7	4309219	3696512	5023485
126.958	152.681	207.393	19384.1	16118.3	23311.6	2460972	2959597	2046354	3559249	3.1E+08	4.5E+08	2.6E+08	5.4E+08	3.8E+08	26330.187	31665	4020130	3342829	4834660
123.808	152.681	201.393	18903.1	15328.4	23311.6	2340356	2886158	1897770	3559249	2.9E+08	4.4E+08	2.3E+08	5.4E+08	3.6E+08	24934.013	30749	3806960	3087024	4694794
132.67	112.752	161.302	14958.8	17601.4	12713	1984586	1686631	2335178	1433409	2.6E+08	1.9E+08	3.1E+08	1.6E+08	2.2E+08	21399.954	18187.1	2412882	2839136	2050625
128.705	112.752	155.28	14511.7	16565	12713	1867735	1636223	2132003	1433409	2.4E+08	1.8E+08	2.7E+08	1.6E+08	2.1E+08	19985.346	17508.1	2253383	2572217	1974069
125.604	112.752	151.132	14162	15776.3	12713	1778806	1596795	1981563	1433409	2.2E+08	1.8E+08	2.5E+08	1.6E+08	2E+08	18982.739	17040.4	2140337	2384303	1921334
152.24	184.726	299.183	28122.7	23177.1	34123.6	4281408	5194983	3528492	6303498	6.5E+08	9.6E+08	5.4E+08	1.2E+09	7.9E+08	45547.665	55266.7	8413822	6934191	1E+07
146.98	184.726	292.99	27151	21603.2	34123.6	3990671	5015494	3175251	6303498	5.9E+08	9.3E+08	4.7E+08	1.2E+09	7.4E+08	43063.746	54122.7	7954978	6329526	9997955
143.883	184.726	283.407	26578.9	20702.4	34123.6	3824261	4909809	2978726	6303498	5.5E+08	9.1E+08	4.3E+08	1.2E+09	7.1E+08	40777.46	52352.5	7532643	5867193	9670844
155.139	152.681	252.704	23686.8	24068.1	23311.6	3674753	3616540	3733903	3559249	5.7E+08	5.5E+08	5.8E+08	5.4E+08	5.6E+08	39204.183	38583.1	5985750	6082098	5890928
151.119	152.681	248.396	23073.1	22837.1	23311.6	3486800	3522839	3451130	3559249	5.3E+08	5.4E+08	5.2E+08	5.4E+08	5.3E+08	37537.451	37925.4	5731271	5672640	5790507
147.149	152.681	242.102	22466.9	21652.7	23311.6	3305967	3430271	3186168	3559249	4.9E+08	5.2E+08	4.7E+08	5.4E+08	5E+08	35625.034	36964.5	5439280	5242175	5643797
158.263	112.752	189.487	17844.4	25047	12713	2824096	2011985	3964005	1433409	4.5E+08	2.3E+08	6.3E+08	1.6E+08	3.2E+08	29988.74	21365	3381283	4746094	2408945
152.95	112.752	187.182	17245.4	23393.7	12713	2637683	1944448	3578070	1433409	4E+08	2.2E+08	5.5E+08	1.6E+08	3E+08	28629.456	21105.1	3228022	4378877	2379634
149.123	112.752	182.223	16813.8	22237.6	12713	2507325	1895790	3316125	1433409	3.7E+08	2.1E+08	4.9E+08	1.6E+08	2.8E+08	27173.557	20545.9	3063866	4052193	2316592
176.124	184.726	344.916	32534.7	31019.7	34123.6	5730141	6009986	5463326	6303498	1E+09	1.1E+09	9.6E+08	1.2E+09	1.1E+09	60748.01	63714.8	1.1E+07	1.1E+07	1.2E+07
171.142	184.726	338.305	31614.2	29289.4	34123.6	5410511	5839961	5012641	6303498	9.3E+08	1.1E+09	8.6E+08	1.2E+09	1E+09	57897.967	62493.5	1.1E+07	9908749	1.2E+07
166.633	184.726	330.01	30781.4	27766.6	34123.6	5129205	5686118	4626838	6303498	8.5E+08	1.1E+09	7.7E+08	1.2E+09	9.5E+08	54990.597	60961.3	1E+07	9163257	1.1E+07
179.557	152.681	295.388	27415	32240.6	23311.6	4922544	4185757	5789022	3559249	8.8E+08	6.4E+08	1E+09	5.4E+08	7.5E+08	53038.918	45100.3	8098056	9523496	6885971
175.494	152.681	285.194	26794.6	30798	23311.6	4702289	4091041	5404864	3559249	8.3E+08	6.2E+08	9.5E+08	5.4E+08	7.2E+08	50049.703	43543.8	7641659	8783409	6648324
170.803	152.681	280.005	26078.4	29173.6	23311.6	4454271	3981691	4982940	3559249	7.6E+08	6.1E+08	8.5E+08	5.4E+08	6.8E+08	47825.7	42751.6	7302095	8168768	6527372
182.893	112.752	220.678	20621.5	33449.8	12713	3771520	2325108	6117720	1433409	6.9E+08	2.6E+08	1.1E+09	1.6E+08	4.3E+08	40360.344	24881.8	4550700	7381615	2805466
178.523	112.752	216.423	20128.7	31870.3	12713	3593435	2269551	5689573	1433409	6.4E+08	2.6E+08	1E+09	1.6E+08	4.1E+08	38636.34	24402	4356315	6897460	2751373
174.041	112.752	209.16	19623.5	30290.4	12713	3415297	2212581	5271785	1433409	5.9E+08	2.5E+08	9.2E+08	1.6E+08	3.9E+08	36402.572	23583.2	4104454	6335554	2659048
4073.46	4051.43	6523.14	609337	625488	631333	9.3E+07	9.5E+07	9.8E+07	9.8E+07	1E+08	1.5E+10	1.5E+10	1.7E+10	1.4E+10	998517.4	1013192	1.5E+08	1.6E+08	1.6E+08
			V			ΣQx				INV(W)					Coef.				
27	4073.46	4051.43	609337	625488	631333	6523.14		385.635	-3.827	-1.3635	0.0043	0.01049	0.0024	*	-47.06187				
4073.46	625488	609337	9.3E+07	9.8E+07	9.5E+07	998517		-3.827	0.04528	0.0061	-4E-05	-0.0001	-2E-06	F	0.4953688				
4051.43	609337	631333	9.5E+07	9.3E+07	1E+08	1013192		-1.3635	0.0061	0.01256	-2E-05	-1E-05	-3E-05	V	0.1635328				
609337	9.3E+07	9.5E+07	1.4E+10	1.5E+10	1.5E+10	1.5E+08		0.0043	-4E-05	-2E-05	1.2E-07	6.6E-08	9.3E-09	FV	0.0105126				
625488	9.8E+07	9.3E+07	1.5E+10	1.5E+10	1.4E+10	1.6E+08		0.01049	-0.0001	-1E-05	6.6E-08	4E-07	1.6E-09	F^2	-0.001549				
631333	9.5E+07	1E+08	1.5E+10	1.4E+10	1.7E+10	1.6E+08		0.0024	-2E-06	-3E-05	9.3E-09	1.6E-09	1E-07	V^2	-0.000613				

EN8

CK950	V	Debit	F <sup>v</sup>	F <sup>2</sup>	v <sup>2</sup>	F <sup>2v</sup>	F <sup>v2</sup>	F <sup>3</sup>	v <sup>3</sup>	F <sup>3v</sup>	F <sup>v3</sup>	F <sup>4</sup>	v <sup>4</sup>	(F <sup>v</sup> ) <sup>2</sup>	D <sup>F</sup>	D <sup>v</sup>	D <sup>Fv</sup>	D <sup>F2</sup>	D <sup>v2</sup>
Forța	184.6	111.583	9904.93	2878.99	34077.2	531461	1828450	154476	6290642	2.9E+07	3.4E+08	8288568	1.2E+09	9.8E+07	5987.131	20598.3	1105224	321247	3802439
53.6562	184.6	107.738	9487.78	2641.59	34077.2	487638	1751444	135769	6290642	2.5E+07	3.2E+08	6978018	1.2E+09	9E+07	5537.325	19888.3	1022190	284599	3671388
51.3964	184.6	103.362	9085.83	2422.51	34077.2	447195	1677244	119234	6290642	2.2E+07	3.1E+08	5868559	1.2E+09	8.3E+07	5087.359	19080.6	939126	250395	3522271
49.219	184.6	97.4424	8497.84	3072.4	23503.9	471028	1302801	170301	3603372	2.6E+07	2E+08	9439623	5.5E+08	7.2E+07	5401.156	14938.9	828050	299382	2290274
55.4292	153.31	94.6321	8098.42	2790.36	23503.9	427790	1241566	147398	3603372	2.3E+07	1.9E+08	7786127	5.5E+08	6.6E+07	4998.834	14508	766370	264058	2224220
52.8239	153.31	90.3121	7787.6	2580.29	23503.9	395583	1193915	131070	3603372	2E+07	1.8E+08	6657876	5.5E+08	6.1E+07	4587.541	13845.7	703315	233031	2122684
50.7965	113.663	72.5888	6376.55	3147.28	12919.2	357728	724777	176564	1468437	2E+07	8.2E+07	9905346	1.7E+08	4.1E+07	4072.274	8250.65	462866	228457	937792
53.22	113.663	70.6585	6049.13	2832.37	12919.2	321935	687562	150738	1468437	1.7E+07	7.8E+07	8022299	1.7E+08	3.7E+07	3760.442	8031.24	427422	200131	912853
51.284	113.663	67.4063	5829.09	2630.05	12919.2	298939	662551	134880	1468437	1.5E+07	7.5E+07	6917178	1.7E+08	3.4E+07	3456.869	7661.59	392917	177282	870898
63.7507	184.6	135.312	11768.4	4064.15	34077.2	750242	2172442	259093	6290642	4.8E+07	4E+08	1.7E+07	1.2E+09	1.4E+08	8626.247	24978.6	1592405	549929	4611054
60.6945	184.6	128.743	11204.2	3683.83	34077.2	680034	2068297	223588	6290642	4.1E+07	3.8E+08	1.4E+07	1.2E+09	1.3E+08	7814.018	23766	1442468	474268	4387207
59.2508	184.6	124.691	10937.7	3510.66	34077.2	648067	2019098	208009	6290642	3.8E+07	3.7E+08	1.2E+07	1.2E+09	1.2E+08	7388.063	23018	1363836	437749	4249127
65.3202	153.31	113.768	10014.2	4266.72	23503.9	654130	1595277	278703	3603372	4.3E+07	2.4E+08	1.8E+07	5.5E+08	1E+08	7431.353	17441.8	1139299	485417	2673992
62.8071	153.31	108.69	9628.94	3944.74	23503.9	604766	1476211	247757	3603372	3.8E+07	2.3E+08	1.6E+07	5.5E+08	9.3E+07	6826.507	16663.2	1046570	428753	2554636
60.5519	153.31	106.754	9283.19	3666.53	23503.9	562114	1423203	222015	3603372	3.4E+07	2.2E+08	1.3E+07	5.5E+08	8.6E+07	6464.145	16366.4	991016	391416	2509129
66.3533	113.663	85.2232	7541.91	4402.77	12919.2	500431	857235	292138	1468437	3.3E+07	9.7E+07	1.9E+07	1.7E+08	5.7E+07	5654.846	9686.71	642746	375218	1101019
63.9412	113.663	83.605	7267.74	4088.48	12919.2	464708	826071	261422	1468437	3E+07	9.4E+07	1.7E+07	1.7E+08	5.3E+07	5345.802	9502.78	607619	341817	1080113
61.6254	113.663	79.5035	7004.52	3797.69	12919.2	431656	796153	234034	1468437	2.7E+07	9E+07	1.4E+07	1.7E+08	4.9E+07	4899.433	9036.59	556883	301929	1027124
72.5529	184.6	154.322	13393.3	5263.92	34077.2	971720	2472396	381913	6290642	7.1E+07	4.6E+08	2.8E+07	1.2E+09	1.8E+08	11196.53	28487.9	2066880	812341	5258867
70.7783	184.6	147.43	13065.7	5009.56	34077.2	924765	2411922	354568	6290642	6.5E+07	4.5E+08	2.5E+07	1.2E+09	1.7E+08	10434.84	27215.6	1926272	738560	5023996
67.6624	184.6	142.641	12490.5	4578.2	34077.2	845136	2305742	309772	6290642	5.7E+07	4.3E+08	2.1E+07	1.2E+09	1.6E+08	9651.444	26331.6	1781656	653040	4860804
74.8073	153.31	133.178	11468.7	5596.14	23503.9	857942	1758262	418632	3603372	6.4E+07	2.7E+08	3.1E+07	5.5E+08	1.3E+08	9962.676	20417.4	1527375	745281	3130194
72.756	153.31	127.627	11154.2	5293.43	23503.9	811534	1710047	385129	3603372	5.9E+07	2.6E+08	2.8E+07	5.5E+08	1.2E+08	9285.597	19566.4	1423572	675583	2999719
69.9807	153.31	121.343	10728.7	4897.29	23503.9	750803	1644816	342716	3603372	5.3E+07	2.5E+08	2.4E+07	5.5E+08	1.2E+08	8491.657	18603	1301854	594252	2852028
76.3525	113.663	100.959	8678.43	5829.7	12919.2	662620	986415	445112	1468437	5.1E+07	1.1E+08	3.4E+07	1.7E+08	7.5E+07	7708.485	11475.3	876168	588562	1304316
73.2399	113.663	95.586	8324.65	5364.08	12919.2	609696	946203	392864	1468437	4.5E+07	1.1E+08	2.9E+07	1.7E+08	6.9E+07	7000.707	10864.6	795720	512731	1234899
70.8069	113.663	92.1045	8048.11	5013.62	12919.2	569862	914771	354999	1468437	4E+07	1E+08	2.5E+07	1.7E+08	6.5E+07	6521.635	10468.9	741267	461777	1189920
1687.16	4064.15	2897.2	253120	107267	634502	1.6E+07	3.9E+07	6932892	1E+08	1E+09	6.3E+09	4.5E+08	1.7E+10	2.5E+09	183592.9	450694	2.8E+07	1.2E+07	7.2E+07
						ΣQx				INV(V)					Coef.				
27	1687.16	4064.15	253120	107267	634502	2897.2		371.286	-8.5562	-1.4607	0.01068	0.05549	0.00267	*	37.21955				
1687.16	107267	253120	1.6E+07	6932892	3.9E+07	183593		-8.5562	0.24039	0.01552	-0.0002	-0.0017	-7E-06	F	-1.18883				
4064.15	253120	634502	3.9E+07	1.6E+07	1E+08	450694		-1.4607	0.01552	0.01355	-5E-05	-6E-05	-3E-05	V	0.0045669				
253120	1.6E+07	3.9E+07	2.5E+09	1E+09	6.3E+09	2.8E+07		0.01068	-0.0002	-5E-05	7.2E-07	8.8E-07	2.8E-08	FV	0.012409				
107267	6932892	1.6E+07	1E+09	4.5E+08	2.5E+09	1.2E+07		0.05549	-0.0017	-6E-05	8.8E-07	1.2E-05	2.5E-08	F2	0.008349				
634502	3.9E+07	1E+08	6.3E+09	2.5E+09	1.7E+10	7.2E+07		0.00267	-7E-06	-3E-05	2.8E-08	2.5E-08	1.1E-07	V2	-0.000248				





EN1A

KC850

Forța	V	Debit	F <sup>v</sup>	F <sup>a2</sup>	v <sup>a2</sup>	F <sup>a2v</sup>	F <sup>v^2</sup>	F <sup>a3</sup>	v <sup>a3</sup>	F <sup>a3v</sup>	F <sup>v^3</sup>	F <sup>a4</sup>	v <sup>a4</sup>	(F <sup>v</sup> ) <sup>a2</sup>	D <sup>f</sup>	D <sup>v</sup>	D <sup>f^2</sup>	D <sup>v^2</sup>	
40.5235	184.6	93.6207	7480.63	1642.15	34077.2	303141	1380924	66545.6	6290642	1.2E+07	2.5E+08	2696657	1.2E+09	5.6E+07	3793.8356	17282.4	700342	153739	3190328
37.9691	184.6	84.5387	7012.79	1443.17	34077.2	266410	1294561	54824.9	6290642	1E+07	2.4E+08	2082748	1.2E+09	4.9E+07	3211.5496	15605.8	592852	122004	2880838
36.0663	184.6	80.423	6657.85	1300.78	34077.2	240124	1229038	46914.4	6290642	8660399	2.3E+08	1692031	1.2E+09	4.4E+07	2900.5619	14846.1	535444	104613	2740586
41.1583	153.31	76.71	6309.96	1694	23503.9	259707	967378	69722.1	3603372	1.1E+07	1.5E+08	2669641	5.5E+08	4E+07	3157.25	11760.4	484037	129947	1602982
38.9747	153.31	72.7681	5975.2	1519.03	23503.9	232881	916056	59203.5	3603372	9076475	1.4E+08	2307438	5.5E+08	3.6E+07	2836.1141	11156.1	434804	110537	1710333
36.7562	153.31	69.9783	5635.08	1351.02	23503.9	207124	863912	49658.2	3603372	7613087	1.3E+08	1825246	5.5E+08	3.2E+07	2572.1342	10728.4	394333	94541.8	1644760
42.3237	113.663	57.6252	4810.63	1791.3	12919.2	203604	546790	75814.3	1468437	8617271	6.2E+07	3208744	1.7E+08	2.3E+07	2438.9106	6549.84	277213	103224	744473
39.1048	113.663	56.6973	4444.76	1529.18	12919.2	173811	505204	59798.3	1468437	6798844	5.7E+07	2338398	1.7E+08	2E+07	2217.135	6444.38	252006	86700.5	732486
37.9021	113.663	55.1442	4308.06	1436.57	12919.2	163284	489666	54448.9	1468437	6188813	5.6E+07	2063725	1.7E+08	1.9E+07	2090.0787	6267.84	237564	79218.3	712421
52.6918	184.6	121	9726.91	2776.43	34077.2	512529	1795588	146295	6290642	2.7E+07	3.3E+08	7708563	1.2E+09	9.5E+07	6375.731	22336.7	1176960	335949	4123348
49.9918	184.6	113.975	9228.49	2499.18	34077.2	461349	1703580	124939	6290642	2.3E+07	3.1E+08	6245923	1.2E+09	8.5E+07	5697.8379	21039.8	1051821	284845	3883956
48.0685	184.6	109.369	8873.45	2310.59	34077.2	426534	1638039	111066	6290642	2.1E+07	3E+08	5338804	1.2E+09	7.9E+07	5257.2059	20189.5	970480	252706	3726982
54.6252	153.31	101.932	8374.57	2983.91	23503.9	457462	1283903	162997	3603372	2.5E+07	2E+08	8903720	5.5E+08	7E+07	5568.0711	15627.2	853639	304157	2395804
51.3936	153.31	99.1483	7879.13	2641.3	23503.9	404937	1207948	135746	3603372	2.1E+07	1.9E+08	6976455	5.5E+08	6.2E+07	5095.5854	15200.4	781203	261880	2330369
48.8836	153.31	93.8913	7494.32	2389.6	23503.9	366349	1148953	116812	3603372	1.8E+07	1.8E+08	5710198	5.5E+08	5.6E+07	4589.7424	14394.5	703652	224363	2206810
54.5969	113.663	77.3919	6205.64	2980.82	12919.2	338808	705350	162743	1468437	1.8E+07	8E+07	8885286	1.7E+08	3.9E+07	4225.3547	8796.58	480266	230691	999844
52.4651	113.663	75.2296	5963.34	2752.59	12919.2	312867	677810	144415	1468437	1.6E+07	7.7E+07	7576762	1.7E+08	3.6E+07	3946.9314	8550.81	448619	207076	971909
49.4309	113.663	70.2321	5618.46	2443.42	12919.2	277725	638610	120780	1468437	1.4E+07	7.3E+07	5970278	1.7E+08	3.2E+07	3471.6389	7982.78	394596	171606	907346
65.3579	184.6	145.92	12065.1	4271.65	34077.2	788547	2227210	279186	6290642	5.2E+07	4.1E+08	1.8E+07	1.2E+09	1.5E+08	9537.0117	26936.8	1760532	623319	4972534
62.0803	184.6	141.626	11460	3853.96	34077.2	711441	2115519	239255	6290642	4.4E+07	3.9E+08	1.5E+07	1.2E+09	1.3E+08	8792.1967	26144.2	1623039	545822	4826220
59.7152	184.6	133.915	11023.4	3565.9	34077.2	658266	2034923	212939	6290642	3.9E+07	3.8E+08	1.3E+07	1.2E+09	1.2E+08	7996.7571	24720.7	1476201	477528	4563441
67.2491	153.31	126.904	10309.9	4522.44	23503.9	693333	1580613	304130	3603372	4.7E+07	2.4E+08	2E+07	5.5E+08	1.1E+08	8534.1921	19455.7	1308375	573916	2982741
64.5229	153.31	120.849	9891.99	4163.21	23503.9	638260	1516538	268622	3603372	4.1E+07	2.3E+08	1.7E+07	5.5E+08	9.8E+07	7797.5467	16527.4	1195440	503120	2840426
61.1676	153.31	115.728	9377.58	3741.47	23503.9	573604	1437675	228857	3603372	3.5E+07	2.2E+08	1.4E+07	5.5E+08	8.8E+07	7078.7823	17742.2	1085246	432992	2720049
68.6822	113.663	94.2722	7806.61	4717.24	12919.2	536175	887321	323990	1468437	3.7E+07	1E+08	2.2E+07	1.7E+08	6.1E+07	6474.8177	10715.2	735946	444705	1217925
65.0441	113.663	89.6807	7393.1	4230.74	12919.2	480877	840320	275184	1468437	3.1E+07	9.6E+07	1.8E+07	1.7E+08	5.5E+07	5833.2029	10193.4	663018	379415	1158607
62.1047	113.663	86.6598	7059	3856.99	12919.2	438397	802345	239537	1468437	2.7E+07	9.1E+07	1.5E+07	1.7E+08	5E+07	5381.9785	9849.99	611731	334246	1119578
1388.87	4064.15	2565.23	208386	74408.6	634502	1.1E+07	3.2E+07	4134424	1E+08	6.2E+08	5.2E+09	2.4E+08	1.7E+10	1.7E+09	136872.15	399045	2.1E+07	7572862	6.4E+07
27	1388.87	4064.15	208386	74408.6	634502	2565.23	128.627	-2.2972	-0.9716	0.00444	0.01564	0.00249	*		-11.94256				
1388.87	74408.6	208386	1.1E+07	4134424	3.2E+07	136872	-2.2972	0.07293	0.00635	-1E-04	-0.0006	-5E-06	F		0.556896				
4064.15	208386	634502	3.2E+07	1.1E+07	1E+08	399045	-0.9716	0.00635	0.01125	-3E-05	-2E-05	-3E-05	V		0.005211				
208386	1.1E+07	3.2E+07	1.7E+09	6.2E+08	5.2E+09	2.1E+07	0.00444	-1E-04	-3E-05	4.3E-07	3.3E-07	1.4E-08	FV		0.0129177				
74408.6	4134424	1.1E+07	6.2E+08	2.4E+08	1.7E+09	7572862	0.01564	-0.0006	-2E-05	3.3E-07	5E-06	2.6E-08	F^2		-0.006641				
634502	3.2E+07	1E+08	5.2E+09	1.7E+09	1.7E+10	6.4E+07	0.00249	-5E-06	-3E-05	1.4E-08	2.6E-08	1.1E-07	V^2		-0.000165				

Coef.

INV(V)

ΣQx

EN1A

KC910	Forța	v	Debit	F*v	F^2	v^2	F^2*v	F*v^2	F^3	v^3	F^3*v	F*v^3	F^4	v^4	(F*v)^2	D^F	D^v	D^F*v	D^F^2	D^v^2	
44.9211	184.6	97.5863	8292.43	2017.91	34077.2	372505	1530783	90646.5	6290642	1.7E+07	2.8E+08	4071941	1.2E+09	6.9E+07	4383.6817	18014.4	809228	196920	3325462		
42.7397	184.6	91.8694	7889.76	1826.69	34077.2	337206	1456449	78072.1	6290642	1.4E+07	2.7E+08	3336779	1.2E+09	6.2E+07	3926.4758	16959.1	724827	167817	3130649		
40.1269	184.6	88.1773	7407.42	1610.16	34077.2	297236	1367409	64610.8	6290642	1.2E+07	2.5E+08	2592630	1.2E+09	5.5E+07	3538.2784	16277.5	653166	141980	3004832		
44.9389	153.31	81.7967	6889.57	2019.5	23503.9	309609	1056237	90754.1	3603372	1.4E+07	1.6E+08	4078388	5.5E+08	4.7E+07	3675.8509	12540.2	563544	165189	1922539		
43.3119	153.31	77.1353	6640.13	1875.92	23503.9	287596	1017996	81249.4	3603372	1.2E+07	1.6E+08	3519063	5.5E+08	4.4E+07	3340.8718	11825.6	512188	144699	1812978		
41.2691	153.31	74.6065	6326.96	1703.14	23503.9	261108	969984	70287.2	3603372	1.1E+07	1.5E+08	2900691	5.5E+08	4E+07	3078.9455	11437.9	472032	127065	1753541		
45.6216	113.663	61.269	5185.48	2081.33	12919.2	236570	589397	94953.8	1468437	1.1E+07	6.7E+07	4331950	1.7E+08	2.7E+07	2795.1899	6964	317709	127521	791548		
43.4922	113.663	59.9391	4943.44	1891.57	12919.2	215001	561886	82268.5	1468437	9350868	6.4E+07	3578035	1.7E+08	2.4E+07	2606.8826	6812.85	296306	113379	774368		
41.5386	113.663	56.7495	4721.4	1725.46	12919.2	196120	536647	71673	1468437	8146559	6.1E+07	2977198	1.7E+08	2.2E+07	2357.297	6450.31	267937	97918.8	733161		
58.6493	184.6	127.161	10826.7	3439.74	34077.2	634976	1998601	201738	6290642	3.7E+07	3.7E+08	1.2E+09	1.2E+09	1.2E+08	7457.9014	23473.9	1376728	437401	4333284		
55.9569	184.6	121.658	10329.6	3131.17	34077.2	578014	1906851	175211	6290642	3.2E+07	3.5E+08	9804241	1.2E+09	1.1E+08	6807.63	22458.2	1256688	380934	4145775		
53.7237	184.6	116.715	9917.39	2886.23	34077.2	532799	1830750	155059	6290642	2.9E+07	3.4E+08	8330341	1.2E+09	9.8E+07	6270.3621	21545.6	1157509	336867	3977317		
59.8344	153.31	110.556	9173.2	3580.16	23503.9	548873	1406341	214217	3603372	3.3E+07	2.2E+08	1.3E+09	5.5E+08	8.4E+07	6615.0773	16949.4	1014156	395809	2598503		
57.0214	153.31	102.62	8741.94	3251.44	23503.9	498478	1340224	185402	3603372	2.8E+07	2.1E+08	1.1E+09	5.5E+08	7.6E+07	5851.5428	15732.7	897098	333663	2411969		
54.6396	153.31	99.9844	8376.79	2985.49	23503.9	457704	1284243	163126	3603372	2.5E+07	2E+08	8913142	5.5E+08	7E+07	5463.1125	15328.6	837548	298502	2350021		
60.804	113.663	83.5873	6911.16	3697.13	12919.2	420226	785542	224800	1468437	2.6E+07	8.9E+07	1.4E+09	1.7E+08	4.8E+07	5082.4444	9500.77	577685	309033	1079884		
58.3337	113.663	78.7421	6630.37	3402.82	12919.2	386774	753627	198499	1468437	2.3E+07	8.6E+07	1.2E+09	1.7E+08	4.4E+07	4593.3179	8950.05	522089	267945	1017288		
55.4632	113.663	74.8597	6304.1	3076.16	12919.2	349645	716542	170614	1468437	1.9E+07	8.1E+07	9462776	1.7E+08	4E+07	4151.9544	8508.76	471923	230281	967130		
73.038	184.6	158.069	13482.8	5334.55	34077.2	984758	2488928	389625	6290642	7.2E+07	4.6E+08	2.8E+09	1.2E+09	1.8E+08	11545.073	29179.5	2131211	843226	5386530		
69.7405	184.6	151.992	12874.1	4863.74	34077.2	897846	2376558	339200	6290642	6.3E+07	4.4E+08	2.4E+09	1.2E+09	1.7E+08	10599.986	28057.7	1956757	739248	5179448		
67.4513	184.6	146.24	12451.5	4549.68	34077.2	839871	2298548	306882	6290642	5.7E+07	4.2E+08	2.1E+09	1.2E+09	1.6E+08	9864.0909	26995.9	1820911	665346	4983449		
75.1335	153.31	132.499	11518.7	5645.04	23503.9	865440	1765928	424132	3603372	6.5E+07	2.7E+08	3.2E+09	5.5E+08	1.3E+08	9955.1166	20313.4	1526216	747963	3114240		
71.3037	153.31	129.387	10931.5	5084.21	23503.9	779459	1675912	362523	3603372	5.6E+07	2.6E+08	2.6E+09	5.5E+08	1.2E+08	9225.8001	19836.4	1414405	657833	3041106		
68.5737	153.31	123.278	10513	4702.35	23503.9	720917	1611748	322458	3603372	4.9E+07	2.5E+08	2.2E+09	5.5E+08	1.1E+08	8453.6609	18899.8	1296028	579699	2897521		
75.6734	113.663	102.341	8601.25	5726.46	12919.2	650885	977642	433340	1468437	4.9E+07	1.1E+08	3.3E+09	7.4E+08	7.4E+07	7744.4964	11632.4	880261	586052	1322169		
73.4896	113.663	99.3931	8353.04	5400.73	12919.2	613862	949430	396897	1468437	4.5E+07	1.1E+08	2.9E+09	1.7E+08	7E+07	7304.3611	11297.3	830234	536795	1284083		
70.0013	113.663	92.5359	7956.54	4900.18	12919.2	556968	904363	343019	1468437	3.9E+07	1E+08	2.4E+09	1.7E+08	6.3E+07	6477.6329	10517.9	736266	453443	1195494		
1546.79	4064.15	2740.75	232190	92409	634502	1.4E+07	3.6E+07	5731257	1E+08	8.6E+08	5.8E+09	3.7E+08	1.7E+10	2.1E+09	163166.99	426460	2.5E+07	1E+07	6.9E+07		
27	1546.79	4064.15	232190	92409	634502	2740.75	ΣQx								Coef.						
1546.79	92409	232190	1.4E+07	5731257	3.6E+07	163167		119.388	-1.8731	-0.9199	0.00351	0.01159	0.00241	*	-2.174245						
4064.15	232190	634502	3.6E+07	1.4E+07	1E+08	426460		-1.8731	0.0556	0.00441	-7E-05	-0.0004	-2E-06	F	0.3344049						
232190	1.4E+07	3.6E+07	2.1E+09	8.6E+08	5.8E+09	2.5E+07		-0.9199	0.00441	0.01103	-2E-05	-1E-05	-3E-05	V	-0.070069						
92409	5731257	1.4E+07	8.6E+08	3.7E+08	2.1E+09	1E+07		0.00351	-7E-05	-2E-05	3.2E-07	1.7E-07	9.4E-09	FV	0.011672						
634502	3.6E+07	1E+08	5.8E+09	2.1E+09	1.7E+10	6.9E+07		0.01159	-0.0004	-1E-05	1.7E-07	3.2E-06	5.6E-09	F^2	-0.002953						
								0.00241	-2E-06	-3E-05	9.4E-09	5.6E-09	1.1E-07	V^2	0.0002045						

Aluminiu 99

K68

Forța	v	Debit	F <sup>v</sup>	F <sup>2</sup>	v <sup>2</sup>	F <sup>2v</sup>	F <sup>v2</sup>	F <sup>3</sup>	v <sup>3</sup>	F <sup>3v</sup>	F <sup>v3</sup>	F <sup>4</sup>	v <sup>4</sup>	(F <sup>v</sup> ) <sup>2</sup>	D <sup>F</sup>	D <sup>v</sup>	D <sup>Fv</sup>	D <sup>F2</sup>	D <sup>v2</sup>
9.3463	184.726	20.6648	1726.5	87.3533	34123.6	16136.4	318929.1	816.43	6303498	150816	5.9E+07	7630.6	1.2E+09	2980807	193.13948	3817.32	35677.8	1805.14	705157
7.9645	184.726	17.1044	1471.25	63.4333	34123.6	11717.8	271777.1	505.214	6303498	93326	5E+07	4023.78	1.2E+09	2164569	136.22833	3159.63	25164.9	1084.99	583665
7.1824	184.726	15.7483	1326.77	51.5869	34123.6	9529.42	245089.1	370.518	6303498	68444.1	4.5E+07	2661.21	1.2E+09	1760328	113.1109	2909.12	20894.5	812.408	537390
9.9363	153.561	18.3659	1525.83	98.7301	23581	15161.1	234307.8	981.011	3621122	150645	3.6E+07	9747.62	5.6E+08	2328153	182.48953	2820.29	28023.3	1813.27	433087
8.9832	153.561	16.4687	1379.47	80.6979	23581	12392.1	211832.8	724.925	3621122	111320	3.3E+07	6512.15	5.6E+08	1902936	147.94161	2528.95	22718.1	1328.99	398348
8.6842	153.561	15.6433	1333.55	75.4153	23581	11580.9	204782.1	654.922	3621122	100570	3.1E+07	5687.47	5.6E+08	1778369	135.84987	2402.21	20861.2	1179.75	368885
10.6498	113.097	13.9251	1204.46	113.418	12791	12827.3	136221.7	1207.88	1446629	136608	1.5E+07	12863.7	1.6E+08	1450734	148.29958	1474.89	16772.3	1579.36	178116
9.4314	113.097	12.4968	1066.67	88.9513	12791	10060.2	120637.1	838.935	1446629	94881.4	1.4E+07	7912.33	1.6E+08	1137777	117.86219	1413.35	13329.9	1111.61	159846
8.8934	113.097	12.2042	1005.82	79.0926	12791	8945.16	113755.5	703.402	1446629	79552.9	1.3E+07	6255.63	1.6E+08	1011674	108.53671	1380.26	12275.2	965.26	156104
11.4598	184.726	24.8393	2116.92	131.327	34123.6	24259.5	391049.2	1504.98	6303498	278009	7.2E+07	17246.8	1.2E+09	4481346	284.65301	4588.45	52582.7	3262.07	847604
10.5456	184.726	22.547	1948.04	111.21	34123.6	20543.3	359853.5	1172.77	6303498	216641	6.6E+07	12367.6	1.2E+09	3794871	237.77143	4165.01	43922.5	2507.44	769383
9.9874	184.726	21.1908	1844.93	99.7482	34123.6	18426	340805.7	996.225	6303498	184028	6.3E+07	9949.7	1.2E+09	3403763	211.64051	3914.48	39095.4	2113.74	723104
11.9863	153.561	21.6944	1840.63	143.671	23581	22062.3	282648.9	1722.09	3621122	264446	4.3E+07	20641.5	5.6E+08	3387914	260.03531	3331.41	39931.3	3116.86	511575
10.9843	153.561	19.5473	1686.76	120.655	23581	18527.9	259020.7	1325.31	3621122	203516	4E+07	14557.6	5.6E+08	2845161	214.71352	3001.71	32971.6	2358.48	460945
10.0834	153.561	18.488	1548.42	101.675	23581	15613.3	237776.6	1025.23	3621122	157435	3.7E+07	10337.8	5.6E+08	2397597	186.42165	2839.03	28627.1	1879.76	435965
12.4546	113.097	16.3934	1408.58	155.117	12791	17543.3	159306.9	1931.92	1446629	218495	1.8E+07	24061.3	1.6E+08	1984103	204.17305	1854.05	23091.4	2542.89	209688
11.9047	113.097	15.7366	1346.39	141.722	12791	16028.4	152273.1	1687.16	1446629	190813	1.7E+07	20085.1	1.6E+08	1812766	187.33908	1779.76	21187.6	2230.22	201287
10.0982	113.097	13.6336	1142.08	101.974	12791	11532.9	129166.1	1029.75	1446629	116462	1.5E+07	10398.6	1.6E+08	1304346	137.67513	1541.93	15570.7	1390.27	174388
15.9824	184.726	34.0098	2952.36	255.437	34123.6	47185.8	545376.5	4082.5	6303498	754142	1E+08	65248.1	1.2E+09	8716425	543.55759	6282.47	100409	8687.35	1160534
14.6451	184.726	31.132	2705.33	214.479	34123.6	39619.8	499743	3141.07	6303498	580235	9.2E+07	46001.2	1.2E+09	7318787	455.93124	5750.88	84222.2	6677.16	1062335
13.7098	184.726	29.01	2532.55	187.959	34123.6	34720.8	467827.3	2576.88	6303498	476015	8.6E+07	35328.4	1.2E+09	6413818	397.72129	5359.89	73469.3	5452.68	989925
16.0345	153.561	28.6677	2462.27	257.105	23581	39481.3	378109.5	4122.55	3621122	633064	5.8E+07	66103.1	5.6E+08	6062796	459.67272	4402.25	70587.8	7370.62	676014
15.2366	153.561	27.7562	2339.75	232.154	23581	35649.8	359294.2	3537.24	3621122	543182	5.5E+07	53895.5	5.6E+08	5474422	422.91034	4262.27	64942.6	6443.72	654519
13.9894	153.561	25.3751	2148.23	195.703	23581	30052.4	329884	2737.77	3621122	420415	5.1E+07	38299.8	5.6E+08	4614879	354.98311	3896.63	54511.6	4966	598371
16.7532	113.097	22.6005	1894.74	280.67	12791	31743	214290.3	4702.12	1446629	531797	2.4E+07	78775.5	1.6E+08	3590048	378.63046	2556.05	42822.1	6343.27	289083
15.9345	113.097	21.269	1802.15	253.908	12791	28716.4	203818.3	4045.9	1446629	457581	2.3E+07	64469.4	1.6E+08	3247743	338.9106	2405.47	38329.9	5400.37	272052
15.0834	113.097	19.5606	1705.89	227.509	12791	25730.7	192931.9	3431.61	1446629	388106	2.2E+07	51760.3	1.6E+08	2910069	295.04095	2212.26	33368.3	4450.22	250200
317.945	4062.46	556.073	47466.3	3950.7	634460	585787	7360508	51576.3	1E+08	7600545	1.2E+09	702822	1.7E+10	9E+07	6855.2392	86149	1055360	88873.9	1.4E+07
27	317.945	4062.46	47466.3	3950.7	634460	556.073	114.023	-8.0665	0.02225	0.01905	0.20911	0.00225	0.00225	*	-3.962445				
317.945	3950.7	47466.3	585787	51576.3	7360508	6855.24	-8.0665	1.09758	0.02032	-0.002	-0.0325	1.2E-05	1.2E-05	F	-0.151187				
4062.46	47466.3	634460	7360508	585787	1E+08	86149	-0.9032	0.02032	0.01087	-1E-04	-0.0002	-3E-05	-3E-05	V	0.0754946				
47466.3	585787	7360508	9E+07	7600545	1.2E+09	1055360	0.01905	-0.002	-1E-04	6.8E-06	3.8E-05	6E-08	6E-08	FV	0.0108739				
3950.7	51576.3	585787	7600545	702822	9E+07	88873.9	0.20911	-0.0325	-0.0002	3.8E-05	0.00109	-9E-07	-9E-07	F <sup>2</sup>	0.0095937				
634460	7360508	1E+08	1.2E+09	9E+07	1.7E+10	1.4E+07	0.00225	1.2E-05	-3E-05	6E-08	-9E-07	1.1E-07	1.1E-07	V <sup>2</sup>	-0.000236				

Coef.  
INV(V)  
ΣQx



Aluminiu 99

KC910

Forța	V	Debit	F*v	F^2	v^2	F^2*v	F*v^2	F^3	v^3	F^3*v	F*v^3	F^4	v^4	(F*v)^2	D^F	D^v	D^F*v	D^F^2	D^v^2
9.55446	184.726	21.3677	1764.95	91.2878	34123.6	16863.2	326032.3	872.206	6303498	161119	6E+07	8333.46	1.2E+09	3115064	204.15698	3947.16	37713	1950.61	729142
8.85217	184.726	18.3117	1635.22	78.3609	34123.6	14475.3	302067.6	693.664	6303498	128138	5.6E+07	6140.43	1.2E+09	2673953	162.09845	3382.64	29943.7	1434.92	624861
7.85583	184.726	19.1142	1451.17	61.714	34123.6	11400.2	268068.8	484.815	6303498	89557.7	5E+07	3808.62	1.2E+09	2105902	150.15762	3530.88	27738	1179.61	652244
9.97952	153.561	19.4929	1532.47	99.5909	23581	15293.3	235327.1	993.869	3621122	152620	3.6E+07	9918.34	5.6E+08	2348452	194.52968	2993.35	29872.2	1941.31	459662
9.9722	153.561	18.8472	1531.34	99.4447	23581	15270.8	235154.3	991.682	3621122	152284	3.6E+07	9889.25	5.6E+08	2345005	187.94808	2894.2	28861.5	1874.26	444436
8.19292	153.561	18.1092	1258.11	67.124	23581	10307.6	193197.2	549.941	3621122	84449.6	3E+07	4505.63	5.6E+08	1582850	148.36765	2780.88	22783.5	1215.56	427034
10.1436	113.097	12.9071	1147.21	102.892	12791	11636.9	129746.8	1043.7	1446629	118040	1.5E+07	10586.9	1.6E+08	1316098	130.92411	1459.76	14807.2	1328.04	165095
9.44052	113.097	13.3922	1067.7	89.1233	12791	10079.6	120753.7	841.37	1446629	95156.7	1.4E+07	7942.97	1.6E+08	1139977	126.42933	1514.62	14298.8	1193.56	171300
8.44925	113.097	12.9364	955.587	71.3898	12791	8073.99	108074.4	603.19	1446629	68219.2	1.2E+07	5096.5	1.6E+08	913147	109.30302	1463.07	12361.9	923.528	165470
12.7639	184.726	31.1612	2357.82	162.918	34123.6	30095.1	435550.4	2079.47	6303498	384131	8E+07	26542.1	1.2E+09	5559329	397.73921	5756.28	73472.6	5076.71	1063332
12.7865	184.726	28.2708	2361.99	163.493	34123.6	30201.4	436319.3	2090.5	6303498	386169	8.1E+07	26730.1	1.2E+09	5579976	361.48302	5222.34	66775.2	4622.09	964700
11.3285	184.726	27.3885	2092.66	128.335	34123.6	23706.8	386568.8	1453.84	6303498	268562	7.1E+07	16469.9	1.2E+09	4379245	310.27021	5059.35	57314.9	3514.9	934592
13.7184	153.561	25.9052	2106.61	188.194	23581	28899.3	323493.1	2581.72	3621122	396451	5E+07	35417	5.6E+08	4437802	355.37779	3978.03	54572.2	4875.21	610871
13.0904	153.561	24.5448	2010.18	171.359	23581	26314.1	308685.5	2243.17	3621122	344484	4.7E+07	29364.1	5.6E+08	4040827	321.30245	3769.13	49339.5	4205.99	578791
11.058	153.561	21.5985	1698.07	122.279	23581	18777.3	260758.1	1352.16	3621122	207639	4E+07	14952.1	5.6E+08	2883456	238.83525	3316.88	36675.8	2641.03	509313
13.632	113.097	17.6522	1541.74	185.831	12791	21017	174366.7	2533.24	1446629	286503	2E+07	34533.1	1.6E+08	2376964	240.63443	1996.42	27215.1	3280.32	225789
12.4249	113.097	16.9123	1405.23	154.379	12791	17459.8	158927.3	1918.14	1446629	216937	1.8E+07	23832.8	1.6E+08	1974660	210.13389	1912.73	23765.6	2610.9	216325
11.8887	113.097	16.5434	1344.58	141.341	12791	15985.2	152068.1	1680.35	1446629	190043	1.7E+07	19977.2	1.6E+08	1807889	196.67943	1871.02	22243.9	2338.26	211607
16.1779	184.726	35.5836	2988.47	261.723	34123.6	48347	552046.6	4234.13	6303498	782152	1E+08	68499.2	1.2E+09	8930937	575.66739	6573.21	106341	9313.07	1214240
15.3505	184.726	33.9435	2835.63	235.638	34123.6	43528.3	523813.6	3617.15	6303498	668181	9.7E+07	55525.1	1.2E+09	8040798	521.05015	6270.24	96251.3	7998.38	1158275
14.2502	184.726	31.9846	2632.38	203.069	34123.6	37512.1	486268.6	2893.78	6303498	534555	9E+07	41237	1.2E+09	6929439	455.78737	5908.37	84195.6	6495.07	1091427
16.7834	153.561	29.0613	2577.27	281.682	23581	43255.3	395768.7	4727.57	3621122	725971	6.1E+07	79344.6	5.6E+08	6642336	487.7465	4462.68	74898.9	8186.03	685294
15.7913	153.561	29.4415	2424.92	249.364	23581	38292.6	372373.6	3937.77	3621122	604688	5.7E+07	62182.4	5.6E+08	5880249	464.91897	4521.07	71393.4	7341.66	694261
14.9364	153.561	28.3334	2293.65	223.097	23581	34258.9	352215.6	3332.26	3621122	511706	5.4E+07	49772.1	5.6E+08	5260840	423.2002	4350.91	64987.1	6321.1	668131
17.5378	113.097	23.2558	1983.48	307.574	12791	34785.8	224326.1	5394.18	1446629	610067	2.5E+07	94602	1.6E+08	3934186	407.85561	2630.17	46127.4	7152.89	297465
15.2752	113.097	21.9272	1727.58	233.332	12791	26389.2	195385.3	3564.19	1446629	403101	2.2E+07	54443.8	1.6E+08	2984550	334.94249	2479.91	37881.1	5116.32	280471
14.2354	113.097	21.758	1609.98	202.646	12791	22918.8	182085	2884.75	1446629	326258	2.1E+07	41065.5	1.6E+08	2592051	309.7343	2460.78	35030.1	4409.19	278307
335.47	4062.46	619.745	50336	4377.18	634460	655145	7839443	59592.8	1E+08	8897159	1.3E+09	840712	1.7E+10	1E+08	8027.2736	96505.9	1246860	108541	1.6E+07
27	335.47	4062.46	50336	4377.18	634460	619.745		100.505	-6.0249	-0.8803	0.01476	0.15122	0.00233	*	-7.840375				
335.47	4377.18	50336	655145	59592.8	7839443	8027.27		-6.0249	0.75366	0.01975	-0.0013	-0.022	-1E-05	F	0.7515961				
4062.46	50336	634460	7839443	655145	1E+08	96505.9		-0.8803	0.01975	0.0106	-9E-05	-0.0003	-3E-05	V	0.0647895				
50336	655145	7839443	1E+08	8897159	1.3E+09	1246860		0.01476	-0.0013	-9E-05	6E-06	1.7E-05	4.5E-08	FV	0.0106863				
4377.18	59592.8	655145	8897159	840712	1E+08	108541		0.15122	-0.022	-0.0003	1.7E-05	0.00078	2E-07	F^2	-0.027953				
634460	7839443	1E+08	1.3E+09	1E+08	1.7E+10	1.6E+07		0.00233	-1E-05	-3E-05	4.5E-08	2E-07	1.1E-07	V^2	-0.000167				

5.4.5. Estimarea preciziei relațiilor  $Q=f(F, v)$  pentru debit

tab. 45

K68		EN24T		KC990	
Debit		Debit		Debit	
260.325	260.493	-0.06%	252.358	252.415	-0.02%
252.161	252.106	0.02%	243.013	244.128	-0.46%
246.933	245.63	0.53%	238.028	237.034	0.42%
218.448	219.639	-0.55%	215.493	214.491	0.46%
214.95	214.027	0.43%	207.393	207.663	-0.13%
207.146	208.286	-0.55%	201.393	202.269	-0.44%
163.887	166.103	-1.35%	161.302	160.578	0.45%
160.997	160.292	0.44%	155.28	155.519	-0.15%
156.334	155.761	0.37%	151.132	151.528	-0.26%
307.034	309.286	-0.73%	299.183	300.818	-0.55%
300.463	299.14	0.44%	292.99	290.435	0.87%
290.179	290.822	-0.22%	283.407	284.282	-0.31%
261.114	260.639	0.18%	252.704	254.544	-0.73%
254.412	253.908	0.20%	248.396	248.008	0.16%
248.179	247.982	0.08%	242.102	241.501	0.25%
197.194	195.76	0.73%	189.487	192.059	-1.36%
192.71	191.873	0.43%	187.182	185.691	0.80%
185.051	185.157	-0.06%	182.223	181.049	0.64%
356.929	355.51	0.40%	344.916	346.885	-0.57%
345.55	346.168	-0.18%	338.305	337.421	0.26%
337.816	338.234	-0.12%	330.01	328.79	0.37%
302.447	301.344	0.36%	295.388	293.176	0.75%
292.588	293.814	-0.42%	285.194	286.876	-0.59%
288.063	287.709	0.12%	280.005	279.539	0.17%
226.348	227.546	-0.53%	220.678	220.442	0.11%
221.79	222.055	-0.12%	216.423	215.543	0.41%
215.821	215.586	0.11%	209.16	210.458	-0.62%
Max	0.73%	Max	0.87%	Max	0.87%
Min	-1.35%	Min	-1.36%	Min	-1.36%

tab. 46

KC950		EN8		KC910	
Debit		Debit		Debit	
111.583	112.784	-1.08%	120.39	119.776	0.51%
107.738	108.311	-0.53%	113.92	115.017	-0.96%
103.362	104.083	-0.70%	107.912	108.094	-0.17%
97.4424	97.3056	0.14%	101.771	101.838	-0.07%
94.6321	93.0917	1.63%	96.0502	96.5988	-0.57%
90.3121	89.8909	0.47%	91.4234	91.3562	0.07%
72.5888	73.2499	-0.91%	76.9971	77.2677	-0.35%
70.6585	69.9822	0.96%	74.1574	73.5636	0.80%
67.4063	67.864	-0.68%	70.6685	69.9053	1.07%
135.312	133.802	1.12%	140.87	139.695	0.83%
128.743	127.259	1.15%	134.847	133.429	1.05%
124.691	124.222	0.38%	128.163	128.466	-0.24%
113.768	114.336	-0.50%	118.203	118.307	-0.09%
108.69	109.854	-1.07%	113.261	113.99	-0.64%
106.754	105.922	0.78%	109.88	110.128	-0.23%
85.232	86.0047	-0.92%	87.5734	89.733	-2.47%
83.605	82.8459	0.91%	85.5782	86.9507	-1.60%
79.5035	79.9048	-0.50%	83.5505	82.8191	0.88%
154.322	153.518	0.52%	159.6	160.894	-0.81%
147.43	149.439	-1.36%	154.571	154.167	0.26%
142.641	142.404	0.17%	147.608	148.344	-0.50%
133.178	132.206	0.73%	135.396	135.556	-0.12%
127.627	128.214	-0.46%	129.213	129.858	-0.50%
121.343	122.926	-1.30%	126.998	124.564	1.92%
100.959	100.135	0.82%	101.565	101.058	0.50%
95.586	95.5572	0.03%	98.0668	97.4008	0.68%
92.1045	92.0919	0.01%	95.722	95.1708	0.58%
Max	1.63%	Max	1.92%	Max	1.92%
Min	-1.36%	Min	-2.47%	Min	-2.47%

tab. 47

EN1A		KC910	
KC850	Debit	Debit	
93.6207	91.6882	97.5863	97.711
84.5387	85.5546	91.8694	92.8461
80.423	80.8442	88.1773	86.9819
76.71	78.1589	81.7967	81.3684
72.7681	73.7802	77.1353	78.3369
69.9783	69.2667	74.6065	74.5087
57.6252	60.3361	61.269	62.1378
56.6973	55.5575	59.9391	59.161
55.1442	53.7368	56.7495	56.4065
121	119.95	127.161	127.682
113.975	113.849	121.658	121.892
109.369	109.444	116.715	117.057
101.932	103.764	110.556	108.395
99.1483	97.8391	102.62	103.392
93.8913	93.1417	99.9844	99.1185
77.3919	77.293	83.5873	82.5853
75.2296	74.4913	78.7421	79.351
70.2321	70.3994	74.8597	75.5475
145.92	147.279	158.069	157.901
141.626	140.412	151.992	151.084
133.915	135.367	146.24	146.313
126.904	125.579	132.499	134.79
120.849	121.047	129.387	128.312
115.728	115.334	123.278	123.642
94.2722	94.2881	102.341	101.292
89.6807	90.1509	99.3931	98.6262
86.6598	86.6798	92.5359	94.3099
	Max	Max	Max
	Min	Min	Min
	2.55%	1.95%	1.95%
	-4.70%	-1.92%	-1.92%

tab. 48

Aluminiu 99		KC910	
K68	Debit	Debit	
20.6648	20.1355	21.3677	22.2689
17.1044	17.3393	18.3117	20.716
15.7483	15.7729	19.1142	18.4657
18.3659	18.1066	19.4929	19.5053
16.4687	16.4862	18.8472	19.4919
15.6433	15.9815	18.1092	16.1383
13.9251	14.1348	12.9071	14.4892
12.4968	12.5858	13.3922	13.4959
12.2042	11.911	12.9364	12.0486
24.8393	24.4832	31.1612	29.0144
22.547	22.5921	28.2708	29.0597
21.1908	21.4453	27.3885	26.0687
21.6944	21.6509	25.9052	25.9742
19.5473	19.9085	24.5448	24.9423
18.488	18.3583	21.5985	21.4514
16.3934	16.4815	17.6522	19.0087
15.7366	15.7599	16.9123	17.5219
13.6336	13.43	16.5434	16.8352
34.0098	34.0746	35.5836	35.5576
31.132	31.1976	33.9435	34.0317
29.01	29.2059	31.9846	31.9432
28.6677	28.8869	29.0613	30.6942
27.7562	27.4358	29.4415	29.2238
25.3751	25.1921	28.3334	27.9128
22.6005	22.3226	23.2558	23.2617
21.269	21.1828	21.9272	20.902
19.5606	20.0115	21.758	19.7215
	Max	Max	Max
	Min	Min	Min
	2.56%	10.88%	10.88%
	-2.30%	-13.13%	-13.13%



## 5.5. REPREZENTAREA GRAFICĂ A RELAȚIILOR

Folosind funcția din anexa 8, s-au obținut următoarele reprezentări grafice pentru relațiile de tipul  $Q=f(F,v)$ :

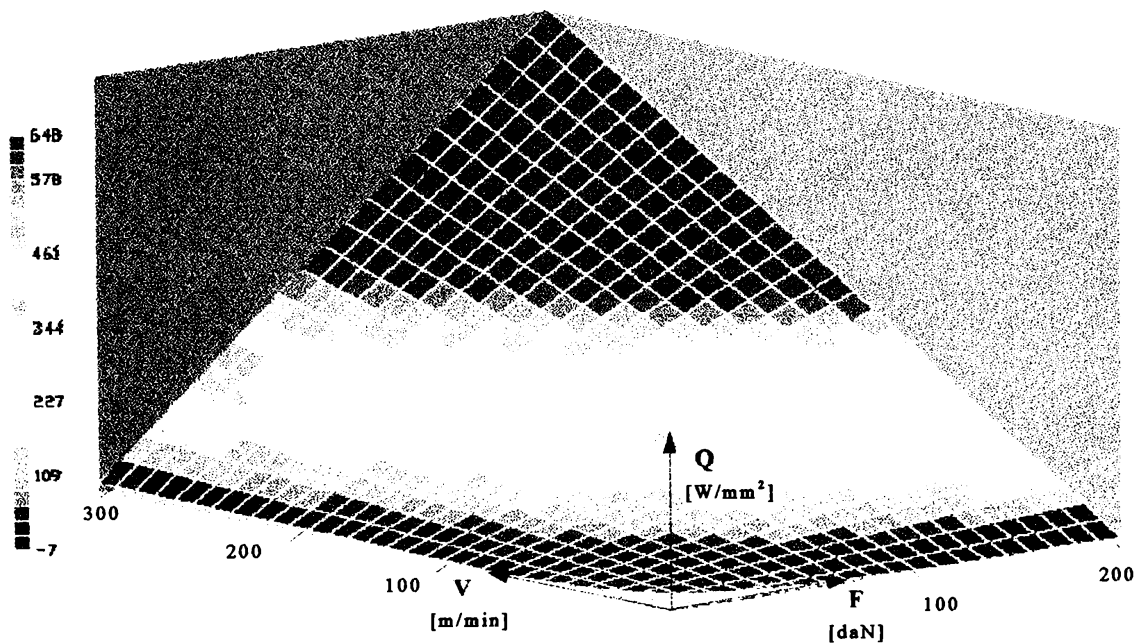


Fig. 96 Grafic de variație  $Q=f(F,v)$  pentru EN24T \ K68

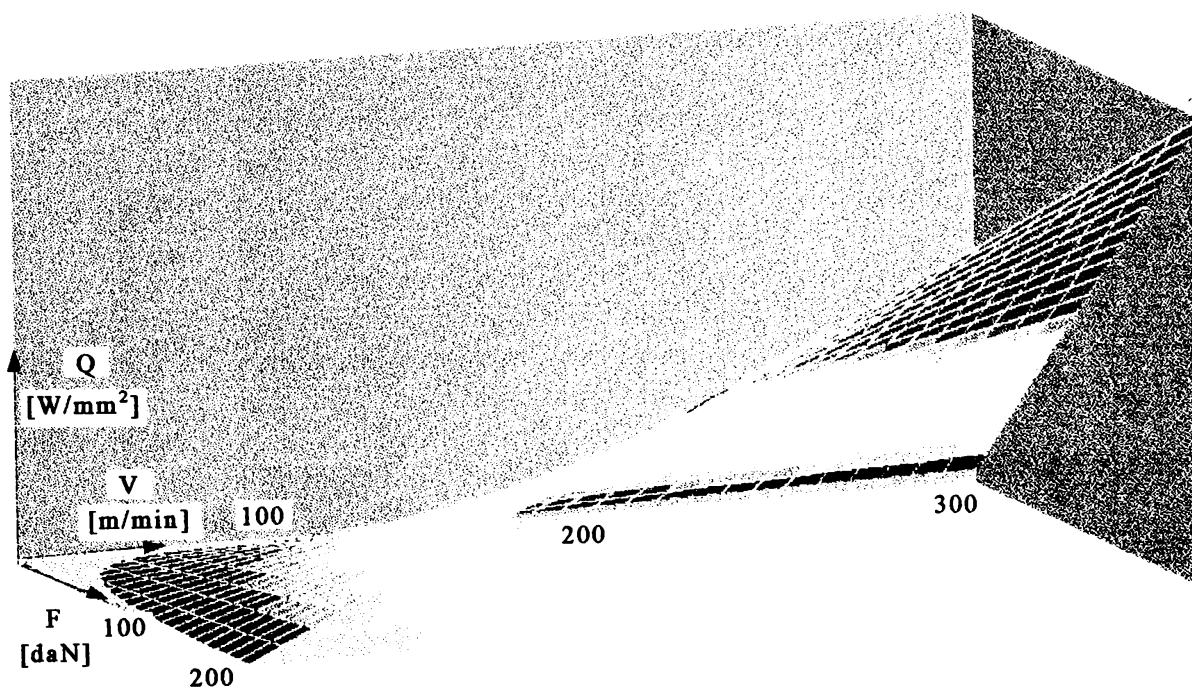


Fig. 97 Grafic de variație  $Q=f(F,v)$  pentru EN24T \ K68<sup>63</sup>

<sup>63</sup> Alt unghi de vedere

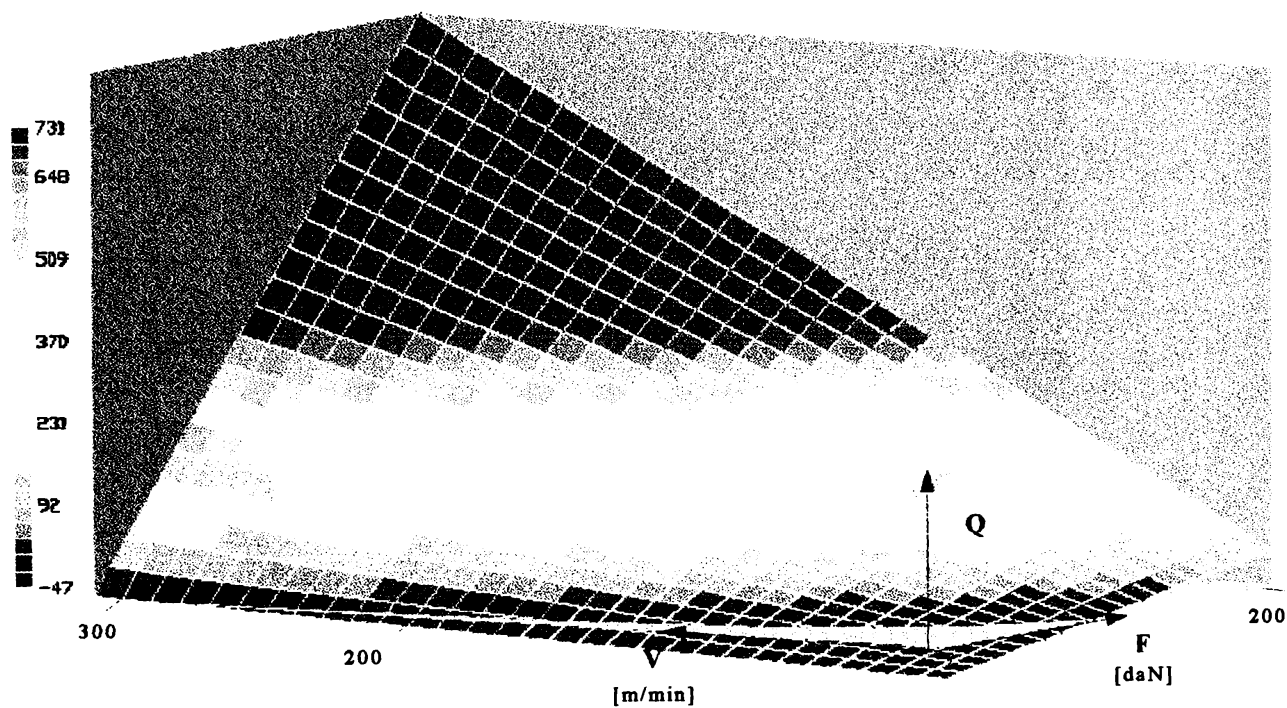


Fig. 98 Grafic de variație  $Q=f(F,v)$  pentru EN24T \ KC990

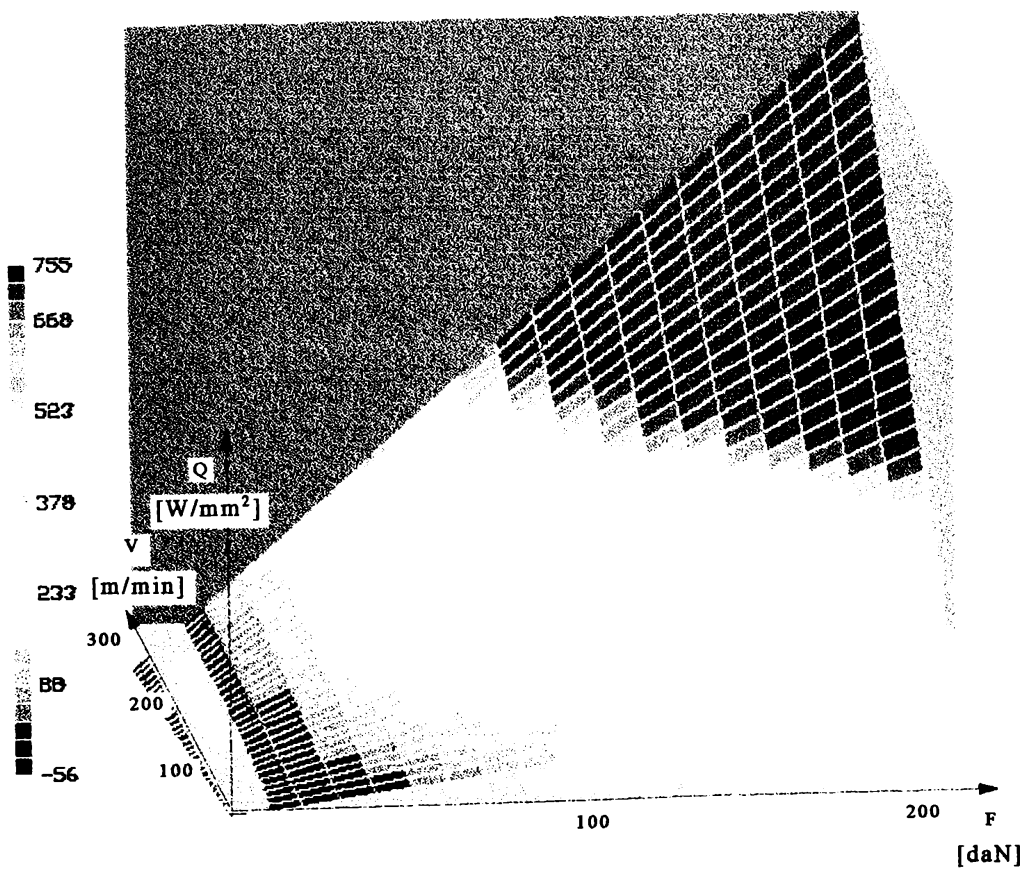


Fig. 99 Grafic de variație  $Q=f(F,v)$  pentru EN8 \ KC910

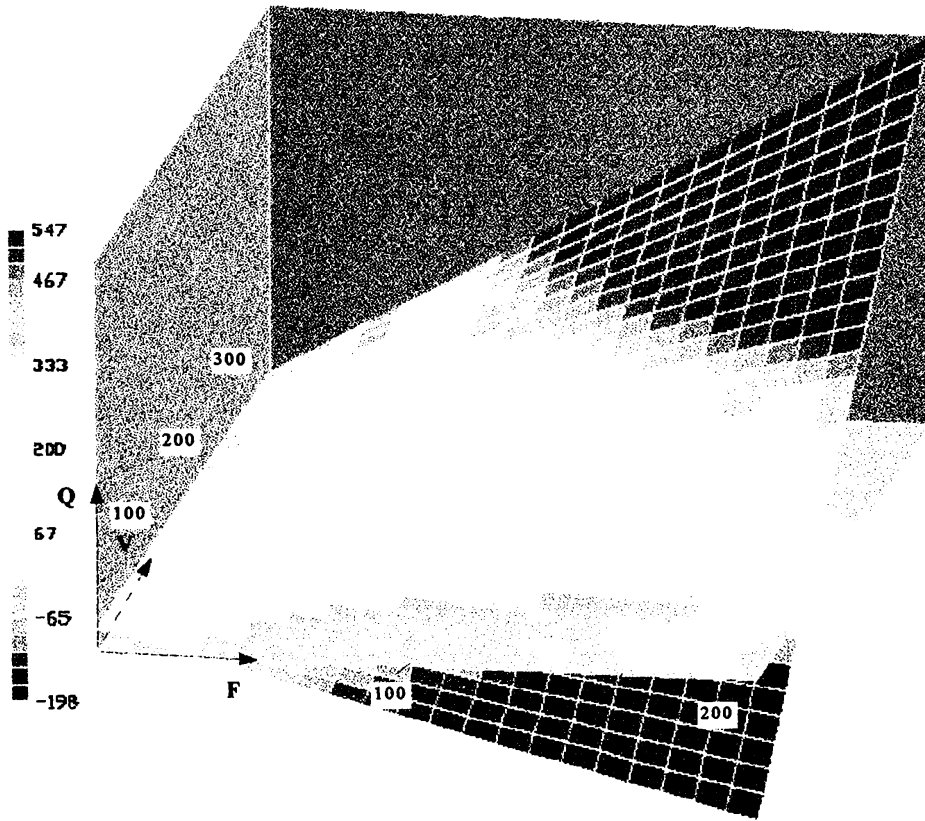


Fig. 100 Grafic de variație  $Q=f(F,v)$  pentru EN8 \ KC950

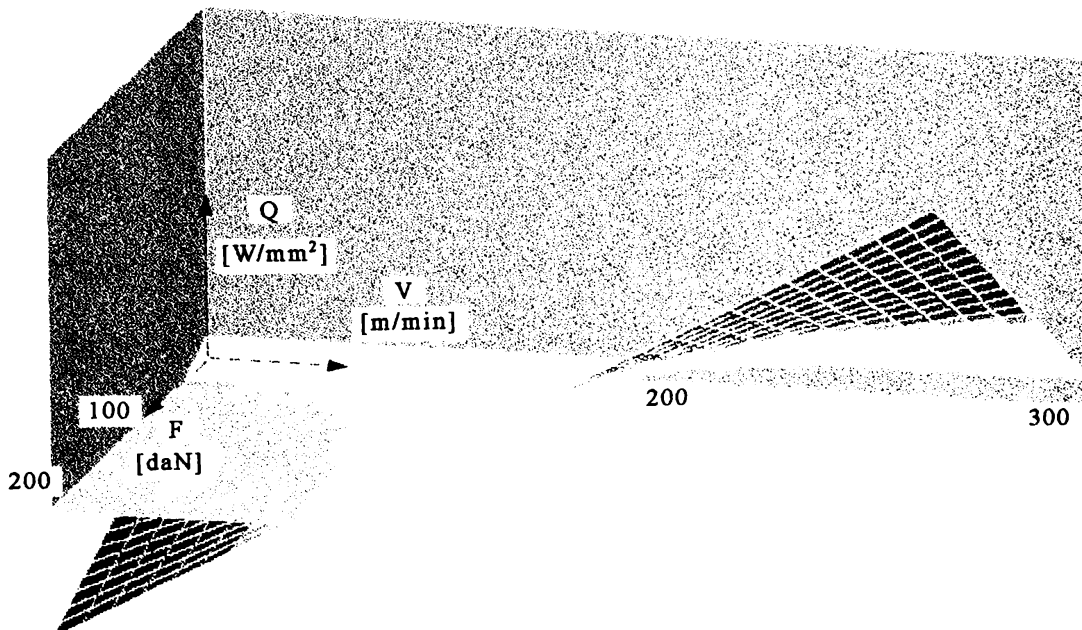


Fig. 101 Grafic de variație  $Q=f(F,v)$  pentru EN8 \ KC950<sup>64</sup>

<sup>64</sup> Alt unghi de vedere



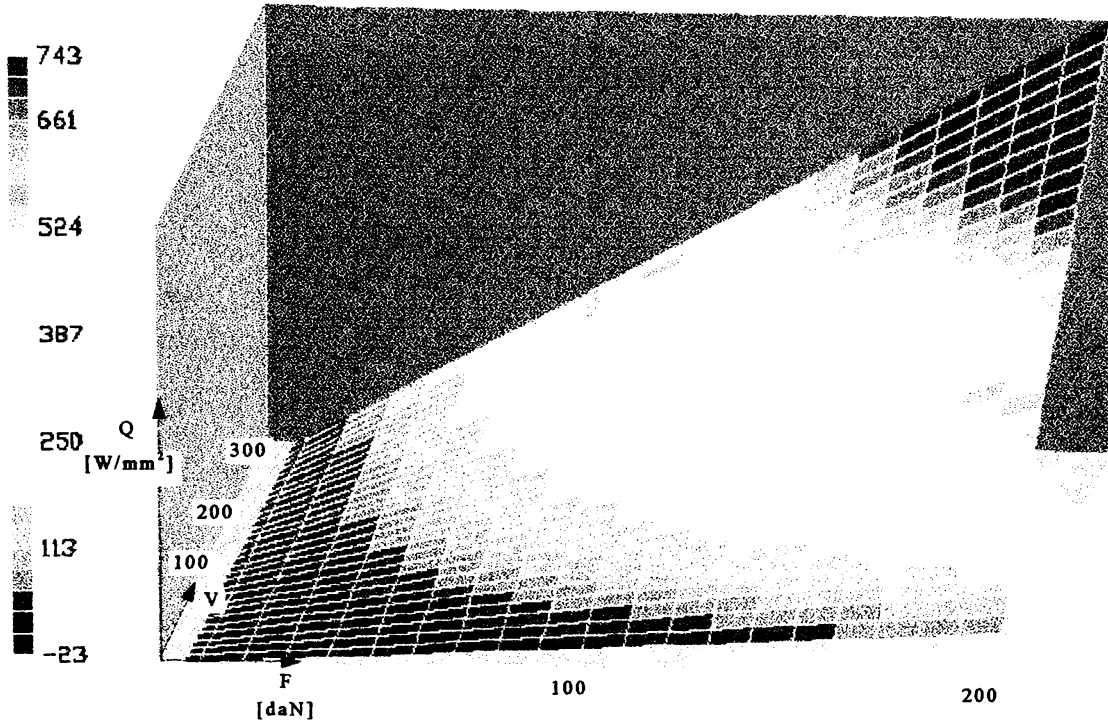


Fig. 102 Grafic de variație  $Q=f(F,v)$  pentru EN1 \ KC910

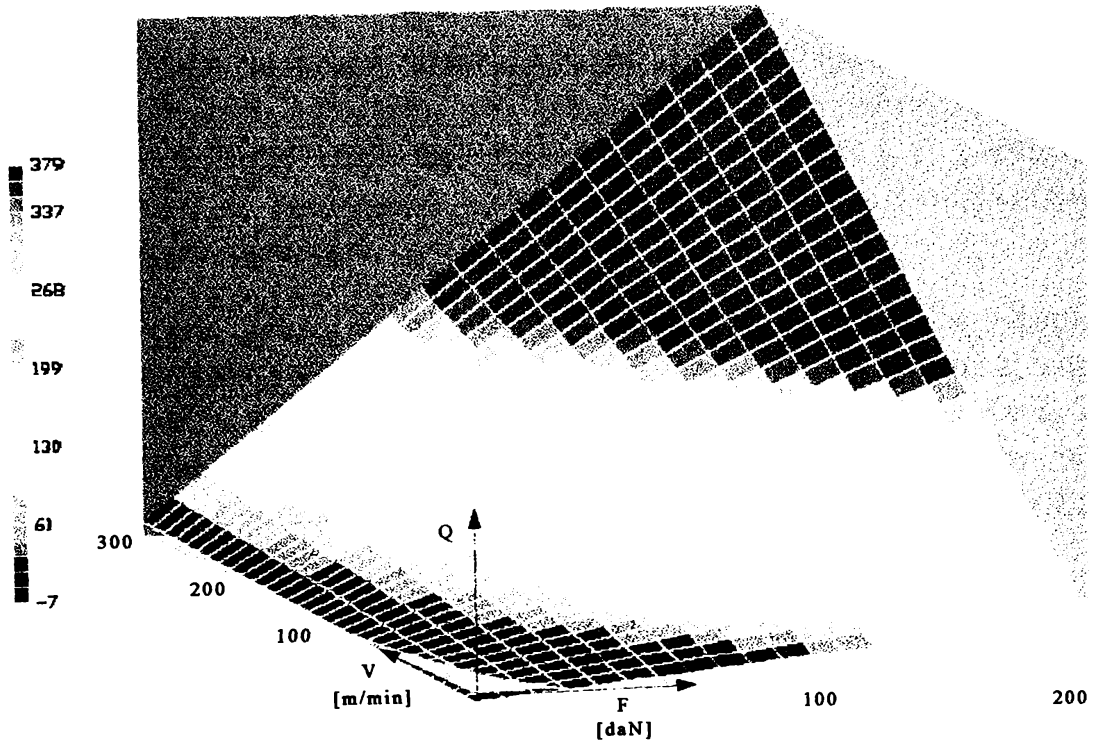


Fig. 103 Grafic de variație  $Q=f(F,v)$  pentru Al \ KC910

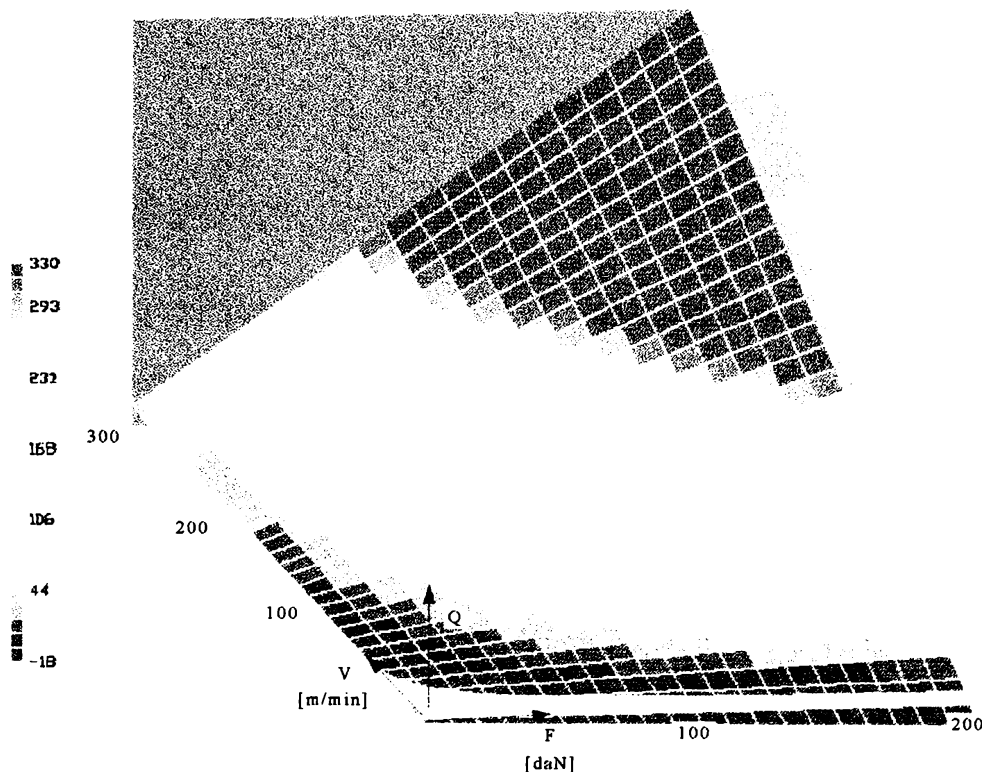


Fig. 104 Grafic de variație  $Q=f(F,v)$  pentru Al \ K68

#### Observații

- După cum se poate observa, la unele dintre graficele prezentate, apare în zona originii o porțiune unde suprafața  $Q(F,V)$  intersectează planul XOY. Porțiunea dintre curba de intersecție și axele OX, OY, conform relațiilor obținute, ar corespunde unor valori negative ale debitului sursei, ceea ce fizic este imposibil.

- Explicația pentru acest fenomen constă în faptul că la elaborarea relațiilor s-au folosit valori mai mari ale vitezelor de așchiere, și combinații de parametri de așchiere, care produc forțe de asemenea mai mari decât cele din zona în discuție. Astfel, pe măsură ce valorile pentru viteze se îndepărtează de intervalul (100-200) [m/min] iar cele pentru forțe de intervalul (10-200) [daN], relațiile conduc la rezultate mai puțin precise.

- Acest lucru nu reduce din utilitatea practică a relațiilor obținute, întrucât combinațiile de viteze și forțe din zonele respective nu pot să apară nici odată în practică. După cum s-a verificat, relațiile oferă o precizie bună pentru valori ale parametrilor apropiate de cele din timpul experimentului.

## 6. CONCLUZII

În urma finalizării cercetării, se pot formula concluzii a căror sinteză este prezentată în cele ce urmează:

### 6.1. Despre utilizarea termografiei în infraroșu

- Este metoda de investigare a câmpului termic la suprafața unui corp, care oferă cel mai mare volum de informație utilă și cea mai bună precizie.

- În procesele de prelucrare, distanța maximă de observare a zonei de așchiere, pentru obținerea unor imagini termice cu rezoluție utilizabilă, este de cca. 0,5 m.

- Pentru calibrarea ulterioară a imaginilor termografice, este foarte importantă cunoașterea emisivității corpurilor. Astfel, pentru temperaturi de 500°C, după aplicarea corecției de emisivitate, la termogramele realizate, s-au obținut diferențe de până la 20% față de termograma necorectată (cea inițială).

- Pentru corpurile din zona de așchiere, unde temperatura este mare, trebuie luată în considerare și dependența  $\varepsilon(T)$ . În acest caz, calibrarea termogramei impune și rezolvarea ecuației

$$\varepsilon(T) \cdot (T_{\text{Reală}}^4 - T_{\text{Mediu}}^4) = T_{\text{Măsurată}}^4 - T_{\text{Mediu}}^4$$

pentru toate punctele studiate.

- Variația emisivității în funcție de temperatură poate să aibă forme foarte diferite de la material la material; pentru plăcuțele de așchiere studiate emisivitatea crește odată cu temperatura, până la o anumită valoare, după care din cauza arderii peliculei de oxizi de la suprafață, apare o tendință de scădere; la unele materiale, cum ar fi cuprul, acest comportament este contrar.

- Pentru plăcuțele de așchiere studiate, la temperaturi de 450°C, s-au obținut variații ale emisivității totale care au ajuns până la 20% față de cea măsurată la temperatura ambiantă.

- La calibrări precise ale termogramelor, pentru corpuri cu suprafețe plane care sunt oblice față de direcția de măsură, trebuie luată în considerare și variația emisivității față de unghiul făcut cu normala la suprafață; pentru plăcuțele de așchiere studiate, s-au obținut variații care la tempera-

tura de circa 400°C au ajuns la valori maxime de 10% între cele trei direcții de măsurare folosite în experiment.

- La măsurarea distribuției câmpului termic în piesă și subansamblurile mașinii unelte, unde diferențele maxime de temperatură față de mediul ambiant sunt mici, imaginile termografice sunt afectate de *zgomotul de fond atmosferic*, produs de microparticulele de praf care absorb și radiază radiația infraroșie primită de la diverse surse.

- Măsurarea câmpului termic din piesă în timpul așchierii prin termografie este dificilă, deoarece suprafața metalului așchiat reflectă foarte bine, pe întreaga lungime a semifabricatului, radiația puternică primită din zona de așchiere. Acest fenomen creează o falsă distribuție a câmpului termic din piesă, așa după cum se poate observa în termogramele prezentate în lucrare.

- Termogramele la care diferențele maxime de temperatură între punctele zonei studiate sunt de circa 150÷200°C, prezintă contururi mai clare, fiind mai ușor de interpretat. Acest lucru depinde și de fereastra de temperatură selectată la operarea scannerului, sensibilitatea și precizia acestuia scăzând odată cu creșterea temperaturii măsurate.

- În cursul experimentului realizat, s-au măsurat temperaturi care, pentru mașina unealtă, în zona păpușii fixe, sunt de circa 40°C la două ore de funcționare.

- Temperaturile maxime măsurate diferite în puncte de pe fața de degajare a sculei, au valori cuprinse între 100°C la prelucrarea aluminiului și 540°C la prelucrarea EN24T.

- Temperaturile maxime indicate în piesă, care s-au măsurat după încheierea prelucrării, (din motivele enumerate anterior), s-au situat între 32°C la prelucrarea aluminiului și 54°C la prelucrarea EN24T.

- Este de remarcat faptul că temperaturile măsurate pe fața de degajare a sculei, sunt mai mici decât cele indicate în unele lucrări de specialitate.

- Imaginile termografice prelucrate și calibrate, oferă un material indispensabil la aflarea soluției reale a ecuației căldurii în anumite condiții. Aceste imagini au fost elementul esențial în stabilirea unor expresii de estimare a puterii sursei termice din procesele de așchiere.



## 6.2. Despre utilizarea măsurătorilor tensometrice

- Echipamentul folosit, cu referire la amplificatorul multicanal realizat, s-a dovedit a fi sensibil și fiabil.

- Pentru etalonarea datelor măsurate dinamic, s-au folosit ecuațiile de etalonare stabilite în urma aplicării unor forțe cunoscute pe un echipament de laborator.

- Considerăm metoda de măsurare a forțelor cu rozetă delta și etalonarea datelor prin procedeul prezentat, ca fiind mai precisă decât cea de plasare a trei timbre pe direcții preferențiale prestabilite, pe care alungirile specifice sunt produse dominant de una din componentele forțelor de așchiere.<sup>65</sup>

- Influența temperaturii asupra indicațiilor tensometrice este vizibilă chiar și în condițiile în care s-au utilizat timbre cu autocompensare termică. Pentru lipirea timbrelor s-a folosit adeziv cu proprietăți izolante iar puntea tensometrică a avut un element inactiv în contact termic cu suprafața, pentru a asigura compensarea efectului termic.

- Valorile forțelor măsurate, sunt cuprinse între 7 daN la prelucrarea aluminiului și 180 daN la prelucrarea EN24T. Experiențele efectuate de autor arată că aceste valori sunt cu circa 10% mai mari decât cele obținute cu relații de calcul din literatura de specialitate.

- Valorile forțelor măsurate au fost prelucrate prin regresie liniară, precizia cea mai bună de  $\approx 2\%$  fiind oferită la prelucrarea oțelurilor; la prelucrarea aluminiului, precizia obținută este de  $\approx 10\%$ . Acest aspect se explică prin influența procentuală mai mare pe care o au factorii perturbatori asupra mărimilor măsurate, care, la prelucrarea aluminiului sunt mai mici.

## 6.3. Despre aplicarea metodei elementului finit la calculul puterii sursei termice

- La rezolvarea ecuației căldurii, pentru cazul staționar, s-a folosit formularea variațională a metodei elementului finit, având ca necunoscută funcția care modelează comportamentul surselor interne și cunoscând distribuția câmpului de temperaturi, deci soluțiile aproximative pentru funcția  $T(x, y, z)$ .

---

<sup>65</sup> La o analiză a stării de deformații produse de forțele de așchiere pe fiecare din direcțiile studiate, se simte și influența unei alte componente a forțelor de așchiere

- Corectitudinea algoritmilor care implementează metoda, a fost verificată, rezultatele fiind foarte precise. În cazul rulării unei probleme inverse cu datele obținute de la rularea unei probleme directe, diferențele finale pentru valorile încărcărilor la nivel de element au fost sub 0,1%.

- Pentru prelucrările efectuate, puterea calculată a sursei termice a variat între 11 [W/mm<sup>2</sup>] la prelucrarea aluminiului și 340 [W/mm<sup>2</sup>] la prelucrarea oțelului EN24T.

- Valorile încărcărilor la nivel de element, au fost tratate prin regresie liniară și de ordinul doi, obținându-se relații care permit evaluarea puterii sursei termice în condiții date.

- Estimările obținute cu aceste relații au fost comparate cu valorile rezultate în urma analizei cu element finit și s-a ajuns la concluzia că precizia cea mai bună o oferă la prelucrarea oțelurilor  $\approx \pm 2,5\%$ ; pentru prelucrarea aluminiului, relațiile obținute sunt mai puțin precise, valorile fiind în jur de  $\pm 12\%$ ; explicația pentru această tendință fiind aceeași ca cea de la relațiile pentru forțe.

- În cazul introducerii condițiilor de contur de tip Dirichlet, rezultatele obținute prin folosirea metodei termenul dominant pe diagonală și a metodei curățirii liniei și coloanei corespunzătoare nodului, au fost foarte apropiate, cu diferențe în limitele  $\pm 1\%$ .

#### **6.4. Despre influența factorilor de prelucrare asupra puterii sursei termice**

- Cea mai puternică influență asupra puterii sursei termice a fost exercitată de proprietățile materialului așchiat. Astfel, oțelul EN24T cu  $\sigma_r=95$  [daN/mm<sup>2</sup>] produce o sursă termică ce ajunge până la 350 [W/mm<sup>2</sup>], în timp ce aluminiul cu  $\sigma_r=7$  [daN/mm<sup>2</sup>], în condiții asemănătoare de prelucrare, produce o sursă termică cu valori maxime sub 50 [W/mm<sup>2</sup>].

- Creșterea adâncimii de așchiere, produce, de asemenea, mărirea puterii sursei termice. Astfel, pentru oțelul EN24T la o viteză de 184 [m/min] și un avans de 0,2 [mm/rot], creșterea adâncimii de așchiere de la 0,5 mm la 0,7 mm provoacă o creștere a puterii sursei termice de la 258 la 355 [W/mm<sup>2</sup>].

- Creșterea avansului provoacă și el creșterea puterii sursei termice, însă într-o măsură mai mică decât adâncimea de așchiere. Astfel, pentru oțelul EN24T la o viteză de 184 [m/min] și o adâncime de așchiere de 0,5 [mm], creșterea avansului de așchiere de la 0,1 mm la 0,2 mm provoacă o creștere a puterii sursei termice de la 245 la 258 [W/mm<sup>2</sup>].

## 7. CONTRIBUȚII

Urmărind derularea cercetărilor, echipamentele folosite și fazele de lucru, contribuțiile autorului, atât în plan teoretic, cât și în plan aplicativ, s-au concretizat în următoarele:

### 7.1. În plan teoretic

- Rezolvarea ecuației conducției căldurii, prin metoda elementului finit, pentru un proces de strunjire, având ca necunoscută funcția ce mod-elează aportul surselor termice interne, în ipoteza cunoașterii dis-tribuției de temperaturi și a condițiilor de contur.
- Elaborarea unor algoritmi și a unui program de calcul complet, pentru rezolvarea problemei de mai sus, folosind formularea variațională a metodei elementului finit, cu elemente triunghiulare liniare ortotrope, în condițiile introducerii condițiilor de contur de tip Dirichlet, Cauchy, transfer convectiv, transfer radiant de căldură și dependență de tem-peratură a proprietăților de material (program FEM).
- Elaborarea algoritmilor și a programului de calcul pentru problema con-ducției căldurii în cazul normal, având ca necunoscută funcția de dis-tribuție a temperaturii, cu același tip de elemente și condiții de contur de mai sus (program FEM).
- Ca element de noutate, s-a folosit alocarea dinamică în modul protejat, ceea ce a simplificat mult algoritmi pentru secvențele de cod ce adre-sează datele din matricele globale, prin eliminarea segmentării acestora, mărind totodată și viteza de rezolvare a sistemului.
- S-au folosit rezultatele obținute cu acest program, pentru deducerea prin regresie a unor relații de predicție a aportului surselor termice în proce-sele de strunjire exterioară.
- O primă grupă de relații estimează puterea sursei în funcție de avans, viteză și adâncimea de așchiere, iar a doua estimează această putere în funcție de viteză și forța de așchiere.

- Se apreciază că utilitatea acestor relații este evidentă, pentru că:
  - ◇ ele oferă o posibilitate de estimare a puterii sursei termice și calcularea ulterioară precisă a câmpului termic;
  - ◇ rezultatele din acest studiu, introduse într-o analiză statică sau dinamică cu elemente finite permit calculul deplasărilor, deci se poate controla astfel unul din factorii care au implicații asupra preciziei de prelucrare;
  - ◇ relațiile stabilite sunt utile și la strunjiri interioare, unde aceste estimări nu ar fi fost posibile prin folosirea relațiilor deja cunoscute din literatura de specialitate.
- S-au stabilit prin regresie liniară, o serie de relații pentru calculul forței de așchiere în funcție de parametrii  $v$ ,  $s$ ,  $t$ .
- Au fost elaborate:
  - ◇ un algoritm și un program de calcul pentru preluarea temperaturii și a fluxului pe contur din termograme în infraroșu, cu suprapunerea rețelei de elemente finite corespunzătoare cazului (program TERMO).
  - ◇ algoritm și program de calcul pentru postprocesarea soluției obținute după rularea problemei de element finit (program POST).
  - ◇ program de captare a datelor experimentale pe 8 canale, din placa de achiziții de date PCL711S (program PCL711).
  - ◇ program de reprezentare grafică pentru funcție de 2 variabile (program GRAFIC).
  - ◇ algoritmi pentru foi de calcul Excel în vederea prelucrării datelor experimentale.

## 7.2. În plan experimental

- Proiectarea și realizarea instalației experimentale pentru determinarea emisivității corpurilor mici.
- Determinarea funcțiilor de distribuție a temperaturii în diverse condiții de prelucrare pentru operații de strunjire.
- Determinarea, pe baza experiențelor făcute, a curbelor de dependență a emisivității în funcție de temperatură, pentru plăcuțe de așchiere din gama Kennametal.

- Elaborarea unei metode originale de etalonare a semnalului tensometric, pe care am descris-o în lucrare.
- Determinarea și măsurarea forțelor de așchiere la operații de strunjire, în diverse condiții de prelucrare.
- Elaborarea unor relații de estimare a debitului sursei termice, de forma  $Q=f(F, v)$  și  $Q=f(v, s, t)$ , pentru o gamă de materiale și plăcuțe de așchiere.

# ANEXE



## 8. ANEXA 1 - PROGRAM ANALIZĂ ELEMENT FINIT

### 8.1. CARACTERISTICI

- Program de analiză cu element finit pentru transfer static de căldură;
- Limbaj de programare folosit: C;
- Compilator: WATCOM 10;
- Tipuri de elemente:
  - triunghiular plan, cu variație liniară, pentru materiale ortotrope;
- Tipuri de neliniarități acceptate: de material, cu dependență de temperatură pentru emisivitate;
- Format fișier executabil - aplicație ADS, rulează în modul protejat sub platforma AutoCAD R12;
- Funcții apelabile la prompterul Command:

#### FEM

Încarcă fișierul cu extensia ADB, unde se află descrisă problema;

#### MODEL

Desenează modelul geometric, cu numerotarea nodurilor și elementelor în straturi separate;

#### SOLVE

Rezolvă problema descrisă în fișierul de intrare, pentru cazul direct, având ca necunoscute temperaturile nodale; creează fișierul de ieșire REZ99.DAT, care poate fi folosit direct la postprocesare cu programul POST din anexa 2;

#### INV

Rezolvă problema descrisă în fișierul de intrare, pentru cazul invers, având ca necunoscute încărcările la nivel de element; creează fișierul de ieșire REZ100.DAT.

#### VER

Verifică validitatea datelor din fișierul de intrare, depistând referiri la noduri sau elemente inexistente, noduri coincidente și lipsa unor secțiuni din datele de intrare;

#### Fișiere de intrare

Fișierul cu descrierea problemei:

format : ASCII

extensia implicită : ADB

structura fișierului: cuprinde mai multe secțiuni care vor fi descrise în continuare:

• Fiecare secțiune începe cu o linie de identificare la care primul caracter este #, în continuare fiind prezentați prin subliniere identificatorii care sunt șiruri de caractere obligatorii la determinarea unor anumite date;

##### 1. secțiunea de titlu a problemei

Linia de identificare : #TITLE

Conținutul secțiunii: o linie cu un șir de maxim 132 de caractere, cu titlul problemei, care va fi trecut în fișierul de ieșire

##### 2. secțiunea de descriere a nodurilor

Linia de identificare : #NODES

Conținutul secțiunii: descrierea secvențială a nodurilor, câte unul pe fiecare linie, cu următorul format:

*NumărNod, CoordX, CoordY, CoordZ*

##### 3. secțiunea de descriere a elementelor

Linia de identificare : #ELEMENTS

Conținutul secțiunii: descrierea secvențială a elementelor, cu următorul format:

*NumărElem, TipElement, NumărMaterial  
Nod1, Nod2, Nod3*

Unde tipul de element actualmente implementat, cel triunghiular plan ortotrop, liniar, are codul 258, iar număr material este numărul unuia dintre cele descrise la secțiunea de mai jos;

##### 4. secțiunea de descriere a materialelor

Linia de identificare : #MATERIALS

Conținutul secțiunii: descrierea proprietăților materialelor existente, în versiunea curentă sunt disponibile: valoare conductivitate pe direcția X respectiv Y și emisivitate dependentă polinomialde temperatură, cu următorul format:

MATID, *NrMaterial*, *NumeMaterial*

MATID, *NrMaterial*, KX, *valoare coeficient de conducție pe direcția X*, KY, *valoare coeficient de conducție pe direcția Y*

MATID, *NrMaterial*, EMIS, *T0, T1, T2, T3, T4, T5, T6*

unde T0...T6 sunt coeficienții polinomului de descriere a variației  $\epsilon=f(T)$ , în ordinea  $T^0, T^1, \dots, T^6$

Obs: Dacă se specifică numai KX materialul se consideră izotrop;

#### 5.secțiunea de descriere a condițiilor de contur

Linia de identificare : #BC

Conținutul secțiunii: descrierea condițiilor de contur, în versiunea curentă fiind acceptate următoarele tipuri de condiții de contur: temperatură impusă în nod, flux impus pe o latură a elementului, schimb prin radiație impus pe o latură a elementului, schimb prin convecție impus pe o latură a elementului, cu următorul format:

TEMPE, *NrNod*, *ValTemp*

DFLUX, *NrElement*, *Nod1*, *Nod2*, *Valoare*

RADIATIE, *NrElement*, *Nod1*, *Nod2*, *Valoare*

CONVECTIE, *NrElement*, *Nod1*, *Nod2*, *Valoare*

Obs: Secțiunea este utilizabilă și la rularea problemei directe și la cea a problemei inverse, unde trebuiesc completate pentru toate nodurile rețelei valorile temperaturilor;

#### 6.secțiunea de descriere a condițiilor de contur inițiale

Linia de identificare : #BCINI

Conținutul secțiunii: descrierea condițiilor de contur de tip temperatură impusă, care trebuie introduse pentru rezolvarea problemei inverse; se introduc în câteva noduri care nu sunt conectate la elemente cu sursă de căldură; formatul este identic cu cel de la secțiunea precedentă:

TEMPE, *NrNod*, *ValTemp*

#### 7.secțiunea de descriere a încărcărilor

Linia de identificare : #LOADS

Conținutul secțiunii: descrierea încărcărilor termice; în această versiune este disponibilă numai încărcarea distribuită în tot elementul; formatul este următorul:

ESOURCE, *Număr element*, *valoare încărcare*

Obs: Secțiunea este utilizabilă atât la rularea problemei directe cât și la cea a problemei inverse, unde servește pentru identificarea elementelor în care se va calcula valoarea încărcării; în această situație câmpul *valoare de încărcare* din fișierul de intrare nu are nici o importanță;

#### 8.secțiunea de descriere a constantelor fizice

Linia de identificare : #CONST

Conținutul secțiunii: descrierea unor constante folosite în calcule; în versiunea curentă sunt disponibile: temperatura mediului și constanta lui Boltzmann, cu următorul format:

TEMPMEDIU, *ValTemp*

BOLTZMANN, *ValBoltzmann*

Obs: Dacă nu sunt precizate, sunt folosite următoarele valori implicite: pentru temperatura mediului 20°C și pentru constanta lui Boltzmann 5.66961E-8 [W/mm<sup>2</sup>/°K<sup>4</sup>]; constanta lui Boltzmann poate fi primi altă valoare în funcție de unitățile de măsură folosite;

#### 9.secțiunea de descriere a elementelor de control a programului

Linia de identificare : #CONTROL

Conținutul secțiunii: descrierea unor elemente opționale de control a execuției programului; în această versiune sunt disponibile: controlul modului de rezolvare a sistemului final; controlul modului de impunere a condițiilor de tip Dirichlet, precizia soluției între două iterații consecutive când apare schimb de căldură prin radiație, numărul maxim de iterații când apare schimb de căldură prin radiație, precizia de verificare a coincidenței dintre două noduri și

precizarea unui factor de scară pentru coordonatele geometrice ale nodurilor; formatul fiind următorul:

GAUSS, *TipSol*

Unde TipSol poate fi: ELABORAT sau SIMPLU, cu semnificațiile: ELABORAT rezolvă sistemul cu tehnica pivotării totale, SIMPLU rezolvă sistemul triangularizând direct matricea și folosind ca pivot elementul curent din linia și coloana corespunzătoare variabilelor de control a ciclului;

Obs. Versiunea de rezolvare a sistemului cu calcularea directă a inversei matricei de conductivitate (identificatorul INVERS) nu este implementată; valoarea implicită este SIMPLU

MODBC, *Mod*

Unde Mod poate fi: GRAND sau ELIMINARE, cu semnificațiile: GRAND introduce condițiile Dirichlet prin metoda termen dominant pe diagonală iar ELIMINARE introduce condițiile Dirichlet prin metoda curățirii liniei și coloanei corespunzătoare nodului;

Obs: valoarea implicită este ELIMINARE

PRECITER, *Precizie*

Unde Precizie este o valoare folosită la verificarea condiției de încheiere a procesului de rezolvare iterativă a sistemului, când emisivitatea depinde de temperatură; reprezintă pragul pentru cea mai mare diferență acceptabilă între soluțiile tuturor temperaturilor nodale, obținute la două iterații consecutive;

Obs: este folosită numai la rezolvarea problemei directe; valoarea implicită este  $10^{-5}$

MAXITER, *NrIter*

Unde NrIter este o valoare folosită la verificarea condiției de încheiere a procesului de rezolvare iterativă a sistemului, când emisivitatea depinde de temperatură; reprezintă numărul maxim de iterații admisibile efectuate la problemele statice cu neliniarități de material;

Obs: este folosită numai la rezolvarea problemei directe; valoarea implicită este 50;

PRECVER, *Precizie*

Unde Precizie este o valoare folosită la verificarea condiției de coincidență a coordonatelor a două noduri;

Obs: valoarea implicită este  $10^{-3}$ ;

SCARAGEO, *Scara*

Unde Scara este un număr folosit pentru a multiplica coordonatele geometrice la alcătuirea matricelor sistemului; se folosește pentru a acorda unitățile de măsură la mărimile implicate în analiză;

Obs: valoarea implicită 1;

### 10. secțiunea de încheiere a datelor de intrare

Linia de identificare : #END

Este singura linie din această secțiune și folosește la marcarea sfârșitului datelor;

Obs: La începutul unei linii se mai poate folosi șirul de caractere // pentru a introduce comentarii în vederea îmbunătățirii clarității datelor de intrare

### Fișiere de ieșire

*La problema directă*

format : ASCII

nume implicit : *REZ99.DAT*

structura fișierului: cuprinde 4 secțiuni : titlul problemei, coordonate noduri, rețea de elemente, soluția problemei - temperaturile nodale;

Obs: Fișierul poate fi folosit direct de programul de postprocesare, prin schimbarea extensiei;

*La problema inversă*

format : ASCII

nume implicit : *REZ100.DAT*

structura fișierului: cuprinde 4 secțiuni : titlul problemei, coordonate noduri, rețea de elemente, soluția problemei - valorile încărcărilor din elementele indicate în secțiunea LOADS, care au produs distribuția câmpului termic cu valorile nodale din secțiunea BC.

## 8.2. FIȘIER HEADER

```

#define Marcator      "#"
#define Comment      "//"
#define NoInit       0
#define IsTitle      1
#define IsNodes      2
#define IsMaterials   4
#define IsBC         8
#define IsLoads      16
#define IsElements   32
#define IsBCIni      64
#define IsConstants  128
#define IsControl    256
#define Title        "TITLE"
#define Nodes        "NODES"
#define Materials    "MATERIALS"
#define BC           "BC"
#define BCIni       "BCINI"
#define Loads        "LOADS"
#define Elements     "ELEMENTS"
#define Constants    "CONST"
#define Control      "CONTROL"
#define EndOfASCII   "END"
#define LgASCIILine  255
#define EROARE       0
#define OK           1
#define English      0
#define Romanian     1
#define LgCR         2
#define CR           "\015\012"
#define Original     0
#define Nou          1

#define Beam         0
#define Truss        1
#define Triangular   2
#define Qadrilater   3
#define Prismatic    4
#define Hexaedric    5
#define Tetraedric   6

#define Liniar       1
#define Quadratic    2
#define Cubic        3

#define LowerMask    0X00FF
#define UpperMask    0XFF00

#define TriangLiniar Triangular & (Liniar << 8) //258
#define QadrilaterLiniar Qadrilater & (Liniar << 8) //259

#define SirMatName   "NAME"
#define SirMatID     "MATID"
#define SirPoissonX  "PX"
#define SirPoissonY  "PY"
#define SirPoissonZ  "PZ"

#define SirYoungX    "YX"
#define SirYoungY    "YY"
#define SirYoungZ    "YZ"

#define SirAlphaX    "ALX"
#define SirAlphaY    "ALY"
#define SirAlphaZ    "ALZ"

#define SirTransversalX "TX"
#define SirTransversalY "TY"
#define SirTransversalZ "TZ"

#define SirDensity    "DENS"
#define SirRefTemp    "REFT"
#define SirSpecificHeat "C"
#define SirKx         "KX"
#define SirKy         "KY"
#define SirKz         "KZ"
#define SirEmissivity "EMIS"

#define SirTemperatura "TEMPE"
#define SirCFlux       "CLFUX"
#define SirDFlux       "DFLUX"

```

```

#define SirRadiatie "RADIATIE"
#define SirConvectie "CONVECTIE"
#define SirTempMediu "TEMPMEDIU"

#define SirElemSource "ESOURCE"
#define SirNodSource "NSOURCE"
#define SirBoltzmann "BOLTZMANN"

#define SirGauss "GAUSS"
#define SirPrecIter "PRECITER"
#define SirMaxIter "MAXITER"
#define SirModBC "MODBC"
#define SirPrecVer "PRECVER"
#define SirScGeo "SCARAGEO"

#define SirElaborat "ELABORAT"
#define SirSimplu "SIMPLU"
#define SirInvers "INVERSA"
#define SirGrand "GRAND"
#define SirElimina "ELIMINARE"

#define GAUSS_SIMPLU 2000
#define GAUSS_ELABORAT 2001
#define GAUSS_INVERS 2002
#define PRECITER 1.0E-5L
#define MAXITER 50
#define BC_ELIMINA 2002
#define BC_GRAND 2003
#define BOLTZMANN 5.66961E-8L // [W/mm2/K4];
#define TEMP_MEDIU 20
#define GDL 3

#define GRAND_VAL 1.0E100L
#define MIN_VAL 1.0E-100L
#define DIF_COOR 1.0E-3L
#define MaxTermPolinom 7

#define Direct 0
#define Invers 1
#define GATA 0x100
#define MAI_DEPARTE 0x200

typedef int FEMType; // Pentru tipul de data atasat nr de nod si element

struct NodesStruct
{
    FEMType Nr;
    ads_real x;
    ads_real y;
    ads_real z;
    ads_name Sys;
};
typedef struct NodesStruct NodesType;

struct NodesList
{
    FEMType Nod;
    struct NodesList *Urm;
};
typedef struct NodesList NodesListType;

struct ElementsStruct
{
    FEMType Nr;
    unsigned char Shape;
    unsigned char Function;
    unsigned char Material;
    ads_real Jacobi;
    NodesListType *NodList;
};

typedef struct ElementsStruct ElementsType;

struct ElementsList
{
    ElementsType *Element;
    struct ElementsList *Urm;
};
typedef struct ElementsList ElementsListType;

struct MaterialsStruct
{
    int MatID;

```

```

        ads_real    PoissonX;
        ads_real    PoissonY;
        ads_real    PoissonZ;
        ads_real    YoungX;
        ads_real    YoungY;
        ads_real    YoungZ;
        ads_real    AlphaX;
        ads_real    AlphaY;
        ads_real    AlphaZ;
        ads_real    TransversalX;
        ads_real    TransversalY;
        ads_real    TransversalZ;
        ads_real    Density;
        ads_real    RefTemp;
        ads_real    SpecificHeat;
        ads_real    Kx;
        ads_real    Ky;
        ads_real    Kz;
        ads_real    Emissivity[7];
        char        Name[15];
        struct MaterialsStruct *Urm;
    };

typedef struct MaterialsStruct MaterialsType;

struct BCStruct
{
    FEMType        NodSauElement;
    unsigned char  TipRestrictie;
    int            Nod1, Nod2;
    ads_real       Impus;
    struct BCStruct *Urm;
};

typedef struct BCStruct BConditionsType;

struct LoadStruct
{
    FEMType        NodSauElement;
    unsigned char  TipRestrictie;
    ads_real       Impus;
    struct LoadStruct *Urm;
};

typedef struct LoadStruct LoadType;

enum LoadIdent {
    NodSource = 1 , ElemSource
};

enum BCIdent {
    Temperatura = 1 , FluxConcentrat , FluxDistribuit,
    Convectie, Radiatie, TranslatieX , TranslatieY , TranslatieZ ,
    RotatieX , RotatieY , RotatieZ
};

struct ListaPost
{
    NodesType      Nod;
    ads_real       Temp;
    struct Lista   *Urm;
};

typedef struct ListaPost ListaPostType;

struct ListaBCSave
{
    ads_real       (*PVectColumn)[1];
    ads_real       Impus;
    struct ListaBCSave *Urm;
};

typedef struct ListaBCSave ListaBCSaveType;

```



### 8.3. FIȘIER SURSĂ

```

/*-----
PROGRAM DE ANALIZĂ CU ELEMENTE FINITE LINIARE ORTOTROPE
PENTRU TRANSFER STATIC DE CĂLDURĂ
© Adrian ROSCA, 1997
Versiunea 1.2
-----*/

#include <stdio.h>
#include <fcntl.h>
#include <sys\stat.h>
#include <adslib.h>
#include <ads.h>
#include <adsdlg.h>
#include <adscodes.h>
#include <string.h>
#include <dos.h>
#include <math.h>
#include <malloc.h>
#include "fem.h"
#include "mes.h"

//
/**** INCEPUT SECȚIUNE VARIABLE GLOBALE ***
//

char JobTitle[132], Cale[60]="D:\\adi\\doct\\programe\\elemen-1\\";
FEMType NrNodes, NrElements, NrLoads, (*MatLoad)[1][1], Dif;
NodesType *PNodes;
ElementsType *PElements;
MaterialsType *PMaterials;
BConditionsType *PBConditions, *PBConditionsIni;
LoadType *PLoad, *PL;
ads_real (*PLoadsVect)[1], (*PSolutionVect)[1], (*PGlobalMatrix)[1][1], (*PTmpVect)[1];
FEMType NrNodes, NrElements, Iter;
int Language, IsDefinedSection = 0, GdlElem = GDL, Dim;
ads_real TemperaturaMediului = TEMP_MEDIU, Boltzmann = BOLTZMANN;
ListaPostType *PPost;
char SirErr[1024], T1[10];
FILE *Err;
int GaussTip = GAUSS_SIMPLU, MaxIter = MAXITER, ModBC = BC_ELIMINA;
ads_real PrecIter = PRECITER, PrecVer = DIF_COOR, ScaraGeometrie = 1;

//
/**** SFARSIT SECȚIUNE VARIABLE GLOBALE ***
//

void Scaleaza(ads_real Scara)
{
/**#1
// Funcție care scalează geometria modelului
//

NodesType *P1 = PNodes, N;
int i;

for(i = 0 ; i < NrNodes ; i++)
{
N = *P1;
N.x *= Scara;
N.y *= Scara;
_fmemcpy(P1, &N, sizeof(NodesType));
P1++;
}
}

ads_real RetEmissiv(int Nr, ads_real Temp)
{
/**#2
// Funcție care calculează emisivitatea pentru un material la o anumită temperatură
//

ads_real Emissiv = 0.0, Curent;
int i;

MaterialsType *PMat = PMaterials;
while(PMat && (PMat->MatID != Nr))
PMat = PMat->Urm;
if(PMat)
{

```

```

    for(i = 0 ; i < MaxTermPolinom ; i++)
        if(Curent = PMat->Emissivity[i])
            Emisiv += pow(Temp , i) * Curent;
    return Emisiv;
}
else
    return 1;
}

void Iese(int CodIesire)
///#3
// Funcție încheiere aplicație ADS
//
    {
        ads_exit(CodIesire);
    }

void VerificaHeap(int unde)
///#4
// Funcție care verifică integritatea zonei de HEAP
//
    {
        int rez;
        char Msg[50], tmp[5];

#ifdef _MSC_VER
        return ;
#else
#ifdef _BORLANDC__
        rez = heapcheck();
        if(rez != _HEAPOK)
#else
        rez = _heapchk();
        if((rez == _HEAPBADBEGIN) || (rez == _HEAPBADNODE))
#endif
            {
                strcpy(Msg , Mes[4][Language]);
                itoa(unde , tmp , 10);
                strcat(Msg , tmp);
                ads_alert(Msg);
                Iese(EROARE);
            }
#endif
    }

void *Aloca(int CiteObj, int DimZona)
///#5
// Funcție care face alocarea memoriei dinamice pentru un obiect
//
    {
        void *Pt;

        Pt = calloc(CiteObj, DimZona);
        if(!Pt)
            {
                ads_alert(Mes[1][Language]);
                Iese(0);
            }
        memset(Pt , '\000' , DimZona * CiteObj);
        return Pt;
    }

long GetLine(char *Destinatie , char *Sursa , long Cit, int *NrLin, char **Addr)
///#6
// Funcție care citește o linie din zona de memorie unde s-a introdus fișierul de intrare
//
    {
        char *PEnd;
        long Lg;
        int IsComment;

        do
            {
                IsComment = FALSE;
                PEnd = strchr(Sursa , 13);
                Lg = PEnd ? PEnd - Sursa : strlen(Sursa);
                Lg = Lg > Cit ? Cit : Lg;
                memcpy(Destinatie , Sursa , Lg);
                Destinatie[Lg] = EOS;
                if(strstr(Destinatie , Comment))
                    {

```

```

        IsComment = TRUE;
        Sursa += Lg + LgCR;
        *Addr = Sursa;
        (*NrLin)++;
    }
}
while (IsComment);

return Lg;
}

FEMType Cite(char *Cap, char *PFinal)
{
###7
// Funcție care calculează câte linii are o secțiune din fișierul de intrare
//
    FEMType Nr = 0;
    char *PTmp = Cap;
    char TmpLine[LgASCIILine + 1];
    int Lg, T1;
    int Iese = FALSE;

    do
    {
        Lg = GetLine(TmpLine , PTmp, LgASCIILine, &T1, &PTmp);
        PTmp += Lg + LgCR;
        if ((TmpLine[0] == Marcator[0]) || (PTmp >= PFinal))
            Iese = TRUE;
        else
            Nr++;
    }
    while (!Iese);
    return Nr;
}

void CutComma (char *ALine)
{
###8
// Funcție care înlocuiește virgulele din sintaxa datelor citite cu spații
//
    char *PCurent;

    do
    {
        if(PCurent = strchr(ALine , ','))
            memcpy(PCurent , " " , 1);
    }
    while(PCurent);
}

MaterialsType *EsteMaterial(int Mat)
###9
// Funcție care verifică dacă este definit un material
//
{
    MaterialsType *PMat = FMaterials;

    while(PMat)
    {
        if(PMat->MatID == Mat)
            return PMat;
        PMat = PMat->Urm;
    }
    return NULL;
}

int InputFromASCII (void)
###10
// Funcție care citește datele de intrare din fișier și alocă memoria pentru structurile
// necesare memorării interne a datelor
//
{
    char NumeFis[20];
    char *PTemp, *PCap , *PEnd, ALine[LgASCIILine + 1], *PSir , *PC1, *PTmp;
    char *Simplu = "%d " , *Dublu = "%d " , *PPrec, Conv[10];
    char *NF1= "%lf %lf %lf" , NodesFormat[18], ElementsFormat[15] , Ident[16], NCale[60];

    int FisASCII, Curent = 0, LiniaCurentaDinASCII = 0;
    int i, NodesPerElem , Mat;
    int FEMDim = sizeof(FEMType);
    ads_real Valoare;

    unsigned int Sepa;

```

```

long LgLine, FisLen;

NodesType *PTempNodes, Nod;
ads_point Pct;
ElementsType Ele, *PTempElements;
BConditionsType *PBCTemp, *PBCond1, BCond;
NodesListType *PCurent, *PT1;
MaterialsType *PTempMat, *PM1;
LoadType *PLoad1, *PLoadTemp;

if (FEMDim > 2)
    PPREC = Dublu;
else
    PPREC = Simplu;

strcpy(NodesFormat, PPREC);
strcat(NodesFormat, NF1);

strcpy(ElementsFormat1, PPREC);
strcat(ElementsFormat1, PPREC);
strcat(ElementsFormat1, PPREC);
ads_getstring(0, "\nDati nume fisier .adb: ", NumeFis);
strcpy(NCale, Cale);
strcat(NCale, strcat(NumeFis, ".adb"));

// *** Citeste din fisierul de intrare ***
#if defined(__BORLANDC__) || defined(_MSC_VER)
    if((FisASCII = open(NCale, O_RDONLY | O_BINARY)) == -1)
        #else
            if((FisASCII = open(NCale, O_RDONLY | O_BINARY, S_IRUSR)) == -1)

#endif
    {
        ads_alert(Mes[0][Language]);
        // ads_alert(itoa(errno, NumeFis, 10));
        return EROARE;
    }
FisLen = filelength(FisASCII);
PCap = (char *) Aloca(1, FisLen);
read(FisASCII, PCap, FisLen);
close(FisASCII);

PTemp = PCap;
PEnd = PCap + FisLen;
do
    {
        LgLine = GetLine(ALine, PTemp, LgASCIILine, &LiniaCurentaDinASCII, &PTemp);
        CutComma(ALine);
        LiniaCurentaDinASCII++;
        if(ALine[0] == Marcator[0])
            {
                if(!(strcmp(&ALine[1], Elements)))
                    {
                        IsDefinedSection != IsElements;
                        Curent = IsElements;
                    }
                else
                    {
                        if(!(strcmp(&ALine[1], Nodes)))
                            {
                                IsDefinedSection != IsNodes;
                                Curent = IsNodes;
                            }
                        else
                            {
                                if(!(strcmp(&ALine[1], Materials)))
                                    {
                                        IsDefinedSection != IsMaterials;
                                        Curent = IsMaterials;
                                    }
                                else
                                    {
                                        if(!(strcmp(&ALine[1], BC)))
                                            {
                                                IsDefinedSection != IsBC;
                                                Curent = IsBC;
                                            }
                                        else
                                            {
                                                if(!(strcmp(&ALine[1], BCIni)))
                                                    {
                                                        IsDefinedSection != IsBCIni;
                                                        Curent = IsBCIni;
                                                    }
                                                else
                                                    {

```

```

        {
            if(!(strcmp(&ALine[1] , Loads)))
            {
                IsDefinedSection |= IsLoads;
                Curent = IsLoads;
            }
            else
            {
                if(!(strcmp(&ALine[1] , Title)))
                {
                    IsDefinedSection |= IsTitle;
                    Curent = IsTitle;
                }
                else
                {
                    if(!(strcmp(&ALine[1] , Constants)))
                    {
                        IsDefinedSection |= IsConstants;
                        Curent = IsConstants;
                    }
                    else
                    {
                        if(!(strcmp(&ALine[1] , Control)))
                        {
                            IsDefinedSection |= IsControl;
                            Curent = IsControl;
                        }
                        else
                        {
                            if(!(strcmp(&ALine[1] , EndOfASCII)))
                                PTemp = PEnd;
                        }
                    }
                }
            }
        }
    }
}
else
{
    switch (Curent)
    {
        case IsTitle      : strcpy(JobTitle , ALine);
                          break;
        case IsNodes      :
            if (!PNodes)
            {
                NrNodes = Cite(PTemp , PEnd);
                PNodes = Aloca(NrNodes , sizeof(NodesType));
                PTempNodes = PNodes;
            }
            sscanf(ALine , NodesFormat , &Nod.Nr , &Nod.x , &Nod.y ,
                &Nod.z);
            //Nod.Sys = NULL;
            memcpy(PTempNodes , &Nod , sizeof(NodesType));
            PTempNodes++;
            VerificaHeap(7);
            break;
        case IsElements   :
            VerificaHeap(6);
            if (!PElements)
            {
                NrElements = (Cite(PTemp , PEnd)) >> 1;
                PElements = (ElementsType *)Aloca(NrElements , sizeof(ElementsType));
                PTempElements = PElements;
            }
            sscanf(ALine , ElementsFormat1 , &Ele.Nr , &Sepa , &Ele.Material);
            Ele.Shape = (char) (Sepa & LowerMask);
            Ele.Function = (char) ((Sepa & UpperMask) >> 8);
            PTemp += LgLine + LgCR;
            LgLine = GetLine(ALine , PTemp , LgASCIILine ,
                &LiniaCurentaDinASCII , &PTemp);
            CutComma(ALine);
            LiniaCurentaDinASCII++;
            switch (Ele.Shape)
            {
                case Beam   :
                case Truss  :
                    NodesPerElem = 2 + Ele.Function - 1;
                    break;
                case Triangular :
                    NodesPerElem = 3 * Ele.Function;
                    break;
                case Qadrilater :
            }
    }
}

```

```

        NodesPerElem = 4 * Ele.Function;
        break;
    case Prismatic :
        NodesPerElem = 6 * Ele.Function;
        break;
    case Hexaedric :
        NodesPerElem = 8 * Ele.Function;
        break;
    case Tetraedric :
        NodesPerElem = 4 + (Ele.Function - 1) * 6;
        break;
    default :
        ads_alert(strcat(Mes[2][Language], (char *)itoa(Ele.Nr,Conv, 10)));
        NodesPerElem = 3;
    }

    Ele.NodList = (NodesListType *)Aloca(1, sizeof(NodesListType));
    PSir = ALine;
    PCurent = Ele.NodList;
    PCurent->Urm = NULL;
    sscanf(PSir, PPrec, PCurent);
    while (*PSir == '\040')
    {
        PSir++;
    }
    for(i = 0 ; i < (NodesPerElem - 1) ; i++)
    {
        if(!(PSir = strchr(PSir, 32)))
        {
            ads_alert(strcat(Mes[3][Language], (char *)itoa(Ele.Nr,Conv,10)));
            break;
        }
        while (*PSir == '\040')
        {
            PSir++;
        }
        PSir = strchr(PSir, 32);
        PT1 = (NodesListType *)Aloca(1, sizeof(NodesListType));
        PT1->Urm = NULL;
        PCurent->Urm = PT1;
        sscanf(PSir, PPrec, PT1);
        PCurent = PT1;
    }
    memcpy(PTempElements, &Ele, sizeof(ElementsType));
    PTempElements++;
    VerificaHeap(11);
    break;

case IsMaterials

//
    PC1 = ALine;
    sscanf(PC1, "%15[^]s %d", &Ident, &Mat);
    sscanf(PC1, "%5s %d", &Ident, &Mat);
    if(!(PTempMat = EsteMaterial(Mat)))
    {
        PTempMat = (MaterialsType *)Aloca(1, sizeof(MaterialsType));
        PTempMat->MatID = Mat;
        if(!FMaterials)
            FMaterials = PTempMat;
        else
            FMaterials->Urm = PTempMat;
        PTempMat->Urm = NULL;
        PM1 = PTempMat;
    }
    PSir = strchr(PC1, 32);
    PSir++;
    PSir = strchr(PSir, 32);
    PSir++;
    while(PSir < (ALine + LgLine))
    {
        PC1 = strchr(PSir, 32);
        PC1++;
        sscanf(PSir, "%15[^]s", Ident);
        sscanf(PC1, "%lf", &Valoare);
#ifdef _MSC_VER
        if(!_fstricmp(SirMatName, Ident))
#else
        if(!strcmpi(SirMatName, Ident))
#endif
        {
            strcpy(PTempMat->Name, PC1);
            PSir = ALine + LgLine + 1;
        }
        else

```



```

{
    if(!strcmpi(SirPoissonX , Ident))
        PTempMat->PoissonX = Valoare;
    else
    if(!strcmpi(SirPoissonY , Ident))
        PTempMat->PoissonY = Valoare;
    else
    if(!strcmpi(SirPoissonZ , Ident))
        PTempMat->PoissonZ = Valoare;
    else

    if(!strcmpi(SirYoungX , Ident))
        PTempMat->YoungX = Valoare;
    else
    if(!strcmpi(SirYoungY , Ident))
        PTempMat->YoungY = Valoare;
    else
    if(!strcmpi(SirYoungZ , Ident))
        PTempMat->YoungZ = Valoare;
    else

    if(!strcmpi(SirAlphaX , Ident))
        PTempMat->AlphaX = Valoare;
    else
    if(!strcmpi(SirAlphaY , Ident))
        PTempMat->AlphaY = Valoare;
    else
    if(!strcmpi(SirAlphaZ , Ident))
        PTempMat->AlphaZ = Valoare;
    else

    if(!strcmpi(SirTransversalX , Ident))
        PTempMat->TransversalX = Valoare;
    else
    if(!strcmpi(SirTransversalY , Ident))
        PTempMat->TransversalY = Valoare;
    else
    if(!strcmpi(SirTransversalZ , Ident))
        PTempMat->TransversalZ = Valoare;
    else

    if(!strcmpi(SirDensity , Ident))
        PTempMat->Density = Valoare;
    else
    if(!strcmpi(SirRefTemp , Ident))
        PTempMat->RefTemp = Valoare;
    else
    if(!strcmpi(SirSpecificHeat , Ident))
        PTempMat->SpecificHeat = Valoare;
    else
    if(!strcmpi(SirKx , Ident))
        PTempMat->Kx = Valoare;
    else
    if(!strcmpi(SirKy , Ident))
        PTempMat->Ky = Valoare;
    else
    if(!strcmpi(SirKz , Ident))
        PTempMat->Kz = Valoare;
    else
    if(!strcmpi(SirEmissivity , Ident))
    {
        PTmp = PC1;
        i = 0;
        while(((isdigit(*PTmp)) || (*PTmp == '-')) && PTmp)
        {
            PTempMat->Emissivity[i] = Valoare;
            PTmp = strchr(PTmp , 32);
            PTmp++;      i ++;
            sscanf(PTmp , "%lg" , &Valoare);
        }
    }
    if(!(PSir = strchr(PC1 , 32)))
        PSir = ALine + LgLine;
    PSir++;
}
}

break;

case IsBC :

    if(!PBConditions)
    {
        PBCond1 = PBCTemp = PBConditions = Aloca(1, sizeof(BConditionsType));

```

```

    }
    else
    {
        PBCond1->Urm = PBCTemp = Aloca(1, sizeof(BConditionsType));
        PBCTemp->Urm = NULL;
        PBCond1 = PBCTemp;
    }

    PC1 = ALine;
    sscanf(PC1, "%15[^\n]s", &Ident);
    if(!strcmp(Ident, SirTemperatura))
    {
        PC1 = strchr(PC1, 32);
        PC1++;
        sscanf(PC1, "%d %lf", &PBCTemp->NodSauElement,
            &PBCTemp->Impus);
        PBCTemp->TipRestrictie = Temperatura;
    }
    if(!strcmp(Ident, SirRadiatie))
    {
        PC1 = strchr(PC1, 32);
        PC1++;
        sscanf(PC1, "%d %d %d %lf", &PBCTemp->NodSauElement,
            &PBCTemp->Nod1, &PBCTemp->Nod2, &PBCTemp->Impus);
        PBCTemp->TipRestrictie = Radiatie;
    }
    if(!strcmp(Ident, SirConvectie))
    {
        PC1 = strchr(PC1, 32);
        PC1++;
        sscanf(PC1, "%d %d %d %lf", &PBCTemp->NodSauElement,
            &PBCTemp->Nod1, &PBCTemp->Nod2, &PBCTemp->Impus);
        PBCTemp->TipRestrictie = Convectie;
    }
    if(!strcmp(Ident, SirDFlux))
    {
        PC1 = strchr(PC1, 32);
        PC1++;
        sscanf(PC1, "%d %d %d %lf", &PBCTemp->NodSauElement,
            &PBCTemp->Nod1, &PBCTemp->Nod2, &PBCTemp->Impus);

        PBCTemp->TipRestrictie = FluxDistribuit;
    }
    break;

case IsBCIni :
    if(!PBConditionsIni)
    {
        PBCond1 = PBCTemp = PBConditionsIni = Aloca(1, sizeof(BConditionsType));
    }
    else
    {
        PBCond1->Urm = PBCTemp = Aloca(1, sizeof(BConditionsType));
        PBCTemp->Urm = NULL;
        PBCond1 = PBCTemp;
    }

    PC1 = ALine;
    sscanf(PC1, "%15[^\n]s", &Ident);
    if(!strcmp(Ident, SirTemperatura))
    {
        PC1 = strchr(PC1, 32);
        PC1++;
        sscanf(PC1, "%d %lf", &PBCTemp->NodSauElement,
            &PBCTemp->Impus);
        PBCTemp->TipRestrictie = Temperatura;
    }
    break;

case IsLoads :
    PC1 = ALine;
    if(!PLoad)
    {
        PLoad1 = PLoadTemp = PLoad = Aloca(1, sizeof(LoadType));
    }
    else
    {
        PLoad1->Urm = PLoadTemp = Aloca(1, sizeof(LoadType));
        PLoadTemp->Urm = NULL;
        PLoad1 = PLoadTemp;
    }
    sscanf(PC1, "%15[^\n]s", &Ident);
    if(!strcmp(Ident, SirElemSource))
    {

```

```

        PC1 = strchr(PC1 , 32);
        sscanf(PC1 , "%d %lf" ,&PLoadTemp->NodSauElement ,
                &PLoadTemp->Impus);
        PLoadTemp->TipRestrictie = ElemSource;
    }
    else
        if(!strcmp(Ident , SirNodSource))
        {
            PC1 = strchr(PC1 , 32);
            sscanf(PC1 , "%d %lf" ,&PLoadTemp->NodSauElement ,
                    &PLoadTemp->Impus);
            PLoadTemp->TipRestrictie = NodSource;
        }

        break;

case IsConstants:
    PC1 = ALine;
    sscanf(PC1 , "%15[^ ]s" ,&Ident);
    if(!strcmp(Ident , SirTempMediu))
    {
        PC1 = strchr(PC1 , 32);
        PC1++;
        sscanf(PC1 , "%lf" , &TemperaturaMediului);
    }
    if(!strcmp(Ident , SirBoltzmann))
    {
        PC1 = strchr(PC1 , 32);
        PC1++;
        sscanf(PC1 , "%lf" , &Boltzmann);
    }

        break;

case IsControl:
    PC1 = ALine;
    sscanf(PC1 , "%15[^ ]s" ,&Ident);
    if(!strcmp(Ident , SirGauss))
    {
        PC1 = strchr(PC1 , 32);
        PC1++;
        sscanf(PC1 , "%15[^ ]s" ,&Ident);
        if(!strcmp(Ident , SirElaborat))
            GaussTip = GAUSS_ELABORAT;
        else
            if(!strcmp(Ident , SirSimplu))
                GaussTip = GAUSS_SIMPLU;
            else
                if(!strcmp(Ident , SirInvers))
                    GaussTip = GAUSS_INVERS;
    }
    if(!strcmp(Ident , SirModBC))
    {
        PC1 = strchr(PC1 , 32);
        PC1++;
        sscanf(PC1 , "%15[^ ]s" ,&Ident);
        if(!strcmp(Ident , SirGrand))
            ModBC = BC_GRAND;
        else
            if(!strcmp(Ident , SirElimina))
                ModBC = BC_ELIMINA;
    }
    if(!strcmp(Ident , SirPrecIter))
    {
        PC1 = strchr(PC1 , 32);
        PC1++;
        sscanf(PC1 , "%lf" , &PrecIter);
    }
    if(!strcmp(Ident , SirMaxIter))
    {
        PC1 = strchr(PC1 , 32);
        PC1++;
        sscanf(PC1 , "%d" , &MaxIter);
    }
    if(!strcmp(Ident , SirPrecVer))
    {
        PC1 = strchr(PC1 , 32);
        PC1++;
        sscanf(PC1 , "%lf" , &PrecVer);
    }
    if(!strcmp(Ident , SirScGeo))
    {
        PC1 = strchr(PC1 , 32);
        PC1++;
        sscanf(PC1 , "%lf" , &ScaraGeometrie);
    }

```

```

    )

    break;

    case NoInit      :
                    break;
    default          ;;
    }
}
PTemp += LgLine + LgCR;
}
while (PTemp < PEnd);
free(PCap);
VerificaHeap(10);
return OK;
}

void CopyNodToPct(ads_point dest , NodesType *PN)
###11
// Funcție care copiază coordonatele unui nod într-un punct ADS
//
{
    char *sursa = (char *)PN + sizeof(FEMType);
    memcpy(dest , sursa , sizeof(ads_point));
}

NodesType *CautaNod(FEMType Nr)
###12
// Funcție care returnează adresa unde este memorat un nod
//
{
    int i;
    NodesType *PN = PNodes;

    for(i=1 ; i <= NrNodes ; i++)
    {
        if(Nr == PN->Nr)
            return PN;
        PN++;
    }
    return NULL;
}

int Model(void)
###13
// Funcție care desenează modelul geometric
//
{
    int i , j;
    ads_real x, y;
    ads_point p[4];
    NodesType *PN = PNodes;
    NodesListType *PNL;
    ElementsType *PE = PElements;
    struct resbuf *prb, sText, sDes, sNod;
    char Tmp[10];

    sNod.restype = RTSTR;
    sNod.resval.rstring = "NOD";
    ads_setvar("CLAYER", &sNod);
    for(i = 1 ; i <= NrNodes ; i++)
    {
        p[0][X] = PN->x;
        p[0][Y] = PN->y;
        p[0][Z] = PN->z;
        ads_command(RTSTR , "POINT" , RT3DPOINT , p[0] , RTNONE);
        // itoa(i, Tmp, 10);
        itoa(PN->Nr, Tmp, 10);
        prb = ads_buildlist(RTSTR , "TEXT" , RT3DPOINT , p[0] ,
            RTSTR , "" , RTSTR , "" , RTSTR , Tmp, RTNONE);
        PN++;
        ads_cmd(prb);
        ads_relrb(prb);
    }
    sText.restype = sDes.restype = RTSTR;
    sText.resval.rstring = "ELEM";
    sDes.resval.rstring = "MODEL";
    ads_setvar("CLAYER", &sDes);

    for(i = 1 ; i <= NrElements ; i++)
    {
        switch (PE->Shape)

```

```

    {
    case Triangular :
    PNL = PE->NodList;
    x = y = 0;
    for (j = 0 ; j < 3 ; j++)
    {
        PN = CautaNod(PNL->Nod);
        CopyNodToPct(p[j] , PN);
        x += p[j][X] ;      y += p[j][Y];
        PNL = PNL->Urm;
    }
    x /= 3;      y /= 3;
    prb = ads_buildlist(RTSTR , "3DFACE" , RT3DPOINT , p[0] ,
        RT3DPOINT , p[1] , RT3DPOINT , p[2] ,
        RTSTR , "" , RTSTR , "" , RTNONE);
    break;

    case Qadrilater:
    PNL = PE->NodList;
    for (j = 0 ; j < 4 ; j++)
    {
        PN = CautaNod(PNL->Nod);
        CopyNodToPct(p[j] , PN);
        PNL = PNL->Urm;
    }
    prb = ads_buildlist(RTSTR , "3DFACE" , RT3DPOINT , p[0] ,
        RT3DPOINT , p[1] , RT3DPOINT , p[2] , RT3DPOINT , p[3] ,
        RTSTR , "" , RTNONE);
    break;
    }
    ads_cmd(prb);
    ads_relrb(prb);

    ads_setvar("CLAYER", &sText);
    // itoa(i, Tmp, 10);
    itoa(PE->Nr, Tmp, 10);
    p[0][X] = x ;      p[0][Y] = y;
    prb = ads_buildlist(RTSTR , "TEXT" , RT3DPOINT , p[0] ,
        RTSTR , "" , RTSTR , "" , RTSTR , Tmp, RTNONE);
    ads_cmd(prb);
    ads_relrb(prb);
    ads_setvar("CLAYER", &sDes);

    PE++;
    }
    return OK;
    }

    void ScrieMat(FEMType Lin , FEMType Col , ads_real Val)
    ///#14
    // Funcție care scrie în matricea globală alocată dinamic
    //
    {
        ads_real *PTmp = PGlobalMatrix;

        PTmp += NrNodes * Lin;
        PTmp += Col;
        *PTmp = Val;
    }

    ads_real CitMat(FEMType Lin , FEMType Col)
    ///#15
    // Funcție care citește din matricea globală alocată dinamic
    //
    {
        ads_real *PTmp = PGlobalMatrix;

        PTmp += NrNodes * Lin;
        PTmp += Col;
        return *PTmp;
    }

    void ScrieInt(FEMType Lin , FEMType Col , FEMType Val, FEMType (*Pointer)
    [1][1], FEMType Lg)
    ///#16
    // Funcție care scrie un întreg în matrice alocată dinamic
    //
    {
        FEMType *PTmp = Pointer;

        PTmp += Lg * Lin;
        PTmp += Col;
        *PTmp = Val;
    }

```

```

FEMType CitInt(FEMType Lin , FEMType Col, FEMType (*Pointer)[1][1], FEMType Lg)
###17
// Funcție care citește un întreg din matricea alocată dinamic
//
{
    FEMType *PTmp = Pointer;

    PTmp += Lg * Lin;
    PTmp += Col;
    return *PTmp;
}

void ScrieSolutiile(FEMType NrEqua, int nr, int Cum)
###18
// Funcție care scrie soluțiile problemei
//
{
    FEMType i , j, F;
    FILE *Fis;
    NodesType *PNTmp = PNodes;
    ElementsType *PETmp = PElements;
    NodesListType *PN;
    char Line[1024] , Temp[80];
    char Nume[14] = "Rez" , *Ext = ".dat", NCale[60];

    itoa(nr, &Nume[3] , 10);
    strcat(Nume , Ext);
    strcpy(NCale, Cale);
    strcat(NCale, Nume);
    Fis = fopen(NCale , "w");
    if( !Fis)
        ads_alert(NCale);

    fputs("---- TITLE ---- \n", Fis);
    fprintf(Fis, "%s\n", JobTitle);
    fputs("---- NODES ----\n", Fis);

    for(i = 0 ; i < NrNodes ; i++)
    {
        fprintf(Fis , "%7d %14f %14f %14f\n" , PNTmp->Nr, PNTmp->x,
            PNTmp->y, PNTmp->z);
        PNTmp++;
    }

    fputs("---- ELEMENTS ----\n", Fis);

    for(i = 0 ; i < NrElements ; i++)
    {
        F = (PETmp->Function << 8) | PETmp->Shape;
        fprintf(Fis , "%5d %5d %5d\n" , PETmp->Nr, F, PETmp->Material);
        PN = PETmp->NodList;
        for(j=0;j < 3 ; j++)
        {
            fprintf(Fis , "%7d" , PN->Nod);
            PN = PN->Urm;
        }
        fprintf(Fis , "\n");
        PETmp++;
    }
    fputs("---- SOLUTII ----\n", Fis);
    if(Cum == Direct)
        for(i = 0 ; i < NrEqua ; i++)
            fprintf(Fis , "%7d %25.16f\n" , i + 1 , *PSolutionVect[i]);
    else
    {
        PL = PLoad;
        for(i = 0 ; i < NrEqua ; i++)
        {
            fprintf(Fis , "%7d %25.16f\n" , PL->NodSauElement , *PSolutionVect[i]);
            PL = PL->Urm;
        }
    }
    fclose(Fis);
    ads_printf("\n%f\n", *PSolutionVect[12]);
}

void CuInvers(FEMType NrEqua, int Cum)
{
    int i;
    return;
}

```

```

void GaussElab(FEMType NrEqua, int Cum)
/**#19
// Funcție care rezolvă sistemul prin tehnica pivotării totale
//
{
  int i, j, k;
  ads_real Pivot, Factor;
  char Tmp[20] = "Pivot zero ";
  FEMType BestPivot;
  ads_real MaxLin, Curent, MaxRap;

  //Triangularizarea matricii globale
  for(i = 0 ; i < (NrEqua - 1) ; i++)
  {
    MaxRap = MIN_VAL;
    for(j = i ; j < NrEqua ; j++)
    {
      MaxLin = MIN_VAL;
      for(k = 0 ; k < NrEqua; k++ )
      {
        Curent = fabs(CitMat(j, k));
        if(Curent > MaxLin)
          MaxLin = Curent;
      }
      Curent = fabs(CitMat(j, i)) / MaxLin;
      if(Curent > MaxRap)
      {
        MaxRap = Curent;
        BestPivot = j;
      }
    }
    if(BestPivot != i)
    {
      for(j = 0 ; j < NrEqua ; j++)
      {
        Factor = CitMat(i, j);
        ScrieMat(i, j, CitMat(BestPivot, j));
        ScrieMat(BestPivot, j, Factor);
      }
      Factor = *PLoadsVect[i];
      *PLoadsVect[i] = *PLoadsVect[BestPivot];
      *PLoadsVect[BestPivot] = Factor;
    }
    Pivot = CitMat(i , i);
    j = i + 1;
    if(!Pivot)
    {
      itoa(i , &Tmp[11] , 10);
      ads_alert(Tmp);
      ScrieSolutiile(NrEqua,i+1, Cum);
      return;
      break;
    }
    else
      for(j = i + 1 ; j < NrEqua ; j++)
      {
        Factor = CitMat(j , i) / Pivot;
        if(Factor)
        {
          *PLoadsVect[j] = *PLoadsVect[j] - Factor * (*PLoadsVect[i]);
          for(k = i; k < NrEqua ; k++)
          {
            ScrieMat(j , k , CitMat(j , k) - Factor * CitMat(i , k));
            VerificaHeap(4);
          }
        }
      }
  }

  //Rezolvarea sistemului
  *PSolutionVect[NrEqua - 1] = *PLoadsVect[NrEqua - 1] /
    CitMat(NrEqua - 1,NrEqua - 1);
  for(i = NrEqua - 2 ; i >= 0 ; i--)
  {
    Factor = 0.0L;
    for(k = i + 1; k < NrEqua ; k++)
      Factor = Factor + CitMat(i, k) * (*PSolutionVect[k]);
    *PSolutionVect[i] = (*PLoadsVect[i] - Factor) / CitMat(i , i);
  }
}

```



```

void Gauss(FEMType NrEqua, int Cum)
///##20
// Funcție care rezolvă sistemul prin triangularizare normală
//
{
  int i, j, k;
  ads_real Pivot, Factor;
  char Tmp[20] = "Pivot zero ";

  //Triangularizarea matricei globale
  for(i = 0 ; i < (NrEqua - 1) ; i++)
  {
    Pivot = CitMat(i , i);
    j = i + 1;
    if(!Pivot)
    {
      itoa(i , &Tmp[11] , 10);
      ads_alert(Tmp);
      ScrieSolutiile(NrEqua,i+1, Cum);
      return;
      break;
    }
    else
    for(j = i + 1 ; j < NrEqua ; j++)
    {
      Factor = CitMat(j , i) / Pivot;
      if(Factor)
      {
        *PLoadsVect[j] = *PLoadsVect[j] - Factor * (*PLoadsVect[i]);
        for(k = i; k < NrEqua ; k++)
        {
          ScrieMat(j , k , CitMat(j , k) - Factor * CitMat(i , k));
          VerificaHeap(4);
        }
      }
    }
  }

  //Rezolvarea sistemului
  *PSolutionVect[NrEqua - 1] = *PLoadsVect[NrEqua - 1] /
                               CitMat(NrEqua - 1,NrEqua - 1);
  for(i = NrEqua - 2 ; i >= 0 ; i--)
  {
    Factor = 0.0L;
    for(k = i + 1; k < NrEqua ; k++)
      Factor = Factor + CitMat(i, k) * (*PSolutionVect[k]);
    *PSolutionVect[i] = (*PLoadsVect[i] - Factor) / CitMat(i , i);
  }
}

BConditionsType *AreBC(FEMType Nr, FEMType TipCond)
///##21
// Funcție care verifică dacă un element are impuse un anume tip de condiții limită
//
{
  BConditionsType *PBC = PBConditions;
  while(PBC)
  {
    if((PBC->NodSauElement == Nr) && (PBC->TipRestrictie == TipCond))
      return PBC;
    PBC = PBC->Urm;
  }
  return NULL;
}

int AreRadiatie(void)
///##22
// Funcție care verifică dacă sunt impuse condiții de contur de tip radiație, pentru
// a rezolva sistemul din mai multe iterații
//
{
  BConditionsType *PBC = PBConditions;

  while(PBC)
  {
    if(PBC->TipRestrictie == Radiatie)
      return 1;
    PBC = PBC->Urm;
  }
  return 0;
}

```

```

ads_real ImpuneBC(ElementsType *PElem, int Cond, int Cum)
//##23
// Funcție care impune condițiile de contur direct în matricea globală
//
{
  BConditionsType *PBC;
  ads_real Lg, Coef, Coef1, Alfa, TempMedieElement, Emisivitatea, TM4, TElem4;

  NodesType *P1, *P2;
  int N1, N2;

  if(PBC = AreBC(PElem->Nr, Cond))
  {
    N1 = PBC->Nod1 - 1;
    N2 = PBC->Nod2 - 1;
    P1 = PNodes + N1;
    P2 = PNodes + N2;
    Lg = sqrt(pow(P1->x - P2->x, 2) + pow(P1->y - P2->y, 2));
    switch (Cond)
    {
      case Convectie:
        Coef = PBC->Impus * Lg;
        Coef1 = Coef * TemperaturaMediului / 2;
        if(Cum == Direct)
        {
          ScrieMat(N1, N1, CitMat(N1, N1) + Coef / 3);
          ScrieMat(N2, N2, CitMat(N2, N2) + Coef / 3);
          ScrieMat(N1, N2, CitMat(N1, N2) + Coef / 6);
          ScrieMat(N2, N1, CitMat(N2, N1) + Coef / 6);
          *PLoadsVect[N1] += Coef1;
          *PLoadsVect[N2] += Coef1;
        }
        else
        {
          *PSolutionVect[N1] -= Coef1;
          *PSolutionVect[N2] -= Coef1;
        }
        return 0;
        break;

      case Radiatie:
        if(Cum == Direct)
          TempMedieElement = (*PSolutionVect[N1] + *PSolutionVect[N2]) / 2;
        else
          TempMedieElement = (*PTmpVect[N1] + *PTmpVect[N2]) / 2;
        Emisivitatea = RetEmissiv(PElem->Material, TempMedieElement);
        TElem4 = pow(TempMedieElement + 273, 4);
        TM4 = pow(TemperaturaMediului + 273, 4);
        Alfa = Emisivitatea * Boltzmann * (TElem4 - TM4) / (TempMedieElement - TemperaturaMediu-
lui);
        Coef = Alfa * Lg;
        Coef1 = Coef * TemperaturaMediului / 2;
        if(Cum == Direct)
        {
          ScrieMat(N1, N1, CitMat(N1, N1) + Coef / 3);
          ScrieMat(N2, N2, CitMat(N2, N2) + Coef / 3);
          ScrieMat(N1, N2, CitMat(N1, N2) + Coef / 6);
          ScrieMat(N2, N1, CitMat(N2, N1) + Coef / 6);
          *PLoadsVect[N1] += Coef1;
          *PLoadsVect[N2] += Coef1;
        }
        else
        {
          *PSolutionVect[N1] -= Coef1;
          *PSolutionVect[N2] -= Coef1;
        }
        return Alfa;
        break;

      case FluxDistribuit:
        Coef = PBC->Impus * Lg / 2;
        if(Cum == Direct)
        {
          *PLoadsVect[N1] += Coef;
          *PLoadsVect[N2] += Coef;
        }
        else
        {
          *PSolutionVect[N1] -= Coef;
          *PSolutionVect[N2] -= Coef;
        }
        return 0;
        break;
    }
  }
}

```

```

}
else
    return 0;
}

```

```

void FaceMatriceaElementara(ElementsType *PElem, NodesType *Nod, ads_real *MatElem)

```

```

///#24
// Funcție care completează matricea elementară de conductivitate
//
{
    ads_real X32, X13, X21, Y23, Y31, Y12, Jacobian, Coef1, Coef2;
    ads_real Kx, Ky, *Mat = MatElem;
    NodesType N1, N2, N3;

    N1 = *Nod;
    N2 = *(Nod + 1);
    N3 = *(Nod + 2);
    X32 = N3.x - N2.x;
    X13 = N1.x - N3.x;
    X21 = N2.x - N1.x;
    Y23 = N2.y - N3.y;
    Y31 = N3.y - N1.y;
    Y12 = N1.y - N2.y;
    PElem->Jacobi = Jacobian = X21 * Y31 - X13 * Y12;
    Kx = (EsteMaterial((int)PElem->Material))->Kx;
    Ky = (EsteMaterial((int)PElem->Material))->Ky;
    Ky = Ky ? Ky : Kx;
    Coef1 = Kx / Jacobian / 2.0L;
    Coef2 = Ky / Jacobian / 2.0L;
    *Mat++ = Y23 * Y23 * Coef1 + X32 * X32 * Coef2;//0,0
    *Mat++ = Y23 * Y31 * Coef1 + X32 * X13 * Coef2;//0,1
    *Mat++ = Y12 * Y23 * Coef1 + X21 * X32 * Coef2;//0,2
    *Mat++ = Y23 * Y31 * Coef1 + X32 * X13 * Coef2;//1,0
    *Mat++ = Y31 * Y31 * Coef1 + X13 * X13 * Coef2;//1,1
    *Mat++ = Y31 * Y12 * Coef1 + X13 * X21 * Coef2;//1,2
    *Mat++ = Y12 * Y23 * Coef1 + X21 * X32 * Coef2;//2,0
    *Mat++ = Y31 * Y12 * Coef1 + X13 * X21 * Coef2;//2,1
    *Mat++ = Y12 * Y12 * Coef1 + X21 * X21 * Coef2;//2,2
}

```

```

int Solve(void)

```

```

///#25
// Funcție care lansează procedura de rezolvare a problemei directe
//
{
    char Tmp[20];
    FEMType NrEqua, NrIter = 0, Converge = 0;
    ElementsType *PElem = PElements, *PE;
    LoadType *PLoadTemp = PLoad;
    NodesType *PN;
    BConditionsType *PBC = NULL;
    ads_real Pivot, Factor, Coef;
    ads_real (*CRVect1)[1], (*CRVect2)[1], MatElem[GDL][GDL];
    FEMType i, j, k, Idx1, Idx2, EsteRad;
    NodesType Nod[GDL];
    NodesListType *PNd, *PNd1;

    NrEqua = NrNodes;
    EsteRad = AreRadiatie();
    CRVect1 = Aloca(NrElements, sizeof(ads_real));
    CRVect2 = Aloca(NrElements, sizeof(ads_real));
    VerificaHeap(8);
    PLoadsVect = Aloca(NrEqua, sizeof(ads_real));
    PSolutionVect = Aloca(NrEqua, sizeof(ads_real));
    PGlobalMatrix = Aloca(NrEqua * NrEqua, sizeof(ads_real));
    VerificaHeap(9);

    if(ScaraGeometrie != 1)
        Scaleaza(ScaraGeometrie);
    while((NrIter++ < MaxIter) && !Converge)
    {
        PElem = PElements;
        for(i = 0; i < NrElements; i++)
        {
            PNd = PElem->NodList;
            for(j = 0; j <= PElem->Shape; j++)
            {
                VerificaHeap(12);
                memcpy((void *)&Nod[j], CautaNod(PNd->Nod), sizeof(NodesType));
                PNd = PNd->Urm;
            }
        }
    }
}

```

```

    }
    FaceMatriceaElementara(PElem, &Nod, &MatElem);
    ImpuneBC(PElem, Convectie, Direct);
    if(NrIter > 1)
        *CRVect2[PElem->Nr - 1] = ImpuneBC(PElem, Radiatie, Direct);
    ImpuneBC(PElem, FluxDistribuit, Direct);
// Calculeaza termenul care se introduce in vectorul incarcrilor
    Coef = 0;
    PLoadTemp = PLoad;
    while (PLoadTemp)
    {
        if((PLoadTemp->TipRestrictie == ElemSource) && (PLoadTemp->NodSauElement == (i+1)))
        {
            Coef = PLoadTemp->Impus * PElem->Jacobi / 6.0L;
            break;
        }
        PLoadTemp = PLoadTemp->Urm;
    }

    VerificaHeap(1);
// Asamblarea matricei globale
    PNd = PElem->NodList;
    for(j = 0 ; j < GdlElem ; j++)
    {
        Idx1 = PNd->Nod - 1;
        *PLoadsVect[Idx1] += Coef;
        PNd1 = PElem->NodList;
        for(k = 0 ; k < GdlElem ; k++)
        {
            Idx2 = PNd1->Nod - 1;
            ScrieMat(Idx1 , Idx2 , CitMat(Idx1 , Idx2) + MatElem[j][k]);
            VerificaHeap(2);
            PNd1 = PNd1->Urm;
        }
        PNd = PNd->Urm;
    }
    PElem++;
}
VerificaHeap(3);
//ScrieSolutiile(NrEqua,1, Direct);

PBC = PBCconditions;
while(PBC)
{
    if(PBC->TipRestrictie == Temperatura)
    {
        Idx1 = PBC->NodSauElement - 1;
        if(ModBC == BC_GRAND)
        {
            *PLoadsVect[Idx1] = PBC->Impus * GRAND_VAL;
            ScrieMat(Idx1 , Idx1 , GRAND_VAL + CitMat(Idx1, Idx1));
        }
        else
        {
            for (i = 0 ; i < NrNodes ; i++)
            {
                *PLoadsVect[i] = *PLoadsVect[i] - PBC->Impus * CitMat(i, Idx1);
                ScrieMat(Idx1 , i , 0);
                ScrieMat(i , Idx1 , 0);
            }
            ScrieMat(Idx1 , Idx1 , 1);
            *PLoadsVect[Idx1] = PBC->Impus;
        }
    }
    VerificaHeap(13);
    PBC = PBC->Urm;
}
//ScrieSolutiile(NrEqua,2, Direct);
switch(GaussTip)
{
    case GAUSS_ELABORAT:
        GaussElab(NrEqua, Direct);
        break;
    case GAUSS_SIMPLU:
        Gauss(NrEqua, Direct);
        break;
    case GAUSS_INVERS:
        CuInvers(NrEqua, Direct);
}

if(EsteRad)
{
//Evalueaza convergenta pentru elemente cu conditii de contur radiatie
    if(NrIter > 1)

```

```

{
    PE = PElements;
    Converge = 1;
    for(j = 0 ; j < NrElements ; j++)
    {
        if(AreBC(PE->Nr, Radiatie))
        {
            if(Converge)
                Converge = fabs(*CRVect1[j] - *CRVect2[j]) < PrecIter;
            *CRVect2[j] = (*CRVect1[j] + *CRVect2[j]) / 2;
        }
        PE++;
    }
    _fmemcpy(CRVect1, CRVect2 , NrElements * sizeof(ads_real));
}
}
else
    Converge = 1;
}

if(ScaraGeometrie != 1)
    Scaleaza(1 / ScaraGeometrie);
ScrieSolutiile(NrEqua,99, Direct);
ads_printf("\nNumar iteratii: %d", NrIter);
free(CRVect1);
free(CRVect2);
/* free(PLoadsVect);
free(PSolutionVect);
free(PGlobalMatrix);*/

return OK;
}

int Identic(int Nr1, int Nr2)
###26
// Funcție care verifică identitatea a două coloane din matricele intermediare de
// la problema inversă
//
{
    int (*Col1)[1], (*Col2)[1], i;

    Col1 = Aloca(NrLoads, sizeof(FEMType));
    Col2 = Aloca(NrLoads, sizeof(FEMType));
    for(i = 0 ; i < NrLoads ; i++)
    {
        *Col1[i] = CitInt(i, Nr1, MatLoad, Dim) != 0;
        *Col2[i] = CitInt(i, Nr2, MatLoad, Dim) != 0;
        if(*Col1[i] ^ *Col2[i])
        {
            free(Col1);
            free(Col2);
            return 0;
        }
    }
    free(Col1);
    free(Col2);
    return 1;
}

int Singular(int (*Coloane)[1])
###27
// Funcție care condiții de singularitate pentru determinantul matricei [V]
//
{
    int i, j;
    int (*Marcat)[1];

    Marcat = Aloca(NrLoads, sizeof(FEMType));
    j = 0;
    for(i = 0 ; i < Dim ; i++)
    {
        if(*Coloane[i])
            *Marcat[j++] = i;
    }
    for(i = 0; i < (NrLoads - 1); i++)
        for(j = i + 1 ; j < NrLoads ; j++)
            if(Identic(*Marcat[i], *Marcat[j]))
            {
                free(Marcat);
                return 1;
            }
}

```

```

for(i = 0 ; i < NrLoads ; i++)
{
    for(j = 0 ; j < NrLoads ; j++)
        if(CitInt(i, *Marcat[j], MatLoad, Dim))
            break;
    else
        if(j == (NrLoads - 1))
        {
            free(Marcat);
            return 1;
        }
}
free(Marcat);
return 0;
}

```

```

int Scoate(int Start, int (*Coloane)[1], int Scoase)

```

```

###28

```

```

// Funcție care scoate o coloană din matricea dreptunghiulară pentru a face sistemul
// pentru problema inversă compatibil determinat
//

```

```

{
    int rez = MAI_DEPARTE, i, j;

```

```

do

```

```

{
    if(Start >= Dim)
        return MAI_DEPARTE;
    *Coloane[Start] = 0;
    Scoase++;
    if(Scoase == Dif)

```

```

    {
        Iter++;

```

```

        if(Singular(Coloane))

```

```

        {
            _fstncpy(SirErr, "");
            for(i = 0 ; i < Dim ; i++)

```

```

//         itoa(*Coloane[i], T1, 10);
//         j = *Coloane[i];

```

```

#ifdef __BORLANDC__
        sprintf(T1, "%2d", j);

```

```

#else
        sprintf(T1, "%2ld", j);

```

```

#endif

```

```

        _fstreat(SirErr, T1);
        _fstreat(SirErr, " ");
    }

```

```

    fprintf(Err, "%s\n", SirErr);

```

```

        *Coloane[Start++] = 1;
        Scoase--;

```

```

    }
    else
        return GATA;

```

```

}
else

```

```

    if((rez = Scoate(++Start, Coloane, Scoase)) == MAI_DEPARTE)

```

```

    {
        *Coloane[Start - 1] = 1;
        Scoase--;
    }

```

```

}
while(rez == MAI_DEPARTE);
return GATA;
}

```

```

int Inv(void)

```

```

###29

```

```

// Funcție care inițiază procedura de rezolvare a problemei inverse
//

```

```

{

```

```

    FEMType NrEqua, Nr;

```

```

    ElementsType *PElem = PElements;

```

```

    LoadType *PLoadTemp = PLoad;

```

```

    NodesType *PN, *P1, *P2;

```

```

    BConditionsType *PBC = PBConditions;

```

```

    ads_real MatElem[GDL][GDL], Coef;

```

```

    ads_real Pivot, Factor;

```

```

    FEMType i, j, k, l, Idx1, Idx2, (*PNL)[1], (*PNLTmp)[1], Lg, (*Coloane)[1];

```

```

    FEMType Este, Rez, Scoase, Start, rez = 0;

```

```

    NodesType Nod[GDL], N;

```

```

NodesListType *PNd, *PNd1;
ElementsListType *PEListCap, *PEList, *PEListCurent;
ElementsType (*PIndexElem)[1] = PElements;
ListaBCSaveType *PCapSave = NULL, *PSave, *PSavel;
ads_real (*PVect)[1], LungElem;
char Tmp[32], Line[1024];
FILE *Fis;

NrEqua = NrNodes;
VerificaHeap(8);
PLoadsVect = Aloca(NrEqua, sizeof(ads_real));
PSolutionVect = Aloca(NrEqua, sizeof(ads_real));
PGlobalMatrix = Aloca(NrEqua * NrEqua, sizeof(ads_real));
PTmpVect = Aloca(NrEqua, sizeof(ads_real));

// Încărcarea soluției câmpului termic [T] din secțiunea BC în vectorii PLoads, PSolution
// trebuie și în PSolution pentru impunerea BC Raditie care are nevoie de temp din noduri
do
{
    if(PBC->TipRestrictie == Temperatura)
    {
        Idx1 = PBC->NodSauElement - 1;
        *PSolutionVect[Idx1] = *PTmpVect[Idx1] = *PLoadsVect[Idx1] = PBC->Impus;
    }
    PBC = PBC->Urm;
}
while(PBC);

if(ScaraGeometrie != 1)
    Scaleaza(ScaraGeometrie);
VerificaHeap(9);
PElem = PElements;
for(i = 0 ; i < NrElements ; i++)
{
    PNd = PElem->NodList;
    for(j = 0 ; j <= PElem->Shape ; j++)
    {
        VerificaHeap(12);
        memcpy((void *)&Nod[j], CautaNod(PNd->Nod), sizeof(NodesType));
        PNd = PNd->Urm;
    }
    FaceMatriceaElementara(PElem, &Nod, &MatElem);
    ImpuneBC(PElem, Convectie, Direct);
    ImpuneBC(PElem, Radiatie, Direct);
    ImpuneBC(PElem, FluxDistribuit, Direct);
// Asamblarea matricei globale
    PNd = PElem->NodList;
    for(j = 0 ; j < GdlElem ; j++)
    {
        Idx1 = PNd->Nod - 1;
        PNd1 = PElem->NodList;
        for(k = 0 ; k < GdlElem ; k++)
        {
            Idx2 = PNd1->Nod - 1;
            ScrieMat(Idx1, Idx2, CitMat(Idx1, Idx2) + MatElem[j][k]);
            VerificaHeap(2);
            PNd1 = PNd1->Urm;
        }
        PNd = PNd->Urm;
    }
    PElem++;
}
VerificaHeap(3);
//Salvează coloanele din matricea globală ce vor fi afectate de impunerea condițiilor Dirichlet
//introduse prin secțiunea de date BCINI
PBC = PBConditionsIni;
while(PBC)
{
    if(PBC->TipRestrictie == Temperatura)
    {
        Idx1 = PBC->NodSauElement - 1;
        PSave = Aloca(1, sizeof(ListaBCSaveType));
        PSave->PVectColumn = Aloca(NrNodes, sizeof(ads_real));
        PSave->Impus = PBC->Impus;
        if(!PCapSave)
        {
            PSavel = PCapSave = PSave;
        }
        else
            PSavel->Urm = PSave;
        PVect = PSave->PVectColumn;
        for(i = 0 ; i < NrNodes ; i++)
            *PVect[i] = CitMat(i, Idx1);
        PSavel = PSave;
    }
}

```



```

    }
    PBC = PBC->Urm;
}

// Impunerea condițiilor de contur Dirichlet originale numai în matricea globală

PBC = PBConditionsIni;
while(PBC)
{
    if(PBC->TipRestrictie == Temperatura)
    {
        Idx1 = PBC->NodSauElement - 1;
        for (i = 0 ; i < NrNodes ; i++)
        {
            ScrieMat(Idxl , i , 0);
            ScrieMat(i , Idxl , 0);
        }
        ScrieMat(Idxl , Idxl , 1);
    }
    PBC = PBC->Urm;
}

// Obținerea vectorului cu încărcările asamblate care produc câmpul termic [Q] = [K] [T]

for(i = 0 ; i < NrEqua ; i++)
{
    Factor = 0.0L;
    for(k = 0 ; k < NrEqua ; k++)
        Factor = Factor + CitMat(i, k) * (*PTmpVect[k]);
    // Factor = Factor + CitMat(i, k) * (*PLoadsVect[k]);
    *PSolutionVect[i] = Factor;
}

// Reconstituirea vectorului [Q] inițial, înaintea aplicării condițiilor Dirichlet

PSave = PCapSave;
while(PSave)
{
    PVect = PSave->PVectColumn;
    for(i = 0; i < NrNodes ; i++)
        *PSolutionVect[i] += *PVect[i] * PSave->Impus;
    PSave = PSave->Urm;
}

// Scade în [Q] aportul condițiilor de contur de tip convecție radiație
// introduse de formularea variațională
PElem = PElements;
for(i = 0 ; i < NrElements ; i++)
{
    ImpuneBC(PElem, Convecție, Invers);
    ImpuneBC(PElem, Radiație, Invers);
    ImpuneBC(PElem, FluxDistribuit, Invers);
    PElem++;
}

PBC = PBConditionsIni;
while(PBC)
{
    if(PBC->TipRestrictie == Temperatura)
    {
        Idx1 = PBC->NodSauElement - 1;
        *PSolutionVect[Idx1] = 0;
    }
    PBC = PBC->Urm;
}

// Face lista cu elementele care au sursa internă de căldură
PEListCurent = PEListCap = PEList = NULL;
NrLoads = 0;
PL = PLoad;
while(PL)
{
    if(PL->TipRestrictie == ElemSource)
    {
        Idx1 = PL->NodSauElement - 1;
        PEListCurent = Aloca(1, sizeof(ElementsListType));
        if(!PEListCap)
            PEListCap = PEListCurent;
        else
            PEList->Urm = PEListCurent;
        PEListCurent->Element = PIndexElem[Idx1];
        PEListCurent->Urm = NULL;
        PEList = PEListCurent;
        NrLoads++;
    }
}

```

```

    PL = PL->Urm;
}

// completează vector cu număr noduri de la elementele ce au sursă internă de căldură
PEList = PEListCap ;
Lg = NrLoads * GdlElem;
PNL = Aloca(Lg , sizeof(int));
PNLTmp = Aloca(Lg , sizeof(int));
for(i = 0 ; i < NrLoads ; i++)
{
    PNd1 = PEList->Element->NodList;
    for(j = 0 ; j < GdlElem ; j++)
    {
        *PNL[i * GdlElem + j] = PNd1->Nod;
        PNd1 = PNd1->Urm;
    }
    PEList = PEList->Urm;
}

// Ordonează vector cu numărul nodurilor legate la elemente cu sursă internă
for(i = 1 ; i < Lg ; i++)
{
    j = i;
    while((*PNL[j] < *PNL[j - 1]) && (j > 0))
    {
        k = *PNL[j];
        *PNL[j] = *PNL[j - 1];
        *PNL[--j] = k;
    }
}

// Elimină aparițiile multiple din vector
Nr = 0;
i = 0;
while (i < Lg )
{
    j = 1;
    while(*PNL[i] == *PNL[i + j])
        j++;
    *PNLTmp[Nr] = *PNL[i];
    i = (j - 1) ? i + j : ++i;
    Nr++;
}

// Face matricea încărcare
if(NrLoads > Nr)
{
    ads_printf("\nRedistribuire incarcarea, NrLoads > Nodes: %d > %d",
        NrLoads, Nr);
    ads_exit(1);
}
Dim = NrLoads > Nr ? NrLoads : Nr;
MatLoad = Aloca(Dim * Dim , sizeof(FEMType));
PEList = PEListCap ;
for(i = 0 ; i < NrLoads ; i++)
{
    PNd1 = PEList->Element->NodList;
    for(j = 0 ; j < Nr; j++)
    {
        PNd = PNd1;
        for(k = 0 ; k < GdlElem ; k++)
        {
            if(Este = (PNd->Nod == *PNLTmp[j]))
                break;
            else
                PNd = PNd->Urm;
        }
        if(Este)
            ScrieInt(i , j , *PNLTmp[j], MatLoad, Dim);
        else
            ScrieInt(i , j , 0, MatLoad, Dim );
    }
    PEList = PEList->Urm;
}

// Alege combinația de coloane care asigură det(V) != 0
Coloane = Aloca(Dim, sizeof(int));
for(i = 0 ; i < Dim ; i++)
    *Coloane[i] = 1;
Dif = Nr - NrLoads;

```

```

Scoase = 0;
Start = 0;
Iter = 0;
Err = fopen("ERR.ERR", "w");
do
{
rez = Scoate(Start, Coloane, Scoase);
Start++;
}
while((rez != GATA) && (Start != Dim));
fclose(Err);
if(rez != GATA)
{
Fis = fopen("MEF.ERR" , "w");
ads_printf("\nMatrice singulara");
fprintf(Fis, "Total iteratii: %d", Iter);
fprintf(Fis, "\nTotal noduri considerate %d\nTotal elemente considerate %d\n",
Nr, NrLoads);
for(i = 0 ; i < NrLoads ; i++)
{
for(j = 0 ; j < Nr ; j++)
{
sprintf(Tmp , "%5d" , CitInt(i , j, MatLoad, Dim));
if(!j)
strcpy(Line , Tmp);
else
strcat(Line , Tmp);
strcat(Line , " ");
}
fprintf(Fis , "%s\n" , Line);
}

fclose(Fis);
ads_exit(1);
}

// Completează matricea globală V
Rez = NrNodes;
NrNodes = NrLoads;
free(PGlobalMatrix);
PGlobalMatrix = Aloca(NrLoads * NrLoads , sizeof(ads_real));
k = 0;
for(i = 0 ; i < Nr ; i++)
{
if(*Coloane[i])
{
PEList = PEListCap;
for(j = 0 ; j < NrLoads ; j ++ )
{
if(CitInt(j, i, MatLoad, Dim))
{
Factor = PEList->Element->Jacobi / 6;
ScrieMat(k, j, Factor);
}
PEList = PEList->Urm;
}
k++;
}
}

// Încarcă vector Q1 cu valorile încărcărilor asamblate
free(PLoadsVect);
PLoadsVect = Aloca(NrLoads, sizeof(ads_real));
j = 0;
for(i = 0 ; i < Nr ; i++)
{
if(*Coloane[i])
{
*PLoadsVect[j] = *PSolutionVect[*PNLTmp[i] - 1];
j++;
}
}

// Rezolva sistemul
free(PSolutionVect);
PSolutionVect = Aloca(NrLoads , sizeof(ads_real));

switch(GaussTip)
{
case GAUSS_ELABORAT:
GaussElab(NrLoads, Invers);
break;
case GAUSS_SIMPLU:

```

```

    Gauss(NrLoads, Invers);
    break;
case GAUSS_INVERS:
    CuInvers(NrLoads, Invers);
}

NrNodes = Rez;
NrEqua = NrLoads;
if(ScaraGeometrie != 1)
    Scaleaza(1 / ScaraGeometrie);
ScrieSolutiile(NrEqua,100, Invers);

//      Eliberează lista cu elemente ce conțin încărcări

free(PTmpVect);
PEListCurent = PEListCap;
while(PEListCurent)
{
    PEList = PEListCurent->Urm;
    free(PEListCurent);
    PEListCurent = PEList;
}
return OK;
}

int Ver(void)
/**#30
// Funcție care verifică datele de intrare
//

{
FEMType i, j, (*PNodTmp)[1];
NodesType *PNodCurent, *PNod = PNodes;
ElementsType *PElem = PElements;
NodesListType *PNL;
FEMType dx, dy, dz;

PNodTmp = Aloca(NrNodes, sizeof(FEMType));
for(i = 0 ; i < (NrNodes - 1) ; i++)
{
    PNodCurent = PNod + 1;
    for(j = i + 1 ; j < NrNodes ; j++)
    {
        dx = fabs(PNodCurent->x - PNod->x) <= PrecVer;
        dy = fabs(PNodCurent->y - PNod->y) <= PrecVer;
        dz = fabs(PNodCurent->z - PNod->z) <= PrecVer;
        if(dx && dy && dz)
            ads_printf("\nNoduri coincidente :%d %d", PNodCurent->Nr, PNod->Nr);
        PNodCurent++;
    }
    PNod++;
}
for(i = 0 ; i < NrElements ; i++)
{
    PNL = PElem->NodList;
    for(j = 0 ; j < GdlElem ; j++)
    {
        if((PNL->Nod > NrNodes) || (PNL->Nod <= 0))
            ads_printf("\nNod inexistent %d la element %d", PNL->Nod, i + 1);
        *PNodTmp[PNL->Nod - 1] = 1;
        PNL = PNL->Urm;
    }
    PElem++;
}
for(i = 0 ; i < NrNodes ; i++)
{
    if(!*PNodTmp[i])
        ads_printf("\nNod nefolosit :%d\n", i + 1);
}
if(!PNodes)
    ads_printf("\nSecțiune noduri nedefinita");
if(!PElements)
    ads_printf("\nSecțiune elemente nedefinita");
if(!PMaterials)
    ads_printf("\nSecțiune materiale nedefinita");
if(!PBConditions)
    ads_printf("\nSecțiune condiții de contur nedefinita");
if(!PLoad)
    ads_printf("\nSecțiune incaracari nedefinita");

free(PNodTmp);
return 1;
}

```

```

int Fem (void)
###31
// Funcție care citește datele de intrare
//
{
    int rez;

    if((rez = InputFromASCII()) == EROARE)
        return EROARE;

    return OK;
}

static int loadfuncs()
###32
// Funcție care încarcă și înregistrează funcțiile care au fost declarate ca și comenzi
//
{
    if (ads_defun("C:Fem" , 0) == RTNORM)
        ads_regfunc(Fem, 0);
    else
        return 0;

    if (ads_defun("C:Model" , 1) == RTNORM)
        ads_regfunc(Model, 1);
    else
        return 0;

    if (ads_defun("C:Solve" , 2) == RTNORM)
        ads_regfunc(Solve, 2);
    else
        return 0;

    if (ads_defun("C:Inv" , 3) == RTNORM)
        ads_regfunc(Inv, 3);
    else
        return 0;

    if (ads_defun("C:Ver" , 4) == RTNORM)
        ads_regfunc(Ver, 4);
    else
        return 0;
    return 1;
}

void main(argc, argv)
###33
// Funcție principală care conține bucla infinită pentru legătura cu nucleul ADS
//
int argc;
char *argv[];
{
    int stat;
    short scode = RSRSLT;

    ads_init(argc, argv);
    for ( ;; ) {

        if ((stat = ads_link(scode)) < 0)
        {
            printf("MONTEAZA ** EROARE CONEXIUNE ADS ** = %d \n", stat);
            fflush(stdout);
            exit(1);
        }
        scode = RSRSLT;
        switch (stat)
        {
            case RQXLOAD:          scode = loadfuncs() ? RSRSLT : RSERR;
                break;
            case RQSUBR:           break;
            case RQXUNLD:         break;
            case RQSAVE:          break;
            case RQQUIT:          break;
            case RQEND:           break;
            default:               break;
        }
    }
}

```

## 9. ANEXA 2 - PROGRAM POSTPROCESARE TERMOGRAME

### 9.1. CARACTERISTICI

- Programul Post preia fișierul de ieșire de la programul Fem, pentru problema directă și afișează soluția problemei folosind o paletă de culori ușor de interpretat;
- Afișează dinamic coordonatele mouse-ului, relativ la scara modelului, precum și temperatura corespunzătoare punctului respectiv în raport cu scara gradată din stânga ecranului
- Format fișier executabil - aplicație Windows;
- Limbaj de programare C;
- Compilator folosit: Microsoft Visual C++1.5;
- structură de meniuri:

#### Fișiere ⇒

Deschide Fișier .....*Apelează dialogul standard pentru deschidere fișier cu soluții*  
Închide .....*Închide program*

#### Control rețea ⇒

##### Noduri⇒

Marcator.....Dacă este bifat face un marcator cerc în nodurile rețelei  
Număr .....Dacă este bifat afișează numărul nodului lângă el  
Font.....Deschide fereastră de dialog pentru controlul fontului cu care se afișează numărul de nod

##### Elemente⇒

Contur .....Dacă este bifat face conturul elementelor rețelei  
Număr .....Dacă este bifat afișează numărul elementului în centrul lui  
Font.....Deschide fereastră de dialog pentru controlul fontului cu care se afișează numărul elementului  
Afișaj dinamic .....Dacă este bifat afișează dinamic coordonatele mouse-ului și valoarea temperaturii punctului curent  
Factor difuzie .....Deschide fereastră de dialog pentru controlul mărimii triunghiurilor de render; valori acceptate 0...6

#### Control imagine ⇒

Reafișează .....Reafișează informația curentă de pe ecran  
Soluție problemă.....Dacă este bifat reprezintă în triunghiuri colorate soluția problemei

#### Help ⇒

Despre program .....Afișează fereastră cu informații despre program

#### Fișiere de intrare

*Fișierul cu soluția problemei:*

format : ASCII

extensia implicită : PST

structura fișierului: este aceeași cu cea a fișierului de ieșire REZ99.DAT creat de programul Fem.

**Fișiere de ieșire :** Nu este cazul

Obs : În anexă nu sunt cuprinse fișierul cu resursele Windows ale acestui program, precum și fișierul de definiție, care nu prezintă interes.

Aspecte cu fereastra controlată de program, se află în anexa 7, precum și în cap.5.2

## 9.2. FIȘIER HEADER

```

enum SectiuniID {IsTitle =0 , IsNodes, IsElements, IsSolutions};

#define      IsEtapa1      1
#define      IsEtapa2      2
#define      IsEtapa3      3
#define      IsEtapa4      4
#define      X              0
#define      Y              1

#define IntType      100
#define LongType     101
#define FloatType    102
#define DoubleType   103
#define CharType     104
#define MaxUint      0xFFFFU
#define ScAjustare   (float)1 // (float)0.9
#define Reface       TRUE
#define IsRender     TRUE
#define Subdiviziuni 4
#define NodPerElement 4

#define      IncPaleta    4
#define      LgColor      256
#define      Dim1          (LgColor / IncPaleta)
#define      DimPaleta    (Dim1 * 3)
#define      ForegroundPalette 0
#define      LegendaSeg   9
#define      BlockFis     1024

struct ListaASCII
{
    HLOCAL      pObj;
    HLOCAL      pObjSir;
    char far    *pSir;
    int         Lg;
    struct ListaASCII *Urm;
};
typedef struct ListaASCII ListaASCIIType;

struct ListaNodes
{
    HLOCAL      pObj;
    float       x, y, z, Sol;
    int         Nr, xScreen, yScreen;
    struct ListaNodes *Urm;
};
typedef struct ListaNodes ListaNodesType;

struct ListaElements
{
    HLOCAL      pObj;
    int         Nr, Nod[4];
    struct ListaElements *Urm;
};
typedef struct ListaElements ListaElementsType;

```



### 9.3. FIȘIER SURSĂ

```

/*-----
POST.C -- Graphic FEA PostProcessor
          © Adrian ROȘCA, 1996
          Version 1.0
-----*/
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <commdlg.h>
#include <string.h>
#include <float.h>
#include <math.h>
#include "resource.h"
#include "post.h"

char Titlu[80];
char ApplicationName[20] = "FEA Postprocessor";
CHOOSECOLOR BackColour , ForeColour , Colour;
LOGPALETTE *Paleta;
HLOCAL pASCIIobj, pObjPaleta;
ListaNodesType *pNodNou, *pNodCap, *pNodCurent;
ListaElementsType *pElementNou, *pElementCap, *pElementCurent;
int OrdinFinal = 2;
float xmax = FLT_MIN, ymax = FLT_MIN, xmin = FLT_MAX, ymin = FLT_MAX;
float SolMax = FLT_MIN, SolMin = FLT_MAX, ScX, ScY, Sc = 1;
HFONT hFontGeneral, hFontNoduri, hFontElemente;
CHOOSEFONT FontNoduri, FontElemente, FontGeneral;
LOGFONT lfNoduri, lfElemente, lfGeneral;
HPALETTE hPal, hPrev;
HDC hdcMem;
char Coord[] = "Coordonate";
HANDLE hInst;
OPENFILENAME ofn;

long FAR PASCAL _export WndProc (HWND , UINT , UINT, LONG );
long FAR PASCAL _export WndProcCoord (HWND , UINT , UINT, LONG );
BOOL FAR PASCAL _export DespreDlgProc (HWND , UINT , UINT , LONG );
BOOL FAR PASCAL _export DifuzieDlgProc (HWND , UINT , UINT , LONG );
void Dealoca(void);

void PopFileInitialize (HWND hwnd , BOOL ForWrite)
// ##1
// Funcție care inițializează structura folosită de dialogul standard de deschidere a fișierului
//
{
    static char *szFilter[] = { "Post Files (*.pst)", "*.pst", "Data Files (*.dat)", "*.dat",
                               "All Files (*.*)", " *.*", "" };
    static char AFile[20] , ACompleteFile[132];
    lstrcpy(ACompleteFile , szFilter[1]);

    ofn.lStructSize = sizeof (OPENFILENAME) ;
    ofn.hwndOwner = hwnd ;
    ofn.hInstance = (HINSTANCE)NULL ;
    ofn.lpstrFilter = szFilter [0] ;
    ofn.lpstrCustomFilter = NULL ;
    ofn.nMaxCustFilter = 0 ;
    ofn.nFilterIndex = 0 ;
    ofn.lpstrFile = ACompleteFile;
    ofn.nMaxFile = _MAX_PATH ;
    ofn.lpstrFileTitle = AFile ;
    ofn.nMaxFileTitle = _MAX_FNAME + _MAX_EXT ;
    ofn.lpstrInitialDir = "" ;
    ofn.lpstrTitle = NULL ;
    ofn.Flags = OFN_OVERWRITEPROMPT ;
    ofn.nFileOffset = 0 ;
    ofn.nFileExtension = 0 ;
    ofn.lpstrDefExt = "cfg" ;
    ofn.lCustData = 0L ;
    ofn.lpfnHook = NULL ;
    ofn.lpTemplateName = NULL ;
}

HFONT FaceFont(CHOOSEFONT *pFont, HWND hwnd)
// ##2
// Funcție care creează un font
//
{
    pFont->lStructSize = sizeof(CHOOSEFONT);
    pFont->hwndOwner = hwnd;
    pFont->iPointSize = pFont->lpLogFont->lfHeight;
}

```

```

return CreateFont(pFont->lpLogFont->lfHeight, 0, 0, 0, pFont->lpLogFont->lfWeight,
                 pFont->lpLogFont->lfItalic, pFont->lpLogFont->lfUnderline,
                 pFont->lpLogFont->lfStrikeOut, 0, 0, 0, 0, 0, pFont->lpLogFont-
>lfFaceName);
}

```

```

ListaNodesType *CautaNod(int Nr)

```

```

{
// ##3
// Funcție care caută un nod în listă
//
ListaNodesType *pNod = pNodCap;

while (pNod)
{
if(pNod->Nr != Nr)
pNod = pNod->Urm;
else
return pNod;
}
return pNod;
}

```

```

void ReadPostFile(char *Name , HWND hwnd)

```

```

{
// ##4
// Funcție care citește datele de intrare
//
HFILE PFile;
int cit, Total = 0, SectiuneaCurenta, NrNod;
static Tmp[BlockFis];
char *Poz , *pSect, *pCR, *pEnd;
char *Sectiuni[] = {"TITLE" , "NODES" , "ELEMENTS", "SOLUTII", "INCARCARI"};
int CurentSection = 0;
ListaASCIIType *pNou, *pCap = NULL , *pCurent, *pASCII;
HLOCAL pH;
char *pFmtNod = "%d %g %g %g%", *pFmtElement1 = "%d", *pFmtElement2 = "%d %d %d %d";
char *pFmtSolutions = "%d %f";
float Sol;

PFile = _lopen(Name, READ);

Zero(Tmp , BlockFis);

cit = _lread(PFile , Tmp , BlockFis);
do
{
if((void *) (pH = LocalAlloc(LPTR, sizeof(ListaASCIIType))) == NULL)
{
EroareAlocare("Citeste Post Fis -> Obj");
return;
}
pNou = (ListaASCIIType *)LocalLock(pH);
pNou->pObj = pH;
pNou->Urm = NULL;
if(!pCap)
{
pCurent = pCap = pNou;
pCurent->Urm = NULL;
}
else
pCurent->Urm = pNou;
if((void *) (pNou->pObjSir = (HLOCAL)LocalAlloc(LPTR, cit)) == NULL)
{
EroareAlocare("Citeste Post Fis -> Sir");
return;
}
pNou->pSir = (char *)LocalLock(pNou->pObjSir);
pNou->Lg = cit;
_fmemcpy(pNou->pSir , Tmp, cit);
Total += cit;
cit = _lread(PFile , Tmp , BlockFis);
pCurent = pNou;
}
while(cit);
_lclose(PFile);

if((void *) (pH = LocalAlloc(LPTR, sizeof(ListaASCIIType))) == NULL)
{
EroareAlocare("Zonă Global Fis -> Obj");
return;
}
pASCII = (ListaASCIIType *)LocalLock(pH);

```

```

pASCII->pObj = pH;
Total += 2;
if((void *) (pASCII->pObjSir = (HLOCAL)LocalAlloc(LPTR , Total)) == NULL)
{
    EroareAlocare("Zonă Global Fis -> Sir");
    return;
}
pASCII->pSir = (char *)LocalLock(pASCII->pObjSir);
pCurent = pCap;
Poz = pASCII->pSir;
do
{
    _fmemcpy(Poz , pCurent->pSir , pCurent->Lg);
    Poz += pCurent->Lg;
    pCap = pCap->Urm;
    LocalUnlock(pCurent->pObjSir);
    LocalFree(pCurent->pObjSir);
    LocalUnlock(pCurent->pObj);
    LocalFree(pCurent->pObj);
    pCurent = pCap;
}
while(pCurent != NULL);

pCR = pASCII->pSir + Total - 1;
*pCR++ = (char)13; *pCR = '\0';
Poz = pASCII->pSir;
pEnd = Poz + Total;
SectiuneaCurenta = IsTitle;

do
{
    if((pCR = _fstrchr(Poz , 13)) != NULL)
        *pCR = '\0';
    if(pSect = _fstrstr(Poz , Sectiuni[SectiuneaCurenta]))
        SectiuneaCurenta++;
}
else
{
    switch (SectiuneaCurenta - 1)
    {
        case IsTitle:
            _fstrcpy(Titlu, Poz);
            break;

        case IsNodes:
            if((void *) (pH = LocalAlloc(LPTR , sizeof(ListaNodesType))) == NULL)
            {
                EroareAlocare("Noduri");
                return;
            }
            pNodNou = (ListaNodesType *)LocalLock(pH);
            pNodNou->pObj = pH;
            pNodNou->Urm = NULL;
            sscanf(Poz, pFmtNod, &pNodNou->Nr, &pNodNou->x, &pNodNou->y, &pNodNou->z);
            if(!pNodCap)
            {
                pNodCurent = pNodCap = pNodNou;
            }
            else
                pNodCurent->Urm = pNodNou;
            pNodCurent = pNodNou;
            break;

        case IsElements:
            if((void *) (pH = LocalAlloc(LPTR , sizeof(ListaElementsType))) == NULL)
            {
                EroareAlocare("Elemente");
                return;
            }
            pElementNou = (ListaElementsType *)LocalLock(pH);
            pElementNou->pObj = pH;
            pElementNou->Urm = NULL;
            sscanf(Poz, pFmtElement1, &pElementNou->Nr);
            Poz = pCR + 2;
            pCR = _fstrchr(Poz , 13);
            *pCR = '\0';
            sscanf(Poz, pFmtElement2, &pElementNou->Nod[0], &pElementNou->Nod[1],
                &pElementNou->Nod[2], &pElementNou->Nod[3]);
            if(!pElementCap)
            {
                pElementCurent = pElementCap = pElementNou;
            }
            else
                pElementCurent->Urm = pElementNou;
            pElementCurent = pElementNou;
    }
}

```

```

        break;
    case IsSolutions:
        sscanf(Poz, pFmtSolutions, &NrNod, &Sol);
        SolMax = (Sol > SolMax) ? Sol : SolMax ;
        SolMin = (Sol < SolMin) ? Sol : SolMin ;
        if(pNodCurent = CautaNod(NrNod))
            pNodCurent->Sol = Sol;
        else
            MessageBox((HWND)NULL , (LPCSTR)_fstreat((char far *)"Nod
inexistent ",
(char far *)itoa(NrNod, (char *)Tmp, 10)), (LPCSTR)"Solutii" , MB_ICONSTOP ||
MB_OK);
        default :break;
    }
    }
    Poz = pCR + 2;
}
while((Poz < pEnd) && pCR);
LocalUnlock(pASCII->pObjSir);
LocalFree(pASCII->pObjSir);
LocalUnlock(pASCII->pObj);
LocalFree(pASCII->pObj);
}

int CoordNod(int Nr, int coord)
// ##5
// Funcție care returnează coordonatele ecran ale unui nod
//
{
    ListaNodesType *Nod = pNodCap;

    while(Nod->Nr != Nr)
        Nod = Nod->Urm;
    switch (coord)
    {
        case X:
            return Nod->xScreen;
            break;
        case Y:
            return Nod->yScreen;
            break;
        default: ;
    }
}

float SolNod(int Nr)
{
// ##6
// Funcție care returnează soluția nodală
//
    ListaNodesType *Nod = pNodCap;

    while(Nod->Nr != Nr)
        Nod = Nod->Urm;
    return Nod->Sol;
}

void CalcCoord(HWND hwnd)
// ##7
// Funcție care calculează coordonatele ecran pentru noduri
//
{
    ListaNodesType *Nod;
    int Lat, Inalt;
    HDC hdc;
    POINT P;
    DWORD Dublu;

    if (xmax != FLT_MIN)
        return;
    Nod = pNodCap;
    while(Nod)
    {
        if(Nod->x > xmax)
            xmax = Nod->x;
        if(Nod->y > ymax)
            ymax = Nod->y;

        if(Nod->x < xmin)
            xmin = Nod->x;
    }
}

```

```

        if(Nod->y < ymin)
            ymin = Nod->y;
        Nod = Nod->Urm;
    }
    hdc = GetDC(hwnd);
    InfoDisplay(hdc, &P);
    Dublu = GetViewportOrg(hdc);
    Lat = (P.x - P.x / 5) * 10;
    Inalt = P.y * 10 - HIWORD(Dublu) * 0.5;
    ScX = Lat / (xmax - xmin);
    ScY = Inalt / (ymax - ymin);
    Sc = (float)(ScX < ScY ? ScX : ScY) * ScAjustare;
    SetViewportOrg(hdc, P.x / 4 + P.x / 3, HIWORD(Dublu) - P.y / 6);
    Nod = pNodCap;
    do
    {
        Nod->xScreen = (int)(Nod->x * Sc) - xmin * Sc;
        Nod->yScreen = (int)(Nod->y * Sc) - ymin * Sc;
        Nod = Nod->Urm;
    }
    while(Nod);
    ReleaseDC(hwnd, hdc);
}

DWORD CuloareMedie(HDC hdc, ListaNodesType v[NodPerElement], int tip)
// ##8
// Funcție care calculează temperatura medie pentru un triunghi de render
//
{
    float Media = 0;
    int i;

    for(i = 0; i < tip; i++)
        Media += v[i].Sol;
    i = (int)((SolMax - Media / tip) / (SolMax - SolMin) * DimPaleta);
    if(i >= DimPaleta)
        i = DimPaleta - 1;
    else
        if(i < 0)
            i = 0;
    return RGB(Paleta->palPalEntry[i].peRed, Paleta->palPalEntry[i].peGreen,
        Paleta->palPalEntry[i].peBlue);
}

float ReturnSolutie(HDC hdc, COLORREF Col)
// ##9
// Funcție care calculează valoarea temperaturii atașată la o culoare
//
{
    float Ret, Fact;
    COLORREF Curent;
    int i;

    for(i = 0; i < DimPaleta; i++)
    {
        Curent = RGB(Paleta->palPalEntry[i].peRed, Paleta->palPalEntry[i].peGreen,
            Paleta->palPalEntry[i].peBlue);
        if(Curent == Col)
        {
            Fact = i / (float)DimPaleta;
            return Ret = (float)SolMax * (1 - Fact) + SolMin * Fact;
        }
    }
    return FLT_MIN;
}

void DivideElement(ListaNodesType pNod[4], int OrdinCurent, HDC hdc, int tip)
// ##10
// Funcție care împarte un element finit în triunghiuri render
//
{
    ListaNodesType L[Subdiviziuni][NodPerElement];
    int i, j;
    POINT Vertex[NodPerElement];
    DWORD Culoare;
    HBRUSH hBrush, hBPrev;
    HPEN hp, hpPrev;
    RECT Rect;

    switch(tip)
    {

```

```

case 3:
    L[0][0].xScreen = (pNod[0].xScreen + pNod[1].xScreen) / 2;
    L[0][0].yScreen = (pNod[0].yScreen + pNod[1].yScreen) / 2;
    L[0][0].Sol = (pNod[0].Sol + pNod[1].Sol) / 2;

    L[0][1].xScreen = (pNod[1].xScreen + pNod[2].xScreen) / 2;
    L[0][1].yScreen = (pNod[1].yScreen + pNod[2].yScreen) / 2;
    L[0][1].Sol = (pNod[1].Sol + pNod[2].Sol) / 2;
    L[0][2].xScreen = (pNod[2].xScreen + pNod[0].xScreen) / 2;
    L[0][2].yScreen = (pNod[2].yScreen + pNod[0].yScreen) / 2;
    L[0][2].Sol = (pNod[2].Sol + pNod[0].Sol) / 2;
    L[1][0] = pNod[0];      L[1][1] = L[0][0];      L[1][2] = L[0][2];
    L[2][0] = pNod[1];      L[2][1] = L[0][1];      L[2][2] = L[0][0];
    L[3][0] = pNod[2];      L[3][1] = L[0][2];      L[3][2] = L[0][1];
    break;
case 4:
    L[0][0] = pNod[0];
    L[0][1].xScreen = (pNod[0].xScreen + pNod[1].xScreen) / 2;
    L[0][1].yScreen = (pNod[0].yScreen + pNod[1].yScreen) / 2;
    L[0][1].Sol = (pNod[0].Sol + pNod[1].Sol) / 2;
    L[0][3].xScreen = (pNod[3].xScreen + pNod[0].xScreen) / 2;
    L[0][3].yScreen = (pNod[3].yScreen + pNod[0].yScreen) / 2;
    L[0][3].Sol = (pNod[3].Sol + pNod[0].Sol) / 2;
    L[1][0] = L[0][1];      L[1][1] = pNod[1];
    L[1][2].xScreen = (pNod[1].xScreen + pNod[2].xScreen) / 2;
    L[1][2].yScreen = (pNod[1].yScreen + pNod[2].yScreen) / 2;
    L[1][2].Sol = (pNod[1].Sol + pNod[2].Sol) / 2;
    L[2][1] = L[1][2];      L[2][2] = pNod[2];
    L[2][3].xScreen = (pNod[2].xScreen + pNod[3].xScreen) / 2;
    L[2][3].yScreen = (pNod[2].yScreen + pNod[3].yScreen) / 2;
    L[2][3].Sol = (pNod[2].Sol + pNod[3].Sol) / 2;
    L[3][0].xScreen = (pNod[0].xScreen + pNod[3].xScreen) / 2;
    L[3][0].yScreen = (pNod[0].yScreen + pNod[3].yScreen) / 2;
    L[3][0].Sol = (pNod[0].Sol + pNod[3].Sol) / 2;
    L[3][1].xScreen = (L[0][1].xScreen + L[2][3].xScreen) / 2;
    L[3][1].yScreen = (L[3][0].yScreen + L[1][2].yScreen) / 2;
    L[3][1].Sol = (L[0][1].Sol + L[1][2].Sol + L[2][3].Sol + L[3][0].Sol) / 4;
    L[3][2] = L[2][3];      L[3][3] = pNod[3];
    L[0][2] = L[1][3] = L[2][0] = L[3][1];
    break;
default:;
}
if(OrdinCurent < OrdinFinal)
{
    DivideElement(L[1], OrdinCurent + 1, hdc, tip);
    DivideElement(L[2], OrdinCurent + 1, hdc, tip);
    DivideElement(L[3], OrdinCurent + 1, hdc, tip);
    DivideElement(L[0], OrdinCurent + 1, hdc, tip);
}
else
{
    Rect.left = Rect.bottom = 0;
    Rect.right = (int)(xmax * ScX);
    Rect.top = (int)(ymax * ScY);

    SelectPalette(hdc, hPal, ForegroundPalette);
    i = RealizePalette(hdc);

    for(i = 0; i < Subdiviziuni; i++)
    {
        for(j = 0; j < tip; j++)
        {
            Vertex[j].x = L[i][j].xScreen;
            Vertex[j].y = L[i][j].yScreen;
        }
        Culoare = CuloareMedie(hdc, L[i], tip);
        hBrush = CreateSolidBrush(Culoare);
        hBPrev = SelectObject(hdc, hBrush);
        hp = CreatePen(PS_SOLID, 1, Culoare);
        hpPrev = SelectObject(hdc, hp);
        Polygon(hdc, Vertex, tip);
        SelectObject(hdc, hBPrev);
        SelectObject(hdc, hpPrev);
        DeleteObject(hBrush);
        DeleteObject(hp);
    }
}
}

```

```

void PlotStruct(HWND hwnd, int Render, int DacaReface)
// ##11
// Funcție care afișează modelul cu elemente finite

```

```

//
{
    HPEN hp;
    HDC hdc;
    HCURSOR hCursor;
    ListaElementsType *Elem;
    RECT Rect;
    int i;
    static POINT Punct;
    ListaNodesType Lista[NodPerElement], *pNod;
    static hMeta;
    POINT P;

    if(pNodCap)
        CalcCoord(hwnd);
    else
        return;
hdc = GetDC(hwnd);
InfoDisplay(hdc, &P);
Rect.left = Rect.bottom = -7;
Rect.right = P.x * 10;
Rect.top = (int)(ymax * ScY) + 60;
FillRect(hdc, &Rect, GetStockObject(WHITE_BRUSH));
if(!Render)
{
    ReleaseDC(hwnd, hdc);
    return;
}
if (TRUE)          //(DacaReface)
{
    SelectObject(hdc, hFontGeneral);
    hp = CreatePen(PS_SOLID, 1, BLACK_PEN);
    SelectObject(hdc, hp);
    Elem = pElementCap;
    hCursor = SetCursor(LoadCursor((HINSTANCE)NULL, IDC_WAIT));
    ShowCursor(TRUE);
    do
    {
        i = 0;
        while(pNod = CautaNod(Elem->Nod[i]))
            Lista[i++] = *pNod;
        DivideElement(Lista, 1, hdc, i);
        SelectObject(hdc, hp);
        Elem = Elem->Urm;
    }
    while(Elem);
    SetCursor(hCursor);
    ShowCursor(FALSE);
    DeleteObject(hp);
}
else
{
    PlayMetaFile(hdc, hMeta);
}
ReleaseDC(hwnd, hdc);
}

void FaceScala(HWND hwnd)
// ##11
// Funcție care afișează scala de temperaturi
//
{
    int Lg, x, y, xIni, yIni, Gros;
    int i;
    HPEN hp, hPrev, hBPrev;
    HGDIOBJ hbr;
    HDC hdc;
    char Sir[22];
    POINT Info;
    RECT Rect;
    DWORD Dublu;
    COLORREF Col;

    float Sol, EcartLegenda, EcartSol;
    hdc = GetDC(hwnd);
    Dublu = GetViewportOrg(hdc);
    GetClientRect(hwnd, &Rect);
    SetViewportOrg(hdc, 0, Rect.bottom);
    SelectObject(hdc, hFontGeneral);
    InfoDisplay(hdc, &Info);
    x = Info.x / 13;
    y = Info.y * 9;
}

```



```

Lg = Info.x / 4;
Gros = (int)(Info.y / 12.5);
xIni = x + 25 + Lg;
yIni = y;
    SelectPalette(hdc, hPal, ForegroundPalette);
    i = RealizePalette(hdc);
hbr = GetStockObject(BLACK_BRUSH);
hBPrev = SelectObject(hdc, hbr);
    for(i = 0 ; i < DimPaleta ; i++,i++)
    {
        Col = RGB(Paleta->palPalEntry[i].peRed,
                Paleta->palPalEntry[i].peGreen, Paleta->palPalEntry[i].peBlue);

        hp = CreatePen(PS_SOLID , Gros , Col);
        hPrev = SelectObject(hdc , hp);
        MoveTo(hdc , x , y);
        LineTo(hdc , x + Lg , y);
        SelectObject(hdc , hPrev);
        DeleteObject(hp);
        y -= Gros;
    }

hp = CreatePen(PS_SOLID , 1 , BLACK_PEN);
hPrev = SelectObject(hdc, hp);
EcartSol = (SolMax - SolMin) / LegendaSeg;
EcartLegenda = (yIni - y) / LegendaSeg;
y = yIni;
x = xIni - 10;
Sol = SolMax;
SetTextAlign(hdc, TA_BASELINE);
for(i = 0 ; i <= LegendaSeg ; i++)
    {
        MoveTo(hdc, x, y);
        LineTo(hdc, xIni + 2, y);
        sprintf(Sir, "%.6f", Sol);
        TextOut(hdc, xIni + 8, y, Sir, _fstrlen(Sir));
        y -= (int)EcartLegenda;
        Sol -= EcartSol;
    }

MoveTo(hdc, xIni, yIni);
LineTo(hdc, xIni , y + (int)EcartLegenda);
SetViewportOrg(hdc, LOWORD(Dublu) , HIWORD(Dublu));
SelectObject(hdc, hBPrev);
SelectObject(hdc, hPrev);
DeleteObject(hp);
ReleaseDC(hwnd , hdc);
}

void ScrieNoduri(HWND hwnd, BOOL IsNumber)
// ##12
// Funcție care scrie număr nod
//
{
    ListaNodesType *pNod = pNodCap;
    char pSir[5];
    HDC hdc;
    int Ante;
    COLORREF ColorPrev;

    hdc = GetDC(hwnd);
    Ante = SetBkMode(hdc, TRANSPARENT);
    ColorPrev = SetTextColor(hdc, FontNoduri.rgbColors);
    SelectObject(hdc, hFontNoduri);

    while(pNod)
    {
        sprintf(pSir, "%d", pNod->Nr);
        Ellipse(hdc, pNod->xScreen - 6, pNod->yScreen + 6, pNod->xScreen + 6, pNod->yScreen - 6);
        if(IsNumber)
            TextOut(hdc , pNod->xScreen , pNod->yScreen + 7, pSir, _fstrlen(pSir));
        pNod = pNod->Urm;
    }
    SetBkMode(hdc, Ante);
    SetTextColor(hdc, ColorPrev);
    ReleaseDC(hwnd, hdc);
}

void FaceContur(HWND hwnd, BOOL IsNumber)
// ##13
// Funcție care afișează contur elemente
//
{
    ListaElementsType *pElem = pElementCap;
    char pSir[5];
    HDC hdc;

```

```

int Ante, xs, ys, x, y, i, xl, yl, tip;
COLORREF ColorPrev;
UINT PrevAlign;

hdc = GetDC(hwnd);
Ante = SetBkMode(hdc, TRANSPARENT);
ColorPrev = SetTextColor(hdc, FontElemente.rgbColors);
SelectObject(hdc, hFontElemente);
PrevAlign = SetTextAlign(hdc, TA_CENTER | TA_BOTTOM );
while(pElem)
{
    sprintf(pSir, "%d", pElem->Nr);
    xs = ys = 0;
    for(i = 0, tip = 0; i < NodPerElement; i++)
    {
        if(pElem->Nod[i] != 0)
        {
            tip++;
            x = CoordNod(pElem->Nod[i], X);
            y = CoordNod(pElem->Nod[i], Y);
            xs += x;
            ys += y;
            if(i == 0)
            {
                MoveTo(hdc, x, y);
                xl = x;
                yl = y;
            }
            else
                LineTo(hdc, x, y);
        }
        LineTo(hdc, xl, yl);
        if(IsNumber)
            TextOut(hdc, xs / tip, ys / tip, pSir, _fstrlen(pSir));
        pElem = pElem->Urm;
    }
    SetBkMode(hdc, Ante);
    SetTextColor(hdc, ColorPrev);
    SetTextAlign(hdc, PrevAlign);
    ReleaseDC(hwnd, hdc);
}

```

```

int PASCAL WinMain (HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR lpszCmdLine, int nCmdShow)

```

```

// ##14
// Funcție principală
//
{
    static char szAppName[] = "Graphic_FEA_PostProcessor" ;
    static char szMenuName[] = "MenuPrincipal";
    static char szIcoName[] = "FEAIcon";

    HWND      hwnd ;
    MSG       msg ;
    WNDCLASS  wndclass ;
    if (!hPrevInstance)
    {
        wndclass.style = CS_HREDRAW | CS_VREDRAW | CS_OWNDC | CS_NOCLOSE | CS_BYTEALIGNCLIENT
        ;
        wndclass.lpfnWndProc = WndProc ;
        wndclass.cbClsExtra = 0 ;
        wndclass.cbWndExtra = 0 ;
        wndclass.hInstance = hInstance ;
        wndclass.hIcon = LoadIcon (hInstance, szIcoName ) ;
        wndclass.hCursor = LoadCursor ((HINSTANCE)NULL, IDC_ARROW) ;
        wndclass.hbrBackground = GetStockObject (WHITE_BRUSH) ;
        wndclass.lpszMenuName = szMenuName;
        wndclass.lpszClassName = szAppName ;
        RegisterClass (&wndclass) ;

        wndclass.style = CS_HREDRAW | CS_VREDRAW | CS_OWNDC;
        wndclass.lpfnWndProc = WndProcCoord ;
        wndclass.hbrBackground = GetStockObject (LTGRAY_BRUSH) ;
        wndclass.cbWndExtra = sizeof(int);
        wndclass.hIcon = (HICON)NULL;
        wndclass.hCursor = (HCURSOR)NULL;
        wndclass.lpszMenuName = (HMENU)NULL;
        wndclass.lpszClassName = Coord;
        RegisterClass (&wndclass) ;
    }
    else
    {

```

```

    MessageBox((HWND)NULL , (LPCSTR)"Este deja lansat" , (LPCSTR)NULL , MB_OK);
    return 0;
}
hInst = hInstance;
hwnd = CreateWindow (szAppName, ApplicationName ,WS_CAPTION | WS_THICKFRAME |
    WS_MINIMIZEBOX | WS_MAXIMIZEBOX | WS_SYSMENU, CW_USEDEFAULT,
    CW_USEDEFAULT, 800 , 600, (HWND)NULL, (HMENU)NULL, hInstance, NULL) ;
ShowWindow (hwnd, SW_SHOWMAXIMIZED) ;
UpdateWindow (hwnd) ;
while (GetMessage (&msg, (HWND)NULL, 0U, 0U))
{
    TranslateMessage (&msg) ;
    DispatchMessage (&msg) ;
}
return msg.wParam ;
}

```

```

long FAR PASCAL _export WndProc (HWND hwnd, UINT message, UINT wParam,
LONG lParam)

```

```

// ##15
// Funcție care tratează mesajele ferestrei principale
//
{
    HDC hdc;
    static HANDLE hInstance;
    static HMENU hMenu;
    PAINTSTRUCT ps ;
    static int Index, Salt;
    static DWORD dwCustColors [16] ;
    static int i,j,R, G, B, x, y,k, IsCheck;
    static char Tmp[100];
    static int xMouse, yMouse;
    static FARPROC lpfnDespreDlgProc, lpfnDifuzieDlgProc;
    static RECT Rect;
    float Valoare;
    static COLORREF Culoare;
    static HWND hCoord1, hCoord2;
    static HINSTANCE hInstCoord;
    POINT Punct;
    static PALETTEENTRY ape[DimPaleta];
    static OFSTRUCT StrFile;

    switch (message)
    {
        case WM_CREATE :
            hInstance = ((LPCREATESTRUCT) lParam)->hInstance;
            lpfnDespreDlgProc = MakeProcInstance((FARPROC)DespreDlgProc, hInstance);
            lpfnDifuzieDlgProc = MakeProcInstance((FARPROC)DifuzieDlgProc, hInstance);
            R = 255; G = B = 0;
            Index = IsEtapa1;
            pObjPaleta = LocalAlloc(LPTR, sizeof(LOGPALETTE) + DimPaleta
                * sizeof(PALETTEENTRY));
            Paleta = (LOGPALETTE*)LocalLock(pObjPaleta);
            for(i = 0 ; i < DimPaleta ; i++)
            {
                ape[i].peRed = Paleta->palPalEntry[i].peRed = R;
                ape[i].peGreen = Paleta->palPalEntry[i].peGreen = G;
                ape[i].peBlue = Paleta->palPalEntry[i].peBlue = B;
                ape[i].peFlags = Paleta->palPalEntry[i].peFlags = PC_NOCOLLAPSE; //PC_RESERVED;
                switch (Index)
                {
                    case IsEtapa1:
                        G += IncPaleta;
                        if(G > 255)
                        {
                            Index = IsEtapa2;
                            G = 255;
                        }
                        break;
                    case IsEtapa2:
                        R -= IncPaleta * 2;
                        if(R < 0)
                        {
                            Index = IsEtapa3;
                            R = 0;
                        }
                        break;
                    case IsEtapa3:
                        B += IncPaleta * 2;
                        if(B > 255)
                        {
                            Index = IsEtapa4;

```

```

        B = 255;
    }
    break;
    case IsEtapa4:
        G -= IncPaleta;
    default ;;
    }
}
hdc = GetDC(hwnd);
Paleta->palVersion = 0x300;
Paleta->palNumEntries = DimPaleta;
if(hPal = CreatePalette(Paleta))
    {
        hPrev = SelectPalette(hdc, hPal, ForegroundPalette);
        j = RealizePalette(hdc);
    }
else
    MessageBox((HWND)NULL, "Nu se poate crea paleta","PALETA", MB_ICONSTOP ||
MB_OK);

    hCoord1 = FaceWindowCoord(hwnd,1);
    hCoord2 = FaceWindowCoord(hwnd,2);
    GetClientRect(hwnd, &Rect);
    SetViewportOrg(hdc, 0, Rect.bottom);
    SetMapMode(hdc, MM_LOMETRIC);
    FontGeneral.lpLogFont = &lfGeneral;
    FontElemente.lpLogFont = &lfElemente;
    FontNoduri.lpLogFont = &lfNoduri;
    FontGeneral.lpLogFont->lHeight = -27;
    FontElemente.lpLogFont->lHeight = -29;
    FontNoduri.lpLogFont->lHeight = -27;
    _fstrcpy(FontNoduri.lpLogFont->lFaceName, "Courier New");
    _fstrcpy(FontElemente.lpLogFont->lFaceName, "Courier New");
    _fstrcpy(FontGeneral.lpLogFont->lFaceName, "Courier New");
    FontNoduri.Flags = CF_INITTOLOGFONTSTRUCT | CF_SCREENFONTS |
CF_EFFECTS;
    FontElemente.Flags = CF_INITTOLOGFONTSTRUCT | CF_SCREENFONTS |
CF_EFFECTS;
    FontGeneral.Flags = CF_INITTOLOGFONTSTRUCT | CF_SCREENFONTS |
CF_EFFECTS;

    FontElemente.lpLogFont->lItalic = 1;
    hFontNoduri = FaceFont(&FontNoduri, hwnd);
    hFontGeneral = FaceFont(&FontGeneral, hwnd);
    hFontElemente = FaceFont(&FontElemente, hwnd);
    ReleaseDC(hwnd, hdc);
    return 0;
    case WM_KILLFOCUS:
        GetUpdateRect(hwnd, &Rect, 1);

        return 0;
    case WM_SETFOCUS:
        RedrawWindow(hwnd, &Rect, (HRGN)NULL, RDW_UPDATENOW | RDW_INVALIDATE | RDW_ERASE);

        return 0;
    case WM_PAINT: ;
        hdc = BeginPaint(hwnd, &ps);
        GetUpdateRect(hwnd, &Rect, 1);
        RedrawWindow(hwnd, &Rect, (HRGN)NULL, RDW_UPDATENOW | RDW_INVALIDATE | RDW_ERASE);
        EndPaint(hwnd, &ps);
        PostMessage(hwnd, WM_SIZE, 0, 0L);
        return 0;

        case WM_SIZE:
            if(pNodCap)
                {
                    SendMessage(hwnd, WM_COMMAND, ID_REALISEA2A, (LPARAM)0);
                    FaceScala(hwnd);
                }
            return 0;

        case WM_MOUSEMOVE:
            if(ofn.lpstrFileTitle == NULL)
                return 0;
            hdc = GetDC(hwnd);
            xMouse = LOWORD(lParam);
            yMouse = HIWORD(lParam);
            Punct.x = xMouse; Punct.y = yMouse;
            DPTtoLP(hdc, &Punct, 1);
            Culoare = GetPixel(hdc, Punct.x, Punct.y);
            Valoare = ReturnSolutie(hdc, Culoare);
            if(Valoare == FLT_MIN)
                _fstrcpy(Tmp, "Val: ***");
            else
                sprintf(Tmp, "Val: %.6f", Valoare);
            SendMessage(hCoord1, WM_USER, (WORD)_fstrlen(Tmp), (DWORD)Tmp);

```

```

Sc + ymin);
    sprintf(Tmp, "X: %6f | Y: %6f  ", Punct.x / Sc + xmin, Punct.y /
    SendMessage(hCoord2, WM_USER, _fstrlen(Tmp), (DWORD)Tmp);
    ReleaseDC(hwnd, hdc);
    return 0;
break;
case WM_COMMAND :
    hMenu = GetMenu(hwnd);
    switch(wParam)
    {
        case ID_DESCHIDE:
            PopFileInitialize(hwnd, FALSE);
            if(GetOpenFileName(&ofn))
            {
                EnableMenuItem(hMenu, ID_DESCHIDE, MF_GRAYED);
                lstrcpy((LPSTR)Tmp, (LPCSTR)ofn.lpstrFileName);
                ReadPostFile(Tmp, hwnd);
                FaceScala(hwnd);
                CheckMenuItem(hMenu, (UINT)ID_SOLUTIE, MF_CHECKED);
                PlotStruct(hwnd, IsRender, Reface);
            }
            break;
        case ID_IESIRE:
            SendMessage(hwnd, WM_CLOSE, 0, 0L);
            break;
        case ID_MARCATOR_NOD:;
            IsCheck = GetMenuState(hMenu, (UINT)ID_MARCATOR_NOD, MF_BYCOMMAND) & MF_CHECKED;
            CheckMenuItem(hMenu, (UINT)ID_MARCATOR_NOD, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED
: MF_CHECKED));
            if(!IsCheck)
                SctieNoduri(hwnd, GetMenuState(hMenu, (UINT)ID_NUMAR_NOD, MF_BYCOMMAND) &
MF_CHECKED);
            else
            {
                CheckMenuItem(hMenu, (UINT)ID_NUMAR_NOD, MF_UNCHECKED);
                SendMessage(hwnd, WM_COMMAND, (UINT)ID_REAFISEAZA, 0L);
            }
            break;
        case ID_NUMAR_NOD:;
            IsCheck = GetMenuState(hMenu, (UINT)ID_NUMAR_NOD, MF_BYCOMMAND) & MF_CHECKED;
            CheckMenuItem(hMenu, (UINT)ID_NUMAR_NOD, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED :
MF_CHECKED));
            CheckMenuItem(hMenu, (UINT)ID_MARCATOR_NOD, MF_CHECKED);
            if(!IsCheck)
                SctieNoduri(hwnd, 1);
            else
                SendMessage(hwnd, WM_COMMAND, (UINT)ID_REAFISEAZA, 0L);
            break;
        case ID_NODURI_FONT:;
            ChooseFont(&FontNoduri);
            DeleteObject(hFontNoduri);
            hFontNoduri = FaceFont(&FontNoduri, hwnd);
            if(GetMenuState(hMenu, (UINT)ID_NUMAR_NOD, MF_BYCOMMAND) & MF_CHECKED)
                SendMessage(hwnd, WM_COMMAND, (UINT)ID_REAFISEAZA, 0L);
            break;
        case ID_CONTUR_ELEMENT:;
            IsCheck = GetMenuState(hMenu, (UINT)ID_CONTUR_ELEMENT, MF_BYCOMMAND) & MF_CHECKED;
            CheckMenuItem(hMenu, (UINT)ID_CONTUR_ELEMENT, MF_BYCOMMAND | (IsCheck ?
MF_UNCHECKED : MF_CHECKED));
            if(!IsCheck)
                FaceContur(hwnd, GetMenuState(hMenu, (UINT)ID_NUMAR_ELEMENT, MF_BYCOMMAND)
& MF_CHECKED);
            else
            {
                CheckMenuItem(hMenu, (UINT)ID_NUMAR_ELEMENT, MF_UNCHECKED);
                SendMessage(hwnd, WM_COMMAND, (UINT)ID_REAFISEAZA, 0L);
            }
            break;
        case ID_NUMAR_ELEMENT:;
            IsCheck = GetMenuState(hMenu, (UINT)ID_NUMAR_ELEMENT, MF_BYCOMMAND) & MF_CHECKED;
            CheckMenuItem(hMenu, (UINT)ID_NUMAR_ELEMENT, MF_BYCOMMAND | (IsCheck ?
MF_UNCHECKED : MF_CHECKED));
            CheckMenuItem(hMenu, (UINT)ID_CONTUR_ELEMENT, MF_CHECKED);
            if(!IsCheck)
                FaceContur(hwnd, 1);
    }
}

```

```

else
    SendMessage(hwnd, WM_COMMAND, (UINT)ID_REAFISEAZA, 0L);
    break;

    case ID_ELEMENTE_FONT:;
    ChooseFont(&FontElemente);
    DeleteObject(hFontElemente);
    hFontElemente = FaceFont(&FontElemente, hwnd);
    if(GetMenuState(hMenu, (UINT)ID_NUMAR_ELEMENT, MF_BYCOMMAND) & MF_CHECKED)
        SendMessage(hwnd, WM_COMMAND, (UINT)ID_REAFISEAZA, 0L);

    break;

    case ID_DINAMIC:;
    IsCheck = GetMenuState(hMenu, (UINT)ID_DINAMIC, MF_BYCOMMAND) & MF_CHECKED;
    CheckMenuItem(hMenu, (UINT)ID_DINAMIC, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED :
MF_CHECKED));
    if(!IsCheck)
    {
        hCoord1 = FaceWindowCoord(hwnd, 1);
        hCoord2 = FaceWindowCoord(hwnd, 2);
    }
    else
    {
        SendMessage(hCoord1, WM_CLOSE, 0, 0L);
        SendMessage(hCoord2, WM_CLOSE, 0, 0L);
    }
    break;

    case ID_DIFUZIE_RETEA:
        x = OrdinFinal;
        DialogBox(hInstance, "DialogDifuzie", hwnd,
lpfnDifuzieDlgProc);
        if(x != OrdinFinal)
            SendMessage(hwnd,
WM_COMMAND, (UINT)ID_REAFISEAZA, Reface);
        break;

    case ID_REAFISEAZA:
        PlotStruct(hwnd, (int)GetMenuState(hMenu,
(UINT)ID_SOLUTIE, MF_BYCOMMAND)
        & MF_CHECKED, (int)lParam);
        if(GetMenuState(hMenu, (UINT)ID_CONTUR_ELEMENT, MF_BYCOMMAND) & MF_CHECKED)
            FaceContur(hwnd, GetMenuState(hMenu, (UINT)ID_NUMAR_ELEMENT, MF_BYCOMMAND)
& MF_CHECKED);
        if(GetMenuState(hMenu, (UINT)ID_MARCATOR_NOD, MF_BYCOMMAND) & MF_CHECKED)
            ScrieNoduri(hwnd, GetMenuState(hMenu, (UINT)ID_NUMAR_NOD, MF_BYCOMMAND) &
MF_CHECKED);
        break;

    case ID_SOLUTIE:;
    IsCheck = GetMenuState(hMenu, (UINT)ID_SOLUTIE, MF_BYCOMMAND) & MF_CHECKED;
    CheckMenuItem(hMenu, (UINT)ID_SOLUTIE, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED :
MF_CHECKED));
    SendMessage(hwnd, WM_COMMAND, (UINT)ID_REAFISEAZA, lParam);

    break;

    case ID_DESPREPROGRAM:
        DialogBox(hInstance, "DialogDespre", hwnd,
lpfnDespreDlgProc);
        break;

    case ID_INFORMATII:
        MessageBox((HWND)NULL,
(LPCSTR)"Acesta este un program de uz intern\nPentru detalii luați legătura cu realizatorul",
(LPCSTR)"Info", MB_ICONINFORMATION |
MB_OK);

    }

    return 0;
case WM_CLOSE :
    Dealloca();
    if(hPal)
        DeleteObject(hPal);
        DeleteObject(hFontGeneral);
        DeleteObject(hFontElemente);
        DeleteObject(hFontNoduri);
        DestroyWindow(hwnd);
    return 0;
case WM_DESTROY:
    PostQuitMessage (0) ;
    return 0 ;
}
return DefWindowProc (hwnd, message, wParam, lParam) ;
}

```

```

BOOL FAR PASCAL _export DifuzieDlgProc (HWND hDlg, UINT message, UINT wParam,
LONG lParam)
// ##16
// Funcție control dialog parametru difuzie pentru triunghiuri render
//
{
    switch (message)
    {
        case WM_INITDIALOG:
            WriteMemoryToItem(hDlg , IDC_DIFUZIE , &OrdinFinal , IntType , 10);
            return TRUE ;
        case WM_COMMAND:
            switch (wParam)
            {
                case IDOK:
                    ReadItemToMemory(hDlg , IDC_DIFUZIE, 2 , &OrdinFinal , IntType ,
10);
                    if(OrdinFinal > 6)
                        OrdinFinal = 6;
                    else
                        if(OrdinFinal < 0)
                            OrdinFinal = 0;
                    EndDialog (hDlg, 0) ;
                    return TRUE ;
                case IDCANCEL:
                    EndDialog (hDlg, 0) ;
                    return TRUE ;
                    break;
            }
            break ;
    }
    return FALSE ;
}

void Dealoca(void)
// ##17
// Funcție care dealocă memoria pentru dat citite
//
{
    pNodNou = pNodCurent = pNodCap;
    pElementNou = pElementCurent = pElementCap;

    while(pNodCurent)
    {
        pNodNou = pNodCurent->Urm;
        LocalUnlock(pNodCurent->pObj);
        LocalFree(pNodCurent->pObj);
        pNodCurent = pNodNou;
    }
    pNodNou = pNodCurent = pNodCap = NULL;

    while(pElementCurent)
    {
        pElementNou = pElementCurent->Urm;
        LocalUnlock(pElementCurent->pObj);
        LocalFree(pElementCurent->pObj);
        pElementCurent = pElementNou;
    }
    pElementNou = pElementCurent = pElementCap = NULL;
}

```

Listă cu funcții folosite în program, care sunt analoage cu cele din anexa 4:

```

18.void Zero(void *Addr , UINT Lg)
19.void ReadItemToMemory(HWND hDlg , UINT Item ,int Lg ,void *Addr , int DataType
, int Base)
20.void WriteMemoryToItem(HWND hDlg , UINT Item ,void *Addr , int DataType , int
Base)
21.void EroareAlocare(char *Mesaj)
22.void InfoDisplay(HDC hdc, POINT *ret)
23.HWND FaceWindowCoord(HWND hwnd, int care)
24.long FAR PASCAL _export WndProcCoord (HWND hwnd, UINT message, UINT wParam,
25.BOOL FAR PASCAL _export DespreDlgProc (HWND hDlg, UINT message, UINT wParam,
LONG lParam)

```



## 10. ANEXA 3 - PROGRAM ACHIZIȚII DATE

### 10.1. CARACTERISTICI

- Programul PCL711 citește datele de pe placa de achiziții de date PCL711
- Format fișier executabil - aplicație Windows;
- Limbaj de programare C;
- Compilator folosit: Microsoft Visual C++1.5;
- structură de meniuri:

#### File⇒

New .....Inițializează programul pornind o nouă sesiune de măsurări  
 ReadConfig .....Citește structura de canale și parametri din fișier de configurare  
 WriteConfig .....Scrie structura de canale și parametri în fișier de configurare  
 Exit .....Închide program

#### Options⇒

General .....Deschide fereastra de control nr.5 din figura 2  
 Window1...Window8 Deschide câte o fereastră de control nr.3 din figura 2, pentru fiecare canal  
 AutoFileName .....Permite autoincrementarea numelui fișierului de ieșire după încheierea unei măsurători  
 Predefined configure Folosește configurări predefinite pentru două serii de măsurători consecutive diferite

#### ChannelConfig⇒

WatchChannel .....Deschide fereastra de control nr.4 din figura 2, de unde se selectează canalele urmărite  
 Channel0...Channel7 Deschide câte o fereastră de control nr.1 din figura 2, pentru fiecare canal

#### DataControl⇒

Go .....Lansează măsurarea efectivă  
 Pause .....Înterupe măsurătoarea  
 End .....Încheie măsurătoarea înainte de expirarea timpului de oprire automată  
 ClearData .....Șterge din memorie datele citite

#### CNC⇒

Parameters .....Deschide fereastra de control nr.2 din figura 2, de unde se poate controla valoarea razei, lungimii de așchiere și a avansului, care sunt înscrise în fișierul PROG.CNC; acesta conține instrucțiuni pentru comanda unui ciclu de strunjire cilindrică exterioară  
 SendProgramme.....Trimite programul din fișierul PROG.CNC la mașina uncală, pe interfața serială COM4.

#### Help ⇒

Despre program .....Afișează fereastră cu informații despre program

#### Fișiere de intrare

Fișierul cu programul CNC:

format : ASCII

extensia implicită : CNC

conține următorul program de prelucrare:

%	G29 LV21=.2	X35
G0 G71	G95	G94
G29 T2002 X-60 Z-5	G0 X40. Z1	T0
T0	G1 Z-340.5	G0 X0 Z0
G0 X0 Z0	G91	M5
M3	G0 X1 Z1	G29 E
T2	G90	%

*Fișierul cu configurarea:*

format : binar

extensia implicită : CFG

conține informații despre structura canalelor măsurate și parametrii lor, la un moment dat;

**Fișiere de ieșire**

*Fișierul cu datele măsurate:*

format : ASCII

extensia implicită : DAT

Structură: conține valorile digitale preluate de pe placa PCL711 în următorul format:

Timp	Ch0	Ch1	Ch2	Ch3	Ch4	Ch5	.....
0.0000	2409	1249	1193	4048	3920	1101	
0.1000	2426	1249	1190	4043	3916	1112	
0.2000	2406	1264	1198	4047	3907	1108	
0.3000	2418	1247	1175	4037	3925	1108	
0.4000	2425	1253	1192	4045	3927	1101	

.....

În figura 1 se poate observa o structură de urmărire pe 6 canale, fiecare reprezentat în câte o fereastră:

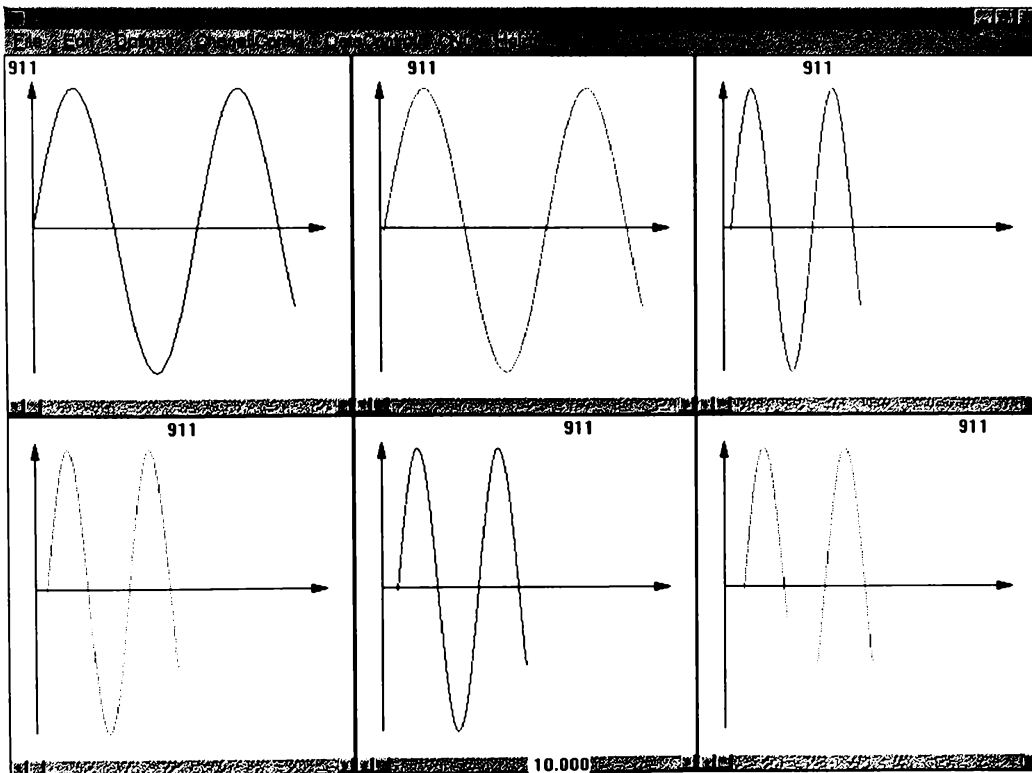


Fig. 1 - Structură de ferestre pentru 6 canale la programul PCL711

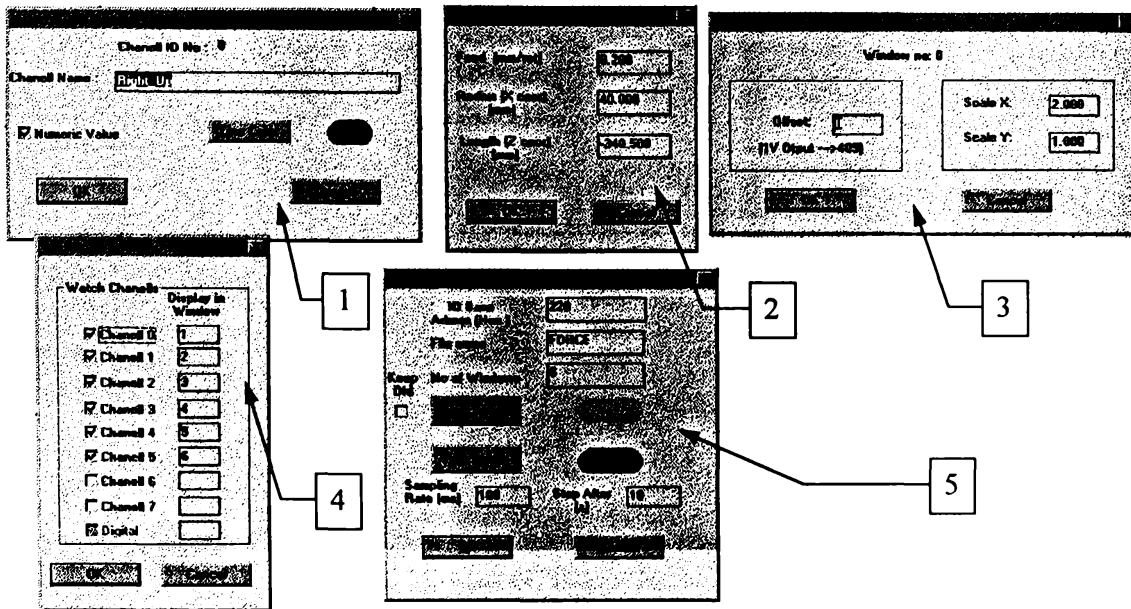


Fig. 2 - Ferestre de control în programul PCL711

## 10.2. FIȘIER HEADER

```

#define MaxDigi                0x00FFFU
#define ID_TIMER                1
#define ID_TIMER_COM4          2
#define DataReadyFlag          16 << 8
#define PCL711MaxChanell       7
#define AppShortName            "Acquisition Program from PCL711 card"
#define Margin                  20
#define TIME_OUT                32768
#define CONTINUOUS              1
#define LG_SIR_TEMP             50
#define Up                       1
#define Right                    2
#define PI                       3.14159265

#define IntType                  100
#define LongType                 101
#define FloatType                102
#define DoubleType               103
#define CharType                 104
#define MaxUInt                  0xFFFFU

#define FeedID                   "LV21"
#define RadiusID                 "GO X"
#define LengthID                 "G1 Z"
#define ConfigNo                 1000
#define ConfigForce              1001
#define ConfigProfile            1002

struct PCL711_WINDOW
{
    double      ScX;
    double      ScY;
    int         Offset;
};
typedef struct PCL711_WINDOW PCL711_WINDOW_TYPE;

struct CHANELL
{
    char        ChName[20];
    int         LineType;
    COLORREF    LineColour;
    BOOL        DisplayNumeric;
    HPEN        hPen;
    int         far *Data;
    int         far *Last;
};

typedef struct CHANELL CHANELLTYPE;
    
```

## 10.3. FIȘIER SURSĂ

```

/*-----
PCL711.C -- Acquisition Program from PCL711 card
© Adrian ROSCA, 1996
Version 1.0
-----*/
#include <stdio.h>
#include <stdlib.h>
#include <windows.h>
#include <commdlg.h>
#include <math.h>
#include <bios.h>
#include <dos.h>
#include "resource.h"
#include "pcl711.h"

static PCL711_WINDOW_TYPE
    DefaultPCL711Window =
    {
        1.0,
        1.0,
        0
    };

static CHANELLTYPE
    DefaultChanell =
    {
        "Chanell ",
        CONTINUOUS,
        RGB(0, 0, 0),
        TRUE,
        (HFEN)NULL,
        NULL,
        NULL
    };

int DSamplingRate = 100 , DMaxChanell = 1 , DMaxWindows = 1 , DStopAfter = 60 ;
int SamplingRate , MaxChanell , MaxWindows , StopAfter , PredefinedConfig;
int ChanellStatus[PCL711MaxChanell + 1], Y, Treceeri = 1;
char *DFilename = "PCL" , FileName[9] , CompleteFN[15], *HeaderCfg = "CONFIGURATION FILE PCL711";
char PathToData[128];
char ApplicationName[100];
char pTemp[2048];
double lTime, GraphScale;
static HWND hwndChild[PCL711MaxChanell + 1];
char szChildClass[] = "Chanell_Child";
static CHOSECOLOR    BackColour , ForeColour , Colour;
static DWORD        dwCustColorsFore[16] , dwCustColorsBack[16] , dwCust[16];
int Curent;

COLORREF    GlobalBackColour = RGB(220 , 220 , 220),
            GlobalForeColour = RGB(0, 0, 0);

UINT DPCLPort = 0x220;
UINT PCLPort , LowData , HighData , ScanChanell , SoftTrigger , DigitalChanell;
BOOL IS_DRAWING;

BOOL FAR PASCAL _export AboutDlgProc (HWND , UINT , UINT ,LONG) ;
BOOL FAR PASCAL _export IndividualDlgProc(HWND , UINT , UINT ,LONG) ;
BOOL FAR PASCAL _export GeneralDlgProc(HWND , UINT , UINT ,LONG) ;
BOOL FAR PASCAL _export WatchDlgProc(HWND , UINT , UINT ,LONG) ;
BOOL FAR PASCAL _export WindowsDlgProc(HWND , UINT , UINT ,LONG) ;
BOOL FAR PASCAL _export CNCProc(HWND , UINT , UINT ,LONG) ;
BOOL FAR PASCAL _export ConfigProc(HWND , UINT , UINT ,LONG) ;
long FAR PASCAL _export WndProc (HWND , UINT , UINT , LONG) ;
long FAR PASCAL _export ChildWndChanell (HWND , UINT , UINT , LONG) ;
void far *_fmemset(void * , int , size_t);
void far *_fmemcpy(void * , void * , size_t);
int _outp(unsigned , int);
unsigned _inpw(unsigned);
unsigned _bios_serialcom(unsigned , unsigned , unsigned);

HANDLE hInst;
OPENFILENAME ofn;

PCL711_WINDOW_TYPE Windows[PCL711MaxChanell + 1];
CHANELLTYPE Chanells[PCL711MaxChanell + 1];

```

```

void MakeFN(int Nr , LPSTR Name)
// ##1
// Construiește nume fișier ieșire
//
{
    LPSTR Ext = ".TXT", s1;
    char FN[15];
    LPSTR i;

    i = lstrcpyn((LPSTR)FN , (LPCSTR)Name , 6);
    s1 = (LPSTR)FN;
    while(*s1 != '\\0')
        s1++;
    itoa(Treceri++, s1 , 10);
    if(lstrlen(FN) > 8)
        FN[7] = '\\0';
    lstrcat((LPSTR)FN , Ext);
    lstrcpy(CompleteFN , FN);
}

void ShowColor(HWND hwnd , COLORREF Col , POINT P1 , POINT P2)
// ##2
// Trasează linie colorată pt selecție foreground și background
//
{
    HDC hdc;
    HPEN hp, hPrev;

    hdc = GetDC(hwnd);
    hp = CreatePen(PS_SOLID , 25 , Col);
    hPrev = SelectObject(hdc, hp);
    MoveTo(hdc , P1.x , P1.y);
    LineTo(hdc , P2.x , P2.y);
    SelectObject(hdc, hPrev);
    DeleteObject(hp);
    ReleaseDC(hwnd , hdc);
}

void MakeChildWindow(HWND hwnd , BOOL Status , BOOL ClearWindow)    ///## 5
// ##3
// Face ferestre child unde se afișează graficele pt fiecare canal
//
{
    div_t Temp;
    int i , NoChanellsWidth;
    RECT WindowRectangle;
    int xWindExt, yWindExt;
    int OriginX , OriginY , SizeX , SizeY;

    Temp = div(MaxWindows , 2);
    NoChanellsWidth = Temp.quot ? Temp.quot : 1;
    if (Temp.rem)
        NoChanellsWidth ++;
    GetClientRect(hwnd , &WindowRectangle);
    xWindExt = WindowRectangle.right - WindowRectangle.left;
    yWindExt = WindowRectangle.bottom - WindowRectangle.top;
    SizeX = MaxWindows > 1 ? xWindExt / NoChanellsWidth : xWindExt;
    SizeY = MaxWindows > 1 ? yWindExt / 2 : yWindExt;
    OriginX = 0;
    OriginY = 0;
    for(i = 0 ; i <= PCL711MaxChanell ; i++)
        if(hwndChild[i])
            SendMessage(hwndChild[i] , WM_CLOSE , 0 , 0L);
    Zero(hwndChild , sizeof(hwndChild));
    if(Status)
        Zero(ChanellStatus , sizeof(ChanellStatus));
    for(i = 0; i < MaxWindows ; i++)
    {
        hwndChild[i] = CreateWindow(szChildClass, NULL, WS_CHILDWINDOW | WS_VISIBLE |
            WS_BORDER | WS_HSCROLL ,OriginX , OriginY , SizeX , SizeY , hwnd , i ,
            GetWindowWord(hwnd , GWW_HINSTANCE), NULL) ;
        if(ClearWindow)
            _fmemcpy(&Windows[i] , &DefaultPCL711Window , sizeof(PCL711_WINDOW_TYPE));
        if ((i + 1) == NoChanellsWidth)
        {
            OriginX = 0;
            OriginY += SizeY;
        }
    }
    else
}

```

```

        OriginX += SizeX;
    };

}

void MakeAddressIO(UINT Base)
// ##4
// Construiește adresele de lucru pt. Dialogul cu placa
//
{
    LowData = Base + 4;
    HighData = Base + 5;
    DigitalChanell = Base + 6;
    ScanChanell = Base + 10;
    SoftTrigger = Base + 12;
}

void InitData(HWND hwnd)
// ##1
// Inițializează structurile pentru canale și ferestre
//
{
    int i;
    HMENU hmenu;
    char S1[4];

    SamplingRate = DSamplingRate;
    MaxChanell = DMaxChanell;
    MaxWindows = DMaxWindows;
    StopAfter = DStopAfter;
    PCLPort = DPCLPort;
    Treceeri = 1;
    PredefinedConfig = ConfigNo;
    lstrcpy(FileName , DFileName);
    MakeFN(Treceeri , (LPSTR)FileName);
    MakeAddressIO(PCLPort);
    for(i = 0 ; i <= PCL711MaxChanell ; i++)
    {
        if(Chanells[i].hPen)
            DeleteObject(Chanells[i].hPen);
        Chanells[i] = DefaultChanell;
        itoa(i , S1 , 10);
        lstrcat(Chanells[i].ChName , (LPCSTR)S1);
        Chanells[i].hPen = CreatePen(PS_SOLID , 1 , GlobalForeColour);
        Chanells[i].LineColour = GlobalForeColour;
        _fmemcpy(&Windows[i] , &DefaultPCL711Window , sizeof(PCL711_WINDOW_TYPE));
    }
    hmenu = GetSubMenu(GetMenu(hwnd), 2);
    EnableMenuItem(hmenu , ID_WINDOW1 , MF_ENABLED );
    for(i = 1 ; i <= PCL711MaxChanell ; i++)
        EnableMenuItem(hmenu , (UINT)(ID_WINDOW1 + i) , MF_GRAYED);

    hmenu = GetSubMenu(GetMenu(hwnd), 3);
    for(i = 0; i <= PCL711MaxChanell ; i++)
        EnableMenuItem(hmenu , (UINT)(ID_CHANELLO + i) , MF_GRAYED);
}

int PASCAL WinMain (HANDLE hInstance, HANDLE hPrevInstance, LPSTR lpszCmdLine, int
nCmdShow)
// ##5
// Inițializează structurile pentru canale și ferestre
//
{
    static char szAppName[] = "PCL711" ;
    static char szMenuName[] = "PCL711";
    static char szIcoName[] = "PCL711Icon";

    HWND        hwnd ;
    MSG         msg ;
    WNDCLASS    wndclass ;

    if (!hPrevInstance)
    {
        wndclass.style          = CS_HREDRAW | CS_VREDRAW ;
        wndclass.lpfnWndProc    = WndProc ;
        wndclass.cbClsExtra     = 0 ;
        wndclass.cbWndExtra     = 0 ;
        wndclass.hInstance     = hInstance ;
        wndclass.hIcon         = LoadIcon (hInstance, szIcoName ) ;
    }
}

```

```

    wndclass.hCursor      = LoadCursor ((HINSTANCE)NULL, IDC_ARROW) ;
    wndclass.hbrBackground = GetStockObject (WHITE_BRUSH) ;
    wndclass.lpszMenuName = szMenuName;
    wndclass.lpszClassName = szAppName ;

    RegisterClass (&wndclass) ;

    wndclass.lpfWndProc   = ChildWndChanell ;
    wndclass.lpszMenuName = NULL;
    wndclass.cbWndExtra   = sizeof(int);
    wndclass.hIcon        = (HICON)NULL ;
    wndclass.hbrBackground = CreateSolidBrush(GlobalBackColour);
    wndclass.lpszClassName = szChildClass ;

    hInst = hInstance;
    RegisterClass (&wndclass) ;

}
hInst = hInstance;
lstrcpy(ApplicationName , AppShortName);
hwnd = CreateWindow (szAppName, ApplicationName ,
    WS_MINIMIZEBOX, WS_OVERLAPPEDWINDOW, CW_USEDEFAULT, CW_USEDEFAULT, CW_USEDEFAULT,
    CW_USEDEFAULT, (HWND)NULL, (HMENU)NULL, hInstance, NULL) ;
ShowWindow (hwnd, SW_SHOWMAXIMIZED) ;
UpdateWindow (hwnd) ;
while (GetMessage (&msg, (HWND)NULL, 0, 0))
{
    TranslateMessage (&msg) ;
    DispatchMessage (&msg) ;
}
return msg.wParam ;
}

void CalcScale(void)
// ##6
// Cacluculează scara de reprezentare
//
{
    RECT rect;

    GetClientRect(hwndChild[0] , &rect);
    GraphScale = (double)(rect.bottom - 2 * Margin) / MaxDigi;
    Y = rect.bottom / 2;
}

void SetWindowName(HWND hwnd)
{
// ##7
// Atașează nume la o fereastră
//
    lstrcpy(ApplicationName , AppShortName);
    lstrcat(ApplicationName , " \273 ");
    lstrcat(ApplicationName , CompleteFN);
    SetWindowText(hwnd , ApplicationName);
}

void ReadConfig(char *ConfigName , HWND hwnd)
// ##8
// Citește structura de canale urmărite și parametri ce le caracterizează din fișierul
// de configurare
//
{
    HFILE PFile;
    int i, iTmp;
    HMENU hmenu;
    char Tmp[50];

    iTmp = PredefinedConfig;
    Zero(Tmp , sizeof(Tmp));
    PFile = _lopen(ConfigName, READ);

    _lread(PFile , Tmp , strlen(HeaderCfg));
    if(lstrcmp(HeaderCfg , Tmp))
    {
        MessageBox(hwnd, "Not Configuration File !",
            "PCL711", MB_ICONEXCLAMATION | MB_OK);
        _lclose(PFile);
        return;
    }
    InitData(hwnd);
    MakeChildWindow(hwnd , TRUE, TRUE);
    for(i = 0 ; i <= PCL711MaxChanell ; i++)

```



```

        DeleteObject(Chanells[i].hPen);
        _hread(PFile , Chanells , sizeof(Chanells));
        _hread(PFile , ChanellStatus , sizeof(ChanellStatus));

        _hread(PFile , Windows , sizeof(Windows));
        _hread(PFile , &SamplingRate , sizeof(SamplingRate));

        _hread(PFile , &MaxChanell , sizeof(MaxChanell));
        _hread(PFile , &MaxWindows , sizeof(MaxWindows));
        _hread(PFile , &StopAfter , sizeof(StopAfter));
        _hread(PFile , &GlobalBackColour , sizeof(GlobalBackColour));
        _hread(PFile , &GlobalForeColour , sizeof(GlobalForeColour));
        _hread(PFile , &PCLPort , sizeof(PCLPort));
        _hread(PFile , &BackColour , sizeof(CHOOSSECOLOR));
        _hread(PFile , dwCustColorsBack , sizeof(dwCustColorsBack));
        _hread(PFile , &ForeColour , sizeof(CHOOSSECOLOR));
        _hread(PFile , dwCustColorsFore , sizeof(dwCustColorsFore));
        _hread(PFile , &Colour , sizeof(CHOOSSECOLOR));
        _hread(PFile , dwCust , sizeof(dwCust));
        _hread(PFile , PathToData , LG_SIR_TEMP);
        _hread(PFile , CompleteFN , sizeof(CompleteFN));
        _hread(PFile , FileName , sizeof(FileName));
        _hread(PFile , &Treceri , sizeof(Treceri));
        _lclose(PFile);
        For(i = 0 ; i <= PCL711MaxChanell ; i++)
        {
            Chanells[i].hPen = CreatePen(PS_SOLID , 1 , Chanells[i].LineColour);
            hmenu = GetSubMenu(GetMenu(hwnd) , 2);
            EnableMenuItem(hmenu , (UINT)(ID_WINDOW1 + i) ,
                (UINT)( i < MaxWindows ? MF_ENABLED : MF_GRAYED));
            hmenu = GetSubMenu(GetMenu(hwnd) , 3);
            if(ChanellStatus[i])
                EnableMenuItem(hmenu , (UINT)(ID_CHANELLO + i) ,MF_ENABLED);
        }
        DeleteObject(SetClassWord(hwndChild[0] , GCW_HBRBACKGROUND ,
            CreateSolidBrush(GlobalBackColour)));
        MakeChildWindow(hwnd , FALSE, FALSE);
        MakeFN(Treceri , (LPSTR)FileName);
        SetWindowName(hwnd);
        PredefinedConfig = iTmp;
    }

long FAR PASCAL _export WndProc (HWND hwnd, UINT message, UINT wParam, LONG lParam)
// ##9
// Funcția de control a ferestrei principale a programului
//
{
    HDC hdc;
    static FARPROC lpfnAboutDlgProc, lpfnIndividualDlgProc, lpfnGeneralDlgProc, lpfnWatchDlgProc,
        lpfnWindowsDlgProc , lpfnCNC , lpfnConfigProc;

    static HANDLE hInstance;
    static HMENU hMenu;
    PAINTSTRUCT ps ;
    double Ftmp;
    static short nCount ;
    static char Sir[LG_SIR_TEMP];
    OFSTRUCT StrFile;
    static int CurentWindow = 1 , DataValue;
    static int Value[PCL711MaxChanell + 1];
    static char Mes[25] = "Can't open file " , *s1, Tmp[30];
    static double RealDimension;
    static HFILE PFile , PCNCFile;
    static DWORD dwCustColors [16] ;
        HMENU hmenul , hmenu;
        static UINT DataDimension;
        BOOL IsCheck;
        LPSTR pl;
    static int i, TimeInc;
    BOOL DataNotReady ;
    static HGLOBAL HGPOinter[PCL711MaxChanell + 1];
    UINT Index;
    int *PSaveChanells[PCL711MaxChanell], Lg, Rez;
    static char OutString[132] , *PString , Format[20];
    int CurentChanell = 0;
    static RECT rect;
    static int ComPort = 3;
        static POINT P, OldP[PCL711MaxChanell + 1];
        static DWORD rr;

    switch (message)
    {
        case WM_CREATE :
            InitData(hwnd);

```

```

SetWindowName(hwnd);
    CurentChanell = 0;
hInstance = ((LPCREATESTRUCT) lParam)->hInstance;
lpfnAboutDlgProc = MakeProcInstance ((FARPROC) AboutDlgProc, hInstance);
lpfnIndividualDlgProc = MakeProcInstance ((FARPROC) IndividualDlgProc, hInstance);
lpfnGeneralDlgProc = MakeProcInstance ((FARPROC) GeneralDlgProc, hInstance);
lpfnWatchDlgProc = MakeProcInstance ((FARPROC) WatchDlgProc, hInstance);
lpfnWindowsDlgProc = MakeProcInstance ((FARPROC) WindowsDlgProc, hInstance);
    lpfnCNC = MakeProcInstance ((FARPROC) CNCProc, hInstance);
    lpfnConfigProc = MakeProcInstance ((FARPROC) ConfigProc, hInstance);
    MakeChildWindow(hwnd, TRUE, TRUE);
return 0;
case WM_PAINT: ;
    hdc = BeginPaint (hwnd, &ps);
    EndPaint (hwnd, &ps);
    return 0;
case WM_SIZE: ;
    MakeChildWindow(hwnd, FALSE, FALSE);
    CalcScale();
    GetClientRect(hwnd, &rect);
    return 0;
case WM_TIMER: ;
    switch(wParam)
    {
        case ID_TIMER: ;
            for(i = 0; i <= PCL711MaxChanell; i++)
            {
                if(ChanellStatus[i])
                {
                    _outp(ScanChanell, i); // Set Chanell No for reading
                    _outp(SoftTrigger, 0); // Set Software Triger to No
                    DataNotReady = TRUE;
                    do
                    {
                        DataValue = _inpw(LowData); // Read data
                        DataNotReady = DataValue & DataReadyFlag;
                    }
                    while(DataNotReady);
                    hdc = GetDC (hwnd);
                    sprintf(Sir, "%7.3f", lTime);
                    SelectObject(hdc, GetStockObject(BLACK_PEN));
                    TextOut(hdc, (rect.left + rect.right) / 2, rect.bottom - 15, (LPSTR)Sir, strlen((LPSTR)Sir));
                    ReleaseDC (hwnd, hdc);
                    sprintf(Sir, "%-5d", DataValue);
                    P.x = (int)(TimeInc * Windows[ChanellStatus[i] - 1].ScX + 3 * i + Margin);
                    *Chanells[i].Last = P.x;
                    Chanells[i].Last++;
                    *Chanells[i].Last = DataValue;
                    Chanells[i].Last++;
                }
            }
            SendMessage(hwndChild[0], WM_PAINT, 0, 0L);
            TimeInc++;
            lTime += (double)SamplingRate / 1000;
            if(lTime > (double)StopAfter)
            {
                SendMessage(hwnd, WM_COMMAND, ID_END, 0L);
                MessageBox(hwnd, "Automatic Stop !", "PCL711", MB_ICONEXCLAMATION |
                MB_OK);
            }
            break;
        case ID_TIMER_COM4: ;
            break;
    }
    return 0;
case WM_COMMAND: ;
    hMenu = GetMenu(hwnd);
    switch(wParam)
    {
        case ID_NEW: ;
            InitData(hwnd);
            MakeChildWindow(hwnd, TRUE, TRUE);
            IS_DRAWING = FALSE;
            break;
        case ID_READ: ;
            PopFileInitialize(hwnd, FALSE);
            if(GetOpenFileName(&ofn))
            {
                strcpy((LPSTR)Tmp, (LPCSTR)ofn.lpstrFileName);
                ReadConfig(Tmp, hwnd);
            }
            break;
        case ID_WRITE: ;
            PopFileInitialize(hwnd, TRUE);
    }

```

```

if(GetSaveFileName(&ofn))
{
    lstrcpy((LPSTR)Tmp, (LPCSTR)ofn.lpstrFileName);
    PFile = OpenFile(Tmp, &StrFile, OF_CREATE);
    lstrcpy((LPSTR)PathToData, (LPCSTR)ofn.lpstrFile, ofn.nFileOffset);
    _lwrite(PFile, HeaderCfg, lstrlen(HeaderCfg));
    _lwrite(PFile, Chanells, sizeof(Chanells));
    _lwrite(PFile, ChanellStatus, sizeof(ChanellStatus));

    _lwrite(PFile, Windows, sizeof(Windows));
    _lwrite(PFile, &SamplingRate, sizeof(SamplingRate));
    _lwrite(PFile, &MaxChanell, sizeof(MaxChanell));
    _lwrite(PFile, &MaxWindows, sizeof(MaxWindows));
    _lwrite(PFile, &StopAfter, sizeof(StopAfter));
    _lwrite(PFile, &GlobalBackColour, sizeof(GlobalBackColour));
    _lwrite(PFile, &GlobalForeColour, sizeof(GlobalForeColour));
    _lwrite(PFile, &PCLPort, sizeof(PCLPort));
    _lwrite(PFile, &BackColour, sizeof(CHOOSSECOLOR));
    _lwrite(PFile, dwCustColorsBack, sizeof(dwCustColorsBack));
    _lwrite(PFile, &ForeColour, sizeof(CHOOSSECOLOR));
    _lwrite(PFile, dwCustColorsFore, sizeof(dwCustColorsFore));
    _lwrite(PFile, &Colour, sizeof(CHOOSSECOLOR));
    _lwrite(PFile, dwCust, sizeof(dwCust));
    _lwrite(PFile, PathToData, LG_SIR_TEMP);
    _lwrite(PFile, CompleteFN, sizeof(CompleteFN));
    _lwrite(PFile, FileName, sizeof(FileName));
    _lwrite(PFile, &Treceeri, sizeof(Treceeri));
    _lclose(PFile);
}
break;
case ID_EXIT :
SendMessage(hwnd, WM_CLOSE, 0, 0L);
break;
case ID_GENERAL :
DialogBox(hInstance, "DIA_GENERAL_CONFIG", hwnd, lpfnGeneralDlgProc);
break;
case ID_AUTO_FILE ;;
hmenu = GetSubMenu(GetMenu(hwnd), 2);
IsCheck = GetMenuState(hmenu, ID_AUTO_FILE, MF_BYCOMMAND) & MF_CHECKED;
CheckMenuItem(hmenu, ID_AUTO_FILE, MF_BYCOMMAND |
(IsCheck ? MF_UNCHECKED : MF_CHECKED));
Treceeri = 1;
MakeFN(Treceeri, (LPSTR)FileName);
SetWindowText(hwnd);
MessageBox(hwnd, CompleteFN, "New file name !", MB_ICONINFORMATION |
MB_OK);
break;
case ID_WINDOW8 : CurentWindow++;
case ID_WINDOW7 : CurentWindow++;
case ID_WINDOW6 : CurentWindow++;
case ID_WINDOW5 : CurentWindow++;
case ID_WINDOW4 : CurentWindow++;
case ID_WINDOW3 : CurentWindow++;
case ID_WINDOW2 : CurentWindow++;
case ID_WINDOW1 : ;
DialogBoxParam(hInstance, "DIA_WINDOWS", hwnd, lpfnWindowsDlgProc,
(LPARAM)CurentWindow);
CurentWindow = 1;
break;
case ID_PREDEFINED :
DialogBox(hInstance, "DIA_CONFIG", hwnd, lpfnConfigProc);
break;
case ID_WATCH_CHANELL :
i = DialogBox(hInstance, "DIA_WATCH_CHANELL", hwnd, lpfnWatchDlgProc);
break;
case ID_DIGITALCHANELL : CurentChanell++;

case ID_CHANELL7 : CurentChanell++;
case ID_CHANELL6 : CurentChanell++;
case ID_CHANELL5 : CurentChanell++;
case ID_CHANELL4 : CurentChanell++;
case ID_CHANELL3 : CurentChanell++;
case ID_CHANELL2 : CurentChanell++;
case ID_CHANELL1 : CurentChanell++;
case ID_CHANELL0 : ;
DialogBoxParam(hInstance, "DIA_INDIVIDUAL_CONFIG", hwnd, lpfnIndividualDlgProc,
(LPARAM)CurentChanell);
break;
case ID_GO :
Curent = 0;
DataNotReady = FALSE;
for(i = 0 ; i <= PCL711MaxChanell ; i++)
DataNotReady |= ChanellStatus[i];
if(!DataNotReady)

```

```

    {
        MessageBox(hwnd, "Select at least one ch.!" ,ApplicationName, MB_ICONSTOP |
MB_OK);
        break;
    }
    lTime = 0.0;
    TimeInc = 0;
    CalcScale();
    if((PFile = OpenFile(CompleteFN, &StrFile, OF_CREATE)) == HFILE_ERROR)
    {
        MessageBox(hwnd, (LPCSTR)lstrcat((LPSTR)Mes, (LPSTR)CompleteFN),
            ApplicationName, MB_ICONSTOP | MB_OK);
        return 0;
    }
    RealDimension = ((double)StopAfter * 1000 / SamplingRate) * sizeof(int) * 2 + 1;
    if(RealDimension > MaxUInt)
    {
        RealDimension = (double)SamplingRate * MaxUInt / 1000;
        StopAfter = (int)RealDimension - 2;
        DataDimension = MaxUInt;
    }
    else
        DataDimension = (int)RealDimension;
    for(i = 0; i <= PCL711MaxChanell; i++)
    {
        if(ChanellStatus[i])
        {
            HGPointer[i] = GlobalAlloc(GPTR, (DWORD)DataDimension);
            rr = GlobalSize(HGPointer[i]);
            Chanells[i].Data = (int far *)GlobalLock(HGPointer[i]);
            if(Chanells[i].Data == NULL)
            {
                MessageBox(hwnd, "Can't alloc memory", "PCL711",
                    MB_ICONEXCLAMATION | MB_OK);
                SendMessage(hwnd, WM_CLOSE, 0, 0);
            }
            Chanells[i].Last = Chanells[i].Data;
        }
        else
            Chanells[i].Last = Chanells[i].Data = NULL;
    }
    hmenu = LoadMenu(hInstance, "PCL711_GO");
    SetMenu(hwnd, hmenu);
    if (!SetTimer(hwnd, ID_TIMER, SamplingRate, NULL))
    {
        MessageBox(hwnd, "Too many timers, close another application!", "PCL711",
            MB_ICONEXCLAMATION | MB_OK);
        return FALSE;
    }
    SendMessage(hwnd, WM_COMMAND, ID_CLEAR, 0);
    hmenu = GetSubMenu(GetMenu(hwnd), 4);
    EnableMenuItem(hmenu, ID_GO, MF_GRAYED);
    EnableMenuItem(hmenu, ID_PAUSE, MF_ENABLED);
    CheckMenuItem(hmenu, ID_PAUSE, MF_BYCOMMAND | MF_UNCHECKED);
    EnableMenuItem(hmenu, ID_CLEAR, MF_GRAYED);
    IS_DRAWING = TRUE;
    break;
case ID_PAUSE :
    hmenu = GetSubMenu(GetMenu(hwnd), 4);
    IsCheck = GetMenuState(hmenu, ID_PAUSE, MF_BYCOMMAND) & MF_CHECKED;
    if (IsCheck)
    {
        IS_DRAWING = TRUE;
        CheckMenuItem(hmenu, ID_PAUSE, MF_BYCOMMAND | MF_UNCHECKED);
        if (!SetTimer(hwnd, ID_TIMER, SamplingRate, NULL))
        {
            MessageBox(hwnd, "Too many timers, close another application!", "PCL711",
                MB_ICONEXCLAMATION | MB_OK);
            return FALSE;
        }
        EnableMenuItem(hmenu, ID_END, MF_ENABLED);
    }
    else
    {
        IS_DRAWING = FALSE;
        CheckMenuItem(hmenu, ID_PAUSE, MF_BYCOMMAND | MF_CHECKED);
        KillTimer(hwnd, ID_TIMER);
        EnableMenuItem(hmenu, ID_END, MF_GRAYED);
    }
    break;
case ID_END : ;
    EnableMenuItem(hmenu, ID_END, MF_DISABLED);

    hmenu = LoadMenu(hInstance, "PCL711");

```

```

SetMenu(hwnd, hmenu);
KillTimer (hwnd, ID_TIMER) ;
hmenu = GetSubMenu(GetMenu(hwnd), 4);
EnableMenuItem(hmenu , ID_GO , MF_ENABLED);
EnableMenuItem(hmenu , ID_PAUSE , MF_GRAYED);
CheckMenuItem(hmenu, ID_PAUSE, MF_BYCOMMAND | MF_UNCHECKED) ;
EnableMenuItem(hmenu , ID_CLEAR , MF_ENABLED);
hmenu = GetSubMenu(GetMenu(hwnd), 3);
hmenu1 = GetSubMenu(GetMenu(hwnd), 2);

for(i = 0 ; i <= PCL711MaxChanell ; i++)
{
if(ChanellStatus[i])
EnableMenuItem(hmenu , (UINT)(ID_CHANELLO + i) , MF_ENABLED);
if(i < MaxWindows)
EnableMenuItem(hmenu1 , (UINT)(ID_WINDOW1 + i) , MF_ENABLED);
}
for(i = 0 ; i <= PCL711MaxChanell; i++)
if(ChanellStatus[i] && (Chanells[i].Data != NULL))
PSaveChanells[i] = Chanells[i].Data;
else
PSaveChanells[i] = NULL;
lstrcpy(Format , "%8d");
DataDimension /= sizeof(int);
FTmp = 0.0;
for(Index = 0 ; (Index < DataDimension) && (FTmp <= lTime); Index++)
{
PString = OutString;
sprintf(PString , "%8.4f" , FTmp);
PString = OutString + strlen(OutString);

FTmp += (double)SamplingRate / 1000;
for(i = 0 ; i <= PCL711MaxChanell; i++)
{
if(ChanellStatus[i])
{
PSaveChanells[i]++;
sprintf(PString , Format , *PSaveChanells[i]);
PString = OutString + strlen(OutString);
PSaveChanells[i]++;
}
}
lstrcat(OutString , "\015\012");
Lg = strlen(OutString);
i = _lwrite(PFile , OutString , Lg);
}
_lclose(PFile);
for(i = 0 ; i <= PCL711MaxChanell; i++)
{
if((ChanellStatus[i]) && (Chanells[i].Data != NULL))
{
Rez = GlobalUnlock(HGPointer[i]);
Rez = GlobalFree(HGPointer[i]);
}
}
hmenu = GetSubMenu(GetMenu(hwnd), 2);
IsCheck = GetMenuState(hmenu, ID_AUTO_FILE, MF_BYCOMMAND) & MF_CHECKED;
if(IsCheck)
{
MakeFN(Treceri , (LPSTR)FileName);
SetWindowName(hwnd);
}
IS_DRAWING = FALSE;
break;
case ID_CLEAR :
MakeChildWindow(hwnd , FALSE, FALSE);
break;
case ID_PARAMETERS
:
DialogBox(hInstance, "DIA_CNC", hwnd,lpfnCNC );
break;
case ID_EDITPROGRAM
:
break;
case ID_SENDPROGRAM
:
if((PCNCFile = _lopen("Prog.cnc" , READ)) == HFILE_ERROR)
MessageBox(hwnd , "Cannot open Prog1.cnc" , "Open CNC" , MB_ICONSTOP);
Zero(pTemp , sizeof(pTemp));
_lread(PCNCFile , pTemp , sizeof(pTemp));
p1 = pTemp;
ComPort = 3;
while(*p1)
{
i = *(char *)p1;
TimeInc = _bios_serialcom( _COM_SEND , ComPort , i );
}
}

```

```

        if(TimeInc & TIME_OUT)
        {
            MessageBox(hwnd, "Cannot send data to COM4:", "PCL711",
                MB_ICONSTOP | MB_OK);
            ComPort = 0;
            break;
        }
        pl++;
    }
    _lclose(PCNCFile);
    MessageBox(hwnd, "Transmission end", "PCL711", MB_ICONINFORMATION |
MB_OK);
        break;
    case ID_INFO : ;
        break;

    case ID_ABOUT : ;
        DialogBox(hInstance, "DIA_ABOUT", hwnd,lpfnAboutDlgProc );
        break;
    }
    break;
    return 0;

    case WM_CLOSE :
        for(i = 0 ; i <= PCL711MaxChanell ; i++)
            DeleteObject (Chanells[i].hPen);
        DestroyWindow(hwnd);
        return 0;

    case WM_DESTROY:
        DeleteObject (GetClassWord(hwnd , GCW_HBRBACKGROUND));
        PostQuitMessage (0) ;
        if(PFile)
            _lclose(PFile);
        return 0 ;
    }
    return DefWindowProc (hwnd, message, wParam, lParam) ;
}

```

```

double DegToRad(double Arg)

```

```

// ##10
// Transformă gradele în radiani
//
{
    double Rez;

    Rez = Arg * PI / 180.0;
    return Rez;
}

```

```

void Arrow(HDC hdc , POINT Origin , int tip, COLORREF Visible)

```

```

// ##11
// Face săgeată
//
{
    POINT p[3];
    int Height = 12, HalfWide;
    double Angle = 30.0 , Real;
    HBRUSH hbrGray, hbrPrevious;
    Angle = DegToRad(Angle) / 2;
    Real = tan(Angle);
    Real *= Height;
    HalfWide = (int)Real;
    p[0] = Origin;
    switch (tip)
    {
        case Up :
            p[1].x = Origin.x - HalfWide;
            p[1].y = Origin.y + Height;
            p[2].x = Origin.x + HalfWide;
            p[2].y = p[1].y;
            break;
        case Right :
            p[1].x = Origin.x - Height;
            p[1].y = Origin.y + HalfWide;
            p[2].x = p[1].x;
            p[2].y = Origin.y - HalfWide;
            break;
    }
    hbrGray = CreateSolidBrush(Visible);
    hbrPrevious = SelectObject(hdc, hbrGray);
    SetPolyFillMode(hdc, WINDING);
    Polygon(hdc, p , 3);
}

```

```

        SelectObject(hdc, hbrPrevious);
        DeleteObject(hbrGray);
    }

long FAR PASCAL _export ChildWndChanell (HWND hwnd, UINT message, UINT wParam,
LONG lParam)
// ##12
// Funcție care tratează mesajele pentru ferestrele child
//
{
    HDC          hdc , hdc1;
    HBRUSH       hbrPrevious;
    PAINTSTRUCT ps ;
    POINT        p1 , p2 ;
    static       int PrevPos , CurentPos, Inc, Pos;
    static RECT  r , InvalidRect;
    static BOOL  First[PCL711MaxChanell];
    static int   i, rez , OffX[PCL711MaxChanell + 1] , j;
    static char  *Sir[20];
    static int   *PD;
    static POINT P , OldP[PCL711MaxChanell + 1];
    static BOOL  Once;
    HWND        hwndTmp;
    static float Rap;

    switch (message)
    {
        case WM_CREATE:
            return 0 ;

        case WM_SIZE :

            Pos = 0;
            SetScrollRange(hwnd , SB_HORZ, 0 , r.right, TRUE);
            SetScrollPos(hwnd , SB_HORZ , 0 , TRUE);
            Inc = r.right / 4;
            Rap = (float)r.right / StopAfter / SamplingRate;
            return 0;

        case WM_PAINT:
            hdc = BeginPaint (hwnd, &ps) ;
            GetClientRect(hwnd , &r);
            if(!GetWindowWord(hwnd , 0))
            {
                p1.x = r.left + Margin;
                p1.y = r.top + Margin;
                MoveTo (hdc, p1.x , p1.y) ;
                Arrow(hdc , p1 , Up , RGB(0, 0, 0));
                LineTo (hdc, r.left + Margin, r.bottom - Margin) ;
                MoveTo (hdc, r.left + Margin, r.bottom / 2) ;
                p2.x = r.right - Margin;
                p2.y = r.bottom / 2;
                LineTo (hdc, p2.x , p2.y) ;
                Arrow(hdc , p2 , Right , RGB(0, 0, 0));
                SetWindowWord(hwnd , 0 , 1);
                Once = TRUE;
            }
            if(IS_DRAWING)
            {
                if(Once)
                    Once = FALSE;
                else
                {
                    for(i = 0 ; i <= PCL711MaxChanell ; i++)
                    {
                        if(ChanellStatus[i])
                        {
                            PD = Chanells[i].Data + Curent * sizeof(int);
                            hdc1 = GetDC(hwndChild[ChanellStatus[i] - 1]);

                            P.x = *PD++;
                            P.y = *PD;
                            P.y = (int)((2 * Y - P.y * GraphScale - Margin) *
                                Windows[ChanellStatus[i] - 1].ScY - Y * (Windows[ChanellStatus[i] - 1].ScY -
                                1) + Windows[ChanellStatus[i] - 1].Offset * GraphScale);
                            SelectObject(hdc1, Chanells[i].hPen);
                            if((P.x - OffX[i]) > r.right)
                            {
                                hwndTmp = hwndChild[ChanellStatus[i] - 1];
                                for(j = 0 ; j <= PCL711MaxChanell ; j++)
                                {
                                    if(hwndTmp == hwndChild[ChanellStatus[j] - 1])
                                    {
                                        OldP[j].x -= Inc;
                                    }
                                }
                            }
                        }
                    }
                }
            }
    }
}

```



```

        OffX[j] += Inc;
    }
    Arrow(hdc1 , p2 , Right , GlobalBackColour);
    ScrollWindow(hwndTmp, -Inc , 0, NULL, NULL) ;
    Once = TRUE;
    hbrPrevious = SelectObject(hdc1 , GetStockObject(BLACK_PEN));
    MoveTo (hdc1, r.left + Margin, r.bottom / 2) ;
    p2.x = r.right - Margin;
    p2.y = r.bottom / 2;
    LineTo (hdc1, p2.x , p2.y) ;
    SelectObject(hdc1 , hbrPrevious);
    Arrow(hdc1 , p2 , Right , RGB(0, 0, 0));
    Pos += (int)(Inc * Rap);
    SetScrollPos(hwndTmp , SB_HORZ , Pos , TRUE);
    P.x -= OffX[i];
    if(lTime == 0.0)
    {
        OffX[i] = 0;
        MoveTo(hdc1 , P.x , P.y);
    }
    else
    {
        MoveTo(hdc1 , OldP[i].x , OldP[i].y);
        LineTo(hdc1 , P.x , P.y);
    }
    OldP[i] = P;
    if(Chanells[i].DisplayNumeric)
    {
        sprintf((char *)Sir , "%-5d" , *PD);
        SetBkColor(hdc1 , GlobalBackColour);
        TextOut(hdc1 , 2 + 40 * i , 1 , (LPSTR)Sir , strlen((LPSTR)Sir));
    }
    ReleaseDC(hwnd , hdc1);
}
Curent++;
}
}
}
EndPaint (hwnd, (PAINTSTRUCT FAR *)hdc) ;
return 0 ;

case WM_HSCROLL :
    if(IS_DRAWING)
        return 0;
    switch (wParam)
    {
        case SB_LINEDOWN      :Pos++;
        break;
        case SB_LINEUP:      Pos--;
        break;
        case SB_PAGEDOWN     :Pos += Inc;
        break;
        case SB_PAGEUP      :Pos -= Inc;
        break;
        case SB_THUMBPOSITION;;
        break;
        case SB_THUMBTRACK;;
        break;
    }
    if(Pos < 0)
        Pos = 0;
    if(Pos > r.right)
        Pos = r.right;
    ScrollWindow(hwnd, Pos - PrevPos, 0, NULL, NULL) ;
    SetScrollPos(hwnd , SB_HORZ , Pos , TRUE);
    PrevPos = Pos;
    return 0;

case WM_DESTROY :
    return 0;
}
return DefWindowProc (hwnd, message, wParam, lParam) ;
}

BOOL FAR PASCAL _export AboutDlgProc (HWND hDlg, UINT message, UINT wParam, LONG
lParam)
// ##13
// Funcția care tratează dialog Despre
//
{
    static DWORD      FreeSp ;
    static UINT FreePercernt , FreeGdi , FreeUser;

```

```

switch (message)
{
    case WM_INITDIALOG:
        FreeSp = GetFreeSpace(1U);
        FreePerernt = GetFreeSystemResources(GFSR_SYSTEMRESOURCES);
        FreeGdi = GetFreeSystemResources(GFSR_GDIRRESOURCES);
        FreeUser = GetFreeSystemResources(GFSR_USERRESOURCES);
        WriteMemoryToItem(hDlg, ID_MEM, &FreeSp, LongType, 10);
        WriteMemoryToItem(hDlg, ID_RES1, &FreePerernt, IntType, 10);
        WriteMemoryToItem(hDlg, ID_RES2, &FreeGdi, IntType, 10);
        WriteMemoryToItem(hDlg, ID_RES3, &FreeUser, IntType, 10);
        return TRUE;

    case WM_COMMAND:
        switch (wParam)
        {
            case IDOK:
            case IDCANCEL:
                EndDialog (hDlg, 0);
                return TRUE;
        }
        break;
}
return FALSE;
}

BOOL FAR PASCAL _export GeneralDlgProc (HWND hDlg, UINT message, UINT wParam,
LONG lParam)
// ##14
// Funcție care trtatează fereastra de setări generale
//
{
    static char Sir[LG_SIR_TEMP];
    static int i;
    static HDC hdc;
    static HMENU hmenu;
    static COLORREF TempFore, TempBack;
    POINT P1 = {200, 120}, P2 = {240, 120}, P3 = {200, 170}, P4 = {240, 170};

    switch (message)
    {
        case WM_INITDIALOG:
            BackColour.lStructSize = sizeof (CHOOSECOLOR);
            BackColour.hwndOwner = hDlg;
            BackColour.hInstance = (HWND)NULL;
            BackColour.rgbResult = RGB (0x80, 0x80, 0x80);
            BackColour.lpCustColors = dwCustColorsBack;
            BackColour.Flags = CC_RGBINIT | CC_FULLOPEN;
            BackColour.lCustData = 0L;
            BackColour.lpfnHook = NULL;
            BackColour.lpTemplateName = NULL;

            ForeColour.lStructSize = sizeof (CHOOSECOLOR);
            ForeColour.hwndOwner = (HWND)NULL;
            ForeColour.hInstance = (HWND)NULL;
            ForeColour.rgbResult = RGB (0x80, 0x80, 0x80);
            ForeColour.lpCustColors = dwCustColorsFore;
            ForeColour.Flags = CC_RGBINIT | CC_FULLOPEN;
            ForeColour.lCustData = 0L;
            ForeColour.lpfnHook = NULL;
            ForeColour.lpTemplateName = NULL;
            TempFore = GlobalForeColour;
            TempBack = GlobalBackColour;

            WriteMemoryToItem(hDlg, IDC_IO_ADDRESS, &PCLPort, IntType, 16);
            WriteMemoryToItem(hDlg, IDC_PATH, FileName, CharType, 0);
            WriteMemoryToItem(hDlg, IDC_NO_OF_WINDOWS, &MaxWindows, IntType, 10);
            WriteMemoryToItem(hDlg, IDC_FREQ, &SamplingRate, IntType, 10);
            WriteMemoryToItem(hDlg, IDC_STOP, &StopAfter, IntType, 10);
            return 0;

        case WM_PAINT:
            ShowColor(hDlg, TempFore, P3, P4);
            ShowColor(hDlg, TempBack, P1, P2);
            return 0;

        case WM_COMMAND:
            switch (wParam)
            {
                case IDC_BACK:
                    i = ChooseColor((CHOOSECOLOR FAR*)&BackColour);
                    TempBack = BackColour.rgbResult;
                    ShowColor(hDlg, TempBack, P1, P2);
                    break;
            }
    }
}

```

```

case IDC_FORE :
i = ChooseColor((CHOOSECOLOR FAR*)&ForeColour);
TempFore = ForeColour.rgbResult;
ShowColor(hDlg , TempFore , P3 , P4);
break;

case IDOK:
GlobalBackColour = TempBack;
GlobalForeColour = TempFore;
if(!IsDlgButtonChecked(hDlg , IDC_OLD))
for(i = 0 ; i <= PCL711MaxChanell ; i++)
{
if(Chanells[i].hPen)
DeleteObject(Chanells[i].hPen);
Chanells[i].hPen = CreatePen(PS_SOLID , 1 , GlobalForeColour);
Chanells[i].LineColour = GlobalForeColour;
}
DeleteObject(SetClassWord(hwndChild[0] , GCW_HBRBACKGROUND ,
CreateSolidBrush(GlobalBackColour)));
ReadItemToMemory(hDlg , IDC_IO_ADDRESS , 5 , &PCLPort , IntType , 16);
ReadItemToMemory(hDlg , IDC_PATH, LG_SIR_TEMP - 1 , FileName , CharType , 0);
MakeFN(Treteri , (LPSTR)FileName);
SetWindowText(GetParent(hDlg));
ReadItemToMemory(hDlg , IDC_NO_OF_WINDOWS , 5 , &MaxWindows , IntType , 10);
if((MaxWindows < 1) || (MaxWindows > (PCL711MaxChanell + 1)))
MaxWindows = 1;
for(i = 0 ; i < PCL711MaxChanell ; i++)
if(ChanellStatus[i] > MaxWindows)
ChanellStatus[i] = MaxWindows;
ReadItemToMemory(hDlg , IDC_FREQ , 5 , &SamplingRate , IntType , 10);
ReadItemToMemory(hDlg , IDC_STOP , 5 , &StopAfter , IntType , 10);
hmenu = GetSubMenu(GetMenu(GetParent(hDlg)), 2);
for(i = 0 ; i <= PCL711MaxChanell ; i++)
if ( i < MaxWindows)
EnableMenuItem(hmenu , (UINT)(ID_WINDOW1 + i) , MF_ENABLED );
else
EnableMenuItem(hmenu , (UINT)(ID_WINDOW1 + i) , MF_GRAYED);
for(i = 0 ; i <= PCL711MaxChanell ; i++)
if ( ChanellStatus[i] > MaxWindows)
ChanellStatus[i] = 1;
MakeChildWindow(GetParent(hDlg) , FALSE, FALSE);
EndDialog (hDlg, 0) ;
break;
case IDCANCEL:
EndDialog (hDlg, 0) ;
break ;
}
return 0 ;
}
return FALSE ;
}

BOOL FAR PASCAL _export IndividualDlgProc (HWND hDlg, UINT message, UINT wParam,
LONG lParam)
// ##15
// Funcție carea tratează fereastra de control individual al canalelor
//
{
static char Sir[LG_SIR_TEMP];
static int CurentChanell;
static COLORREF TempColour;
POINT P1 = {320 , 100} , P2 = {340 , 100};

switch (message)
{
case WM_INITDIALOG:
Colour.lStructSize = sizeof (CHOOSECOLOR) ;
Colour.hwndOwner = (HWND)NULL ;
Colour.hInstance = (HWND)NULL ;
Colour.rgbResult = RGB (0x80, 0x80, 0x80) ;
Colour.lpCustColors = dwCust;
Colour.Flags = CC_RGBINIT | CC_FULLOPEN ;
Colour.lCustData = 0L ;
Colour.lpfHook = NULL ;
Colour.lpTemplateName = NULL ;

CurentChanell = LOWORD(lParam);
TempColour = Chanells[CurentChanell].LineColour;
WriteMemoryToItem(hDlg , ID_CHNO , &CurentChanell , IntType , 10);
WriteMemoryToItem(hDlg , ID_CHANELL_NAME , &Chanells[CurentChanell].ChName ,
CharType , 0);
if(Chanells[CurentChanell].DisplayNumeric)
SendDlgItemMessage(hDlg, ID_NUMERIC , BM_SETCHECK , 1, 0L);
return TRUE ;
}
}

```

```

break;
case WM_PAINT :
    ShowColor(hDlg , TempColour , P1 , P2);
case WM_COMMAND:
    switch (wParam)
    {
        break;
    case ID_LINECOLOR :
        ChooseColor((CHOOSECOLOR FAR*)&Colour);
        TempColour = Colour.rgbResult;
        ShowColor(hDlg , TempColour , P1 , P2);
        break;
    case IDOK:
        ReadItemToMemory(hDlg , ID_CHANELL_NAME , 19 , &Chanells[CurentChanell].ChName ,
            CharType , 0);
        Chanells[CurentChanell].DisplayNumeric = IsDlgButtonChecked(hDlg , ID_NUMERIC);
        EndDialog (hDlg , 0) ;
        Chanells[CurentChanell].LineColour = TempColour;
        DeleteObject(Chanells[CurentChanell].hPen);
        Chanells[CurentChanell].hPen = CreatePen(PS_SOLID , 1 , TempColour);
        break;
    case IDCANCEL:
        EndDialog (hDlg , 0) ;
        return TRUE ;
    }
    break ;
    }
return FALSE ;
}

```

```

BOOL FAR PASCAL _export WatchDlgProc(HWND hDlg, UINT message, UINT wParam, LONG lParam)

```

```

// ##16
// Funcție care tratează fereastra de urmărire a canalelor
//
{
    int i;
    UINT Rez;
    char Sir[LG_SIR_TEMP];
    HMENU hmenu;
    BOOL IsCheck;

    switch (message)
    {
        case WM_INITDIALOG:
            for(i = 0 ; i <= PCL711MaxChanell ; i++)
            {
                if(ChanellStatus[i])
                {
                    SendDlgItemMessage(hDlg, ID_CHECK_CHO + i , BM_SETCHECK , 1, 0L);
                    _itoa(ChanellStatus[i] , Sir , 10);
                    SendDlgItemMessage(hDlg , IDC_WINO + i , WM_SETTEXT , 0 , (LPARAM)((LPCSTR)Sir));
                }
            }
            return TRUE ;
        case WM_COMMAND:
            switch (wParam)
            {
                case IDOK:
                    hmenu = GetSubMenu(GetMenu(GetParent(hDlg)), 3);
                    for(i = 0 ; i <= PCL711MaxChanell ; i++)
                    {
                        Chanells[i].Data = NULL;
                        Chanells[i].Last = NULL;
                        IsCheck = IsDlgButtonChecked(hDlg , ID_CHECK_CHO + i);
                        if(IsCheck)
                        {
                            SendDlgItemMessage(hDlg , IDC_WINO + i , WM_GETTEXT,5,
                                (LPARAM)((LPCSTR)Sir));
                            Rez = atoi(Sir);
                            if(((int)Rez > MaxWindows) || (Rez < 1))
                                ChanellStatus[i] = 1;
                            else
                                ChanellStatus[i] = Rez;
                        }
                        else
                            ChanellStatus[i] = 0;
                        EnableMenuItem(hmenu , (UINT)(ID_CHANELLO + i) , IsCheck ?
                            MF_ENABLED : MF_GRAYED);
                    }
                    EndDialog (hDlg , 0) ;
                    break;
                case IDCANCEL:

```

```

        EndDialog (hDlg, 0) ;
        return TRUE ;
    }
    break ;
}
return FALSE ;
}

BOOL FAR PASCAL _export WindowsDlgProc(HWND hDlg, UINT message, UINT wParam,
LONG lParam)
// ##17
// Funcție care tratează fereastra de control a parametrilor ferestrelor child
//
{
    static char Sir[LG_SIR_TEMP];
    static int Index;

    _ltoa(lParam, Sir, 10);
    switch (message)
    {
        case WM_INITDIALOG:
            WriteMemoryToItem(hDlg, ID_WIN_NO, &Index, IntType, 0);
            Index = (int)(0x00FF & lParam) - 1;
            WriteMemoryToItem(hDlg, ID_OFFSET, &(Windows[Index].Offset), IntType, 0);
            WriteMemoryToItem(hDlg, ID_SCALEX, &(Windows[Index].ScX), DoubleType, 0);
            WriteMemoryToItem(hDlg, ID_SCALEY, &(Windows[Index].ScY), DoubleType, 0);
            return TRUE ;
        case WM_COMMAND:
            switch (wParam)
            {
                case IDOK:
                    ReadItemToMemory(hDlg, ID_OFFSET, 7, &(Windows[Index].Offset), IntType, 10);
                    ReadItemToMemory(hDlg, ID_SCALEX, 7, &(Windows[Index].ScX), DoubleType, 10);
                    ReadItemToMemory(hDlg, ID_SCALEY, 7, &(Windows[Index].ScY), DoubleType, 10);
                    EndDialog (hDlg, 0) ;
                    break;
                case IDCANCEL:
                    EndDialog (hDlg, 0) ;
                    return TRUE ;
            }
            break ;
    }
    return FALSE ;
}

LPSTR FindString(LPSTR Source, LPSTR Target)
// ##18
// Caută un șir
//
{
    LPSTR p1 = Source;
    char Temp[20];
    int Lg = sizeof(Target)+1;

    do
        lstrcpyn(Temp, p1++, Lg);
    while(lstrcmp(Temp, Target));
    p1 += Lg - 2;
    return p1;
}

void CutZero (LPSTR Dest)
// ##19
// Taie zeoruri dintr-un șir
//
{
    char *p1 = Dest + strlen(Dest) - 1;

    while(*p1 == '0')
        p1--;

    p1++;
    *p1 = '\0';
}

BOOL FAR PASCAL _export CNCProc(HWND hDlg, UINT message, UINT wParam, LONG
lParam)
// ##20
// Funcție de control a parametrilor CNC
//
{
    static double Feed = 0, Radius = 0, Length = 0;

```

```

static HFILE PCNCFile;
int iTemp;
static char cTemp[25];
static char *p1 , *p2 , *p3 , *pFeed , *pRadius , *pLength;
static char pDest[2048];
OFSTRUCT FStruct;

switch (message)
{
    case WM_INITDIALOG:
        if((_lclose(PCNCFile) == HFILE_ERROR) == HFILE_ERROR)
        {
            MessageBox(hDlg , "Cannot open Progl.cnc" , "Open CNC" , MB_ICONSTOP);
            EndDialog(hDlg , 0);
        }
        Zero(pTemp , sizeof(pTemp));
        _lread(PCNCFile , pTemp , sizeof(pTemp));
        Feed = atod(FindString(pTemp , FeedID));
        Length = atod(FindString(pTemp , LengthID));
        p1 = FindString(pTemp , RadiusID);
        Radius = atod(FindString(p1 , RadiusID));
        WriteMemoryToItem(hDlg , IDC_FEED , &Feed , DoubleType , 0);
        WriteMemoryToItem(hDlg , IDC_DIAMETER , &Radius , DoubleType , 0);
        WriteMemoryToItem(hDlg , IDC_LENGTH , &Length , DoubleType , 0);
        _lclose(PCNCFile);
        return TRUE ;

    case WM_COMMAND:
        switch (wParam)
        {
            case IDOK:
                ReadItemToMemory(hDlg , IDC_FEED , 7 , &Feed , DoubleType , 10);
                ReadItemToMemory(hDlg , IDC_DIAMETER , 7 , &Radius , DoubleType , 10);
                ReadItemToMemory(hDlg , IDC_LENGTH , 7 , &Length , DoubleType , 10);
                p1 = pTemp;
                p2 = pDest;
                iTemp = 1;
                Zero(pDest , sizeof(pDest));
                pFeed = FindString(pTemp , FeedID);
                pRadius = FindString(pTemp , RadiusID);
                pRadius = FindString(pRadius , RadiusID);
                pLength = FindString(pTemp , LengthID);
                while(*p1 != '\0')
                {
                    *p2++ = *p1++;
                    if(p1 == pFeed)
                    {
                        sprintf(cTemp , "%Ff" , Feed);
                        CutZero(cTemp);
                        p3 = cTemp;
                        if(Feed < 1)
                        {
                            p3++;
                            *p2++ = '=';
                            while(*p3 != '\0')
                            {
                                *p2++ = *p3++;
                                iTemp++;
                            }
                            p1++;
                            while(IsNum(p1++));
                            p1--;
                        }
                    }
                    if(p1 == pRadius)
                    {
                        sprintf(cTemp , "%Ff" , Radius);
                        CutZero(cTemp);
                        p3 = cTemp;
                        while(*p3 != '\0')
                        {
                            *p2++ = *p3++;
                            iTemp++;
                        }
                        while(IsNum(p1++));
                        p1--;
                    }
                    if(p1 == pLength)
                    {
                        sprintf(cTemp , "%Ff" , Length);
                        CutZero(cTemp);
                        p3 = cTemp;
                        while(*p3 != '\0')
                        {
                            *p2++ = *p3++;
                            iTemp++;
                        }
                    }
                }
            }
        }

```

```

        }
        while(IsNum(pl++));
        pl--;
    }
    iTemp++;
}
PCNCFile = OpenFile("Progl.cnc" ,&FStruct, OF_CREATE);
_lwrite(PCNCFile , pDest , iTemp);
_lclose(PCNCFile);
EndDialog (hDlg, 0) ;
break;
case IDCANCEL:
    EndDialog (hDlg, 0) ;
    return TRUE ;
}
break ;
}
return FALSE ;
}

BOOL FAR PASCAL _export ConfigProc(HWND hDlg, UINT message, UINT wParam, LONG
lParam)
// ##21
// Funcție de control a ferestrei Profil
//
{
    static int PrevConfig , iTemp;
    char ConfigName[15];
    HWND hwnd;

    switch (message)
    {
        case WM_INITDIALOG:
            PrevConfig = PredefinedConfig;
            switch(PredefinedConfig)
            {
                case ConfigNo :
                    break;
                case ConfigForce :
                    SendDlgItemMessage(hDlg , IDC_FORCE , BM_SETCHECK , TRUE , 0);
                    break;
                case ConfigProfile :
                    SendDlgItemMessage(hDlg , IDC_PROFILE , BM_SETCHECK , TRUE , 0);
                    break;
            }
            return TRUE ;
        case WM_COMMAND:
            switch (wParam)
            {
                case IDOK:
                    if(SendDlgItemMessage(hDlg , IDC_PROFILE , BM_GETCHECK , 0 , 0))
                        PredefinedConfig = ConfigProfile;
                    else
                        if(SendDlgItemMessage(hDlg , IDC_FORCE , BM_GETCHECK , 0 , 0))
                            PredefinedConfig = ConfigForce;
                    else
                        PredefinedConfig = ConfigNo;
                    if(PrevConfig != PredefinedConfig)
                    {
                        iTemp = Treceri;
                        iTemp--;
                        if(PredefinedConfig == ConfigProfile)
                        {
                            iTemp--;
                            strcpy(ConfigName , "prof.cfg");
                        }
                        else
                            strcpy(ConfigName , "force.cfg");
                        hwnd = GetParent(hDlg);
                        ReadConfig(ConfigName , hwnd);
                        Treceri = iTemp;
                        MakeFN(Treceri , (LPSTR)FileName);
                        SetWindowName(hwnd);
                    }
                    EndDialog (hDlg, 0) ;
                    return TRUE ;
                case IDCANCEL:
                    EndDialog (hDlg, 0) ;
                    return TRUE ;
            }
            }
        return FALSE ;
}

```



Listă cu funcții definite în anexele 2 și 4:

```
22.void PopFileInitialize (HWND hwnd , BOOL ForWrite)
23.void Zero(void *Addr , UINT Lg)
24.void ReadItemToMemory(HWND hDlg , UINT Item ,int Lg ,void *Addr , int DataType
    , int Base)
25.void WriteMemoryToItem(HWND hDlg , UINT Item ,void *Addr , int DataType , int
    Base)
```

## 11. ANEXA 4 - PROGRAM TRATARE TERMOGRAME

### 11.1. CARACTERISTICI

- Programul Termo tratează fișierele bitmap, în format DIB, care conțin termograme prelucrate;
- Format fișier executabil - aplicație Windows;
- Limbaj de programare C;
- Compilator folosit: Microsoft Visual C++1.5;
- structură de meniuri:

#### Fișiere ⇒

Deschide DIB .....*Apelează dialogul standard pentru deschidere fișier cu termogramă*  
 Asociază model.....*Apelează dialogul standard pentru deschidere fișier cu modelul cu elemente finite*  
 Scrie fișier ADB .....*Apelează dialogul standard pentru creare fișier ieșire cu extensia ADB*  
 Ieșire .....*Închide program*

#### Model ⇒

Punctează *Permite punctarea cu mouse-ul, pentru citirea coordonatelor a trei puncte de definire a poziției modelului*  
 Contur .....*Dacă este bifat trasează numai conturul modelului cu elemente finite*  
 Model complet.....*Dacă este bifat trasează tot modelul cu elemente finite*  
 Rota muchie .....*Dacă este bifat calculează poziția modelului după rotația în jurul muchiei determinată de primele două puncte indicate*

#### Termograma ⇒

Temperatri extreme..*Deschide fereastră pentru precizarea temperaturilor extreme de pe scala gradată*  
 Marchează scala.....*Permite punctarea cu mouse-ul, pentru citirea coordonatelor a două puncte de definire a poziției scalei*  
 Reafișează .....*Reafișează informația curentă de pe ecran*

#### Flux ⇒

Flux contur ..*Dacă este bifat afișează linia de calcul a fluxului*  
 Calcul flux ..*Dacă este bifat scrie în fișierul de ieșire informațiile legate de flux*  
 Distanța flux ..*Deschide fereastră pentru precizarea distanței în coordonate ecran pentru calculul fluxului*

#### Help ⇒

Despre program.....*Afișează fereastră cu informații despre program*

#### Fișiere de intrare

*Fișierul cu rețeaua de elemente finite:*

format : ASCII

extensia implicită : TRM

structura fișierului: cuprinde 4 secțiuni : titlu, descriere noduri, descriere elemente, descriere contur model. Fiecare secțiune începe cu o linie cu trei caractere - urmate de identificatorul secțiunii, ca în exemplul de mai jos. Structura datelor din secțiunile TITLE, NODES, ELEMENTS este aceeași ca cea de la programul de analiză cu elemente finite. Pentru secțiunea CONTUR, după linia cu identificatorul de secțiune, urmează pe câte o linie numărul nodurilor care definesc conturul modelului.

*Fișierul cu termograma:*

format : binar

extensia implicită : DIB

structura fișierului: este în concordanță cu standardul care îl reglementează;

*Exemplu fragment cu fișier de intrare:*

---	TITLE	1 258 1
---	STRUCTURA COMPLETA CAP SCULA	2 104 6
---	NODES	2 258 1
1	-1.00000000 21.0000000 0.000000000E+00	101 13 18
2	10.0000000 12.0000000 0.000000000E+00	3 258 1
3	1.20000000 19.2000000 0.000000000E+00	7 31 58
.....	ș.a.m.d.	..... ș.a.m.d.
---	ELEMENTS	— CONTUR

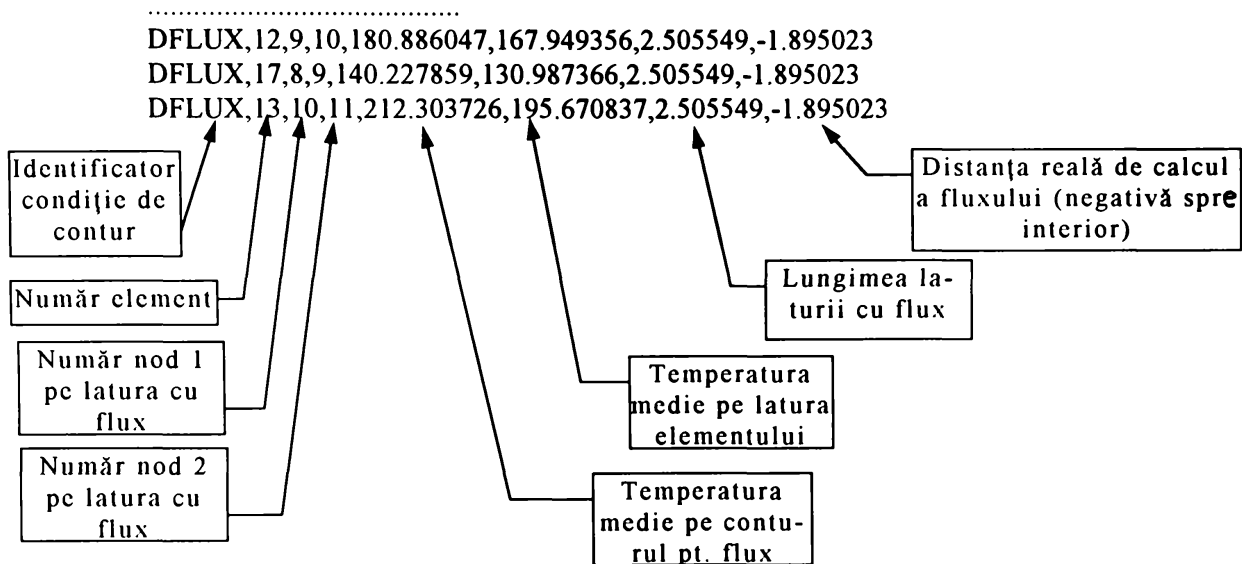
**Fișiere de ieșire**

*Fișierul cu temperaturile nodale*

format : ASCII

extensia implicită : ADB

structura fișierului: cuprinde 2 secțiuni fără identificator de secțiune: prima secțiune conține temperaturile nodale cu structura identică cu cea pentru condițiile de contur Dirichlet (temperatură impusă) din programul de analiză cu element finit; a doua secțiune are următoarea structură:



Obs : Fișierul de ieșire al acestui program, este tratat într-o foaie de calcul Excel cu funcția din anexa 6, pentru a corecta temperaturile cu emisivitatea materialului respectiv.

În anexă nu sunt cuprinse fișierul cu resursele Windows ale acestui program, precum și fișierul de definiție, care nu prezintă interes deosebit

## 11.2. FIȘIER HEADER

```

/*-----
  TERMO.H -- TERMOGRAME Window
            (c) Adrian ROSCA, 1997
-----*/
#define IDM_OPEN      10
#define IDM_ACTUAL    20
#define IDM_STRETCH  21
#define RZ            0
#define RY            1
#define RX            2
#define M_PI          3.14159265358979323846
#define BlockFis      1024
#define MaskRGB       0xFFFFFF

enum SectiuniID {IsTitle =0 , IsNodes, IsElements, IsFlux, IsContur};
#define X            0
#define Y            1

#define IntType      100
#define LongType     101
#define FloatType    102
#define DoubleType   103
#define CharType     104
#define MaxUint      0xFFFFU
#define NodPerElement 4
#define SCREEN       1000
#define REALITATE    2000
#define COMPLET      5000
#define CONTUR       6000
#define NORMAL       7000
struct ListaASCII
{
    HLOCAL      pObj;
    HLOCAL      pObjSir;
    char far    *pSir;
    int         Lg;
    struct ListaASCII *Urm;
};
typedef struct ListaASCII ListaASCIIType;

struct ListaNodes
{
    HLOCAL      pObj;
    float       x, y, Sol;
    int         Nr, xScreen, yScreen;
    struct ListaNodes *Urm;
};
typedef struct ListaNodes ListaNodesType;

struct ListaElements
{
    HLOCAL      pObj;
    int         Nr, Nod[3];
    struct ListaElements *Urm;
};
typedef struct ListaElements ListaElementsType;

struct ListaFlux
{
    HLOCAL      pObj;
    int         Nr;
    ListaNodesType Nod[2];
    struct ListaFlux *Urm;
    struct ListaElements *Urm;
};
//
//
typedef struct ListaFlux ListaFluxType;

struct ListaContur
{
    HLOCAL      pObj;
    int         Nr;
    struct ListaContur *Urm;
};
typedef struct ListaContur ListaConturType;

#define MIN(a,b) ((a) < (b) ? (a) : (b))
enum PARAMETRI {ALFA = 0, GAMA, SCARA, TX, TY};

```

## 11.3. FIȘIER SURSĂ

```

/*-----
  TERM.C -- TERMOGRAME Window
          (c) Adrian ROSCA, 1997
-----*/
#include <windows.h>
#include <windowsx.h>
#include <commdlg.h>
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <math.h>
#include "resource.h"
#include "termo.h"

long FAR PASCAL _export WndProc (HWND, UINT, UINT, LONG) ;
long FAR PASCAL _export WndProcCoord (HWND, UINT, UINT, LONG lParam);
BOOL FAR PASCAL _export DespreDlgProc (HWND, UINT, UINT, LONG) ;
BOOL FAR PASCAL _export DistDlgProc (HWND, UINT, UINT, LONG);

char Coord[] = "Coordonate";
char szAppName [] = "Termograme" ;
char szIconName[] = "CUTIT";
char szMenuName[] = "TERMOGRAMA";
int NrNodes, NrElements, MaxTemp, MinTemp, NrFlux, NrContur;
int IsRotaMuchie = 1, FaceFlx = 1;
int DistFlux = 20;
double Scara;
HDC hdc ;
ListaNodesType      *pNodNou, *pNodCap, *pNodCurent;
ListaElementsType   *pElementNou, *pElementCap, *pElementCurent;
ListaFluxType       *pFluxNou, *pFluxCap, *pFluxCurent;
ListaConturType     *pConturNou, *pConturCap, *pConturCurent;
BOOL FAR PASCAL _export TemperaturiDlgProc (HWND, UINT, UINT, LONG);
char Lin[255];
HGLOBAL hl;

void Dealoca(void)
// ##1
// Dealocă structurile alocate dinamic
//
{
    pNodNou = pNodCurent = pNodCap;
    while(pNodCurent)
    {
        pNodNou = pNodCurent->Urm;
        LocalUnlock(pNodCurent->pObj);
        LocalFree(pNodCurent->pObj);
        pNodCurent = pNodNou;
    }
    pNodCap = NULL;
    pElementCurent = pElementNou = pElementCap;
    while(pElementCurent)
    {
        pElementNou = pElementCurent->Urm;
        LocalUnlock(pElementCurent->pObj);
        LocalFree(pElementCurent->pObj);
        pElementCurent = pElementNou;
    }
    pElementCap = NULL;
    pFluxCurent = pFluxNou = pFluxCap;
    while(pFluxCurent)
    {
        pFluxNou = pFluxCurent->Urm;
        LocalUnlock(pFluxCurent->pObj);
        LocalFree(pFluxCurent->pObj);
        pFluxCurent = pFluxNou;
    }
    pFluxCap = NULL;
    pConturCurent = pConturNou = pConturCap;
    while(pConturCurent)
    {
        pConturNou = pConturCurent->Urm;
        LocalUnlock(pConturCurent->pObj);
        LocalFree(pConturCurent->pObj);
        pConturCurent = pConturNou;
    }
    pConturCap = NULL;
}

void EroareAlocare(char *Mesa)

```

```

{
// ##2
// Scrie mesaj de eroare
//
    MessageBox((HWND)NULL , (LPCSTR)"Nu se poate aloca memorie" ,
               (LPCSTR)Mesaj , MB_ICONSTOP );
}

void Zero(void *Addr , UINT Lg)
{
// ##3
    _fmemset(Addr , '\0' , Lg);
}

void ReadItemToMemory(HWND hDlg , UINT Item ,int Lg ,void *Addr , int DataType , int Base)
// ##4
// Citește date la o adresă de memorie dintr-o cutie de dialog
//
{
    char Sir[100];
    char Format[8];
    char far *PBlanc;
    int Rez;

    SendDlgItemMessage(hDlg , Item , WM_GETTEXT , Lg ,(LPARAM)((LPCSTR)Sir));
    switch(DataType)
    {
        case IntType :
            if(Base == 16)
                lstrcpy(Format , "%Fh");
            else
                lstrcpy(Format , "%Fd");
            break;
        case LongType :
            if(Base == 16)
                lstrcpy(Format , "%Flh");
            else
                lstrcpy(Format , "%Fld");
            break;
        case FloatType :
            lstrcpy(Format , "%Ff");
            break;
        case DoubleType :
            lstrcpy(Format , "%Ffh");
            break;
        case CharType :
            lstrcpy(Format , "%Fs");
            break;
    }
    PBlanc = Sir;
    while(*PBlanc != '\0')
    {
        if(*PBlanc == ' ')
            *PBlanc = '_';
        PBlanc++;
    }
    Rez = sscanf(Sir , Format , Addr);
}

void WriteMemoryToItem(HWND hDlg , UINT Item ,void *Addr , int DataType , int Base)
// ##5
// Scrie date de la o adresă de memorie intr-o cutie de dialog
//
{
    static char Sir[100];

    switch(DataType)
    {
        case IntType :
            if(Base == 16)
                sprintf(Sir , "%Fh" , *(int *)Addr);
            else
                sprintf(Sir , "%Fd" , *(int *)Addr);
            break;
        case LongType :
            if(Base == 16)
                sprintf(Sir , "%Flh" , *(long *)Addr);
            else
                sprintf(Sir , "%Fld" , *(long *)Addr);
            break;
        case FloatType :
            sprintf(Sir , "%Ff" , *(float *)Addr);
            break;
        case DoubleType :

```

```

        sprintf(Sir , "%1.3Flf" , *(double *)Addr);
        break;
    case CharType :
        sprintf(Sir , "%Fs`" , (char *)Addr);
        break;
    }
    SendDlgItemMessage(hDlg , Item , WM_SETTEXT , 0 ,(LPARAM)((LPCSTR)Sir));
}

```

```

void CautaNod(ListaNodesType *P, int Nr, int CoordType)

```

```

{
    // ##6
    // Caută nodul Nr în lista globală și copiază la adresa P coordonatele
    // ecran sau reale
    //

    ListaNodesType *pNod = pNodCap;
    while(pNod)
    {
        if(pNod->Nr == Nr)
        {
            if(CoordType == SCREEN)
            {
                P->xScreen = pNod->xScreen;
                P->yScreen = pNod->yScreen;
            }
            else
            {
                P->x = pNod->x;
                P->y = pNod->y;
            }
            return;
        }
        pNod = pNod->Urm;
    }
}

```

```

void CalcCoordFlux(ListaFluxType *pFlx, int CoordType)

```

```

// ##7
// Calculează coordonatele punctelor unde se va evalua fluxul pe o latură
// paralelă cu punctele P1, P2, din structura indicată de pFlx
//

{
    ListaNodesType P1, P2;
    double m, n, alfa;

    CautaNod(&P1, pFlx->Nod[0].Nr, CoordType);
    CautaNod(&P2, pFlx->Nod[1].Nr, CoordType);
    if(CoordType == REALITATE)
    {
        alfa = atan2(P2.y - P1.y , P2.x - P1.x ) - M_PI / 2;
        pFlx->Nod[0].x = P1.x + DistFlux * cos(alfa);
        pFlx->Nod[0].y = P1.y + DistFlux * sin(alfa);
        pFlx->Nod[1].x = P2.x + DistFlux * cos(alfa);
        pFlx->Nod[1].y = P2.y + DistFlux * sin(alfa);
    }
    else
    {
        alfa = atan2(P2.yScreen - P1.yScreen , P2.xScreen - P1.xScreen ) - M_PI / 2;
        pFlx->Nod[0].xScreen = P1.xScreen + DistFlux * cos(alfa);
        pFlx->Nod[0].yScreen = P1.yScreen + DistFlux * sin(alfa);
        pFlx->Nod[1].xScreen = P2.xScreen + DistFlux * cos(alfa);
        pFlx->Nod[1].yScreen = P2.yScreen + DistFlux * sin(alfa);
    }
}

```

```

void CitesteModel(char *Name , HWND hwnd)

```

```

{
    // ##8
    // Citește din fișierul de intrare modelul cu elemente finite atașat termogramei
    //

    HFILE PFile;
    int cit, Total = 0, SectiuneaCurenta;
    char *Poz , *pSect, *pCR, *pEnd;
    char *Sectiuni[] = {"TITLE" , "NODES" , "ELEMENTS" , "FLUX" , "CONTUR"};
    int CurentSection = 0;
    ListaASCIIType *pNou, *pCap = NULL , *pCurent , *pASCII;
    HLOCAL pH;
    char *pFmtNod = "%d %g %g" , *pFmtElement1 = "%d" , *pFmtElement2 = "%d %d %d %d";
    char *pFmtFlux = "%d %d %d";
}

```



```

static Tmp[BlockFis];

PFile = _lopen(Name, READ);
Zero(Tmp, BlockFis);

cit = _lread(PFile, Tmp, BlockFis);
do
{
    if((void *) (pH = LocalAlloc(LPTR, sizeof(ListaASCIIType))) == NULL)
    {
        EroareAlocare("Citeste Post Fis -> Obj");
        return;
    }
    pNou = (ListaASCIIType *) LocalLock(pH);
    pNou->pObj = pH;
    pNou->Urm = NULL;
    if(!pCap)
    {
        pCurent = pCap = pNou;
        pCurent->Urm = NULL;
    }
    else
        pCurent->Urm = pNou;
    if((void *) (pNou->pObjSir = (HLOCAL) LocalAlloc(LPTR, cit)) == NULL)
    {
        EroareAlocare("Citeste Post Fis -> Sir");
        return;
    }
    pNou->pSir = (char *) LocalLock(pNou->pObjSir);
    pNou->Lg = cit;
    _fmemcpy(pNou->pSir, Tmp, cit);
    Total += cit;
    cit = _lread(PFile, Tmp, BlockFis);
    pCurent = pNou;
}
while(cit);
_lclose(PFile);

if((void *) (pH = LocalAlloc(LPTR, sizeof(ListaASCIIType))) == NULL)
{
    EroareAlocare("Zonă Global Fis -> Obj");
    return;
}
pASCII = (ListaASCIIType *) LocalLock(pH);
pASCII->pObj = pH;
Total += 2;
if((void *) (pASCII->pObjSir = (HLOCAL) LocalAlloc(LPTR, Total)) == NULL)
{
    EroareAlocare("Zonă Global Fis -> Sir");
    return;
}
pASCII->pSir = (char *) LocalLock(pASCII->pObjSir);
pCurent = pCap;
Poz = pASCII->pSir;
do
{
    _fmemcpy(Poz, pCurent->pSir, pCurent->Lg);
    Poz += pCurent->Lg;
    pCap = pCap->Urm;
    LocalUnlock(pCurent->pObjSir);
    LocalUnlock(pCurent->pObj);
    LocalFree(pCurent->pObjSir);
    LocalFree(pCurent->pObj);
    pCurent = pCap;
}
while(pCurent != NULL);

pCR = pASCII->pSir + Total - 1;
*pCR++ = (char)13; *pCR = '\0';
Poz = pASCII->pSir;
pEnd = Poz + Total;
SectiuneaCurenta = IsTitle;

do
{
    if((pCR = _fstrchr(Poz, 13)) != NULL)
        *pCR = '\0';
    if((pSect = _fstrstr(Poz, Sectiuni[SectiuneaCurenta])) && *Sectiuni[SectiuneaCurenta])
        SectiuneaCurenta++;
}
else
{
    switch (SectiuneaCurenta - 1)
    {
        case IsTitle:
    }
}

```

```

        break;

    case IsNodes:
        if((void *) (pH = LocalAlloc(LPTR , sizeof(ListaNodesType))) == NULL)
        {
            EroareAlocare("Noduri");
            return;
        }
        pNodNou = (ListaNodesType *)LocalLock(pH);
        pNodNou->pObj = pH;
        pNodNou->Urm = NULL;
        sscanf(Poz, pFmtNod, &pNodNou->Nr, &pNodNou->x, &pNodNou->y);
        if(!pNodCap)
        {
            pNodCurent = pNodCap = pNodNou;
            NrNodes = 0;
        }
        else
            pNodCurent->Urm = pNodNou;
        pNodCurent = pNodNou;
        NrNodes++;
        break;

    case IsElements:
        if((void *) (pH = LocalAlloc(LPTR , sizeof(ListaElementsType))) == NULL)
        {
            EroareAlocare("Elemente");
            return;
        }
        pElementNou = (ListaElementsType *)LocalLock(pH);
        pElementNou->pObj = pH;
        pElementNou->Urm = NULL;
        sscanf(Poz, pFmtElement1, &pElementNou->Nr);
        Poz = pCR + 2;
        pCR = _fstrchr(Poz , 13);
        *pCR = '\0';
        sscanf(Poz, pFmtElement2, &pElementNou->Nod[0], &pElementNou->Nod[1],
            &pElementNou->Nod[2], &pElementNou->Nod[3]);
        if(!pElementCap)
        {
            pElementCurent = pElementCap = pElementNou;
            NrElements = 1;
        }
        else
            pElementCurent->Urm = pElementNou;
        pElementCurent = pElementNou;
        NrElements++;
        break;

    case IsFlux:
        if((void *) (pH = LocalAlloc(LPTR , sizeof(ListaFluxType))) == NULL)
        {
            EroareAlocare("Flux");
            return;
        }
        pFluxNou = (ListaFluxType *)LocalLock(pH);
        pFluxNou->pObj = pH;
        pFluxNou->Urm = NULL;
        sscanf(Poz, pFmtFlux, &pFluxNou->Nr, &pFluxNou->Nod[0].Nr, &pFluxNou->Nod[1].Nr);
        if(!pFluxCap)
        {
            pFluxCurent = pFluxCap = pFluxNou;
            NrFlux = 1;
        }
        else
            pFluxCurent->Urm = pFluxNou;
        pFluxCurent = pFluxNou;
        NrFlux++;
        break;

    case IsContur:
        if((void *) (pH = LocalAlloc(LPTR , sizeof(ListaConturType))) == NULL)
        {
            EroareAlocare("Contur");
            return;
        }
        pConturNou = (ListaFluxType *)LocalLock(pH);
        pConturNou->pObj = pH;
        pConturNou->Urm = NULL;
        sscanf(Poz, "%d" , &pConturNou->Nr);
        if(!pConturCap)
        {
            pConturCurent = pConturCap = pConturNou;
            NrContur = 1;
        }

```

```

        }
        else
            pConturCurent->Urm = pConturNou;
            pConturCurent = pConturNou;
            NrContur++;
            break;

        default :break;
    }
    }
    Poz = pCR + 2;
}
while((Poz < pEnd) && pCR);
pFluxCurent = pFluxCap;
while(pFluxCurent)
{
    CalcCoordFlux(pFluxCurent, REALITATE);
    pFluxCurent = pFluxCurent->Urm;
}
LocalUnlock(pASCII->pObjSir);
LocalUnlock(pASCII->pObj);
LocalFree(pASCII->pObjSir);
LocalFree(pASCII->pObj);
}

DWORD GetDibInfoHeaderSize (BYTE huge * lpDib)
{
###9
// Extrage dimensiune header la fișier bitmap în format DIB
//

    return ((BITMAPINFOHEADER huge *) lpDib)->biSize ;
}

WORD GetDibWidth (BYTE huge * lpDib)
{
###10
// Extrage dimensiune lățime bitmap în format DIB
//

    if (GetDibInfoHeaderSize (lpDib) == sizeof (BITMAPCOREHEADER))
        return (WORD) (((BITMAPCOREHEADER huge *) lpDib)->bcWidth) ;
    else
        return (WORD) (((BITMAPINFOHEADER huge *) lpDib)->biWidth) ;
}

WORD GetDibHeight (BYTE huge * lpDib)
{
###11
// Extrage dimensiune înălțime bitmap în format DIB
//

    if (GetDibInfoHeaderSize (lpDib) == sizeof (BITMAPCOREHEADER))
        return (WORD) (((BITMAPCOREHEADER huge *) lpDib)->bcHeight) ;
    else
        return (WORD) (((BITMAPINFOHEADER huge *) lpDib)->biHeight) ;
}

RGBQUAD *GetStartPalette(BYTE huge * lpDib)
{
    if (GetDibInfoHeaderSize (lpDib) == sizeof (BITMAPCOREHEADER))
        return (RGBQUAD *) (lpDib + (((BITMAPCOREHEADER huge *) lpDib)->bcSize));
    else
        return (RGBQUAD *) (lpDib + (((BITMAPINFOHEADER huge *) lpDib)->biSize));
}

BYTE huge * GetDibBitsAddr (BYTE huge * lpDib, int *NrColors)
{
###12
// Extrage adresa informației grafice din fișier bitmap în format DIB
//

    DWORD dwNumColors, dwColorTableSize ;
    WORD wBitCount ;

    if (GetDibInfoHeaderSize (lpDib) == sizeof (BITMAPCOREHEADER))
    {
        wBitCount = ((BITMAPCOREHEADER huge *) lpDib)->bcBitCount ;

        if (wBitCount != 24)
            dwNumColors = 1L << wBitCount ;
        else

```

```

        dwNumColors = 0 ;
        *NrColors = dwNumColors;
        dwColorTableSize = dwNumColors * sizeof (RGBTRIPLE) ;
    }
    else
    {
        wBitCount = ((BITMAPINFOHEADER huge *) lpDib)->biBitCount ;

        if (GetDibInfoHeaderSize (lpDib) >= 36)
            dwNumColors = ((BITMAPINFOHEADER huge *) lpDib)->biClrUsed ;
        else
            dwNumColors = 0 ;
        *NrColors = dwNumColors;
        if (dwNumColors == 0)
        {
            if (wBitCount != 24)
                dwNumColors = 1L << wBitCount ;
            else
                dwNumColors = 0 ;
        }

        dwColorTableSize = dwNumColors * sizeof (RGBQUAD) ;
    }
    return lpDib + GetDibInfoHeaderSize (lpDib) + dwColorTableSize ;
}

```

**BYTE huge \* ReadDib (char \* szFileName)**

```

///##13

```

```

// Citește conținut fișier în format DIB

```

```

//

```

```

{
    BITMAPFILEHEADER bmfh ;
    BYTE huge *      lpDib ;
    DWORD           dwDibSize, dwOffset, dwHeaderSize ;
    int             hFile ;
    WORD            wDibRead ;

    if (-1 == (hFile = _lopen (szFileName, OF_READ | OF_SHARE_DENY_WRITE)))
        return NULL ;

    if (_read (hFile, (LPSTR) &bmfh, sizeof (BITMAPFILEHEADER)) !=
        sizeof (BITMAPFILEHEADER))
    {
        _lclose (hFile) ;
        return NULL ;
    }

    if (bmfh.bfType != * (WORD *) "BM")
    {
        _lclose (hFile) ;
        return NULL ;
    }

    dwDibSize = bmfh.bfSize - sizeof (BITMAPFILEHEADER) ;
    if (hl = GlobalAlloc (GPTR, dwDibSize))
        lpDib = (BYTE huge * ) GlobalLock (hl) ;
    else
    {
        _lclose (hFile) ;
        return NULL ;
    }

    dwOffset = 0 ;

    while (dwDibSize > 0)
    {
        wDibRead = (WORD) MIN (65535ul, dwDibSize) ;
        if (wDibRead != _read (hFile, (LPSTR) (lpDib + dwOffset), wDibRead))
        {
            _lclose (hFile) ;
            GlobalUnlock (hl) ;
            GlobalFree (hl) ;
            MessageBox (NULL, "Free", "", MB_OK) ;
            return NULL ;
        }
        dwDibSize -= wDibRead ;
        dwOffset += wDibRead ;
    }

    _lclose (hFile) ;
}

```

```

dwHeaderSize = GetDibInfoHeaderSize (lpDib) ;

if (dwHeaderSize < 12 || (dwHeaderSize > 12 && dwHeaderSize < 16))
{
    GlobalUnlock(h1);
    GlobalFree(h1);
    return NULL ;
}
return lpDib ;
}

double ModulReal(ListaNodesType p1, ListaNodesType p2)
{
    ///#14
    // Calculează distanța reală dintre două noduri
    //
    return sqrt(pow(p1.x - p2.x, 2) + pow(p1.y - p2.y,2));
}

double Modul(POINT p1, POINT p2)
{
    ///#15
    // Calculează distanța reală dintre două puncte
    //
    return sqrt(pow(p1.x - p2.x, 2) + pow(p1.y - p2.y,2));
}

double Distanta(ListaNodesType P, double m, double n)
{
    ///#16
    // Calculează distanța de la un punct pe o direcție perpendiculară dată
    //
    return fabs((P.y - m * P.x - n) / sqrt(m * m + 1));
}

void Translatie(ListaNodesType *PDest, ListaNodesType PSource, double Tx, double Ty, double Sc)
{
    ///#17
    // Translatează noduri în spațiul real
    //
    double x, y;

    x = PSource.x / Sc;
    y = PSource.y / Sc;
    PDest->x = (int)(x + Tx);
    PDest->y = (int)(y + Ty);
}

void RotaOZ(ListaNodesType *PDest, ListaNodesType PSource, double Rota)
///#18
// Rotește un nod în jurul OZ
//
{
    double Initial, x, y, Lg;
    ListaNodesType Origine;

    Origine.x = Origine.y = 0;
    Initial = atan2(PSource.y, PSource.x);
    Lg = ModulReal(PSource, Origine);
    x = Lg * cos(Rota + Initial);
    y = Lg * sin(Rota + Initial);
    PDest->x = (float)x;
    PDest->y = (float)y;
}

void RotaMuchie(ListaNodesType *PDest, ListaNodesType PSource, double Rota, double m, double n)
///#19
// Rotește un nod în jurul muchiei plăcuței determinată de punctele P1, P2
//
{
    double Initial, R, d, m1, n1, delta;

    R = Distanta(PSource, m, n);
    d = R * cos(Rota);
    m1 = - 1 / m;
    n1 = PSource.y - m1 * PSource.x;
    delta = atan(m1);
    PDest->x = PSource.x - (R - d) * cos(delta);
    PDest->y = PSource.y - (R - d) * sin(delta);
}

```

```

void FaceElement(int Nr)
{
  ///#20
  // Construiește pe ecran conturul unui element
  //
  ListaElementsType *pElem = pElementCap;
  ListaNodesType P[3];
  int i;

  while(pElem)
  {
    if(pElem->Nr == Nr)
    {
      for(i = 0 ; i < 3 ; i++)
      {
        CautaNod(&P[i], pElem->Nod[i], SCREEN);
        if(i == 0)
          MoveTo(hdc, P[i].xScreen, P[i].yScreen);
        else
          LineTo(hdc, P[i].xScreen, P[i].yScreen);
      }
      LineTo(hdc, P[0].xScreen, P[0].yScreen);
      pElem = pElem->Urm;
    }
  }
}

void Contur(HWND hwnd, POINT Punct[3], double (*Parametri)[1], int Select)
///#21
// Construiește pe ecran conturul modelului cu elemente finite
//
{
  HPEN hPrev, hFlux;
  static double Sc, x, y, alfa, gama, R, d, mSpatiu, nSpatiu, mScreen, nScreen;
  ListaNodesType P[6], P1, Original[15], Ptmp;
  int i;
  double Tx, Ty;
  ListaNodesType *pNod;

  switch (Select)
  {
    case NORMAL:
      pConturCurent = pConturCap;
      for(i = 0 ; i < 15 ; i++)
      {
        CautaNod(&Original[i], pConturCurent->Nr, REALITATE);
        if(!(pConturCurent = pConturCurent->Urm))
          break;
      }

      Sc = ModulReal(Original[0], Original[1]) / Modul(Punct[0], Punct[1]);
      alfa = atan2(Punct[0].y - Punct[1].y, Punct[0].x - Punct[1].x) - M_PI / 2;

      for(i = 0 ; i < 6 ; i++)
      {
        P[i] = Original[i + 3];
        P[i].x += 1;
      }
      hPrev = SelectObject(hdc, GetStockObject(BLACK_PEN));

      // Coordonate Vârf după rotație OZ

      Tx = Punct[0].x - (int)((Original[0].y * cos(M_PI / 2 + alfa) + 1) / Sc);
      Ty = Punct[0].x - (int)(Original[0].y * sin(M_PI / 2 + alfa) / Sc);
      RotaOZ(&P[0], Original[0], alfa);
      Translatie(&P[0], P[0], Tx, Ty, Sc);
      Tx += Punct[0].x - P[0].x;
      Ty += Punct[0].y - P[0].y;

      RotaOZ(&Original[0], Original[0], alfa);
      RotaOZ(&Original[1], Original[1], alfa);
      if(Original[0].x != Original[1].x)
        mSpatiu = (Original[0].y - Original[1].y) / (Original[0].x - Original[1].x);
      else
        mSpatiu = atan(M_PI / 2);
      nSpatiu = Original[0].y - mSpatiu * Original[0].x;
      if(Punct[0].x != Punct[1].x)
        mScreen = (Punct[0].y - Punct[1].y) / (Punct[0].x - Punct[1].x);
      else
        mScreen = atan(M_PI / 2);
      nScreen = Punct[0].y - mScreen * Punct[0].x;
  }
}

```

```

R = Distanta(Original[2] , mSpatiu , nSpatiu) / Sc;
Ptmp.x = Punct[2].x ; Ptmp.y = Punct[2].y;
d = Distanta(Ptmp, mScreen, nScreen);

if((x = d / R) > 1)
    gama = 0;
else
    gama = acos(x);

pNod = pNodCap;
for(i = 0 ; i < NrNodes ; i++)
{
    Ptmp = *pNod;
    RotaOZ(&Ptmp, Ptmp, alfa);
    if(IsRotaMachie)
        RotaMachie(&Ptmp, Ptmp, gama, mSpatiu, nSpatiu);
    Translatie(&Ptmp, Ptmp, Tx, Ty, Sc);
    pNod->xScreen = Ptmp.x;
    pNod->yScreen = Ptmp.y;
    pNod = pNod->Urm;
}

*Parametri[ALFA] = alfa;
*Parametri[GAMA] = gama;
*Parametri[SCARA] = Sc;
*Parametri[TX] = Tx;
*Parametri[TY] = Ty;
break;

case CONTUR :
    pConturCurent = pConturCap;
    CautaNod(&Ptmp, pConturCurent->Nr , SCREEN);
    MoveTo(hdc , Ptmp.xScreen , Ptmp.yScreen );
    while(pConturCurent->Urm)
    {
        pConturCurent = pConturCurent->Urm;
        CautaNod(&Ptmp, pConturCurent->Nr , SCREEN);
        LineTo(hdc , Ptmp.xScreen , Ptmp.yScreen );
    }
    break;

case COMPLET:
    for(i = 0 ; i < NrElements ; i++)
        FaceElement(i);
    break;
}

if(FaceFlx)
{
    hFlux = CreatePen(PS_SOLID, 1, RGB(180, 180, 180));
    SelectObject(hdc, hFlux);
    pFluxCurent = pFluxCap;
    while(pFluxCurent)
    {
        CalcCoordFlux(pFluxCurent, SCREEN);
        MoveTo(hdc , pFluxCurent->Nod[0].xScreen , pFluxCurent->Nod[0].yScreen );
        LineTo(hdc , pFluxCurent->Nod[1].xScreen , pFluxCurent->Nod[1].yScreen );
        pFluxCurent = pFluxCurent->Urm;
    }
    SelectObject(hdc , GetStockObject(BLACK_PEN));
    DeleteObject(hFlux);
}

}

void Marcator(POINT P)
{
    int Lg = 16;

    MoveTo(hdc, P.x - Lg / 2 , P.y);
    LineTo(hdc, P.x + Lg / 2 , P.y);
    MoveTo(hdc, P.x , P.y - Lg / 2);
    LineTo(hdc, P.x , P.y + Lg / 2);
}

int CompColor(COLORREF C1, COLORREF C2)
{
    ///#23
    // Compară valorile a două culori in reprezentare RGB
    //

```



```

    {
        return (C1 & MaskRGB) == (C2 & MaskRGB);
    }

float CautaTemp(ListaNodType *pNod, float (*Paleta)[1], RGBQUAD (*StartPalette)[1], int NrColors)
///##24
// Returnează valoarea temperaturii corespunzătoare unui punct din termogramă,
// raportată la scara gradată
//
{
    static COLORREF Col, CurentColor;
    static RGBQUAD CurentRGB;
    int i;

    Col = GetPixel(hdc, pNod->xScreen, pNod->yScreen);
    for(i = 0 ; i < NrColors ; i++)
    {
        CurentRGB = *StartPalette[i];
        CurentColor = RGB(CurentRGB.rgbRed, CurentRGB.rgbGreen, CurentRGB.rgbBlue);
        if(CompColor(CurentColor, Col))
            return *Paleta[i];
    }
    return -50.0;
}

void Paleta2Tempe(POINT P[2], float (*Paleta)[1], RGBQUAD (*StartPalette)[1], int NrColors)
{
    ///##25
    // Completează vectorul cu valorile temperaturilor corespunzătoare culorilor din
    // paletă
    //
    static POINT P1, P2;
    static float Flux, T1, T2, T3, T4;
    static COLORREF CuloareMax, CuloareMin, CurentColor;
    static int i, IndexMinim = -1, IndexMaxim = -1, R1, G1, B1, R2, G2, B2;
    static RGBQUAD CurentRGB;
    static float LgTemp, TempCurent, Increment;
    static ListaNodType N1, N2;

    LgTemp = MaxTemp - MinTemp;
    if(P[0].y < P[1].y)
    {
        P1 = P[0];
        P[0] = P[1];
        P[1] = P1;
    }
    CuloareMax = GetPixel(hdc, P[0].x, P[0].y);
    CuloareMin = GetPixel(hdc, P[1].x, P[1].y);

    R1 = GetRValue(CuloareMax); G1 = GetGValue(CuloareMax);
    B1 = GetBValue(CuloareMax);
    R2 = GetRValue(CuloareMin); G2 = GetGValue(CuloareMin);
    B2 = GetBValue(CuloareMin);
    for(i = 0 ; i < NrColors ; i++)
    {
        CurentRGB = *StartPalette[i];
        CurentColor = RGB(CurentRGB.rgbRed, CurentRGB.rgbGreen, CurentRGB.rgbBlue);
        if(CompColor(CurentColor, CuloareMax) && (IndexMaxim == -1))
        {
            IndexMaxim = i;
            if(IndexMinim != -1)
                break;
        }
        else
            if(CompColor(CurentColor, CuloareMin) && (IndexMinim == -1))
            {
                IndexMinim = i;
                if(IndexMaxim != -1)
                    break;
            }
    }
    if((IndexMinim == -1) || (IndexMaxim == -1))
    {
        MessageBox((HWND)NULL, (LPCSTR)"Culoare inexistentă în paletă", "Paleta", MB_ICONSTOP
    );
        return;
    }
    if(IndexMinim >= IndexMaxim)

```

```

    {
        MessageBox((HWND)NULL , (LPCSTR)"Index incorect in paletă" , "Paleta" , MB_ICONSTOP );
        return;
    }
    Increment = LgTemp / (IndexMaxim - IndexMinim);
    TempCurent = MinTemp;
    for(i = 0 ; i < NrColors ; i++)
    {
        if((i >= IndexMinim) && (i <= IndexMaxim))
        {
            *Paleta[i] = TempCurent;
            TempCurent += Increment;
        }
    }
    pNodCurent = pNodCap;
    for(i = 0 ; i < NrNodes ; i++)
    {
        pNodCurent->Sol = CautaTemp(pNodCurent, Paleta, StartPalette, NrColors);
        pNodCurent = pNodCurent->Urm;
    }
    if(FaceFlx)
    {
        P1.x = pNodCap->xScreen;          P1.y = pNodCap->yScreen;
        P2.x = pNodCap->Urm->xScreen;     P2.y = pNodCap->yScreen;
        Scara = ModulReal(*pNodCap , *pNodCap->Urm) / Modul(P1, P2);
        pFluxCurent = pFluxCap;
        T1, T3 pe linia de flux și T2, T4 pe contur
        while(pFluxCurent)
        {
            T1 = CautaTemp(&(pFluxCurent->Nod[0]), Paleta, StartPalette, NrColors);
            CautaNod(&N1, pFluxCurent->Nod[0].Nr, SCREEN);
            T2 = CautaTemp(&N1 , Paleta, StartPalette, NrColors);

            T3 = CautaTemp(&(pFluxCurent->Nod[1]), Paleta, StartPalette, NrColors);
            CautaNod(&N2, pFluxCurent->Nod[1].Nr, SCREEN);
            T4 = CautaTemp(&N2 , Paleta, StartPalette, NrColors);
            pFluxCurent->Nod[0].Sol = (T1 + T3) / 2;      // Media t° pe linia de flux
            pFluxCurent->Nod[1].Sol = (T2 + T4) / 2;      // Media t° pe contur
            pFluxCurent = pFluxCurent->Urm;
        }
    }
}

int PASCAL WinMain (HANDLE hInstance, HANDLE hPrevInstance,
                    LPSTR lpszCmdLine, int nCmdShow)
{
    ///#26
    // Funcție principală
    //
    {
        HWND      hwnd ;
        MSG       msg ;
        WNDCLASS  wndclass ;

        if (!hPrevInstance)
        {
            wndclass.style          = CS_HREDRAW | CS_VREDRAW | CS_DBLCLKS ;
            wndclass.lpfnWndProc    = WndProc ;
            wndclass.cbClsExtra     = 0 ;
            wndclass.cbWndExtra     = 0 ;
            wndclass.hInstance      = hInstance ;
            wndclass.hIcon          = LoadIcon (hInstance, szIconName ) ;
            wndclass.hCursor        = LoadCursor (NULL, IDC_ARROW) ;
            wndclass.hbrBackground  = GetStockObject (WHITE_BRUSH) ;
            wndclass.lpszMenuName   = szMenuName ;
            wndclass.lpszClassName  = szAppName ;
            RegisterClass (&wndclass) ;

            wndclass.style          = CS_HREDRAW | CS_VREDRAW | CS_OWNDC;
            wndclass.lpfnWndProc    = WndProcCoord ;
            wndclass.hbrBackground  = GetStockObject (LTGRAY_BRUSH) ;
            wndclass.cbWndExtra     = sizeof(int);
            wndclass.hIcon          = (HICON)NULL;
            wndclass.hCursor        = (HCURSOR)NULL;
            wndclass.lpszMenuName   = (HMENU)NULL;
            wndclass.lpszClassName  = Coord;
            RegisterClass (&wndclass) ;
        }

        _fstrcpy(Lin , lpszCmdLine);
        hwnd = CreateWindow (szAppName, "Termograme",
                            WS_OVERLAPPEDWINDOW,
                            CW_USEDEFAULT, CW_USEDEFAULT,
                            CW_USEDEFAULT, CW_USEDEFAULT,
                            NULL, NULL, hInstance, NULL) ;
    }
}

```

```

ShowWindow (hwnd, SW_SHOWMAXIMIZED) ;
UpdateWindow (hwnd) ;

while (GetMessage (&msg, NULL, 0, 0))
{
    TranslateMessage (&msg) ;
    DispatchMessage (&msg) ;
}
return msg.wParam ;
}

void InfoDisplay (POINT *ret)
{
    ///##27
    // Returnează dimensiunile curente ale display-ului
    //

    ret->x = GetDeviceCaps(hdc, HORZSIZE);
    ret->y = GetDeviceCaps(hdc, VERTSIZE);
}

HWND FaceWindowCoord (HWND hwnd, int care)
{
    ///##28
    // Creează ferestre pentru informații auxiliare
    //

    POINT P;
    int Start, Lat, Inalt;

    InfoDisplay (&P);
    LPtoDP (hdc, &P, 1);
    Start = (int)(P.x * 2.37);    Lat = (int)(P.x * 0.47);    Inalt = (int)(P.x * 0.095);
    switch (care)
    {
        case 1:
            return CreateWindow (Coord, NULL, WS_CHILDWINDOW | WS_VISIBLE
                , Start, 0, 1.5 * Lat, Inalt, hwnd, (HMENU)NULL, GetWindowWord (hwnd,
GWW_HINSTANCE), NULL) ;
        case 2:
            return CreateWindow (Coord, NULL, WS_CHILDWINDOW | WS_VISIBLE
                , Start + 1.5 * Lat, 0, 2 * Lat, Inalt, hwnd, (HMENU)NULL,
                GetWindowWord (hwnd, GWW_HINSTANCE), NULL) ;
        case 3:
            return CreateWindow (Coord, NULL, WS_CHILDWINDOW | WS_VISIBLE
                , Start + 2 * Lat, Inalt, Lat, Inalt, hwnd, (HMENU)NULL, GetWindowWord (hwnd,
GWW_HINSTANCE), NULL) ;
        case 4:
            return CreateWindow (Coord, NULL, WS_CHILDWINDOW | WS_VISIBLE
                , Start + 2 * Lat, 2 * Inalt, Lat, Inalt, hwnd, (HMENU)NULL, GetWindowWord (hwnd,
GWW_HINSTANCE), NULL) ;
        case 5:
            return CreateWindow (Coord, NULL, WS_CHILDWINDOW | WS_VISIBLE
                , Start + 2 * Lat, 3 * Inalt, Lat, Inalt, hwnd, (HMENU)NULL, GetWindowWord (hwnd,
GWW_HINSTANCE), NULL) ;
        case 6:
            return CreateWindow (Coord, NULL, WS_CHILDWINDOW | WS_VISIBLE
                , Start + 2 * Lat, 4 * Inalt, Lat, Inalt, hwnd, (HMENU)NULL, GetWindowWord (hwnd,
GWW_HINSTANCE), NULL) ;
    }
}

long FAR PASCAL _export WndProc (HWND hwnd, UINT message, UINT wParam,
LONG lParam)

///##29
// Funcția care tratează mesajele ferestrei principale
//

{
    static int        IsCheck, i, NrColors;
    static char       szFileName [ _MAX_PATH ],
    static char       szTitleName [ _MAX_FNAME + _MAX_EXT ] ;
    static char       *szFilter[] = {"DIB Files (*.DIB)", "*.dib",
        "Bitmap Files (*.BMP)", "*.bmp", "Termo Files (*.TRM)", "*.trm",
        "ASCII DataBaseFile FEM (*.ADB)", "*.adb", "" } ;

    static OFSTRUCT   StFile;
    static LPBITMAPINFO lpDib;
    static RGBQUAD    (*StartPalette)[1];

```

```

static COLORREF    Culoare;
static OPENFILENAME ofn ;
static short      cxClient, cyClient ;
static WORD       wDisplay = IDM_ACTUAL ;
static POINT      Punctate[3], CapScala[2], Ptmp;
int               Lg;
HDC               hdc1 ;
static HMENU      hMenu ;
BYTE huge *      lpDibBits ;
    static HWND    hCoord1, hCoord2, hDIB, hPST, hMax, hMin;
PAINTSTRUCT      ps ;
short            cxDib, cyDib ;
static int        Punct = 0, LastCommand, Titlu = 0;
static float      TempInPaleta[256], Valoare;
static double     Parametri[5];
static RECT       Rect;
static FARPROC    lpfnTemperaturiDlgProc, lpfnDespreDlgProc, lpfnDistDlgProc;
static HANDLE     hInstance;
static char       Lin[80], NumeDIB[15], NumePST[15];
static HFILE      PFile;
static ListaNodesType N1, N2;

switch (message)
{
    case WM_CREATE:
        hInstance = ((LPCREATESTRUCT) lParam)->hInstance;
        ofn.lStructSize = sizeof (OPENFILENAME) ;
        ofn.hwndOwner = hwnd ;
        ofn.lpstrFilter = szFilter [0] ;
        ofn.lpstrFile = szFileName ;
        ofn.nMaxFile = _MAX_PATH ;
        ofn.lpstrFileTitle = szTitleName ;
        ofn.nMaxFileTitle = _MAX_FNAME + _MAX_EXT ;
        ofn.lpstrDefExt = "dib" ;
        if( _fstrlen(Lin))
        {
            PostMessage(hwnd, WM_COMMAND, IDM_OPEN, 1L);
        }
        lpfnTemperaturiDlgProc = MakeProcInstance((FARPROC)TemperaturiDlgProc, hInstance);
        lpfnDespreDlgProc = MakeProcInstance((FARPROC)DespreDlgProc , hInstance);
        lpfnDistDlgProc = MakeProcInstance((FARPROC)DistDlgProc , hInstance);
        hdc = GetDC(hwnd);
            hCoord1 = FaceWindowCoord(hwnd,1);
            hCoord2 = FaceWindowCoord(hwnd,2);
            hDIB = FaceWindowCoord(hwnd,3);
            hPST = FaceWindowCoord(hwnd,4);
            hMax = FaceWindowCoord(hwnd,5);
            hMin = FaceWindowCoord(hwnd,6);

        GetClientRect(hwnd, &Rect);
        SetViewportOrg(hdc , 0 , Rect.bottom);
        SetMapMode(hdc, MM_LOMETRIC);
        return 0 ;
    case WM_SIZE:
        cxClient = LOWORD (lParam) ;
        cyClient = HIWORD (lParam) ;
        return 0 ;
    case WM_COMMAND:
        hMenu = GetMenu (hwnd) ;
        switch (wParam)
        {
            case ID_OPEN:
                LastCommand = ID_OPEN;
                ofn.lpstrDefExt = "dib" ;
                ofn.lpstrFilter = szFilter [0] ;

                if(lParam == 1L)
                {
                    lpDib = ReadDib (Lin) ;
                    InvalidateRect (hwnd, NULL, TRUE) ;
                }
                else
                {
                    if (GetOpenFileName (&ofn))
                    {
                        if (lpDib != NULL)
                        {
                            GlobalUnlock(h1);
                            GlobalFree(h1);
                            lpDib = NULL ;
                        }
                    }
                }
            }
        }

```

```

        lpDib = ReadDib (szFileName) ;
        if (lpDib == NULL)
            MessageBox (hwnd, szAppName,
                "Nu se poate deschide fişier DIB",
                MB_ICONEXCLAMATION | MB_OK) ;
        InvalidateRect (hwnd, NULL, TRUE) ;
        _fstrcpy(NumeDIB , _fstrchr(szFileName, '\\') + 1);
        SetTimer(hwnd, 5, 500, NULL);
    }
}
return 0 ;

case ID_MODEL:
    LastCommand = ID_MODEL;
    ofn.lpstrDefExt      = "trm" ;
    ofn.lpstrFilter      = szFilter [4] ;

    if (GetOpenFileName (&ofn))
    {
        if(pNodCap)
            Dealloc();
        CitesteModel(szFileName, hwnd);
        _fstrcpy(NumePST , _fstrchr(szFileName, '\\') + 1);
        SetTimer(hwnd, 5, 500, NULL);
    }
    return 0 ;

case ID_ADB :
    LastCommand = ID_ADB;
    CheckMenuItem(hMenu, (UINT)ID_CONTUR , MF_BYCOMMAND | MF_UNCHECKED);
    CheckMenuItem(hMenu, (UINT)ID_COMPLET, MF_BYCOMMAND | MF_UNCHECKED);
    SendMessage(hwnd, WM_PAINT, 0, 0L);
    IsCheck = GetMenuState(hMenu, (UINT)ID_FLUX_CALCUL, MF_BYCOMMAND) & MF_CHECKED;
    strcpy(szFileName, "");
    ofn.lpstrDefExt      = "adb" ;
    ofn.lpstrFilter      = szFilter [6] ;
    if(GetSaveFileName(&ofn))
    {
        if(PFile = OpenFile(szFileName, &StFile , OF_CREATE))
        {
            pNodCurent = pNodCap;
            for(i = 0 ; i < NrNodes; i++)
            {
                lstrcpy(Lin, "");
                sprintf(Lin, "TEMPE,%d,%f", pNodCurent->Nr, pNodCurent->Sol);
                lstrcat(Lin , "\015\012");
                Lg = strlen(Lin);
                _lwrite(PFile , Lin , Lg);
            }
            pNodCurent = pNodCurent->Urm;
            pFluxCurent = pFluxCap;
            if(IsCheck)
            {
                while(pFluxCurent)
                {
                    lstrcpy(Lin, "");
                    CautaNod(&N1, pFluxCurent->Nod[0].Nr, REALITATE);
                    CautaNod(&N2, pFluxCurent->Nod[1].Nr, REALITATE);
                    Valoare = ModulReal(N1, N2); //
                    Lungimea laturii
                    sprintf(Lin, "DFLUX,%d,%d,%d,%f,%f,%f,%f", pFluxCurent->Nr, pFluxCurent->Nod[0].Nr,
                        pFluxCurent->Nod[1].Nr, pFluxCurent->Nod[0].Sol, pFluxCurent->Nod[1].Sol, Valoare, DistFlux * Scara);
                    lstrcat(Lin , "\015\012");
                    Lg = strlen(Lin);
                    _lwrite(PFile , Lin , Lg);
                    pFluxCurent = pFluxCurent->Urm;
                }
            }
            _lclose(PFile);
        }
    }
    else
        MessageBox(hwnd, "Nu se poate scrie in fişier adb", "Fişier ADB", MB_ICONSTOP);
}

return 0 ;
case ID_SAV_DIB:
    LastCommand = ID_SAV_DIB;
    return 0 ;

case ID_IESIRE:

```

```

        LastCommand = ID_IESIRE;
        SendMessage(hwnd, WM_DESTROY, 0, 0L);
        return 0;

    case ID_PUNCT:
        LastCommand = ID_PUNCT;
        Punct = 1;
        return 0;

//
    case ID_NOD:
        LastCommand = ID_NOD;
        return 0;

    case ID_CONTUR:
        LastCommand = ID_CONTUR;
        IsCheck = GetMenuState(hMenu, (UINT)ID_CONTUR, MF_BYCOMMAND) & MF_CHECKED;
        CheckMenuItem(hMenu, (UINT)ID_CONTUR, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED : MF_CHECKED));
        if(!IsCheck)
            CheckMenuItem(hMenu, (UINT)ID_COMPLET, MF_BYCOMMAND | IsCheck);
        InvalidateRect(hwnd, NULL, TRUE);
        return 0;

    case ID_COMPLET:
        LastCommand = ID_COMPLET;
        IsCheck = GetMenuState(hMenu, (UINT)ID_COMPLET, MF_BYCOMMAND) & MF_CHECKED;
        CheckMenuItem(hMenu, (UINT)ID_COMPLET, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED : MF_CHECKED));
        if(!IsCheck)
            CheckMenuItem(hMenu, (UINT)ID_CONTUR, MF_BYCOMMAND | IsCheck);
        InvalidateRect(hwnd, NULL, TRUE);
        return 0;

    case ID_ROTA:
        LastCommand = ID_ROTA;
        IsCheck = GetMenuState(hMenu, (UINT)ID_ROTA, MF_BYCOMMAND) & MF_CHECKED;
        IsRotaMuchia = !IsCheck;
        CheckMenuItem(hMenu, (UINT)ID_ROTA, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED : MF_CHECKED));
        break;

    case ID_TEMPERATURI:
        LastCommand = ID_TEMPERATURI;
        DialogBox(hInstance, "DialogTemperaturi", hwnd, lpfnTemperaturiDlgProc);
        sprintf(Lin, "TMax: %d °C", MaxTemp);
        SendMessage(hMax, WM_USER, (WORD)fstrlen(Lin), (DWORD)Lin);
        sprintf(Lin, "TMin: %d °C", MinTemp);
        SendMessage(hMin, WM_USER, (WORD)fstrlen(Lin), (DWORD)Lin);
        return 0;

    case ID_SCALA:
        LastCommand = ID_SCALA;
        Punct = 4;
        return 0;

    case ID_REFRESH:
        LastCommand = ID_REFRESH;
        InvalidateRect(hwnd, NULL, TRUE);
        return 0;

    case ID_FLUX_CONTUR:
        LastCommand = ID_FLUX_CONTUR;
        IsCheck = GetMenuState(hMenu, (UINT)ID_FLUX_CONTUR, MF_BYCOMMAND) & MF_CHECKED;
        CheckMenuItem(hMenu, (UINT)ID_FLUX_CONTUR, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED : MF_CHECKED));
        FaceFlx = !IsCheck;
        return 0;

    case ID_FLUX_CALCUL:
        IsCheck = GetMenuState(hMenu, (UINT)ID_FLUX_CALCUL, MF_BYCOMMAND) & MF_CHECKED;
        CheckMenuItem(hMenu, (UINT)ID_FLUX_CALCUL, MF_BYCOMMAND | (IsCheck ? MF_UNCHECKED : MF_CHECKED));
        return 0;

    case ID_FLUX_DISTANTA:
        LastCommand = ID_FLUX_DISTANTA;
        if(DialogBox(hInstance, "DIALOGDISTANTA", hwnd, lpfnDistDlgProc)
            Paleta2Tempe(CapScala, &TempInPaleta, StartPalette, NrColors);

        return 0;

    case ID_DESPRE:
        LastCommand = ID_DESPRE;
        DialogBox(hInstance, "DIALOGDESPRE", hwnd, lpfnDespreDlgProc);
        return 0;
    }
    return 0;

case WM_TIMER:

```

```

        Titlu = 1;
        return 0;

    case WM_PAINT:
        hdc1 = BeginPaint (hwnd, &ps) ;
        SetTimer(hwnd, 5, 500, NULL);
        if (lpDib != NULL)
        {
            lpDibBits = GetDibBitsAddr (lpDib, &NrColors);
            StartPalette = GetStartPalette(lpDib);
            cxDib = GetDibWidth (lpDib) ;
            cyDib = GetDibHeight (lpDib) ;
            SetStretchBltMode (hdc1, STRETCH_ANDSCANS);
            if (wDisplay == IDM_ACTUAL)
                i = SetDIBitsToDevice (hdc1, 0, 0, cxDib, cyDib, 0, 0,
                    0, cyDib, (LPSTR) lpDibBits,
                    (LPBITMAPINFO) lpDib,
                    DIB_RGB_COLORS) ;
            else
                StretchDIBits (hdc1, 0, 0, cxClient, cyClient,
                    0, 0, cxDib, cyDib,
                    (LPSTR) lpDibBits,
                    (LPBITMAPINFO) lpDib,
                    DIB_RGB_COLORS, SRCCOPY) ;
        }
        if (Parametri[TX] != 0)
        {
            if (GetMenuState(hMenu, (UINT)ID_COMPLET, MF_BYCOMMAND) & MF_CHECKED)
                Contur(hwnd, Punctate, &Parametri, COMPLET);
            else
                if (GetMenuState(hMenu, (UINT)ID_CONTUR, MF_BYCOMMAND) & MF_CHECKED)
                    Contur(hwnd, Punctate, &Parametri, CONTUR);
        }
        EndPaint (hwnd, &ps) ;
        return 0 ;

    case WM_RBUTTONDOWN:
        SendMessage(hwnd, WM_COMMAND, LastCommand, 0L);
        return 0;

        case WM_MOUSEMOVE :
            if (Titlu)
            {
                Titlu = 0;
                KillTimer(hwnd, 5);
                SendMessage(hDIB, WM_USER, _fstrlen(NumeDIB) , (DWORD)NumeDIB);
                SendMessage(hPST, WM_USER, _fstrlen(NumePST) , (DWORD)NumePST);
                sprintf(Lin, "TMax: %d °C", MaxTemp);
                SendMessage(hMax, WM_USER, (WORD)_fstrlen(Lin), (DWORD)Lin);
                sprintf(Lin, "TMin: %d °C", MinTemp);
                SendMessage(hMin, WM_USER, (WORD)_fstrlen(Lin), (DWORD)Lin);
            }

            Ptmp.x = LOWORD(lParam);
            Ptmp.y = HIWORD(lParam);
            DPToLP(hdc, &Ptmp, 1);
            Culoare = GetPixel(hdc, Ptmp.x, Ptmp.y);
            sprintf(Lin, "R:%3d|G:%3d|B:%3d      ", GetRValue(Culoare),
                GetGValue(Culoare), GetBValue(Culoare));
            SendMessage(hCoord1, WM_USER, (WORD)_fstrlen(Lin), (DWORD)Lin);
            sprintf(Lin, "X: %6d | Y: %6d      ", Ptmp.x , Ptmp.y);
            SendMessage(hCoord2, WM_USER, _fstrlen(Lin), (DWORD)Lin);

            return 0;

    case WM_LBUTTONDOWN:
        switch (Punct)
        {
            case 1:
                Punctate[0].x = LOWORD(lParam);
                Punctate[0].y = HIWORD(lParam);
                DPToLP(hdc, &Punctate, 1);
                Marcator(Punctate[0]);
                Punct = 2;
                break;

            case 2:
                Punctate[1].x = LOWORD(lParam);
                Punctate[1].y = HIWORD(lParam);
                DPToLP(hdc, &Punctate[1], 1);
                Marcator(Punctate[1]);
                Punct = 3;
                break;

            case 3:

```



```

Punct = 0;
Punctate[2].x = LOWORD(lParam);
Punctate[2].y = HIWORD(lParam);
DPToLP(hdc, &Punctate[2], 1);
Marcator(Punctate[2]);
Contur(hwnd, Punctate, &Parametri, NORMAL);
    if(GetMenuState(hMenu, (UINT)ID_COMPLET, MF_BYCOMMAND) & MF_CHECKED)
        Contur(hwnd, Punctate, &Parametri, COMPLET);
    else
        if(GetMenuState(hMenu, (UINT)ID_CONTUR, MF_BYCOMMAND) & MF_CHECKED)
            Contur(hwnd, Punctate, &Parametri, CONTUR);

break;

case 4:
Punct = 5;
CapScala[0].x = LOWORD(lParam);
CapScala[0].y = HIWORD(lParam);
//     Ptmp = CapScala[0];
//     DPToLP(hdc, &CapScala[0], 1);
//     Marcator(Ptmp);
break;
case 5:
Punct = 0;
CapScala[1].x = LOWORD(lParam);
CapScala[1].y = HIWORD(lParam);
Ptmp = CapScala[1];
//     DPToLP(hdc, &CapScala[1], 1);
//     Marcator(Ptmp);
Paleta2Tempe(CapScala, &TempInPaleta, StartPalette, NrColors);
    EnableMenuItem(hMenu, ID_ADB, MF_ENABLED);

break;

default:;
}
return 0;

case WM_LBUTTONDOWN:
    InvalidateRect(hwnd, NULL, TRUE);
    return 0;

case WM_DESTROY:
    ReleaseDC(hwnd, hdc);
    if (lpDib != NULL)
        GlobalFreePtr(lpDib);

    PostQuitMessage(0);
    return 0;
}
return DefWindowProc(hwnd, message, wParam, lParam);
}

```

```

BOOL FAR PASCAL _export TemperaturiDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
{
    ///#30
    // Funcție care trtatează dialog temperaturi
    //

    switch (message)
    {
        case WM_INITDIALOG:
            WriteMemoryToItem(hDlg, IDC_MAX, &MaxTemp, IntType, 10);
            WriteMemoryToItem(hDlg, IDC_MIN, &MinTemp, IntType, 10);

            return TRUE;

        case WM_COMMAND:
            switch (wParam)
            {
                case IDOK:
                    ReadItemToMemory(hDlg, IDC_MAX, 5, &MaxTemp, IntType, 10);
                    ReadItemToMemory(hDlg, IDC_MIN, 5, &MinTemp, IntType, 10);

                    EndDialog(hDlg, 0);

                    return TRUE;

                case IDCANCEL:
                    EndDialog(hDlg, 0);

                    return TRUE;
                    break;
            }
            break;
    }
    return FALSE;
}

```

```

BOOL FAR PASCAL _export DistDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)
///#31
// Funcție care tratează dialog distanță calcul flux
//

{
    switch (message)
    {
        case WM_INITDIALOG:
            WriteMemoryToItem(hDlg, IDC_DIST, &DistFlux, IntType, 10);
            return TRUE;
        case WM_COMMAND:
            switch (wParam)
            {
                case IDOK:
                    ReadItemToMemory(hDlg, IDC_DIST, 5, &DistFlux, IntType, 10);
                    EndDialog (hDlg, 1);
                    return TRUE;
                break;
                case IDCANCEL:
                    EndDialog (hDlg, 0);
                    return TRUE;
                break;
            }
            break;
    }
    return FALSE;
}

```

```

long FAR PASCAL _export WndProcCoord (HWND hwnd, UINT message, UINT wParam,
LONG lParam)

```

```

///#32
// Funcție care tratează mesaje pentru ferestre auxiliare
//

{
    HDC hdc;
    static HANDLE hInstCoord;
    static int Index = 0;

    switch (message)
    {
        case WM_CREATE :
            switch(Index)
            {
                case 0:
                    hdc = GetDC(hwnd);
                    Index++;
                    SetTextColor(hdc, RGB(255, 0, 0));
                    ReleaseDC(hwnd, hdc);
                    break;
                case 5:
                    hdc = GetDC(hwnd);
                    Index++;
                    SetTextColor(hdc, RGB(0, 0, 255));
                    ReleaseDC(hwnd, hdc);
                    Index++;
                    break;
                case 4:
                    hdc = GetDC(hwnd);
                    SetTextColor(hdc, RGB(255, 0, 0));
                    ReleaseDC(hwnd, hdc);
                case 3:
                case 2:
                case 1:
                    Index++;
                    break;
            }
            hInstCoord = ((LPCREATESTRUCT) lParam)->hInstance;
            return 0;

        case WM_USER :
            hdc = GetDC(hwnd);
            TextOut(hdc, 4, 2, (char *)lParam, wParam);
            ReleaseDC(hwnd, hdc);
            return 0;

        case WM_DESTROY:
            Index = 0;
            return 0;
    }
    return DefWindowProc (hwnd, message, wParam, lParam);
}

```

```
}  
  
BOOL FAR PASCAL _export DespreDlgProc (HWND hDlg, UINT message, UINT wParam, LONG lParam)  
//###33  
// Funcție care tratează dialog despre  
//  
{  
    switch (message)  
    {  
        case WM_INITDIALOG:  
            return TRUE ;  
        case WM_COMMAND:  
            switch (wParam)  
            {  
                case IDOK:  
                case IDCANCEL:  
                    EndDialog (hDlg, 0) ;  
                    return TRUE ;  
            }  
            break ;  
    }  
    return FALSE ;  
}
```

## 12. ANEXA 5



## 13. ANEXA 6 - CORECȚIA DE EMISIVITATE

### 13.1. FUNCȚIA CARE REZOLVĂ ECUAȚIA DE CORECȚIE

**Sub CautaEmisivitatea()**

```

Dim Wk As Worksheet
Dim Dia As DialogSheet
NumeDia$ = "DiaNod"
Plăcuță = 1
Corp = 2

Set Dia = DialogSheets(NumeDia$)

If Dia.Show <> False Then
Tabel$ = Dia.EditBoxes("NumeWk").Text
NrNodes = Val(Dia.EditBoxes("NrNod").Text)
Set Wk = Worksheets(Tabel$)
For i = 2 To NrNodes + 1
Wk.Cells(i, 6) = 1
Curent$ = "$F" & LTrim(Str$(i))
Tmediu = Wk.Cells(5, 8) ^ 4
ValoareMăsurată = Wk.Cells(i, 3) ^ 4
TLiber = ValoareMăsurată - Tmediu
If Wk.Cells(i, 2) = Plăcuță Then
Sir$ = "=(H3*" & Curent$ & "6+I3*" & Curent$ & "5+J3*" & Curent$
Sir$ = Sir$ & "4+K3*" & Curent$ & "3+L3*" & Curent$ & "2+M3*" & Curent$ & "+N3)*"
Emisivitatea$ = Right(Sir$, Len(Sir$) - 1)
Sir$ = Sir$ & Curent$ & "4"
Sir$ = Sir$ & "-" & Emisivitatea$ & "$H5^4" 'Str$(Tmediu)
Else
Sir$ = "=(H4*" & Curent$ & "6+I4*" & Curent$ & "5+J4*" & Curent$
Sir$ = Sir$ & "4+K4*" & Curent$ & "3+L4*" & Curent$ & "2+M4*" & Curent$ & "+N4)*"
Emisivitatea$ = Right(Sir$, Len(Sir$) - 1)
Sir$ = Sir$ & Curent$ & "4"
Sir$ = Sir$ & "-" & Emisivitatea$ & "$H5^4"
End If
Wk.Cells(i, 4).Formula = Sir$
Wk.Cells(i, 4).GoalSeek TLiber, Wk.Cells(i, 6)
Wk.Cells(i, 5).Formula = "=POWER($C" & LTrim(Str$(i)) & ", 1/4)"
Next i
End If
End Sub

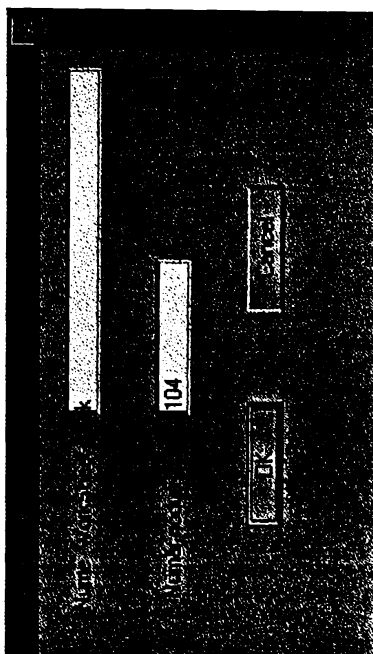
```

**Observație :** Fereastra de dialog de unde se citesc variabilele NrNodes și Tabel\$ este vizibilă în pagina 277

### 13.2. FOAIA DE CALCUL PENTRU CORECȚIA DE EMISIVITATE

Nod	Material	T° Măsurată	T° Reală	Emisivități						Diferența	Procent
				x^6	x^5	x^4	x^3	x^2	x^1		
1	1	192.658264	1.38E+09	3.725606	251.4875					58.82924	23.39%
2	1	138.987366	3.73E+08	3.433554	182.2723					43.28466	23.75%
3	1	179.24054	1.03E+09	3.658972	234.3135					55.07297	23.50%
4	1	172.531677	8.86E+08	3.624242	225.6897					53.16803	23.55%
5	1	152.40509	5.39E+08	3.513558	199.6624					47.28732	23.68%
6	1	149.050659	4.93E+08	3.494085	195.3434					46.2927	23.70%
7	1	196.012695	1.48E+09	3.741718	255.7643					59.75163	23.36%
8	1	155.759521	5.88E+08	3.532756	204.0372					48.27766	23.66%
9	1	142.341797	4.1E+08	3.454086	186.6331					44.29134	23.73%
10	1	169.177246	8.19E+08	3.606496	221.3694					52.19216	23.58%
11	1	172.531677	8.86E+08	3.624242	225.6897					53.16803	23.55%
12	1	179.24054	1.03E+09	3.658972	234.3135					55.07297	23.50%
13	1	266.45575	5.04E+09	4.040232	343.5649					77.10912	22.44%
14	1	222.848145	2.47E+09	3.863369	299.7017					66.95351	23.08%

KC990/D1  
Cuțit  
Tmediu



94	2	102.088623	1.08E+08	3.178662	106.8276					4.738874	4.44%
95	2	95.379761	82600846	3.125101	99.807					4.427241	4.44%
96	2	105.443054	1.23E+08	3.204457	110.3375					4.894469	4.44%
97	2	102.088623	1.08E+08	3.178662	106.8276					4.738874	4.44%
98	2	98.734192	94872095	3.152223	103.3173					4.583136	4.44%
99	2	112.151917	1.58E+08	3.254256	117.3672					5.205267	4.44%
100	2	105.443054	1.23E+08	3.204457	110.3375					4.894469	4.44%
101	1	263.101318	4.79E+09	4.027456	339.4869					76.38657	22.50%
102	2	92.02533	71558225	3.097254	86.2965					4.271171	4.44%
103	2	91	68414961	3.088591	95.27343					4.223428	4.44%
104	1	162.468384	6.97E+08	3.570197	212.7131					50.24469	23.62%
										77.82084	23.75%
										22.39%	

21

3

## 14. ANEXA 7 - EXEMPLU DE TRATARE A UNEI TERMOGRAMME

Pentru exemplificare s-a ales termograma din figura 79, care introdusă în programul Termo, arată ca în figura 3:



Fig. 3 - Termograma ENIA \ KC850 \ 153 \ 0,25 \ 0,5 în programul Termo

În paragraful 5.2 se pot observa variațiile de temperatură înainte și după corecția de emisivitate, precum și variația temperaturii pe conturul modelului și pe linia de calcul a fluxului în cazul acestei termograme.

După introducerea acestor date, obținute din foaia de calcul Excel, în programul FEM și rularea problemei inverse, se obțin pentru elementele 20, 20, 23, 64, 40, o valoare de  $70[\text{W}/\text{mm}^2]$ , pentru densitatea fluxului termic. Această valoare ulterior a fost folosită pentru obținerea prin regresie a relațiilor de estimare.

În scopul verificării datelor, cu aceste valori s-a creat fișierul de intrare de mai jos, pentru problema directă.

Pentru că la descrierea modelului geometric s-a folosit ca unitate de măsură milimetri iar valorile surselor termice sunt exprimate în  $[\text{W}/\text{m}^2]$ , fluxul pe contur este exprimat în  $[\text{W}/\text{m}]$ <sup>66</sup>, constantele de material și mediu sunt exprimate în SI, a fost nevoie de corelarea unităților de măsură, care s-a făcut cu linia de control a execuției programului SCARAGEO<sup>67</sup>,  $1.0\text{E}-3$ , care asigură acest lucru.

<sup>66</sup> Conturul este liniar, fiind în discuție o analiză a unei probleme plane.

<sup>67</sup> Linia de control SCARAGEO afectează numai coordonatele nodale



```
#TITLE
PBL DIRECTA - EN1 \ KC850 \ 153 \ 0,25 \ 0,5
//...
// DEFINIRE 104 NODURI
//...
#NODES
1,-1.00000000,21.0000000,0.00000000E+00
2,10.0000000,12.0000000,0.00000000E+00
3,1.20000000,19.2000000,0.00000000E+00
4,3.40000000,17.4000000,0.00000000E+00
5,5.60000000,15.6000000,0.00000000E+00
6,7.80000000,13.8000000,0.00000000E+00
7,11.0000000,27.0000000,0.00000000E+00
8,10.16666667,14.5000000,0.00000000E+00
9,10.33333333,17.0000000,0.00000000E+00
10,10.50000000,19.5000000,0.00000000E+00
11,10.66666667,22.0000000,0.00000000E+00
12,10.83333333,24.5000000,0.00000000E+00
13,-1.00000000,36.0000000,0.00000000E+00
14,8.60000000,28.8000000,0.00000000E+00
15,6.20000000,30.6000000,0.00000000E+00
16,3.80000000,32.4000000,0.00000000E+00
17,1.40000000,34.2000000,0.00000000E+00
18,-1.00000000,33.0000000,0.00000000E+00
19,-1.00000000,30.0000000,0.00000000E+00
20,-1.00000000,27.0000000,0.00000000E+00
21,-1.00000000,24.0000000,0.00000000E+00
22,7.50000000,23.5000000,0.00000000E+00
23,5.00000000,26.0000000,0.00000000E+00
24,6.76776695,25.2677670,0.00000000E+00
25,5.00000000,21.0000000,0.00000000E+00
26,6.76776695,21.7322330,0.00000000E+00
27,2.50000000,23.5000000,0.00000000E+00
28,3.23223305,21.7322330,0.00000000E+00
29,3.23223305,25.2677670,0.00000000E+00
30,1.99756595,20.3350318,0.00000000E+00
31,8.59530874,25.7556485,0.00000000E+00
32,1.14585625,22.5950860,0.00000000E+00
33,3.96873320,19.8141154,0.00000000E+00
34,8.63103471,18.4714706,0.00000000E+00
35,8.24924941,20.9654792,0.00000000E+00
36,3.93381300,29.5277777,0.00000000E+00
37,1.72421719,27.8970603,0.00000000E+00
38,8.12357529,16.3318147,0.00000000E+00
39,9.14523857,23.3239121,0.00000000E+00
40,1.43794000,31.2083333,0.00000000E+00
41,6.41667678,27.8483764,0.00000000E+00
42,1.07416069,25.0977851,0.00000000E+00
43,6.15707189,18.7496501,0.00000000E+00
44,6.52356920,20.5718572,0.00000000E+00
45,4.44619916,27.0071479,0.00000000E+00
46,0.00000000E+00,0.0000E+00,0.00000000E+00
47,25.0000000,0.00000000E+00,0.000000E+00
48,3.57142857,0.00000000E+00,0.000000E+00
49,7.14285714,0.00000000E+00,0.000000E+00
50,10.7142857,0.00000000E+00,0.000000E+00
51,14.2857143,0.00000000E+00,0.000000E+00
52,17.8571429,0.00000000E+00,0.000000E+00
53,21.4285714,0.00000000E+00,0.000000E+00
54,25.0000000,14.0000000,0.00000000E+00
55,25.0000000,3.50000000,0.00000000E+00
56,25.0000000,7.00000000,0.00000000E+00
57,25.0000000,10.5000000,0.00000000E+00
58,10.9277583,25.9163743,0.00000000E+00
59,22.1855517,16.3832749,0.00000000E+00
60,19.3711033,18.7665497,0.00000000E+00
61,16.5566550,21.1498246,0.00000000E+00
62,13.7422066,23.5330994,0.00000000E+00
63,0.192675736E-13,20.1818182,0.00000000E+00
64,0.160563114E-13,16.8181818,0.00000000E+00
65,0.128450491E-13,13.4545455,0.00000000E+00
66,0.963378682E-14,10.0909091,0.000000E+00
67,0.642252455E-14,6.72727273,0.000000E+00
68,0.321126227E-14,3.36363636,0.000000E+00
69,21.9836803,12.5994059,0.00000000E+00
70,8.54763666,3.11774117,0.00000000E+00
71,12.3498836,3.08544909,0.00000000E+00
72,16.0348675,3.01144592,0.00000000E+00
73,16.0795120,17.3109916,0.00000000E+00
74,2.59178452,11.6097096,0.00000000E+00
75,2.73475383,8.40817174,0.00000000E+00
76,13.4581535,20.0166346,0.00000000E+00
77,2.51724161,14.8598934,0.00000000E+00
```

```
78,4.62807094,3.09893130,0.00000000E+00
79,19.1513436,2.82851618,0.00000000E+00
80,21.7680303,4.70632165,0.00000000E+00
81,21.8622773,8.73689610,0.00000000E+00
82,19.1762865,14.9514115,0.00000000E+00
83,2.70120435,5.54639072,0.00000000E+00
84,13.0235824,16.7766668,0.00000000E+00
85,5.06231240,12.9809687,0.00000000E+00
86,13.0904534,13.4540659,0.00000000E+00
87,7.55192012,11.1881523,0.00000000E+00
88,18.1482386,6.53763382,0.00000000E+00
89,18.8016145,11.1002054,0.00000000E+00
90,5.93892577,6.45759586,0.00000000E+00
91,14.7750397,15.2362880,0.00000000E+00
92,5.25323376,9.88719809,0.00000000E+00
93,14.1598084,6.20565613,0.00000000E+00
94,16.1715050,13.6949469,0.00000000E+00
95,10.1837984,6.44268136,0.00000000E+00
96,15.1560885,9.93427593,0.00000000E+00
97,7.80437436,8.69634581,0.00000000E+00
98,10.0321115,9.59149478,0.00000000E+00
99,12.2056324,10.6394501,0.00000000E+00
100,12.3530406,8.53328210,0.00000000E+00
101,0.2,33.6,0.0E+00
102,1.77,1.68,0.0E+00
103,23.2,1.75,0.0E+00
104,9.0,14.1,0.0E+00
//...
// DEFINIRE 166 ELEMENTE
//...
#ELEMENTS
1,258,1
2,104,6
2,258,1
101,13,18
3,258,1
7,31,58
4,258,1
14,31,7
5,258,1
4,30,3
6,258,1
58,31,12
7,258,1
32,28,27
8,258,1
30,28,32
9,258,1
33,28,30
10,258,1
25,28,33
11,258,1
34,10,35
12,258,1
10,34,9
13,258,1
11,35,10
14,258,1
16,36,15
15,258,1
20,37,19
16,258,1
1,30,32
17,258,1
9,38,8
18,258,1
4,33,30
19,258,1
6,104,38
20,258,1
101,18,40
21,258,1
12,39,11
22,258,1
17,40,16
23,258,1
19,40,18
24,258,1
15,41,14
25,258,1
31,14,41
26,258,1
21,42,20
27,258,1
```

37,20,42	50,70,49
28,258,1	70,258,2
6,38,5	51,71,50
29,258,1	71,258,2
38,9,34	52,72,51
30,258,1	72,258,2
35,11,39	61,73,60
31,258,1	73,258,2
39,12,31	66,74,65
32,258,1	74,258,2
41,15,36	67,75,66
33,258,1	75,258,2
36,16,40	71,51,72
34,258,1	76,258,2
40,19,37	74,66,75
35,258,1	77,258,2
1,32,21	76,62,11
36,258,1	78,258,2
4,43,33	62,76,61
37,258,1	79,258,2
42,21,32	65,77,64
38,258,1	80,258,2
5,43,4	4,64,77
39,258,1	81,258,2
5,38,43	102,78,68
40,258,1	82,258,2
36,40,37	53,79,52
41,258,1	83,258,2
38,34,43	49,78,48
42,258,1	84,258,2
42,32,27	72,52,79
43,258,1	85,258,2
25,33,43	56,80,55
44,258,1	86,258,2
42,27,29	53,103,80
45,258,1	87,258,2
31,24,22	81,57,69
46,258,1	88,258,2
37,42,29	57,81,56
47,258,1	89,258,2
26,25,44	60,82,59
48,258,1	90,258,2
25,43,44	69,59,82
49,258,1	91,258,2
31,41,24	68,83,67
50,258,1	92,258,2
45,36,37	75,67,83
51,258,1	93,258,2
41,36,45	79,53,80
52,258,1	94,258,2
45,37,29	49,70,78
53,258,1	95,258,2
39,31,22	80,56,81
54,258,1	96,258,2
41,45,24	82,60,73
55,258,1	97,258,2
45,29,23	73,61,76
56,258,1	98,258,2
34,35,43	76,11,10
57,258,1	99,258,2
35,39,22	5,4,77
58,258,1	100,258,2
45,23,24	77,65,74
59,258,1	101,258,2
35,22,26	83,68,78
60,258,1	102,258,2
35,44,43	84,10,9
61,258,1	103,258,2
35,26,44	76,10,84
62,258,2	104,258,2
47,103,53	5,77,85
63,258,2	105,258,2
46,102,68	6,5,85
64,258,1	106,258,1
13,101,17	3,30,63
65,258,1	107,258,2
17,101,40	84,9,86
66,258,2	108,258,2
59,69,54	87,6,85
67,258,2	109,258,2
54,69,57	2,6,87
68,258,2	110,258,2
70,50,71	73,76,84
69,258,2	111,258,2

85,77,74	98,95,100
112,258,2	150,258,2
88,79,80	93,96,100
113,258,2	151,258,2
72,79,88	96,86,99
114,258,2	152,258,2
89,69,82	99,98,100
115,258,2	153,258,2
81,69,89	96,99,100
116,258,2	154,258,2
90,83,78	64,3,63
117,258,2	155,258,2
75,83,90	3,64,4
118,258,2	156,258,2
88,80,81	2,99,86
119,258,2	157,258,2
92,74,75	2,86,8
120,258,2	158,258,2
70,90,78	8,86,9
121,258,2	159,258,2
73,84,91	12,11,62
122,258,2	160,258,2
85,74,92	58,12,62
123,258,2	161,258,2
93,72,88	48,102,46
124,258,2	162,258,2
71,72,93	48,78,102
125,258,2	163,258,2
94,82,73	55,103,47
126,258,2	164,258,2
89,82,94	103,55,80
127,258,2	165,258,1
88,81,89	8,104,2
128,258,2	166,258,1
94,73,91	8,38,104
129,258,2	#MATERIALS
91,84,86	//K - > [W/m/K]
130,258,2	MATID,1,NAME,PLACUTA
87,85,92	MATID,1,KX,10.2
131,258,2	MATID,1,EMIS,0.3331,-7.00E-07,2.00E-07,-
92,75,90	1.00E-09,5.00E-12,-6.00E-15,3.00E-18
132,258,2	MATID,2,NAME,CORP
70,95,90	MATID,2,KX,19.0
133,258,2	MATID,2,EMIS,0.8328,1.00E-05,3.00E-08,-
93,88,96	6.00E-10,3.00E-12,-4.00E-15,2.00E-18
134,258,2	#CONST
70,71,95	TEMPMEDIU,25
135,258,2	BOLTZMANN,5.66961E-8
71,93,95	#BC
136,258,2	TEMPE,47,36
88,89,96	TEMPE,53,36
137,258,2	TEMPE,55,36
92,90,97	DFLUX,2,13,18,-3.0E4
138,258,2	DFLUX,23,18,19,-1.0E4
96,89,94	DFLUX,15,19,20,-3.0E3
139,258,2	DFLUX,3,58,7,-4.0E3
94,91,86	DFLUX,14,15,16,-4.0E3
140,258,2	DFLUX,22,16,17,-4.0E4
87,92,97	DFLUX,64,17,13,-2.0E4
141,258,2	#LOADS
97,90,95	ESOURCE,2,7.0E7
142,258,2	ESOURCE,20,7.0E7
98,87,97	ESOURCE,23,7.0E7
143,258,2	ESOURCE,64,7.0E7
2,87,98	ESOURCE,40,7.0E7
144,258,2	#CONTROL
99,2,98	GAUSS,ELABORAT
145,258,1	PRECITER,1.0E-2
63,30,1	MAXITER,30
146,258,2	MODBC,ELIMINARE
96,94,86	PRECVER,1.0E-02
147,258,2	SCARAGEO,1.0E-3
98,97,95	#END
148,258,2	
93,100,95	
149,258,2	

După rularea problemei de mai sus, se obțin următoarele soluții pentru temperaturile nodale:

Anexa 7 - Exemplu tratare termogramă

---- SOLUTII ----			
1	118.0178237	53	36.13414938
2	49.38864579	54	34.3582126
3	71.79256581	55	36.13414938
4	74.00700424	56	28.98417447
5	65.13160638	57	37.90521586
6	56.186252	58	74.00700424
7	67.35532541	59	32.57599667
8	49.38864579	60	39.67245361
9	62.90370241	61	43.19838629
10	62.90370241	62	43.19838629
11	82.84427698	63	91.65820865
12	78.42923809	64	51.9833881
13	185.6952894	65	43.19838629
14	62.90370241	66	39.67245361
15	71.79256581	67	41.43664503
16	124.593504	68	37.90521586
17	181.3501722	69	36.13414938
18	181.3501722	70	37.90521586
19	126.7842647	71	36.13414938
20	80.63755314	72	37.90521586
21	80.63755314	73	41.43664503
22	60.67085011	74	43.19838629
23	115.8247823	75	39.67245361
24	67.35532541	76	44.95813741
25	78.42923809	77	44.95813741
26	62.90370241	78	37.90521586
27	91.65820865	79	36.13414938
28	76.2191339	80	39.67245361
29	113.6311561	81	39.67245361
30	89.45642955	82	39.67245361
31	60.67085011	83	41.43664503
32	65.13160638	84	43.19838629
33	91.65820865	85	37.90521586
34	85.04957608	86	39.67245361
35	74.00700424	87	41.43664503
36	93.85900971	88	32.57599667
37	93.85900971	89	37.90521586
38	76.2191339	90	37.90521586
39	71.79256581	91	43.19838629
40	157.3934055	92	41.43664503
41	71.79256581	93	36.13414938
42	78.42923809	94	41.43664503
43	85.04957608	95	39.67245361
44	82.84427698	96	37.90521586
45	124.593504	97	39.67245361
46	39.67245361	98	39.67245361
47	37.90521586	99	37.90521586
48	39.67245361	100	41.43664503
49	36.1341493	101	183.5231802
50	30.7855608	102	44.95813741
51	34.3582126	103	37.90521586
52	39.67245361	104	53.93179244

\*Cu valorile temperaturilor nodale astfel calculate, s-a obținut postprocesarea din figura 4, folosind programul Post. După cum se poate observa, distribuția de temperaturi este asemănătoare cu cea din figura unde este reprezentată termograma, iar valorile temperaturilor nodale sunt apropiate de cele obținute după aplicarea corecției de emisivitate. Valorile calculate aici s-au introdus apoi în fișierul următor, unde se face rularea aceleiași probleme pentru cazul invers, când se caută valorile densității fluxului termic ce au produs distribuția câmpului termic.

```
#TITLE
PBL INVERSA -EN1 \ KC850 \ 153 \ 0,25 \ 0,5
// DEFINIRE 104 NODURI
///...
#NODES
1,-1.00000000,21.0000000,0.000000000E+00
2,10.00000000,12.0000000,0.000000000E+00
3,1.20000000,19.2000000,0.000000000E+00
4,3.40000000,17.4000000,0.000000000E+00
5,5.60000000,15.6000000,0.000000000E+00
6,7.80000000,13.8000000,0.000000000E+00
7,11.0000000,27.0000000,0.000000000E+00
8,10.1666667,14.5000000,0.000000000E+00
9,10.3333333,17.0000000,0.000000000E+00
10,10.5000000,19.5000000,0.000000000E+00
11,10.6666667,22.0000000,0.000000000E+00
12,10.8333333,24.5000000,0.000000000E+00
13,-1.00000000,36.0000000,0.000000000E+00
14,8.60000000,28.8000000,0.000000000E+00
15,6.20000000,30.6000000,0.000000000E+00
16,3.80000000,32.4000000,0.000000000E+00
17,1.40000000,34.2000000,0.000000000E+00
18,-1.00000000,33.0000000,0.000000000E+00
19,-1.00000000,30.0000000,0.000000000E+00
20,-1.00000000,27.0000000,0.000000000E+00
21,-1.00000000,24.0000000,0.000000000E+00
22,7.50000000,23.5000000,0.000000000E+00
23,5.00000000,26.0000000,0.000000000E+00
24,6.76776695,25.2677670,0.000000000E+00
25,5.00000000,21.0000000,0.000000000E+00
26,6.76776695,21.7322330,0.000000000E+00
27,2.50000000,23.5000000,0.000000000E+00
28,3.23223305,21.7322330,0.000000000E+00
29,3.23223305,25.2677670,0.000000000E+00
30,1.99756595,20.3350318,0.000000000E+00
31,8.59530874,25.7556485,0.000000000E+00
32,1.14585625,22.5950860,0.000000000E+00
33,3.96873320,19.8141154,0.000000000E+00
34,8.63103471,18.4714706,0.000000000E+00
35,8.24924941,20.9654792,0.000000000E+00
36,3.93381300,29.5277777,0.000000000E+00
37,1.72421719,27.8970603,0.000000000E+00
38,8.12357529,16.3318147,0.000000000E+00
39,9.14523857,23.3239121,0.000000000E+00
40,1.43794000,31.2083333,0.000000000E+00
41,6.41667678,27.8483764,0.000000000E+00
42,1.07416069,25.0977851,0.000000000E+00
43,6.15707189,18.7496501,0.000000000E+00
44,6.52356920,20.5718572,0.000000000E+00
45,4.44619916,27.0071479,0.000000000E+00
46,0.00000000,0.000000000E+00,0.000000000E+00
47,25.0000000,0.000000000E+00,0.000000000E+00
48,3.57142857,0.000000000E+00,0.000000000E+00
49,7.14285714,0.000000000E+00,0.000000000E+00
50,10.7142857,0.000000000E+00,0.000000000E+00
51,14.2857143,0.000000000E+00,0.000000000E+00
52,17.8571429,0.000000000E+00,0.000000000E+00
53,21.4285714,0.000000000E+00,0.000000000E+00
54,25.0000000,14.0000000,0.000000000E+00
55,25.0000000,3.50000000,0.000000000E+00
56,25.0000000,7.00000000,0.000000000E+00
57,25.0000000,10.5000000,0.000000000E+00
58,10.9277583,25.9163743,0.000000000E+00
59,22.1855517,16.3832749,0.000000000E+00
60,19.3711033,18.7665497,0.000000000E+00
61,16.5566550,21.1498246,0.000000000E+00
62,13.7422066,23.5330994,0.000000000E+00
63,0.192675736E-13,20.1818182,0.000000000E+00
64,0.160563114E-13,16.8181818,0.000000000E+00
65,0.128450491E-13,13.4545455,0.000000000E+00
66,0.963378682E-14,10.0909091,0.000000000E+00
67,0.642252455E-14,6.72727273,0.000000000E+00
68,0.321126227E-14,3.36363636,0.000000000E+00
69,21.9836803,12.5994059,0.000000000E+00
70,8.54763666,3.11774117,0.000000000E+00
71,12.3498836,3.08544909,0.000000000E+00
72,16.0348675,3.01144592,0.000000000E+00
73,16.0795120,17.3109916,0.000000000E+00
74,2.59178452,11.6097096,0.000000000E+00
75,2.73475383,8.40817174,0.000000000E+00
76,13.4581535,20.0166346,0.000000000E+00
77,2.51724161,14.8598934,0.000000000E+00
78,4.62807094,3.09893130,0.000000000E+00
```

```
79,19.1513436,2.82851618,0.000000000E+00
80,21.7680303,4.70632165,0.000000000E+00
81,21.8622773,8.73689610,0.000000000E+00
82,19.1762865,14.9514115,0.000000000E+00
83,2.70120435,5.54639072,0.000000000E+00
84,13.0235824,16.7766668,0.000000000E+00
85,5.06231240,12.9809687,0.000000000E+00
86,13.0904534,13.4540659,0.000000000E+00
87,7.55192012,11.1881523,0.000000000E+00
88,18.1482386,6.53763382,0.000000000E+00
89,18.8016145,11.1002054,0.000000000E+00
90,5.93892577,6.45759586,0.000000000E+00
91,14.7750397,15.2362880,0.000000000E+00
92,5.25323376,9.88719809,0.000000000E+00
93,14.1598084,6.20565613,0.000000000E+00
94,16.1715050,13.6949469,0.000000000E+00
95,10.1837984,6.44268136,0.000000000E+00
96,15.1560885,9.93427593,0.000000000E+00
97,7.80437436,8.69634581,0.000000000E+00
98,10.0321115,9.59149478,0.000000000E+00
99,12.2056324,10.6394501,0.000000000E+00
100,12.3530406,8.53328210,0.000000000E+00
101,0.2,33.6,0.0E+00
102,1.77,1.68,0.0E+00
103,23.2,1.75,0.0E+00
104,9.0,14.1,0.0E+00
///...
// DEFINIRE 166 ELEMENTE
///...
#ELEMENTS
1,258,1
2,104,6
2,258,1
101,13,18
3,258,1
7,31,58
4,258,1
14,31,7
5,258,1
4,30,3
6,258,1
58,31,12
7,258,1
32,28,27
8,258,1
30,28,32
9,258,1
33,28,30
10,258,1
25,28,33
11,258,1
34,10,35
12,258,1
10,34,9
13,258,1
11,35,10
14,258,1
16,36,15
15,258,1
20,37,19
16,258,1
1,30,32
17,258,1
9,38,8
18,258,1
4,33,30
19,258,1
6,104,38
20,258,1
101,18,40
21,258,1
12,39,11
22,258,1
17,40,16
23,258,1
19,40,18
24,258,1
15,41,14
25,258,1
31,14,41
26,258,1
21,42,20
27,258,1
37,20,42
```

28,258,1	70,258,2
6,38,5	51,71,50
29,258,1	71,258,2
38,9,34	52,72,51
30,258,1	72,258,2
35,11,39	61,73,60
31,258,1	73,258,2
39,12,31	66,74,65
32,258,1	74,258,2
41,15,36	67,75,66
33,258,1	75,258,2
36,16,40	71,51,72
34,258,1	76,258,2
40,19,37	74,66,75
35,258,1	77,258,2
1,32,21	76,62,11
36,258,1	78,258,2
4,43,33	62,76,61
37,258,1	79,258,2
42,21,32	65,77,64
38,258,1	80,258,2
5,43,4	4,64,77
39,258,1	81,258,2
5,38,43	102,78,68
40,258,1	82,258,2
36,40,37	53,79,52
41,258,1	83,258,2
38,34,43	49,78,48
42,258,1	84,258,2
42,32,27	72,52,79
43,258,1	85,258,2
25,33,43	56,80,55
44,258,1	86,258,2
42,27,29	53,103,80
45,258,1	87,258,2
31,24,22	81,57,69
46,258,1	88,258,2
37,42,29	57,81,56
47,258,1	89,258,2
26,25,44	60,82,59
48,258,1	90,258,2
25,43,44	69,59,82
49,258,1	91,258,2
31,41,24	68,83,67
50,258,1	92,258,2
45,36,37	75,67,83
51,258,1	93,258,2
41,36,45	79,53,80
52,258,1	94,258,2
45,37,29	49,70,78
53,258,1	95,258,2
39,31,22	80,56,81
54,258,1	96,258,2
41,45,24	82,60,73
55,258,1	97,258,2
45,29,23	73,61,76
56,258,1	98,258,2
34,35,43	76,11,10
57,258,1	99,258,2
35,39,22	5,4,77
58,258,1	100,258,2
45,23,24	77,65,74
59,258,1	101,258,2
35,22,26	83,68,78
60,258,1	102,258,2
35,44,43	84,10,9
61,258,1	103,258,2
35,26,44	76,10,84
62,258,2	104,258,2
47,103,53	5,77,85
63,258,2	105,258,2
46,102,68	6,5,85
64,258,1	106,258,1
13,101,17	3,30,63
65,258,1	107,258,2
17,101,40	84,9,86
66,258,2	108,258,2
59,69,54	87,6,85
67,258,2	109,258,2
54,69,57	2,6,87
68,258,2	110,258,2
70,50,71	73,76,84
69,258,2	111,258,2
50,70,49	85,77,74

112,258,2  
 88,79,80  
 113,258,2  
 72,79,88  
 114,258,2  
 89,69,82  
 115,258,2  
 81,69,89  
 116,258,2  
 90,83,78  
 117,258,2  
 75,83,90  
 118,258,2  
 88,80,81  
 119,258,2  
 92,74,75  
 120,258,2  
 70,90,78  
 121,258,2  
 73,84,91  
 122,258,2  
 85,74,92  
 123,258,2  
 93,72,88  
 124,258,2  
 71,72,93  
 125,258,2  
 94,82,73  
 126,258,2  
 89,82,94  
 127,258,2  
 88,81,89  
 128,258,2  
 94,73,91  
 129,258,2  
 91,84,86  
 130,258,2  
 87,85,92  
 131,258,2  
 92,75,90  
 132,258,2  
 70,95,90  
 133,258,2  
 93,88,96  
 134,258,2  
 70,71,95  
 135,258,2  
 71,93,95  
 136,258,2  
 88,89,96  
 137,258,2  
 92,90,97  
 138,258,2  
 96,89,94  
 139,258,2  
 94,91,86  
 140,258,2  
 87,92,97  
 141,258,2  
 97,90,95  
 142,258,2  
 98,87,97  
 143,258,2  
 2,87,98  
 144,258,2  
 99,2,98  
 145,258,1  
 63,30,1  
 146,258,2  
 96,94,86  
 147,258,2  
 98,97,95  
 148,258,2  
 93,100,95  
 149,258,2  
 98,95,100  
 150,258,2  
 93,96,100  
 151,258,2  
 96,86,99  
 152,258,2  
 99,98,100  
 153,258,2  
 96,99,100

154,258,2  
 64,3,63  
 155,258,2  
 3,64,4  
 156,258,2  
 2,99,86  
 157,258,2  
 2,86,8  
 158,258,2  
 8,86,9  
 159,258,2  
 12,11,62  
 160,258,2  
 58,12,62  
 161,258,2  
 48,102,46  
 162,258,2  
 48,78,102  
 163,258,2  
 55,103,47  
 164,258,2  
 103,55,80  
 165,258,1  
 8,104,2  
 166,258,1  
 8,38,104  
 #MATERIALS  
 //K - > [W/m/K]  
 MATID,1,NAME,PLACUTA  
 MATID,1,KX,10.2  
 MATID,1,EMIS,0.3331,-7.00E-07,2.00E-07,-  
 1.00E-09,5.00E-12,-6.00E-15,3.00E-18  
 MATID,2,NAME,CORP  
 MATID,2,KX,19.0  
 MATID,2,EMIS,0.8328,1.00E-05,3.00E-08,-  
 6.00E-10,3.00E-12,-4.00E-15,2.00E-18  
 #CONST  
 TEMPEDIU,25  
 BOLTZMANN,5.66961E-8  
 #BCINI  
 TEMPE,47,36  
 TEMPE,53,36  
 TEMPE,55,36  
 #BC  
 TEMPE,1,107.8747000331354000  
 TEMPE,2,73.2272019272310200  
 TEMPE,3,95.7619007641456000  
 TEMPE,4,87.9726419426191500  
 TEMPE,5,82.2851893361119000  
 TEMPE,6,77.5034193172925500  
 TEMPE,7,105.7375623236474000  
 TEMPE,8,75.9834263384789400  
 TEMPE,9,79.5051260037760400  
 TEMPE,10,83.7829642718583300  
 TEMPE,11,89.2234404836073600  
 TEMPE,12,96.1937971539488400  
 TEMPE,13,184.8867642342144000  
 TEMPE,14,123.1355340607034000  
 TEMPE,15,142.5746224440110000  
 TEMPE,16,159.1641113112272000  
 TEMPE,17,177.7203166136674000  
 TEMPE,18,184.7387595694353000  
 TEMPE,19,167.9741040980162000  
 TEMPE,20,145.3470693432035000  
 TEMPE,21,125.2270355212297000  
 TEMPE,22,100.6153575858926000  
 TEMPE,23,132.1259217076186000  
 TEMPE,24,117.1624430672072000  
 TEMPE,25,93.7005483184555800  
 TEMPE,26,92.6103904663561000  
 TEMPE,27,120.7691719032179000  
 TEMPE,28,103.6921836666650000  
 TEMPE,29,134.3974089185557000  
 TEMPE,30,99.8021272803789600  
 TEMPE,31,110.4536836522337000  
 TEMPE,32,115.1890393183852000  
 TEMPE,33,94.0909173080104400  
 TEMPE,34,84.5444308095643300  
 TEMPE,35,90.2316779159819600  
 TEMPE,36,151.7188748659260000  
 TEMPE,37,150.8876680175544000  
 TEMPE,38,81.2322167560755300  
 TEMPE,39,96.9322962522118400  
 TEMPE,40,171.6520760717116000



TEMPE, 41, 130.6025220712512000	TEMPE, 84, 76.8036837642222600
TEMPE, 42, 132.4315153469641000	TEMPE, 85, 78.8888948832621100
TEMPE, 43, 88.1421290550035800	TEMPE, 86, 72.2346045140420600
TEMPE, 44, 91.0937627437118400	TEMPE, 87, 74.7284589832847400
TEMPE, 45, 137.9583230216248000	TEMPE, 88, 57.6520189217022900
TEMPE, 46, 71.3455305251684600	TEMPE, 89, 63.4171728200824000
TEMPE, 47, 36.0000000000000000	TEMPE, 90, 71.7162569381308700
TEMPE, 48, 70.6869314132272600	TEMPE, 91, 72.8878218355153700
TEMPE, 49, 68.5535001720325600	TEMPE, 92, 75.0537995129562500
TEMPE, 50, 64.9586115240638800	TEMPE, 93, 63.1601136405773000
TEMPE, 51, 59.6083265372816100	TEMPE, 94, 69.4589872941418200
TEMPE, 52, 51.5587471917076500	TEMPE, 95, 68.1678758526021800
TEMPE, 53, 36.0000000000000000	TEMPE, 96, 65.9326454715399100
TEMPE, 54, 62.4363767779396600	TEMPE, 97, 72.0885261333354800
TEMPE, 55, 36.0000000000000000	TEMPE, 98, 70.9068737006802900
TEMPE, 56, 50.8640460046059600	TEMPE, 99, 69.8477415678396900
TEMPE, 57, 58.4819430914020900	TEMPE, 100, 67.5657009544633400
TEMPE, 58, 100.7591450477308000	TEMPE, 101, 183.1220583518614000
TEMPE, 59, 66.2843433427690300	TEMPE, 102, 71.3072565736584600
TEMPE, 60, 71.8990125610024400	TEMPE, 103, 38.3523534616385000
TEMPE, 61, 78.6201336917637800	TEMPE, 104, 76.7737993153398000
TEMPE, 62, 87.1335747444860000	DFLUX, 2, 13, 18, -3.0E4
TEMPE, 63, 101.1682034302856000	DFLUX, 23, 18, 19, -1.0E4
TEMPE, 64, 88.7084981118561000	DFLUX, 15, 19, 20, -3.0E3
TEMPE, 65, 81.6989885399162700	DFLUX, 3, 58, 7, -4.0E3
TEMPE, 66, 77.0020654798382200	DFLUX, 14, 15, 16, -4.0E3
TEMPE, 67, 73.8313383475386800	DFLUX, 22, 16, 17, -4.0E4
TEMPE, 68, 72.0742692001878400	DFLUX, 64, 17, 13, -2.0E4
TEMPE, 69, 62.6177260260813100	#LOADS
TEMPE, 70, 67.9183539428831200	ESOURCE, 2, 7.0E7
TEMPE, 71, 63.5046086326233200	ESOURCE, 20, 7.0E7
TEMPE, 72, 57.3018353359807800	ESOURCE, 23, 7.0E7
TEMPE, 73, 74.1782947864210600	ESOURCE, 64, 7.0E7
TEMPE, 74, 78.4262052949616800	ESOURCE, 40, 7.0E7
TEMPE, 75, 74.8249073422827800	#CONTROL
TEMPE, 76, 81.4103712990497400	GAUSS, ELABORAT
TEMPE, 77, 83.5286061867482800	PRECITER, 1.0E-2
TEMPE, 78, 70.7713737002842200	MAXITER, 30
TEMPE, 79, 50.0953797601682200	MODBC, ELIMINARE
TEMPE, 80, 48.2085854141935400	PRECVER, 1.0E-02
TEMPE, 81, 56.8042104437237500	SCARAGEO, 1.0E-3
TEMPE, 82, 67.8608145041370200	#END
TEMPE, 83, 72.6344565545777400	

După rularea acestui fișier, se obțin pentru sursele din elementele amintite valorile de mai jos, care sunt exprimate în [W/m<sup>2</sup>].

```

---- SOLUTII ----
 2      70000000.0000003700000000
20      70000000.0000027700000000
23      69999999.9999993900000000
64      69999999.9999901700000000
40      69999999.9999982100000000
    
```

După cum se observă, diferențele față de valorile din fișierul direct, unde s-au folosit surse cu puterea termică de  $7 \cdot 10^7$  [W/m<sup>2</sup>], sunt ne semnificative, ceea ce dovedește corectitudinea algoritmilor de rezolvare pentru problema inversă.

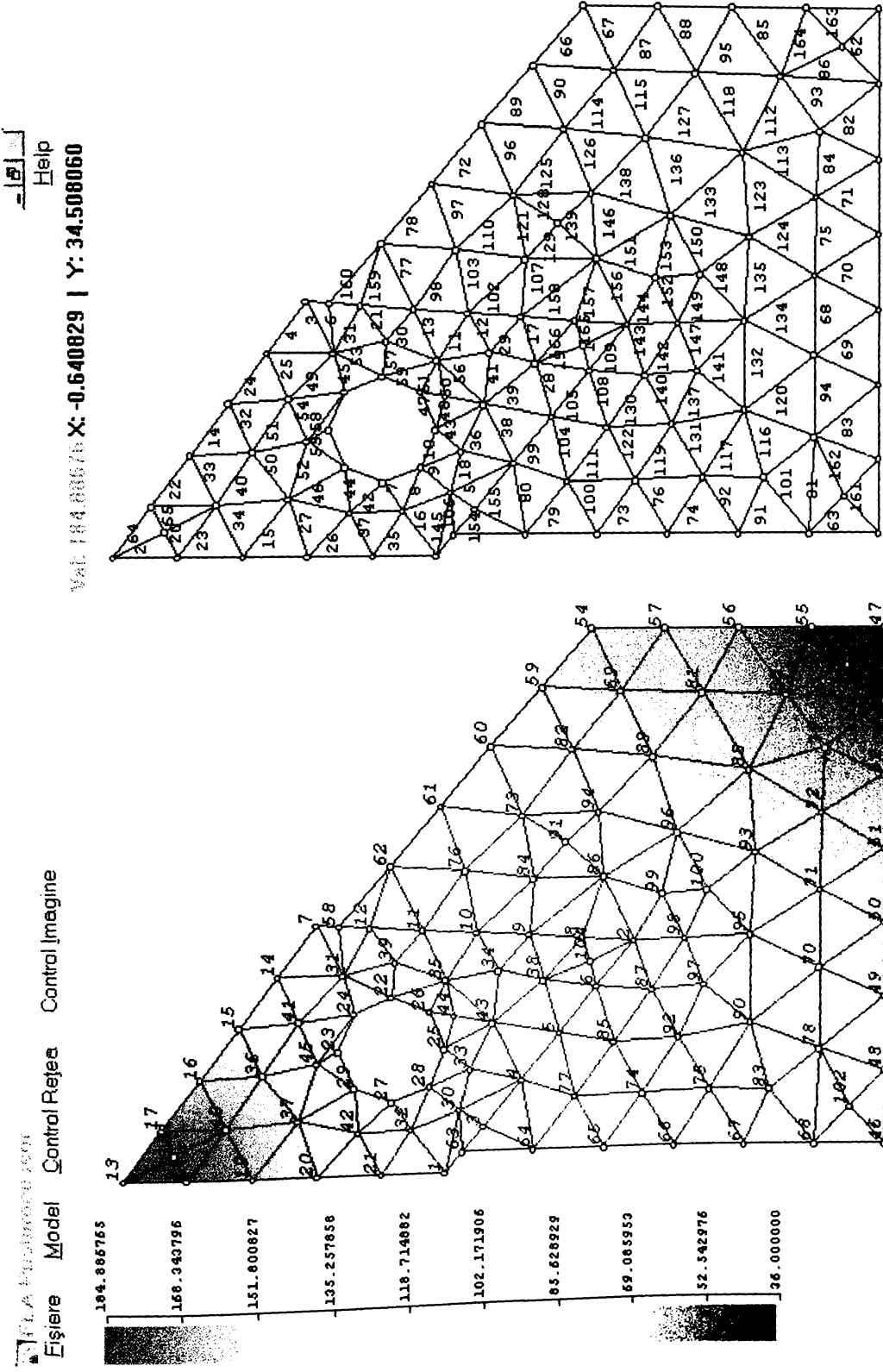


Fig. 4 - Postprocesare pentru rularea problemei cu datele anterioare

## 15. ANEXA 8 - PROGRAM REPREZENTARE GRAFICĂ

### 15.1. CARACTERISTICI

- Programul trasează graficul unei funcții de două variabile, folosind entități 3DSurface din AutoCAD;
- Format fișier Text - funcție externă AutoLisp
- Limbaj de programare Lisp;
- structură de meniuri;
- funcția de comandă : GF
- funcția care implementează relația de doua varianile reprezentată grafic: FUNC

#### Fișiere de intrare

Fișierul cu codul funcției:

format : ASCII

extensia implicită : LSP

Fișiere de ieșire : Nu este cazul

Obs: Cu acest program s-au trasat graficele ce rprezintă variația  $Q=f(F,v)$  din finalul cap. 5;  
În continuare este prezentat codul, cu valorile parametrilor pentru aproximarea debitului la  
strunjirea aluminiului cu plăcuța K68.

```
(DEFUN FUNC (XL YL)
  (SETQ RET (+ (* V2 YL) (* F2 XL) (* FV XL YL)
  (* V YL) (* F XL) LIB))
)

(DEFUN BLOC (NUME MULTIME)
  (IF (= (TBLSEARCH "BLOCK" NUME) nil)
    (COMMAND "BLOCK" NUME "0,0,0" MULTIME "")
    (COMMAND "BLOCK" NUME "Y" "0,0,0" MULTIME
  ""))
)

(DEFUN STRAT (NUME)
  (IF (NOT (TBLSEARCH "LAYER" NUME))
    (COMMAND "LAYER" "M" NUME ""))
)

(DEFUN SCALA()
  (STRAT "SCALA")
  (SETVAR "CLAYER" "SCALA")
  (SETVAR "CECOLOR" "BYLAYER")
  (SETQ P1 (LIST 0 (* -1 LAT)))
  (SETQ P2 (LIST (* XMAX X_FACT) (* -1 LAT)))
  (COMMAND "LINE" P1 P2 "")

  (SETQ Y1 (- (* -1 LAT) (/ LAT 2)))
  (SETQ Y2 (+ (* -1 LAT) (/ LAT 2)))
  (SETQ Y3 (* 2.5 (* -1 LAT)))
  (SETQ PAS_SC 5)
  (SETQ V1 XMIN)
  (SETQ X 0 XV 0)
  (WHILE (<= XV XMAX)
    (SETQ P1 (LIST X Y1))
    (SETQ P2 (LIST X Y2))
    (SETQ P3 (LIST X Y3))
    (COMMAND "LINE" P1 P2 "")
    (SETQ SIR (ITOA (FIX XV)))
    (COMMAND "TEXT" P3 HTEXT 0 SIR "" "")
    (SETQ XV (+ XV (* PAS_X PAS_SC)))
    (SETQ X (* XV X_FACT))
  )
  (SETQ SCL (SSGET "X" (LIST (CONS 8
"SCALA"))))
  (BLOC "SCL" SCL)
  (SETQ P1 (LIST (CAR P3) 0))
  (COMMAND "INSERT" "SCL" "0,0,0" "1" "1" "0")
  (COMMAND "INSERT" "SCL" P1 "1" "1" "90")
)

(DEFUN LEGENDA ()
  (SETQ PAS_LEG 5)
  (SETQ SC_LEG 0.5)
  (SETQ SC_TEXT 0.7)

  (STRAT "XY")
  (STRAT "XZ")
  (STRAT "XZ1")
  (STRAT "YZ")
  (STRAT "YZ1")
  (STRAT "LEGENDA")

  (SETVAR "CLAYER" "LEGENDA")
  (SETQ LAT (* PAS_X X_FACT SC_LEG))
  (SETQ X (* -10 LAT) Y 0)
  (SETQ HTEXT (* LAT SC_TEXT))
  (SETQ INDEX 0)
  (SETQ NR MIC)
  (SETQ CURENT PAS_LEG)
  (REPEAT LGP
    (SETQ P1 (LIST X Y))
    (SETQ P2 (LIST (+ X LAT) Y))
    (SETQ P3 (LIST (+ X LAT) (+ Y LAT)))
    (SETQ P4 (LIST X (+ Y LAT)))
    (SETQ P5 (LIST (+ X LAT (/ LAT 2)) (+ Y (/
LAT -3))))
    (SETQ COL (ITOA (NTH INDEX PALETA)))
    (SETVAR "CECOLOR" COL)
    (COMMAND "3DFACE" P1 P2 P3 P4 "")
    (SETVAR "CECOLOR" "BYLAYER")
    (SETQ SIR (ITOA (FIX NR)))
    (IF (= CURENT PAS_LEG)
      (PROGN
        (COMMAND "TEXT" P5 HTEXT 0 SIR "" "")
        (SETQ CURENT 1)
      )
      (PROGN (SETQ CURENT (+ CURENT 1))
    )
    (SETQ Y (+ Y LAT))
    (SETQ INDEX (+ INDEX 1))
    (SETQ NR (+ NR ECART))
  )
  (SETQ SIR (ITOA (FIX NR)))
  (SETQ P5 (LIST (+ X LAT (/ LAT 2)) (+ Y (/
LAT -3))))
  (COMMAND "TEXT" P5 HTEXT 0 SIR "" "")
  (SETQ P1 (LIST 0 0 0))
  (SETQ P2 (LIST (* XMAX X_FACT) 0 0))
  (SETQ P3 (LIST (* XMAX X_FACT) (* YMAX
Y_FACT) 0))
)

```

Anexa 8 - Program reprezentare grafică

```

(SETQ P4 (LIST 0 (* YMAX Y_FACT) 0))
(SETQ P5 (LIST 0 (* YMAX Y_FACT) (* MARE
Z_FACT)))
(SETQ P6 (LIST 0 0 (* MARE Z_FACT)))
(SETQ P7 (LIST (* XMAX X_FACT) 0 (* MARE
Z_FACT)))
(SETQ P8 (LIST (* XMAX X_FACT) (* YMAX
Y_FACT) (* MARE Z_FACT)))

(SETVAR "CECOLOR" "254")
(SETVAR "CLAYER" "XY")
(COMMAND "3DFACE" P1 P2 P3 P4 "")
(SETVAR "CECOLOR" "252")
(SETVAR "CLAYER" "YZ")
(COMMAND "3DFACE" P1 P4 P5 P6 "")
(SETVAR "CLAYER" "YZ1")
(COMMAND "3DFACE" P2 P3 P8 P7 "")
(SETVAR "CECOLOR" "251")
(SETVAR "CLAYER" "XZ1")
(COMMAND "3DFACE" P3 P8 P5 P4 "")
(SETVAR "CLAYER" "XZ")
(COMMAND "3DFACE" P1 P2 P7 P6 "")

(SETQ LEG (SSGET "X" (LIST (CONS 8
"LEGENDA"))))
(BLOC "LEG" LEG)
(COMMAND "UCS" "V")
(COMMAND "INSERT" "LEG" "0,0,0" "1" "1" "0")
(COMMAND "UCS" "P")
(SETQ LAT (/ LAT SC_LEG))
(SCALA)
)

(DEFUN C:GF ()
(SETQ PALETA (LIST 10 20 30 11 21 31 40 41 50
51 61 60 70 81 90 100 110 120
131 130 141 140 151 161 171 160
170 180))
(SETQ PALETA (REVERSE PALETA))
(SETQ LGP (LENGTH PALETA))
(SETQ MARE -1.0E100)
(SETQ MIC 1.0E100)
(SETQ CMD_VECHI (GETVAR "CMDECHO"))
(SETQ COL_VECHI (GETVAR "CECOLOR"))
(SETQ PREC_VECHI (GETVAR "LUPREC"))
(SETQ STR_VECHI (GETVAR "CLAYER"))
(SETVAR "CMDECHO" 0)
(SETVAR "LUPREC" 4)

(SETQ V2 -0.00024 F2 0.009594)
(SETQ FV 0.010874 V 0.075495 F -0.15119 LIB -
3.96254)

; F -> X
; V -> Y
; Q -> Z

(SETQ XMAX 100)
(SETQ XMIN 0.0)
(SETQ YMAX 300)
(SETQ YMIN 0.0)

(SETQ PAS_X 5.0)
(SETQ PAS_Y 10.0)

(SETQ X_FACT 10.0)
(SETQ Y_FACT 10.0)
(SETQ Z_FACT 2.2)

(STRAT "GRAFIC")
(SETVAR "CLAYER" "GRAFIC")

(SETQ Y YMIN)
(SETQ L_TOTAL '())
(WHILE (<= Y YMAX)
(SETQ X XMIN)
(SETQ L_LOCAL '())
(WHILE (<= X XMAX)
(SETQ Z (FUNC X Y))
(IF (< Z MIC)
(SETQ MIC Z)
)
(IF (> Z MARE)
(SETQ MARE Z)
)
(SETQ P_CURENT (LIST (* X X_FACT) (* Y
Y_FACT) (* Z Z_FACT)))
(SETQ L_LOCAL (APPEND L_LOCAL (LIST
P_CURENT)))
(SETQ X (+ X PAS_X))
)
(SETQ L_TOTAL (APPEND L_TOTAL (LIST
L_LOCAL)))
(SETQ Y (+ Y PAS_Y))
)
(SETQ LGSC (- MARE MIC))
(SETQ ECART (/ (ABS LGSC) LGP))
(COMMAND "VPOINT" "0.46,-0.81,0.34")
(SETQ PLAN 0)
(SETQ CIT_PFY (- (LENGTH L_TOTAL) 1))
(REPEAT CIT_PFY
(SETQ W1 (NTH PLAN L_TOTAL))
(SETQ W2 (NTH (+ PLAN 1) L_TOTAL))
(SETQ CURENT 0)
(SETQ CIT_PFX (- (LENGTH W1) 1))
(REPEAT CIT_PFX
(SETQ P1 (NTH CURENT W1))
(SETQ P2 (NTH (+ CURENT 1) W1))
(SETQ P3 (NTH (+ CURENT 1) W2))
(SETQ P4 (NTH CURENT W2))
(SETQ CURENT (+ CURENT 1))
(SETQ Z (/ (+ (CADDR P1) (CADDR P2) (CADDR
P3) (CADDR P4)) 4))
(SETQ INDEX (- (FIX (/ (- Z MIC) ECART))
1))
(IF (< INDEX 0) (SETQ INDEX 0))
(IF (>= INDEX LGP) (SETQ INDEX (- LGP 1)))
(SETQ COL (ITOA (NTH INDEX PALETA)))
(SETVAR "CECOLOR" COL)
(COMMAND "3DFACE" P1 P2 P3 P4 ""))
)
(SETQ PLAN (+ PLAN 1))
)
(LEGENDA)
(SETVAR "CMDECHO" CMD_VECHI)
(SETVAR "CECOLOR" COL_VECHI)
(SETVAR "LUPREC" PREC_VECHI)
(SETVAR "CLAYER" STR_VECHI)
(COMMAND "ZOOM" "EXTENTS")
)

```

## 16. BIBLIOGRAFIE

1. AKIYAMA, T., Analiza deformațiilor termice ale unei scule cu metoda elementului finit, Bulletin of the Japan Society of Precision Engineering, 3, 1977
2. AKSENOV, V., Legătura regimurilor de prelucrare cu fenomenele termice din timpul rectificării, Izvestia Vissih ucebnih zavesenii - mașinostroenie, 8, 1988
3. AMES, W., F., Nonlinear partial differential equations in engineering, Academic Press, 1965
4. BAIK, B. C., Măsurarea temperaturii la strunjirea straturilor de acoperire dure, Industrie-Anzeiger, 21, 1989
5. BAIRD, D. C., An introduction to measurement theory and experiment design, Prentice-Hall, 1982
6. BAYLEY, F., J., Heat transfer, Allyn & Bacon, 1984
7. BEŇO, J., Identificare statistică și analiză microscopică a fenomenelor deformațiilor în zona de tăiere, Strojirenstvi, 10, oct., 1989
8. BORNHAUSER, A., C., Material science, Springer Verlag, 1985
9. BRAUER, J. R., What every engineer should know about finite elements analysis, John Wiley & Sons, 1988
10. BRĂȚIANU, C., Metode cu elemente finite în dinamica fluidelor, Editura Academiei, 1983
11. BRĂȚIANU, C., Modelarea cu elemente finite a transferului de căldură, Iceenerg, 1985
12. BRJOZOVSKII, B. M., Aprecierea influenței perturbațiilor de temperatură asupra preciziei dimensionale, Izvestia Vissih ucebnih zavesenii - mașinostroenie, 8, 1988
13. BUDOVICK, F. S., Matematică finită cu aplicații, McGraw-Hill, 1985
14. BURNHAM, Don, Measurement of temperature without contact, Processing, Oct, 1992
15. BUZDUGAN, Gh., Rezistența materialelor, Editura tehnică, 1980
16. CAPRILI, M., Elemente cuadrilaterale curbe în metoda elementelor finite, Facolta di Ingegneria, Pisa, 1976
17. CHEN, T. Y., Fenomenul de fisurare termomecanică la inițierea unui gol lângă suprafață datorat unei sarcini cu deplasare rapidă, Sovremennoe mașinostroenie, 3, 1989
18. CONWAY, T. G., Folosirea imaginilor termice pentru măsurarea temperaturii, Mechanical Engineering, 6, 1987
19. DALLY, James W., Experimental stress analysis, McGraw-Hill, 1991
20. DEARNLEY, P. A., New technique for determining thermal distribution in cemented carbide cutting tools, Metal Technology, June, 1983
21. DHATT, Gouri, The finite element method displayed, John Wiley & Sons, 1984
22. DI STASIO, J., Considerations on remote temperature measurement, Measurement Science and Technology, Dec, 1994
23. DRĂGHICI, Gh., Bazele teoretice ale proiectării proceselor tehnologice în construcția de mașini, E. D. P., 1974
24. DRĂGHICI, Gh., Tehnologia construcțiilor de mașini, E. D. P., 1984
25. DUMITRAȘ, C., Așchieria metalelor și fiabilitatea sculelor așchietoare, Editura tehnică, 1983
26. EBĂNCĂ, D., Metode de calcul numeric, Sitech, 1994
27. EDEN, R. C., Application of synthetic diamond substrates for thermal management of high performance electronic multi-chip modules, First International Conference on the Applications on Diamond Films and Related Materials, August, 1991
28. ERNST, L. I., O teorie a învelișurilor geometrice neliniare cu elemente finite, teză de doctorat, Delft Univ. of Technology, 1981
29. FAGAN, M., Finite elements analysis, theory and practice, McGraw-Hill, 1992
30. FRANCK, W., Soluții de sporire a preciziei la mașinile unelte de așchiere, Maschinenbautechnik, 8, 1989
31. GĂRBEA, Dan, Analiza cu elemente finite, Editura tehnică, 1990
32. GOULD, Lawrence, Sensors for Machine Tool Monitoring, Sensors, 2, 1989
33. GRABCSENKO, A. I., Rolul fenomenelor termice la rectificarea cu diamant a cristalelor superdure, Gepyartastechnologia, 8, 1986
34. GURUMOORTHY, B., Lifetime predictions for a ceramic cutting tool material at high temperatures, Journal of Material Science, 6, 1987
35. HANDHAKUMAR, J., Integrated model of temperature measurement and visual images, IEEE Computer Vision and Pattern Recognition Conference, June, 1989
36. HARTLEY, Peter, Numerical modelling of material deformation processes, Springer Verlag, 1992
37. HOFMANN, E., Compensarea deformării axiale și unghiulare la mașinile de strunjit prin încălzire, Industrie-Anzeiger, 3, ian., 1989
38. HOMIAKOV, V. S., Utilizarea analizei teoretice modale la calculul câmpurilor de temperaturi ale, Mașinostroenie, 9, sept., 1989

39. HUMMEL, Müller, Infrared temperature measurement on diamond-coated tools during machining, *Diamond and Related Materials*, 1, 1994
40. HUMPEL, W., Utilizarea unor senzori de temperatură cu aparate de măsură multipunct pentru determinarea comportării termice a mașinilor unelte, *Feingeraetetechnik*, 1, ian., 1990
41. HUSSAIN, Zaid, Digital image processing, John Wiley & Sons, 1989
42. IACOB, Caius, *Mecanică teoretică*, E. D. P., 1980
43. JAEGER, J. C., *Elasticity fracture and flow*, Pergamonn Press, 1989
44. JEDREZEJEWSKI, J., Analiza și optimizarea proprietăților termice ale mașinilor unelte, *Mechanik*, 6, iun, 1989
45. JU, F. D., Parametrii care influențează fenomenul de crăpare termomecanică la deplasarea rapidă a sarcinii datorită forțelor de frecare pe suprafața corpurilor cu acoperiri, *Sovremennoe mașinostroenie*, 3, 1989
46. JUNGNIKEL, G., Zur Beurteilung des termischen Verhaltens von Werkzeugmaschinen, *Maschinenbautechnik*, 12, 1974
47. KAUFELD, M., Frezarea rapidă de finisare la aliajele de metale ușoare, *Industrie-Anzeiger*, 40, 1989
48. KAY, J., M., An introduction to fluid mechanics and heat transfer, Cambridge University Press, 1974
49. KITAGAWA, T., Estimarea analitică a uzurii flancurilor la sculele din carburi metalice utilizate la strunjirea oțelurilor carbon, *Bulletin of the Japan Society of Precision Engineering*, 22 nr. 4 dec, 1989
50. KORMEEVA, S. M., Influența condițiilor de prelucrare a metalelor cu viteze foarte mari, asupra temperaturii, *Izvestia Vissih ucebniih zavesenii - mașinostroenie*, 11, 1986
51. KORNEEVA, V. M., Uzura sculei așchietoare în procesul de așchiere discontinuu cu viteze foarte mari, *Izvestia Vissih ucebniih zavesenii - mașinostroenie*, 9, 1989
52. KREYSZIG, Erwin, *Advanced engineering mathematics*, ed. 7, John Wiley & Sons, 1993
53. KUDINOV, A. V., *Dinamica mașinilor unelte*, E. D. P., 1970
54. LANCZOS, Cornelius, *The variational principles of mechanics*, Pergamonn Press, 1982
55. LARRY, I., Stress difference Elasticity and its application to Photomechanics, *Experimental mechanics*, sept., 1972
56. LAVINE, A. S., Un model simplu pentru răcirea convectivă în procesul rectificării, *Konstruirovanie i tehnologiiia mașinostroenia*, 4, oct-dec, 1988
57. LĂZĂRESCU, I., Teoria așchierii metalelor și proiectarea sculelor, E. D. P., 1964
58. LEWIS, J., *Finite elements method principles and applications*, Thomas Nelson and Sons Ltd., 1991
59. LI, Y. Y., Modelarea procesului de rectificare, *Sovremennoe mașinostroenie*, 8, 1989
60. LIN, Zong-Ching, A Thermo-Elastic-Plastic Model with Special Elements in Cutting Process with Tool Flank Wear, *International Journal of Machine Tools and Manufacturing*, Jan, 1994
61. LIVER, B., *Finite elements, an introduction for engineers*, Prentice-Hall, 1983
62. MARCIUK, G., I., *Metode de analiză numerică*, Editura Academiei, 1983
63. MIED, P. B., *Computer Aided Rapid Prototyping*, *Engineering Designer*, March/April, 1996
64. MILITARU, C., *Fiabilitatea și precizia în t. c. m.*, Editura tehnică, 1987
65. MURGULESCU, E., *Geometrie analitică și diferențială*, E. D. P., 1965
66. MUSKHEISHVILI, N. I., *Some basic problems of the mathematics*, McGraw-Hill, 1982
67. NOSCENKO, A. N., Calculul numeric al câmpurilor termice neliniare staționare tridimensional în zona așchierii, *Fizika I Himiia obrabotka materialov*, 20 nr. 1, 1986
68. OLARIU, V., *Modelare numerică cu elemente finite*, Editura tehnică, 1985
69. OSTAFIEV, V. A., Metode moderne de calcul a câmpurilor termice în zona așchierii *Fizika I Himiia obrabotka materialov*, *Fizika I Himiia obrabotka materialov*, 20 nr. 2, 1986
70. OZHAN, F., Temperaturile de așchiere la rectificarea materialelor ușor magnetice, *Industrie-Anzeiger*, 5, 1986
71. PACOSTE, I., *Metode moderne în mecanica structurilor*, Editura științifică și enciclopedică, 1988
72. PANZERT, B., Micșorarea deplasărilor de natură termică la centrele de prelucrare și la celulele de fabricație pentru piese prismatice, *Maschinenbautechnik*, 3, 1989
73. PASCARIU, Ioan, *Elemente finite concepte aplicații*, Editura militară, 1985
74. PASCHER, M., Compensarea deplasărilor generate de fenomene termice la mașini unelte, *Industrie-Anzeiger*, 107 nr. 103-104, 1985
75. PETHÖ, L., Analiza solicitării termice și mecanice prin metoda elementului finit a unui cuțit de strung asupra preciziei de prelucrare, *Comunicările celui de-al III-lea simpozion de tensometrie*, Timișoara 1983, 1983
76. PETHÖ, L., Cercetări cu privire la influența deformațiilor elastice și termice asupra preciziei de prelucrare și posibilități de compensare a acestora, teză de doctorat, Univ. Cluj Napoca, 1984
77. PETRICEANU, Gh., Contribuții la stabilirea variației temperaturii în elementele sistemului tehnologic al mașinii de rectificat roți dințate cilindrice și deformațiile termice ale acestora, *Buletinul științific al Institutului Politehnic Cluj Napoca seria mecanică aplicată*, 14, 1971
78. PETRILA, T., *Metode cu elemente finite și aplicații*, Editura Academiei, 1987



79. PETZOLD, Charles, Programming windows 3.1, Microsoft Press, 1992
80. POPA, M., Utilizarea radiațiilor infraroșii în studiul termografic al sistemului tehnologic, Buletinul științific al Institutului Politehnic Cluj Napoca seria mecanică aplicată, 29, 1986
81. POPESCU, I., Optimizarea proceselor de așchiere, Scrisul românesc, 1987
82. POPESCU, I., Teoria așchierii, Reprografia Universității din Craiova, 1994
83. POPOVICI, V., Tehnologii în optica laser, Mirton, 1995
84. PRAMILA, A., Aplicații ale metodei elementelor finite la proiectarea asistată de calculator a formei unei nave, teză de doctorat, Helsinki Univ. of Technology, 1981
85. PRENTER, P. M., Splines and variational methods, John Wiley & Sons, 1975
86. RAO, S. S., Metoda elementului finit în inginerie, Pergamonn Press, 1981
87. ROSS, L. J., Experiments in heat transfer and thermodynamics, Cambridge University Press, 1994
88. ROȘCA, A. S., "DIS\_MOD" aplicație pentru proiectarea asistată a dispozitivelor modulare, Analele Universității Craiova, seria Colegiu, Timis, 1995
89. ROȘCA, A. S., Analiză statică cu metoda elementului finit la structura ramei electrice suburbane RES 2720, Tema cercet., 1988
90. ROȘCA, A. S., Aplicații didactice ale programelor de calcul tabelar la studiul metodei elementului finit, 1997
91. ROȘCA, A. S., Considerații despre formularea variațională la modelarea cu elemente finite a transferului termic, 1997
92. ROȘCA, A. S., Considerații despre tratarea singularităților la modelarea cu elemente finite în cazul plan, 1, 1997
93. ROȘCA, A. S., Considerații despre utilizarea fotografiei termice în infraroșu, 1, 1997
94. ROȘCA, A. S., Considerații privind studiul câmpului termic din procesele de prelucrare mecanică, Analele Universității Craiova, seria Colegiu, 1995
95. ROȘCA, A. S., Metodă pentru determinarea emisivității corpurilor mici în termografie, 1, 1997
96. ROȘCA, A. S., Metode de analiză a câmpului termic în procesele de așchiere, Referat nr. 2, pentru stagiul doctorat, Univ. Tehnică din Timișoara, 1995
97. ROȘCA, A. S., Orientări și direcții în studiul câmpului termic din procesele de șchiere, Referat nr. 1, pentru stagiul doctorat, Univ. Tehnică din Timișoara, 1994
98. ROȘCA, A. S., Orientări și direcții în studiul câmpului termic din sistemele de prelucrare prin așchiere, Analele Universității Craiova, seria Colegiu, 4, 1993
99. ROȘCA, A. S., Proiectarea în plan cu autocad release 12, Certi, 1995
100. ROȘCA, A. S., Rezistența materialelor, probleme rezolvate cu metoda elementului finit, Certi, 1996
101. ROȘCA, A. S., Studiul câmpului termic la așchiera oțelului 34CrNiMo6, 1997
102. ROȘCA, A. S., Utilizarea elementelor finite liniare la rezolvarea ecuației lui Poisson,
103. RUSU, O., Determinarea prin interferometrie holografică a deplasării lagărului față a strungului SN500 ca urmare a încălzirii în timpul funcționării, Construcția de mașini, 5, 1975
104. SAFIULKIN, V. N., Metode de răcire a zonei de lucru cu scula așchietoare lamelează, Izvestia Vissih ucebniih zavesenii - mașinostroenie, 9, 1987
105. SANFORD, R. J., Hologrames for flames temperature, Combustion Flame, 49 Jan, 1983
106. SCARBOROUGH, J. B., Numerical mathematical analysis, Oxford University Press, 1992
107. SEDOV, L., Similitudine și mărimi în mecanică, Mir, 1977
108. SHIRAKASHI, T., Flow stress of low carbon steel at high temperature and strain rate, Bulletin of the Japan Society of Precision Engineering, 17, 1983
109. SINES, George, Elasticity and strength, Athlone Press, Univ. of London, 1984
110. SPUR, G., Turning of fiber reinforced plastics, Manufacturing Review, June, 1988
111. STASSA, W., Applied finite elements analysis, Prentice-Hall, 1985
112. STEELE, J. M., Applied finite elements modeling, practical problem solving for engineers, McGraw-Hill, 1989
113. STENGERG, R., Analiza convergenței metodelor cu elemente finite, Helsinki Univ. of Technology, 1984
114. STENGERG, R., Analiza convergenței metodelor cu elemente finite mixte, teză de doctorat, Helsinki Univ. of Technology, 1984
115. STREKOWSKI, J. S., A finite element model in orthogonal cutting, SME North American Manufacturing Research Conference, 349, 1985
116. STREKOWSKI, J. S., Finite element prediction of chip geometry and tool workpiece temperature distributions on orthogonal metal cutting, SME North American Manufacturing Research Conference, 112, 1990
117. STROUSTRUP, Bjarne, The c++ programming language, Addison-Wesley Publishing Company, 1994
118. SUMMER, W., Ultraviolet and infrared engineering, John Wiley & Sons, 1989
119. SZEPESSY, A., Convergența metodei elementelor finite pentru curgere laminară aplicată la legi de conservare, teză de doctorat, Chalmers University of Technology, Göteborg, 1989
120. TÄGNFORS, H., Software for finite element calculations, teză de doctorat, Chalmers University of Technology, Göteborg, 1987
121. TIMOSHENKO, S. P., Theory of elasticity, McGraw-Hill, 1970



122. TÎRPE, Gh., Precizia prelucrării prin aşchiere, E.D.P., 1987
123. TONSHOFF, H.K., Tribological Aspects of Interrupted Cutting with Ceramic Tool Materials, The North American Manufacturing Research, 19-th Conference, 1991
124. TRENT, E.M., Aşchiera metalelor şi tribologia gripării III. Temperatura în procesurile de aşchiere a metalelor, Wear, 1, 1988
125. ŢIŢEICA, Gh., Fizica generală, E.D.P., 1985
126. VASILESCU, Al., Analiza dimensională şi teoria similitudinii, Mir, 1969
127. VASILIEV, A.N., Aprecierea influenţei temperaturii asupra uzurii sculelor diamantate la strunjirea fină a metalelor în optică, Optico-mehaneskaia proislennost, 9, 1989
128. VOKE, Janet, Color vision, Pergamonn Press, 1992
129. WANG-CHI, T., Applied elasticity, Allyn & Bacon, 1986
130. WECK, M., Tehnici de măsurare pentru cercetarea şi evaluarea preciziei de prelucrare a maşinilor unelte, Industrie-Anzeiger, 72, sept., 1985
131. WERNER, G., Rectificarea adâncă a canelurilor cu ajutorul discurilor abrazive, Industrial Diamond Review, 48 nr. 6, 1988
132. WIELLE, H., Comportamentul termic al pieselor în prelucrările de precizie, Fertigungstechnik und Betrieb, 3, 1986
133. WOOD, W.G., Experimental method, Athlone Press, Univ. of London, 1984
134. YAMANE, Y., Sisteme de calibrare pentru termocuple cu încălzire infraroşii pentru măsurarea temperaturii sculelor, Bulletin of the Japan Society of Precision Engineering, 1, mar., 1989
135. YAZU, S., Tool application of diamond and cBN, First International Conference on the Applications on Diamond Films and Related Materials, August, 1991
136. YOKOYAMA, K., Analiza deformaţiilor termice a unei piese uzinate în procesul de honuire, Bulletin of the Japan Society of Precision Engineering, 22 nr. 1, 1988
137. ZEID, Ibrahim, CAD/CAM Theory and practice, McGraw-Hill, 1991
138. ZEJES, I., Funcţii spline în teoria mecanismelor, \*\*\*, 1982
139. ZIENKIEWICZ, O., C., The finite element method, vol. 1, 2 ed. 4, McGraw-Hill, 1989
140. ZIRBO, Gh., Consideraţii asupra dimensionării lingotierei studiul câmpului de temperatură, Buletinul ştiinţific al Institutului Politehnic Cluj Napoca seria mecanică aplicată, 29-30, 1986
141. ZORICTUEV, V.T., Temperaturile pe suprafeţele de contact ale sculei şi media energiei termice de contact sculă-pieşă, Izvestia Vissih ucebnih zavesenii - maşinostroenie, 10, 1985
142. \*\*\*, Autocad development system programmer's reference, \*\*\*, 1992
143. \*\*\*, Cercetări asupra temperaturii procesului de aşchiere discontinuu şi cu viteze foarte mari, Izvestia Vissih ucebnih zavesenii - maşinostroenie, 3, 1989
144. \*\*\*, Color filter glasses, Galvoptics Ltd., 1986
145. \*\*\*, Emrc nisa display iii, user's manual, vol. i, ii, \*\*\*, 1983
146. \*\*\*, Emrc nisa, user's manual, michigan usa, \*\*\*, 1993
147. \*\*\*, Filters and safelights, Galvoptics Ltd., 1992
148. \*\*\*, Infrared theory and measurement, Agema Infrared Systems, 1990
149. \*\*\*, Measurements group - technical notes tn501-tn515, \*\*\*, 1980
150. \*\*\*, Metals Reference Book, Addison-Wesley Publishing Company, 1975
151. \*\*\*, Metals reference book, \*\*\*, 1978
152. \*\*\*, Msurarea digitală a temperaturilor, Maschine und Werkzeug, 12, mai, 1990
153. \*\*\*, Prelucrări privind aşchiera cu viteze mari de prelucrare, Rivista di Meccanica, 39 nr. 903, 1988
154. \*\*\*, Proprieties of materials, Pergamonn Press, 1991
155. \*\*\*, Reference manual, Ansys Co., 1992
156. \*\*\*, Student manual for strain gage technology, Measurements Group, 1992
157. \*\*\*, Thermal Photography, Kodak Tech Bits, Vol. No. 1, 1976
158. \*\*\*, Tools, tooling systems and services for the global metalworking industry, Kennametal, 1991