

TEZA DE DOCTORAT

**Studiul modelării câmpurilor termice la
încălzirea oțelurilor pentru deformare plastică**

Conducător științific:

Prof.Dr.Ing. **ILCA IOAN**

Membru corespondent al Academiei de Științe Tehnice din România

Doctorand:

Ing. **RAȚIU SORIN AUREL**

369⁶⁴².440
E

2004

Argument

Ideea acestui studiu pleacă de la problemele apărute în timpul procesului de deformare plastică la cald a semifabricatelor, probleme datorate în cea mai mare măsură încălzirii neuniforme în cuptoarele secțiilor respective.

Cunoașterea distribuției spațiale și temporale a câmpului termic în interiorul incintelor cuptoarelor cu flacără reprezintă punctul de plecare în procesul de optimizare a încălzirii materialului metalic care urmează a fi deformat plastic la cald.

Întrucât realizarea unor măsurători în interiorul spațiului de lucru al acestor cuptoare și, în mod particular, în interiorul semifabricatului supus încălzirii reprezintă o sarcină foarte dificilă, uneori imposibil de realizat, nu putem obține decât informații cu caracter global asupra fenomenelor de transfer termic care au loc în astfel de situații. Cu alte cuvinte, datele culese sunt utile îndeosebi conducerii procesului tehnologic propriu-zis și în mai mică măsură optimizării fenomenului de încălzire.

În ultimii ani, așa-zisul „experiment numeric” începe să fie utilizat din ce în ce mai mult și cu rezultate tot mai promițătoare în evaluarea transferului de căldură în diferite situații particulare. Acest lucru se datorează în primul rând îmbunătățirii algoritmilor numerici și a tehnicilor de simulare.

Prezenta teză se înscrie în tendințele moderne de înlocuire treptată a experimentelor fizice cu cele numerice în domeniul proiectării și optimizării proceselor termice din cuptoarele de încălzire cu flacără.

Metoda de cercetare aleasă - simularea numerică - este una din cele mai moderne și mai folosite la ora actuală în munca de cercetare inginerescă, cu o mare perspectivă în viitor.

Prin natura sa, această lucrare necesită o abordare cu caracter multidisciplinar, începând cu o înțelegere profundă a fenomenelor complexe de transfer de căldură ce au loc în spațiul de lucru al cuptorului și continuând cu solide cunoștințe legate de termodinamică și transmiterea căldurii, algoritmi și metode numerice, precum și de tehnicile moderne de programare.

Doresc să adresez cele mai calde mulțumiri conducătorului științific, domnului prof.dr.ing. Ilca Ioan, membru corespondent al Academiei Tehnice din România, pentru sugestiile, sfaturile pline de profesionalism și încrederea acordată pe întreg parcursul elaborării tezei. Experiența de o viață a dânsului în acest domeniu a fost și va rămâne un model pentru mine.

De asemenea doresc să mulțumesc domnului prof.dr. Maksay Ștefan pentru sprijinul acordat în elaborarea suportului matematic al tezei, doamnei conf.dr.ing. Vîlceanu Lucia pentru sfaturile prețioase, domnului conf.dr.ing. Toader Ștefan și domnului tehnician Popa Aron pentru ajutorul dat în efectuarea măsurătorilor experimentale și tuturor colegilor de la Facultatea de Inginerie din Hunedoara care în mod direct sau indirect au fost alături de mine în acest demers.

Mulțumesc domnului dr.ing. Muntean Sebastian, cercetător principal în cadrul Centrului Național pentru Ingineria Sistemelor cu Fluide Complexe din cadrul Universității „Politehnica” Timișoara, căruia îi datorez inițierea în problematica simulării numerice și domnului dr.ing. Ioan Romulus de la S.C. „Siderurgica” S.A. Hunedoara pentru sprijinul acordat în conceperea și definitivarea instalației experimentale.

În final doresc să mulțumesc soției mele Ligia pentru dragostea, răbdarea și înțelegerea de care a dat dovadă, sprijinul și încurajările permanente și, nu în ultimul rând, părinților mei fără de care nu aș fi ajuns aici.

ing. Sorin A. Rațiu

Lista simbolurilor utilizate

A	Domeniul de analiză
a	Difuzivitatea termică [m^2/s]
α_C	Coeficient parțial de transmitere a căldurii prin convecție [$W/(m^2 \cdot \text{grad})$]
α_R^{g-z}	Coeficient parțial de transfer termic prin radiație de la gaze la zidărie [$W/(m^2 \cdot \text{grad})$]
α_R^{gzm}	Coeficient parțial de transfer termic prin radiație de la gaze, considerate la temperatura pereților, la suprafața materialului [$W/(m^2 \cdot \text{grad})$]
α_R^{z-m}	Coeficient parțial de transfer termic prin radiație de la zidărie la material [$W/(m^2 \cdot \text{grad})$]
α_R^{g-m}	Coeficient parțial de transfer termic prin radiație de la gaze la material [$W/(m^2 \cdot \text{grad})$]
B	Semilățimea blumului [m]
[B]	Matricea derivatelor funcțiilor de formă în raport cu coordonatele
β	Coeficientul de dilatare termică liniară [1/K]
[C]	Matricea capacităților calorice sau a căldurilor specifice pentru un element finit
c	Căldura specifică [$J/(kg \cdot \text{grad})$]
[D]	Matricea coeficienților de conductivitate termică
ΔT	Diferența de temperatură [K]
Δt	Diferența de temperatură [$^{\circ}C$]
$\Delta \tau$	Interval de timp [s]
Δn	Distanța între două suprafețe izoterme [m]
E	Modulul de elasticitate longitudinal [N/mm^2]
ε_m	Gradul de negreală (sau factorul energetic de emisie) suprafeței materialului [-]
ε_z	Gradul de negreală (sau factorul energetic de emisie) al zidăriei [-]
ε_{z-m}	Coeficientul reciproc de negreală între suprafața zidăriei și cea a materialului [-]
ε_r	Eroare relativă [%]
F	Funcția ce precizează distribuția temperaturii în blum în momentul inițial
f,g	Funcții oarecare
{f}	Vectorul încărcărilor termice pentru un element finit
{F}	Vectorul încărcărilor termice pentru întreg domeniul
Φ	Temperatura redusă,
ϕ	Coeficient unghiular sau de iradiere [-]
φ	Funcția de variație a temperaturii în timp pentru punctele suprafeței laterale a blumului
Gr	Criteriul de similitudine Grashof [-]
H	Semiînălțimea blumului [m] și entalpia [kJ/kg]
h	Grosimea elementului finit [m]
$\bar{i}, \bar{j}, \bar{k}$	Versori
J	Funcțională
[K]	Matricea de rigiditate (conductivitate) a întregului domeniu
[k]	Matricea de caracteristici fizice a elementului finit
L	Semilungimea blumului [m]
λ	Conductivitatea termică [$W/(m \cdot \text{grad})$]
[N]	Matricea funcțiilor de formă

N_i, N_j, N_m	Funcții de formă corespunzătoare unui element finit
Nu	Criteriul de similitudine Nusselt [-]
\bar{n}_0	Vector unitar
n_x, n_y, n_z	Cosinusuri directoare [-]
ν	Coefficientul de contracție transversală a lui Poisson [-]
Pr	Criteriul de similitudine Prandtl [-]
\dot{Q}_{g-z}	Flux de căldură transmis de la gaze la zidărie [W]
\dot{Q}_{z-m}	Flux de căldură transmis de la zidărie la metal [W]
$\dot{Q}_{abs.g}$	Flux de căldură absorbit de gazele arse [W]
\dot{q}	Densitatea fluxului termic la cazul general [W/m^2]
\dot{q}_{f-s}	Densitatea fluxului termic, transmis prin convecție, între un fluid și o suprafață la cazul general [W/m^2]
\dot{q}_{z-m}	Densitatea fluxului de căldură transmis de la pereți la material [W/m^2]
\dot{q}_{g-m}	Densitatea fluxului de căldură transmis de la gaze la material [W/m^2]
\dot{q}_{tot}	Densitatea fluxului termic total primit de suprafața materialului [W/m^2]
\dot{q}_v	Debitul surselor interioare de căldură [W/m^3]
Re	Criteriul de similitudine Reynolds [-]
R^2	Coefficient de corelație [-]
ρ	Densitatea [kg/m^3]
S	Suprafața în general [m^2]
S_m	Suprafața exterioară a materialului supus încălzirii [m^2]
S_z	Suprafața interioară liberă a zidăriei [m^2]
$\sigma_1, \sigma_2, \sigma_3$	Tensiuni normale principale [N/mm^2]
σ_c	Limita de curgere aparentă [N/mm^2]
σ_p	Limita de proporționalitate [N/mm^2]
σ_r	Rezistența la rupere [N/mm^2]
T	Temperatura [K]
\tilde{t}	Funcția de temperatură aproximată
$\{t\}_{(e)}$	Vectorul temperaturilor nodale
$\{t\}$	Vectorul temperaturilor necunoscute pentru tot domeniul de analiză
t	Temperatura în general [$^{\circ}C$]
t_{exp}	Temperatura obținută prin măsurători experimentale [$^{\circ}C$]
$t_{analitic, C++, ALGOR}$	Temperatura obținută prin calcul analitic, cu C++ și respectiv cu ALGOR [$^{\circ}C$]
t_g	Temperatura gazelor arse [$^{\circ}C$]
t_z	Temperatura suprafeței interioare a zidăriei [$^{\circ}C$]
t_s	Temperatura unei suprafețe în general [$^{\circ}C$]
t_f	Temperatura unui fluid în general [$^{\circ}C$]
t_m	Temperatura materialului [$^{\circ}C$]
τ	Timpul [s]
x, y, z	Coordonate carteziene: abscisă [m], ordonată [m] și respectiv cotă [m]
r, φ, z	Coordonate cilindrice: raza [m], latitudinea [rad] și respectiv cota [m]
r, φ, ψ	Coordonate sferice: raza [m], latitudinea [rad] și respectiv azimutul [rad]
$\{u\}$	Funcție oarecare sub formă de vector coloană
$V^{(e)}$	Volumul elementului finit

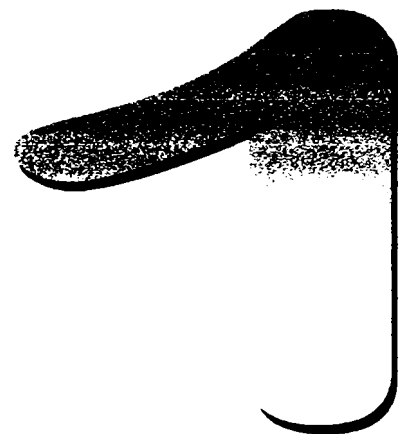
Cuprins

Argument.....	i
Lista simbolurilor utilizate.....	ii
Cuprins.....	iv
Capitolul 1. Introducere.....	1
1.1. Concepte legate de modelare și simulare	1
1.2. Etape în rezolvarea problemelor prin simulare numerică.....	3
1.3. Simularea numerică în transferul termic.....	5
1.4. Obiectivele și scopul tezei.....	6
Capitolul 2. Bazele teoretice ale transferului termic în cuptoare industriale.....	7
2.1. Considerații generale.....	7
2.2. Mărimi caracteristice transferului termic.....	8
2.3. Transferul termic în cuptoarele industriale.....	9
2.3.1. Transferul termic în camera de lucru a cuptoarelor cu flacără.....	10
2.3.2. Problema conducției la încălzire.....	12
2.3.3. Problema convecției la încălzire.....	16
2.3.4. Problema radiației la încălzire.....	18
2.4. Tensiuni interne la încălzire.....	19
Capitolul 3. Metode numerice în tratarea problemelor de transfer termic.....	23
3.1. Metode pentru rezolvarea ecuațiilor diferențiale caracteristice proceselor termogazodinamice.....	23
3.2. Metoda diferențelor finite.....	24
3.2.1. Aproximarea prin diferențe ascendente.....	25
3.2.2. Aproximarea prin diferențe descendente.....	26
3.2.3. Aproximarea prin diferențe centrate.....	26
3.3. Metoda elementelor finite.....	28
3.3.1. Tipuri de probleme rezolvabile prin metoda elementelor finite.....	29
3.3.2. Ipoteze de lucru.....	30
3.3.3. Stabilitatea domeniului de analiză.....	30
3.3.4. Discretizarea domeniului de analiză.....	31
3.3.5. Obținerea modelului numeric cu elemente finite.....	33
Capitolul 4. Studii și cercetări experimentale.....	36
4.1. Prezentarea instalației experimentale pilot.....	36
4.2. Modul de achiziție al datelor.....	39
4.3. Metodologia de experimentare.....	41
4.4. Concluzii.....	42

Capitolul 5. Modelul analitic al transferului termic în interiorul blumului.....	44
5.1. Aparatul matematic.....	44
5.2. Soluția analitică. Rezultate.....	50
5.3. Concluzii.....	52
Capitolul 6. Modelul numeric cu diferențe finite.....	57
6.1. Considerații teoretice.....	57
6.1.1. Cazul regimului nestaționar tridimensional.....	57
6.1.2. Criteriul de stabilitate a ecuațiilor cu diferențe finite	60
6.2. Implementarea modelului numeric cu diferențe finite.....	61
6.2.1. Modelul tridimensional al încălzirii blumului	61
6.2.2. Conductivitatea termică.....	62
6.2.3. Temperatura redusă.....	63
6.2.4. Entalpia.....	63
6.2.5. Ipoteze simplificatoare.....	64
6.3. Descrierea și funcționarea programului sursă pentru simularea câmpului termic.....	65
6.4. Prezentarea rezultatelor.....	67
6.4.1. Câmpul termic.....	67
6.4.2. Evoluția temperaturii punctelor caracteristice.....	70
6.5. Concluzii.....	72
Capitolul 7. Modelul numeric cu elemente finite.....	76
7.1. Considerații teoretice.....	76
7.2. Implementarea modelului numeric cu elemente finite.....	83
7.2.1. Etapele obținerii modelului numeric.....	83
7.2.2. Preprocesarea.....	83
7.2.3. Procesarea.....	89
7.2.4. Postprocesarea.....	90
7.3. Prezentarea rezultatelor.....	91
7.3.1. Câmpul termic.....	91
7.3.2. Tensiunile interne datorate gradientilor termici.....	96
7.3.3. Deformațiile produse datorită încălzirii.....	100
7.3.4. Evoluția temperaturii nodurilor caracteristice.....	101
7.4. Concluzii	103
Capitolul 8. Concluzii finale, contribuții personale, perspective.....	108
8.1. Concluzii finale.....	108
8.2. Contribuții personale.....	111
8.3. Perspective.....	112
Bibliografie.....	113
Anexa.....	a

partea întâi

- **Introducere**
- **Bazele teoretice ale transferului termic în cuptoare industriale**
- **Metode numerice în tratarea problemelor de transfer termic**



Capitolul 1

Introducere

- 1.1. Concepte legate de modelare și simulare
- 1.2. Etape în rezolvarea problemelor prin simulare numerică
- 1.3. Simularea numerică în transferul termic
- 1.4. Obiectivele și scopul tezei

Capitolul 1

Introducere

1.1. Concepte legate de modelare și simulare

Inițiată în urmă cu patru decenii ca o nouă tehnică de studiu, simularea a început să devină un instrument tot mai util în analiza, proiectarea și conducerea sistemelor complexe.

Simularea este un domeniu de investigare aflat la confluența matematicii, teoriei sistemelor și informaticii, cu o metodologie proprie (bazată pe identificarea proceselor, modelarea sistemelor și teoria algoritmilor) și cu mijloace specifice (oferite de echipamentul de calcul automat, de aparatura electronică de măsură, de prelevare a datelor și de verificare a rezultatelor). Ea pune la dispoziția cercetătorului o metodă prin care acesta să-și poată verifica puterea de înțelegere a fenomenelor din jurul său și îl ajută să obțină rezultate mai bune într-un timp mult mai scurt. De asemenea reprezintă un instrument eficient de comunicare, arătând modul de desfășurare al unui proces și stimulează gândirea creativă în legătură cu posibilitățile de îmbunătățire și optimizare a acestuia.

Deoarece simularea își are originea în studiul sistemelor, se definește noțiunea de sistem într-o formulare mai largă: *o colecție de elemente diferite în interacțiune, cuprinzând oameni și mașini, integrate pentru îndeplinirea unui obiectiv dorit, prin manipularea și controlul materialelor, informației, energiei și activității umane*. Cu alte cuvinte, numim sistem colecția de elemente capabile de interacțiune într-un astfel de mod încât să fie realizat programul cerut.

Metoda de studiu recomandată este de a efectua experimentările direct pe sistemul original (dacă ele sunt posibile). Din păcate, nu orice sistem poate suporta sau admite încercările directe. Dacă experimentările de acest fel pot „vătăma” ireversibil sistemul, în sensul modificării stării reale, duratei, rezultatelor în timp util și preciziei informatice necesare, ș.a.m.d, atunci singura cale de studiu și analiză a sistemului este simularea.

Pentru a realiza o simulare, trebuie ca în prealabil să se stabilească un model al sistemului studiat, ce se obține prin tehnica identificării proceselor din acel sistem (prin așa numita modelare). Modelarea se definește ca o procedură a analizei de sistem ce se realizează prin idealizarea matematică a întregului sistem sau a unor părți din acesta. Rezultatul ei este elaborarea modelului sistemului, sub forma unei reprezentări matematice a relațiilor din sistem. La studierea unui fenomen fizic se face o abstractizare și se stabilește o legătură esențială, reproductibilă și necesară între mărimile care guvernează fenomenul. Această legătură reprezintă legea de desfășurare a fenomenului. În cazul fenomenelor fizice, legile se exprimă cu ajutorul unor relații matematice, denumite modele matematice. Cu alte cuvinte, simularea nu este altceva decât o reprezentare matematică a interacțiunii obiectelor din lumea reală. Studiul modelelor matematice conduce la rezultate care, desigur, trebuie să fie verificate în practică.

Există fenomene care au expresiile matematice ce le guvernează de aceeași formă. Acestea permit introducerea noțiunii de fenomene calitativ identice, sau cu alte cuvinte, fenomene ce posedă același model matematic. Această clasă de fenomene, dacă cuprinde fenomene din domenii diferite, spunem că formează o clasă de analogie. Utilizarea analogiei se practică pe scară largă în momentul

în care experimentul dintr-un domeniu devine dificil de realizat, iar în alt domeniu, unde avem un fenomen calitativ identic, este mai ușor de lucrat.

Modelele pot fi statice sau dinamice. Mărirea puterii și vitezei de procesare a calculatoarelor, împreună cu nevoia de răspunsuri tot mai exacte, au propulsat modelele dinamice în fața celor statice în ceea ce privește alegerea lor ca instrumente de studiu.

Modelele statice (permanente) descriu un sistem din punct de vedere matematic prin intermediul unei ecuații sau a unui sistem de ecuații a căror soluție, pentru anumite situații particulare, se obține printr-o singură rezolvare a ecuațiilor respective (rezolvare clasică). Variabilele utilizate în calcule sunt introduse ca valori medii. Comportamentul sistemului este determinat prin însumarea efectelor individuale.

Modelele statice nu iau în considerare comportamentul în timp al sistemelor. Spre exemplu, acestea nu pot fi folosite pentru a determina influența momentului în care are loc un eveniment asupra desfășurării altuia.

Modelarea dinamică este o reprezentare software a fenomenelor și proceselor care depind de timp. Dacă un model static presupune o singură rezolvare a sistemului de ecuații caracteristic fenomenului, modelarea dinamică implică o metodă iterativă. Modelul dinamic recalculează soluția pe măsura trecerii timpului. Acesta poate anticipa posibile modificări în cursul desfășurării proceselor și poate lua în considerare efecte datorate dispersiilor și fenomenelor cu caracter aleator.

Scopul simulării constă în determinarea stării reale a sistemului analizat prin rezolvarea modelului ce-l reprezintă. Există două căi distincte de rezolvare a ecuațiilor unui model: prin calcule analitice/numerice și prin dispozitive analogice. Astfel procedeele de simulare se clasifică după cum urmează:

Simularea analogică (simularea pe sistem) – se bazează pe utilizarea exclusiv a unor mijloace (echipamente) care „înlocuiesc” sistemul original cu un alt sistem fizic efectiv (denumit simulator analogic). Simulatorul analogic pe care se fac încercările experimentale este descris formal de același model ca și originalul. De aceea simularea analogică este o simulare de tip «hard» ce se realizează cu echipamente specifice care alcătuiesc instalațiile (simulatoarele). Aceste simulatoare specializate trebuie „pregătite” pentru specificul fiecărei probleme ce se dorește a fi rezolvată (schemă, structură internă de material, forma frontierei, puncte de alimentare și măsură, etc.). Toate acestea necesită, pentru fiecare problemă, noi construcții și conexiuni, deci noi cheltuieli de materiale (unele recuperabile), însă timpul de pregătire mare face ca simulatorul analogic să fie mai puțin versatil și mai scump în exploatare cu cât este mai amplu, iar dacă este de mai mică anvergură devine mai puțin fidel originalului și mai puțin precis.

Simularea numerică necesită doar mijloace (echipamente) de calcul (calculatoare). Aceasta pentru că ea constă în înlocuirea modelului original (ecuații diferențiale cu derivate parțiale) printr-un alt model, format în funcție de tipul de rezolvare al problemei (ecuații cu diferențe finite, etc.) care este, în esență, un sistem de ecuații algebrice și care formează modelul de simulare. De aceea putem spune că simularea numerică este o simulare de tip «soft». Dezvoltarea tehnicilor de calcul automat, atât în ceea ce privește creșterea în dimensiuni a volumului memoriei operative cât și a celei externe, de stocare, reducerea timpului de prelucrare prin creșterea puterii de calcul, perfecționarea algoritmilor de rezolvare numerică a sistemelor de ecuații algebrice, etc., precum și posibilitățile multiple de vizualizare a rezultatelor în vederea „corectării” în timp real a soluției, a făcut ca simularea numerică asistată de calculator să fie în prezent cel mai utilizat procedeu de simulare aplicat în obținerea soluțiilor sistemelor.

În perioada anilor 70 s-au construit și simulatoare *hibride*. Acestea se utilizau atunci când complexitatea ridicată la care s-a ajuns cu simulatoarele analogice făcea extrem de greoaie manipularea lor (ajustări, modificări, colectarea datelor măsurate, etc.) și când sistemele de calcul automat (ajunse atunci la a treia generație, în zilele de astăzi sunt la a cincia) erau mai simplu de programat și puteau lucra în timp real. Dar și aceste simulatoare hibride, cu instalațiile specializate și destul de scumpe din cauza simulatorului analogic și a interfeței «hard» (simulator analogic –

sistem de calcul), au fost repede depășite în performanțe de simularea numerică bazată pe utilizarea sistemelor de calcul universale [1].

Instrumentele modelării (sub forma pachetelor software) facilitează construirea modelelor specifice proceselor studiate. Un astfel de instrument este performant în condițiile în care este suficient de flexibil pentru a se putea adapta unor situații particulare. De asemenea trebuie să permită explorarea unor noi alternative de abordare și să arate, unde este cazul, care sunt pașii inutili pentru a putea fi eliminați.

Pachetele software pentru modelarea dinamică domină piața. Pornind de la cele cu scop general până la aplicații specializate este acoperit un domeniu extrem de vast care cuprinde atât limbajele de simulare cât și simulatoarele grafice. În timp ce în trecut modelele pentru simulare au fost dezvoltate în limbaje de programare universale, tendința actuală este de a realiza instrumente care să simplifice cât mai mult construirea modelelor, în sensul combinării facilităților puse la dispoziție de către simulatoarele grafice cu flexibilitatea limbajelor de programare. Cele mai recente tendințe în domeniul simulării vizează limbajele de simulare hibride. Acestea pot modela, la ora actuală, evenimente continue, discrete sau combinații ale acestora.

1.2. Etape în rezolvarea problemelor prin simulare numerică

Inventarea calculatoarelor digitale și introducerea lor în lumea științei și tehnologiei a impus apariția conceptului de *aproximare discretă*. Acesta cuprinde teoria aproximării numerice a unui set de ecuații, luând în considerare modelul matematic al unui sistem fizic. De asemenea, noțiunea de aproximare implică definirea modelului matematic în raport cu complexitatea lumii fizice.

Această complexitate a fenomenelor reale impune introducerea unor ipoteze simplificatoare care conduc la modele matematice cu considerarea nivelului de aproximare impus pentru a atinge o precizie acceptabilă, luând în calcul un set de variabile dependente și independente. Odată cu evoluția tehnologiei calculatoarelor ipotezele simplificatoare sunt eliminate treptat ajungându-se la simulări numerice care redau din ce în ce mai fidel fenomenul real.

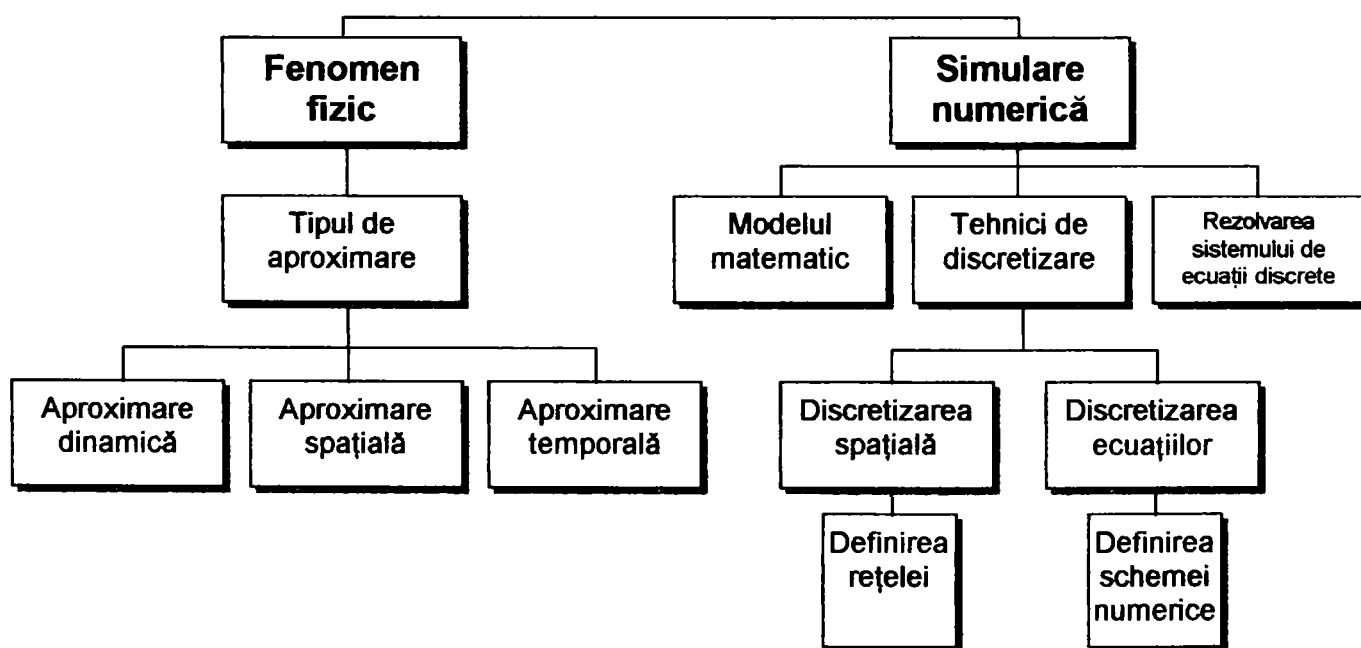


Fig. 1.1. Construirea unui model de simulare numerică

Pentru a stabili modelul matematic trebuie fixat nivelul «scării reale», cu alte cuvinte dimensiunea caracteristică (subatomică, atomică, microscopică, macroscopică și până la

astronomică). Pentru fiecare nivel al scării reale se pot asocia un set de variabile fundamentale (primare) din care se deduc celelalte variabile (caracteristice).

Întâlnim următoarele tipuri de aproximări:

- *Aproximarea temporală* implică estimarea diferitelor variabile și coeficienți dependenți de timp și alegerea pasului de timp optim pentru diferite cazuri particulare;
- *Aproximarea spațială* definește numărul de variabile spațiale utilizate în modelare. La acest nivel se stabilește dacă descrierea monodimensională (1D) sau cea bidimensională (2D), ca alternative la cea tridimensională (3D), vor furniza suficientă precizie. Deoarece procesul de mediere implică o pierdere de informație în procesul de aproximare, această informație, în cele mai multe cazuri, va fi disponibilă din exteriorul modelului, de exemplu din datele empirice. Este evident că modelele simple (de exemplu descrierea conducției termice unidirecționale 1D) necesită mai multe date de intrare empirice sau externe decât conducția 3D, dacă contribuțiile 3D sunt luate în calcul.
- *Aproximarea dinamică* este legată de estimarea influenței relative a diferitelor forțe și componentelor lor asupra comportării sistemului. Evoluția dinamică a sistemului termic, în ansamblu, este determinată de echilibrul diferitelor forțe ce acționează asupra lui, dar rareori toate componentele au o importanță egală. De aceea, unul dintre primii pași pentru estimarea modelelor matematice ce descriu sistemul este realizarea evaluării componentelor forțelor ce acționează, pentru a simplifica modelul cât mai mult posibil.

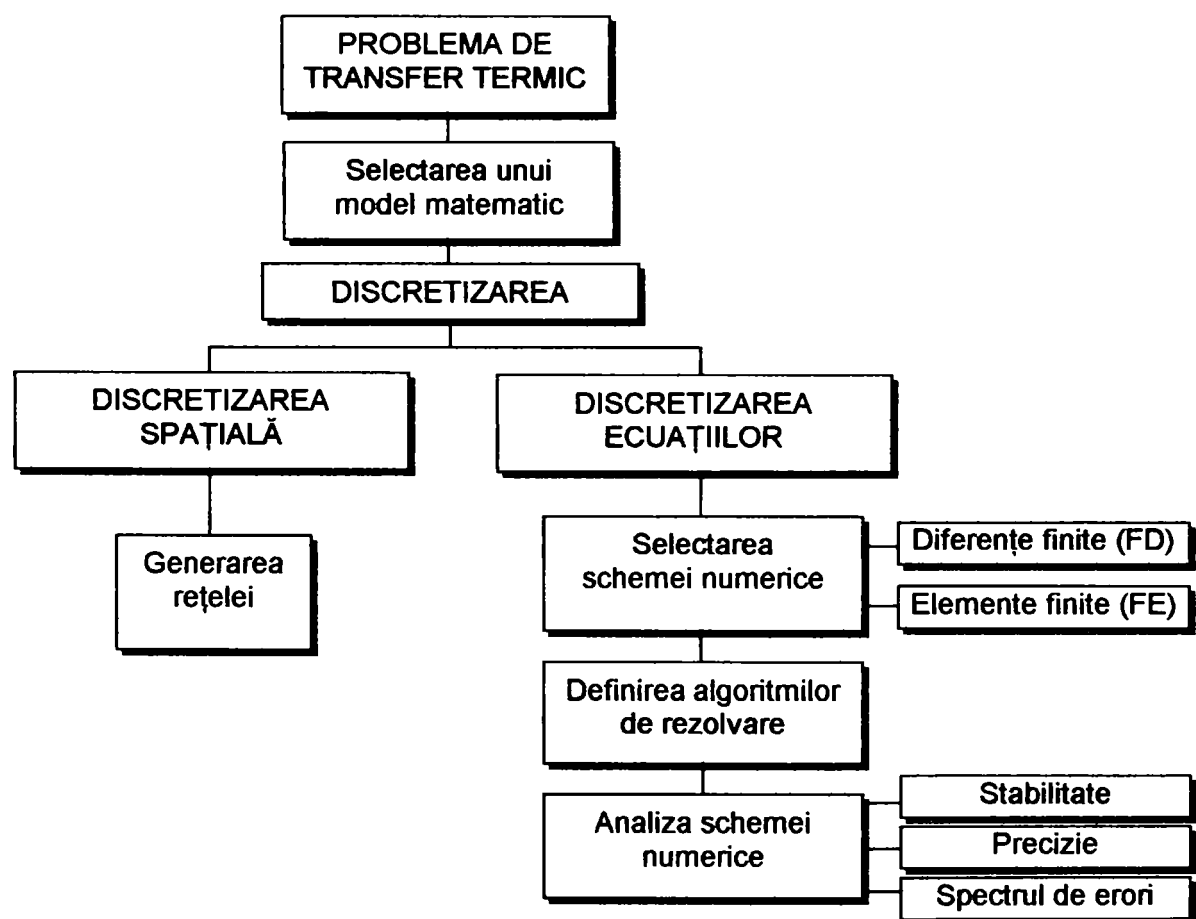


Fig.1.2. Structura unei simulări numerice

Figura 1.1. prezintă schematic interacțiunea între nivelele de aproximare ce definesc modelul matematic al unui sistem și simularea numerică în ansamblu.

Combinând diferitele tipuri de discretizări: cea spațială, care definește *rețeaua spațială* și cea a ecuațiilor, care specifică *schema numerică* se obține structura unei simulări numerice, prezentată în figura 1.2.[1].

1.3. Simularea numerică în transferul termic

În ceea ce privește domeniul extrem de vast și totodată complex al transferului termic putem spune că, la ora actuală, există un număr foarte mare de instrumente software care permit simularea efectelor termice apărute în cele mai diverse situații. De obicei, aceste instrumente vin în întâmpinarea utilizatorilor sub forma unor pachete de programe (sau module) care pot fi achiziționate în funcție de necesități concrete. Trebuie să amintim, totuși, faptul că variantele comerciale ale acestor programe au un preț destul de mare, putând varia între 1.000 și 50.000 \$. Pe lângă aceste costuri, inevitabil se vor lua în considerare și cele legate de stațiile de lucru pe care rulează aceste programe. Analize satisfăcătoare pot fi realizate pe computere care costă între 2.000 și 20.000 \$. Bine-nțeles, prețurile ridicate presupun existența unor sisteme grafice care includ monitoare cu diagonala cât mai mare, plăci grafice performante, memorie de lucru și de stocare cât mai mari, procesoare rapide, de ultimă generație și imprimante de înaltă rezoluție. Tendința actuală este de a folosi sisteme multiprocesor sau sisteme distribuite de tipul rețelelor client-server.

Pe de altă parte, funcție de complexitatea și importanța fenomenelor studiate, precum și de volumul de muncă depus în soluționarea problemelor respective, aceste costuri pot fi considerate mici comparativ cu efectele rezultatelor obținute, rezultate care se reflectă în economii de material, energie, combustibili și nu în ultimul rând timp.

În general, pachetele de programe profesionale, specializate în simularea transferului termic, sunt capabile să realizeze analize ale acestuia atât în regim staționar sau permanent, cât și în regim tranzitoriu. De asemenea pot fi efectuate analize liniare și neliniare ale acestor regimuri. Funcție de complexitatea programului respectiv, acesta poate să conțină un modul de preprocesare care să permită generarea geometriei modelului, sau o interfață care să faciliteze interacțiunea cu programe de tip CAD/CAE pentru Import/Export în formate Autodesk Inventor, CADKEY, Mechanical Desktop, ProENGINEER, SolidEdge, SolidWorks, Rhinoceros, IGES, STL, ACIS, STEP, Parasolid, FEM (diverse programe ce utilizează analiza cu elemente finite), etc.

De asemenea, aceste programe includ module care permit generarea și modificarea automată a rețelei de discretizare specifică modelului studiat. Este posibilă implementarea condițiilor la limită spațiale de ordinul I, II și III, precum și luarea în considerare a surselor interne de generare a energiei termice în cazul transferului conductiv. Poate fi realizată o analiză a fenomenului de radiație termică între două corpuri calculându-se automat coeficienții unghiulari corespunzători suprafețelor respective. Unele programe permit combinarea analizei transferului termic cu alte fenomene, cum ar fi, spre exemplu, curgerea fluidelor. De asemenea se poate realiza un transfer automat al rezultatelor obținute într-o analiză termică spre un modul care permite simularea evenimentelor mecanice, în scopul determinării tensiunilor induse de fenomenul termic.

În fine, aceste programe sunt capabile să pună la dispoziția utilizatorilor medii grafice incorporate în scopul vizualizării rezultatelor sub forma unor „hărți de culori” statice, în formate BMP, JPG, TIF, PCX, TAG, etc., sau dinamice, animate, în format AVI.

Ca exemple de astfel de pachete de programe profesionale putem aminti: ALGOR, ANSYS, COSMOS/M, etc.

1.4. Obiectivele și scopul tezei de doctorat

Printre factorii de natură tehnologică care concură la obținerea unor produse prin deformare plastică de calitate superioară se numără și modul în care semifabricatele inițiale au fost încălzite în cuptoarele care deservește secțiile respective.

Optimizarea procesului de încălzire a semifabricatelor este strâns legată de cunoașterea și stăpânirea fenomenelor complexe de transfer termic ce au loc în spațiul de lucru al cuptoarelor. Controlul fluxului de căldură transmis spre piesă se poate realiza în condițiile în care se pot obține informații suficiente și exacte cu privire la evoluția câmpului de temperaturi în interiorul incintei cuptorului. Aceste informații, la ora actuală provin, în majoritatea cazurilor, din măsurători directe de temperaturi și, eventual, debite, în puncte caracteristice ale zonelor de lucru. De obicei, aceste date se dovedesc a fi insuficiente pentru optimizarea transferului termic spre piesă. Natura fenomenelor reale care au loc în cuptor, precum și particularitățile constructive ale agregatelor de încălzire, nu permit efectuarea anumitor măsurători care ar furniza informații utile în scopul optimizării procesului în discuție. Pe de altă parte, realizarea unor instalații experimentale la scară industrială, cu ajutorul cărora să se poată studia, în particular, toate situațiile reale intervenite în practică, ar fi extrem de costisitoare și nu s-ar justifica din punct de vedere economic.

Considerațiile de mai sus au condus la ideea studiului fenomenelor de transfer termic la încălzirea oțelurilor dintr-o altă perspectivă. Prezenta teză de doctorat își propune să abordeze această problemă prin prisma unei noi tehnici de cercetare - *simularea numerică*.

Scopul final al acestui studiu constă în obținerea unui model de simulare numerică, general valabil și aplicabil tuturor cazurilor particulare de încălzire a oțelurilor în cuptoare cu flacără, model cu ajutorul căruia să poată fi studiată, prin intermediul computerului, posibilitatea optimizării transferului termic în cazurile respective, cu un cost minim (necesar acoperirii cheltuielilor cu echipamentul de calcul și software) și într-un timp foarte scurt.

Atingerea obiectivului amintit a condus la realizarea unei structuri logice a tezei în trei părți. Partea întâi, cuprinzând capitolele 1, 2 și 3 prezintă conceptele legate de modelare și simulare, structurează bazele teoretice ale transferului termic în cuptoarele industriale și evidențiază fundamentele metodelor numerice utilizate în tratarea acestor probleme.

În partea a doua, cuprinzând capitolul 4, s-a urmărit realizarea unei instalații experimentale pilot care să reproducă, la scară redusă, fenomenele specifice încălzirii unui semifabricat din oțel și care să permită efectuarea unor măsurători experimentale, dificil sau chiar imposibil de realizat la scară industrială, pe modele reale.

Parte a treia a tezei, în speță capitolele 5, 6 și 7 prezintă elaborarea a trei modele de simulare a fenomenului de transfer termic. Modelul analitic, folosind metoda clasică a seriilor Fourier este rezolvat folosind programul **MATLAB**. Sunt construite apoi două modele numerice de simulare a fenomenului de transfer termic. Primul, având la bază metoda diferențelor finite, este o realizare proprie a autorului, modelul fiind implementat cu ajutorul mediului de programare **C++**. Al doilea, folosind metoda elementelor finite, reprezintă un model construit cu ajutorul pachetului **ALGOR**.

Datele de ieșire ale celor două modele numerice, precum și cele furnizate de rezolvarea analitică vor fi comparate cu datele rezultate în urma măsurătorilor experimentale pe instalația pilot. Validarea rezultatelor modelelor numerice de către datele experimentale indică folosirea acestora în studiul optimizării proceselor de încălzire, funcție de specificul fiecărui caz în parte.

Capitolul 2

Bazele teoretice ale transferului termic în cuptoare industriale

- 2.1. Considerații generale
- 2.2. Mărimi caracteristice transferului termic
- 2.3. Transferul termic în cuptoarele industriale
 - 2.3.1. Transferul termic în camera de lucru a cuptoarelor cu flacără
 - 2.3.2. Problema conducției la încălzire
 - 2.3.3. Problema convecției la încălzire
 - 2.3.4. Problema radiației la încălzire
- 2.4. Tensiuni interne la încălzire

Capitolul 2

Bazele teoretice ale transferului termic în cuptoare industriale

2.1. Considerații generale

Transferul de căldură este disciplina care tratează procesele spontane, ireversibile în care energia se schimbă între corpuri (sisteme) sub formă de căldură. Astfel de procese au o mare răspândire în mediul înconjurător și în domeniile tehnicii actuale. Studiul transferului de căldură își propune să evidențieze efectele fizice asociate căldurii și să dezvolte metode științifice de analiză care, deși nu întotdeauna exacte, oferă informații legate de performanța sau proiectarea unui anume sistem sau proces.

Obiectivele ingineresti privind studiul transferului de căldură se pot formula astfel:

- Determinarea sau asigurarea cantității de energie termică transferată în unitatea de timp între două sisteme aflate la temperaturi date;
- Determinarea sau asigurarea unei distribuții de temperaturi care să fie compatibilă cu normele de siguranță în exploatare și de protecția muncii;
- Găsire metodelor și procedeelelor de intensificare sau frânare a transferului de energie sub formă de căldură [2].

Încercând o definiție cât mai elocventă, putem spune că transmiterea căldurii reprezintă procesul de transfer al energiei interne în cadrul aceluiași sistem termodinamic, din zonele cu temperaturi mai mari spre zonele cu temperaturi mai mici, sau între sisteme diferite, aflate la temperaturi diferite, de la sistemul cu temperatură mai mare spre cel cu temperatură mai mică. În consecință, *căldura* poate fi identificată numai la granița dintre sistemele care interacționează și există numai atâta timp cât există o diferență de temperatură între acestea. Cu toate că noțiuni ca „schimb de căldură” sau „transfer de căldură” sunt frecvent utilizate, trebuie reținut că ceea ce se transferă în mod obișnuit este energia internă termică și/sau latentă (adică energia cinetică și respectiv potențială a particulelor microscopice). Deci, căldura este o formă de transfer de energie la granița sistemelor ce conduce la schimbarea stării (energiei) acestora. Transmiterea căldurii este un fenomen prin excelență ireversibil și dependent de timp, fenomen care respectă principiile termodinamicii.

Cercetările teoretice și experimentale au pus în evidență caracterul deosebit de complex al acestui fenomen. Din această cauză studiul transmiterii căldurii se împarte în studiul a trei mari capitole corespunzătoare celor trei fenomene simple, principial diferite de transmitere a căldurii: *conducția* (energia termică este transportată de la un punct al mediului spre altul prin intermediul interacțiunii dintre atomii și moleculele materiei fără a fi implicată o mișcare în ansamblu a acesteia), *convecția* (energia termică este transportată de un fluid în mișcare, particulele fluidului acționând ca niște transportori de energie termică) și *radiația* (energia termică este transportată de unde electromagnetice și nu presupune existența unui suport material pentru transferul de căldură). Deoarece în fenomenele reale de transmiterea căldurii pot interveni, în proporții care variază de la caz la caz, toate cele trei moduri de propagare, pentru descrierea procesului global se utilizează

termenul de transmiterea căldurii, rezervându-se întrebuințarea termenilor de conducție, convecție și radiație pentru partea din proces care se desfășoară potrivit mecanismului descris mai sus.

Datorită dificultății pe care o prezintă studiul simultan al celor trei moduri de propagare a căldurii, care au de altfel legi de transmitere diferite, în cadrul procesului complex al transferului termic atenția se concentrează asupra aceluși mod de propagare care se manifestă în mod pregnant [3].

2.2. Mărimi caracteristice transferului termic

Procesul transmiterii căldurii este strâns legat de distribuția temperaturii în sistemul considerat. Totalitatea valorilor de temperatură prin care se caracterizează punctele unui spațiu la un moment dat se numește *câmp de temperaturi*. Temperatura este o mărime de stare care caracterizează gradul de încălzire a unui corp. În cazul general temperatura este o funcție de coordonate și timp. Expresiile câmpului de temperatură în coordonate carteziane, cilindrice și respectiv sferice sunt:

$$t = t(x, y, z, \tau) \quad t = t(r, \varphi, z, \tau) \quad t = t(r, \varphi, \psi, \tau) \quad (2.1)$$

unde x, y, z sunt abscisa, ordonata și respectiv cota punctului, iar r – raza, φ - latitudinea și ψ - azimutul acestuia.

Dacă temperatura variază în timp, câmpul de temperaturi se numește *nestaționar* (sau tranzitoriu, $\frac{\partial t}{\partial \tau} \neq 0$), iar dacă nu variază în timp se numește *staționar* (sau permanent, $\frac{\partial t}{\partial \tau} = 0$).

Locul geometric al punctelor care au aceeași temperatură la un moment de timp dat reprezintă o *suprafață izotermă*. Deoarece un punct dintr-un spațiu nu poate avea simultan două valori diferite ale temperaturii, rezultă că suprafețele izoterme sunt suprafețe continue care nu se intersectează între ele. Ele sunt suprafețe închise sau limitate de marginile corpului.

Să considerăm un câmp tridimensional de temperaturi, din care sunt reprezentate în figura 2.1 două suprafețe izoterme de temperaturi t și $(t + \Delta t)$. Distanța cea mai mică între cele două suprafețe, Δn , este după normala la suprafața t . Deci cea mai mare variație de temperatură pe unitatea de lungime, $\Delta t / \Delta n$, între cele două suprafețe izoterme se produce după direcția normală.

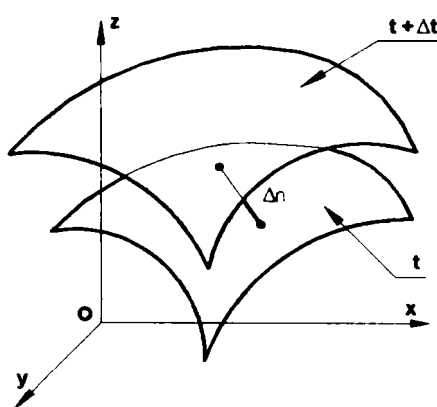


Fig. 2.1. Suprafețe izoterme

Se definește *gradientul de temperatură* drept un vector normal la suprafața izotermă, care numeric este egal cu limita raportului $\Delta t / \Delta n$, când $\Delta n \rightarrow 0$, deci cu derivata temperaturii după direcția normală:

$$\text{grad } t = \bar{n}_0 \lim_{\Delta n \rightarrow 0} \frac{\Delta t}{\Delta n} = \bar{n}_0 \frac{\partial t}{\partial n} = \bar{i} \frac{\partial t}{\partial x} + \bar{j} \frac{\partial t}{\partial y} + \bar{k} \frac{\partial t}{\partial z} = \nabla t \quad (2.2)$$

unde \bar{n}_0 este vectorul unitar normal, pozitiv în sensul de creștere a temperaturii, iar scalarul $\Delta t/\Delta n$ reprezintă valoarea gradientului de temperatură în [K/m], iar $\bar{i}, \bar{j}, \bar{k}$ sunt versorii vectorului unitar normal.

Gradientul de temperatură nu este de regulă constant de-a lungul unei izoterme, el fiind mai mare acolo unde distanța dintre izoterme este mai mică. Izotermele echidistante sunt caracteristice numai câmpului unidirecțional de temperatură.

Fluxul de căldură (sau fluxul termic), notat \dot{Q} [W] reprezintă energia termică ce se transferă printr-un corp sau prin suprafața unui corp (sistem) altui corp (sistem), în unitatea de timp.

Densitatea fluxului termic (sau fluxul unitar de căldură), notat \dot{q} [W/m²] este o mărime vectorială și reprezintă fluxul de căldură care traversează unitatea de suprafață [4].

2.3. Transferul termic în cuptoarele industriale

În conformitate cu scopul urmărit în acest studiu, vom face referiri, în continuare, la transferul de căldură, în interiorul spațiului de lucru al unui cuptor, spre piesa supusă încălzirii. Acest transfer termic poate avea loc prin conducție, prin convecție cu circulație forțată sau naturală, prin radiație de la suprafețe calde sau de la gaze, sau, de cele mai multe ori, prin mai multe din aceste moduri simultan. Mai trebuie menționate aici încălzirile inductive și dielectrice, la care însă căldura ia naștere chiar în piesă și la care, deci nu se poate vorbi de un transfer, în înțelesul adevărat al cuvântului.

Vorbim despre un transfer termic prin conducție atunci când ne referim la transferul în interiorul materialului sau când piesa este încălzită sprijinindu-se direct pe vatra cuptorului. Transferul convectiv este prezent în orice cuptor. Într-adevăr, produsul ce urmează a fi încălzit se află întotdeauna înconjurat de un anumit gaz (aer, gaze arse, gaz protector, gaz de reacție), exceptând evident cuptoarele care funcționează cu vacuum, la care procesul termic are loc într-un spațiu vidat.

Dacă conținutul gazos al spațiului cuptorului este aproximativ staționar, așa cum se întâmplă în interiorul mufelor, a cuptoarelor cu raze infraroșii, în multe cuptoare electrice și cu tuburi radiante, fără ventilatoare de recirculare, atunci avem de-a face cu convecție cu circulație naturală. Acest mod de transfer termic poate fi considerat ca o prețioasă completare a transmisiei căldurii prin radiație, dar în general nu este mare.

În cazul cuptoarelor încălzite cu combustibili, în care gazele arse se preling de-a lungul piesei în spațiul de lucru, circulația este forțată. Într-adevăr gazele arse sunt vehiculate, antrenate, exhaustate, adică «forțate» să circule. Dacă se obține o schimbare bruscă de direcție a acestei curgeri, turbulența va crește și evident va crește și transferul termic. În aceste cazuri, viteza de curgere a gazelor prin cuptor și în jurul piesei nu este prea mare și este greu de determinat din cauza diverselor modificări de secțiuni ale spațiului cuptorului (datorate unei forme mai mult sau mai puțin accidentate a piesei de încălzit). În cazul unor temperaturi ridicate, în cuptor, curgerea dirijată este însoțită și de circulație liberă. Ca urmare, în majoritatea cazurilor, nu este posibil să se determine cu precizie coeficientul de transmitere a căldurii prin convecție de la gaze spre piesă și de la gaze la pereții incintei cuptorului.

Trebuie menționat faptul că, la temperaturi de peste 800 °C, fracțiunea din căldura totală transmisă, care se datorează convecției, este în general redusă și din acest motiv precizia cu care se apreciază coeficientul de transmitere a căldurii prin convecție nu joacă un rol prea important în calculul transmisiei totale de căldură, în cuptoarele industriale cu temperaturi de regim ridicate [5].

2.3.1. Transferul termic în camera de lucru a cuptoarelor cu flacără

În camera de lucru a cuptorului, după cum se vede și din cele prezentate mai sus, schimbul de căldură are un caracter extrem de complex, prin faptul că îmbracă toate formele de transmitere a căldurii (conducție, convecție și radiație). El poate fi separat de asemenea în două fenomene distincte și anume:

- un fenomen *exterior*, de la mediul de încălzire la piesă și
- un fenomen *interior*, de la suprafața piese spre interiorul acesteia.

Se consideră spațiul de lucru al unui cuptor cu flacără, în interiorul căruia se află o piesă, sprijinită pe vatră, ce urmează a fi încălzită. Gazele arse cu temperatura t_g transmit căldura prin radiație și convecție suprafețelor interioare libere ale zidăriei cuptorului și suprafeței exterioare libere a metalului. Căldura primită de suprafața interioară a zidăriei trece, o parte, prin zidărie în mediul înconjurător, iar partea cea mai mare se transmite prin radiație suprafeței materialului. Deoarece gazele arse care umplu spațiul camerei de lucru nu sunt transparente pentru radiațiile calorice din cauza prezenței de CO_2 și H_2O , o anumită parte din radiația emisă de pereți este absorbită de gazele amintite. Fluxul de căldură transmis de gazele arse prin convecție și radiație zidăriei cuptorului este dat de relația:

$$\dot{Q}_{g-z} = \alpha_c S_z (t_g - t_z) + \alpha_R^{g-z} \varepsilon_z S_z (t_g - t_z) \quad [\text{W}] \quad (2.3)$$

unde: α_c [$\text{W}/(\text{m}^2 \cdot \text{grad})$] reprezintă coeficientul parțial de transmitere a căldurii prin convecție de la gazele arse la zidărie și suprafața piesei, S_z [m^2] – suprafața interioară liberă a zidăriei, t_g , t_z [$^\circ\text{C}$] – temperatura gazelor respectiv a interiorului zidăriei, α_R^{g-z} [$\text{W}/(\text{m}^2 \cdot \text{grad})$] reprezintă coeficientul parțial de transfer prin radiație termică de la gazele arse la zidărie, ε_z este factorul energetic de absorbție (coeficientul sau gradul de negreală) al zidăriei.

Fluxul termic din relația (2.3) se transmite, la rândul său, la suprafața metalului de încălzit prin radiație, iar o parte servește la acoperirea pierderilor prin pereți în mediul ambiant, adică:

$$\dot{Q}_{g-z} = \dot{Q}_{z-m} + S_z \dot{q}_p \quad [\text{W}] \quad (2.4)$$

unde \dot{Q}_{z-m} [W] este fluxul termic transmis prin radiație de către zidărie metalului, iar \dot{q}_p [W/m^2] este densitatea fluxului de căldură pierdut spre exterior prin zidărie.

Fluxul de căldură transmis de la zidărie la metal se consumă o parte pentru încălzirea acestuia și o parte, așa cum s-a arătat, este absorbit de gazele arse din spațiul de lucru al cuptorului. Acest ultim flux este dat de relația:

$$\dot{Q}_{\text{abs.g}} = \alpha_R^{g-m} \varepsilon_z S_m (t_z - t_m) \quad [\text{W}] \quad (2.5)$$

unde α_R^{g-m} [$\text{W}/(\text{m}^2 \cdot \text{grad})$] este coeficientul parțial de transmitere a căldurii prin radiație de la gaze, considerate la temperatura pereților, la suprafața materialului, S_m [m^2] este suprafața exterioară a materialului, t_m [$^\circ\text{C}$] – temperatura suprafeței materialului.

Pentru exprimarea fluxului de căldură transmis de la pereți la suprafața materialului de încălzit se introduce coeficientul reciproc de negreală:

$$\varepsilon_{z-m} = \frac{1}{\varepsilon_m + \frac{S_m}{S_z} \begin{pmatrix} 1 & \\ & -1 \end{pmatrix}} \quad (2.6)$$

unde ε_m este factorul energetic de absorbție (gradul de negreală) al suprafeței materialului.

Cu aceste date se poate scrie cantitatea de căldură primită de material de la pereții cuptorului:

$$\dot{Q}_{z-m} = \alpha_R^{z-m} \varepsilon_{z-m} S_m (t_z - t_m) - \alpha_R^{zm} \varepsilon_z S_m (t_z - t_m) \quad [W] \quad (2.7)$$

relație în care primul termen reprezintă fluxul de căldură emis de suprafața pereților, iar al doilea – fluxul de căldură absorbit de gazele arse.

Înlocuind relațiile (2.3) și (2.7) în relația (2.4) se obține o relație care leagă între ele cele trei temperaturi: t_g , t_z și t_m . Notând raportul $\frac{S_m}{S_z} = \Psi$ și ordonând în funcție de cele trei temperaturi se obține:

$$t_g (\alpha_R^{g-z} \varepsilon_z + \alpha_C) = t_z (\alpha_R^{g-z} \varepsilon_z + \alpha_C + \Psi \alpha_R^{z-m} \varepsilon_{z-m} - \Psi \alpha_R^{zm} \varepsilon_z) - t_m \Psi (\alpha_R^{z-m} \varepsilon_{z-m} - \alpha_R^{zm} \varepsilon_z) + \dot{q}_p \quad (2.8)$$

Suprafața liberă a materialului primește în ansamblu următoarele densități de flux de căldură:

- prin radiație de la pereți:

$$\dot{q}_{z-m} = (t_z - t_m) (\alpha_R^{z-m} \varepsilon_{z-m} - \alpha_R^{zm} \varepsilon_z) \quad [W/m^2] \quad (2.9)$$

- prin convecție și radiație de la gazele arse:

$$\dot{q}_{g-m} = (t_g - t_m) (\alpha_C + \alpha_R^{g-m} \varepsilon_m) \quad [W/m^2] \quad (2.10)$$

Densitatea fluxului termic total primit de suprafața materialului va fi atunci:

$$\dot{q}_{tot} = \dot{q}_{z-m} + \dot{q}_{g-m} \quad [W/m^2] \quad (2.11)$$

Relațiile de bilanț termic stabilite permit determinarea coeficientului complex de transfer de căldură α în spațiul de lucru al cuptorului. În urma unor calcule laborioase se obțin relații de forma:

$$\alpha = f(\alpha_R^{g-m}, \alpha_R^{g-z}, \alpha_R^{zm}, \alpha_R^{z-m}, \alpha_C, \varepsilon_z, \varepsilon_{z-m}, \varepsilon_m, \Psi, t_m, t_z, t_g, \dot{q}_p) \quad [W/(m^2 \cdot \text{grad})] \quad (2.12)$$

După cum se observă, coeficientul complex α este funcție, printre altele, de cinci coeficienți parțiali de transfer termic, specifici unor fenomene cu caracter diferit. De aceea studiul transferului de căldură în interiorul cuptorului se va elabora în mod distinct pentru fiecare formă particulară, insistându-se asupra implicațiilor termo-tehnologice specifice [6].

2.3.2. Problema conducției la încălzire

a) Ecuația diferențială generală a conducției termice

La încălzirea unui material într-un cuptor, o deosebită importanță o are rezolvarea problemei de bază a teoriei conducției termice, rezumată la determinarea distribuției temperaturilor în corp în funcție de coordonate și timp. Legătura între variația temperaturii în spațiu și în timp este exprimată prin ecuația diferențială a conducției termice. Pentru cazul cel mai general această ecuație, în coordonate carteziane, are forma următoare:

$$\frac{\partial t}{\partial \tau} = \frac{1}{\rho c} \left[\frac{\partial}{\partial x} \left(\lambda \frac{\partial t}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda \frac{\partial t}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial t}{\partial z} \right) \right] + \frac{\dot{q}_v}{\rho c} \quad (2.13)$$

unde $x[m], y[m], z[m]$ sunt coordonatele carteziane, $t [^{\circ}C]$ – temperatura și $\tau[s]$ – timpul.

sau:

$$\frac{\partial t}{\partial \tau} = \frac{1}{\rho c} [\text{div}(\lambda \text{grad}t) + \dot{q}_v] \quad (2.14)$$

Relațiile (2.13) și (2.14) exprimă ecuația diferențială în cazul cel mai general al conducției termice nestaționare (variabile în timp), tridimensionale, cu surse interne de căldură, caracterizate prin densitatea volumică de flux (sau debitul surselor interioare de căldură) $\dot{q}_v [W/m^3]$, reprezentând fluxul termic degajat în volumul unitar. De obicei se lucrează cu variante simplificate ale ecuației generale. Dacă, spre exemplu, proprietățile termo-fizice ale materialului supus încălzirii și anume: densitatea - $\rho [kg/m^3]$, căldura specifică - $c [J/(kg \cdot \text{grad})]$ și conductivitatea termică $\lambda [W/(m \cdot \text{grad})]$ sunt considerate constante cu temperatura, ecuația diferențială a conducției termice devine:

$$\frac{\partial t}{\partial \tau} = \frac{\lambda}{\rho c} \left(\frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} + \frac{\partial^2 t}{\partial z^2} \right) + \frac{\dot{q}_v}{\rho c} \quad (2.15)$$

Raportul $\frac{\lambda}{\rho c} = a [m^2 / s]$ se numește *difuzivitate termică* și reprezintă unitatea de măsură a vitezei

de nivelare a temperaturii în diferite puncte ale corpului.

Introducând operatorul lui Laplace ∇^2 relația (2.15) devine:

$$\frac{\partial t}{\partial \tau} = a \nabla^2 t + \frac{\dot{q}_v}{\rho c} \quad (2.16)$$

În cazul în care nu există surse interne de căldură și deci $\dot{q}_v = 0$, relația (2.16) se reduce la:

$$\frac{\partial t}{\partial \tau} = a \nabla^2 t \quad (2.17)$$

Din (2.17) se vede că viteza de variație a temperaturii într-un punct este direct proporțională cu curbura câmpului de temperaturi în punctul respectiv, coeficientul de proporționalitate fiind

difuzivitatea termică a . Cu cât a este mai mare uniformizarea câmpului de temperaturi se face mai repede. Valori mari ale lui a se întâlnesc la metale.

Atunci când domeniul de analiză prezintă o simetrie axială, este mai convenabil ca ecuația diferențială a conducției termice să fie exprimată în coordonate cilindrice sau sferice.

În coordonate cilindrice ecuația se scrie astfel:

$$\frac{\partial t}{\partial \tau} = \frac{1}{\rho c} \frac{1}{r} \frac{\partial}{\partial r} \left(\lambda r \frac{\partial t}{\partial r} \right) + \frac{1}{r^2} \frac{\partial}{\partial \varphi} \left(\lambda \frac{\partial t}{\partial \varphi} \right) + \frac{\partial}{\partial z} \left(\lambda \frac{\partial t}{\partial z} \right) + \frac{\dot{q}_v}{\rho c} \quad (2.18)$$

unde r [m] este raza, φ este latitudinea, iar z [m] este cota punctului considerat.

În coordonate sferice ecuația are forma:

$$\frac{\partial t}{\partial \tau} = \frac{1}{\rho c} \frac{1}{r^2} \frac{\partial}{\partial r} \left(\lambda r^2 \frac{\partial t}{\partial r} \right) + \frac{1}{r^2 \sin^2 \psi} \frac{\partial}{\partial \varphi} \left(\lambda \frac{\partial t}{\partial \varphi} \right) + \frac{1}{r^2 \sin \psi} \frac{\partial}{\partial \psi} \left(\lambda \sin \psi \frac{\partial t}{\partial \psi} \right) + \frac{\dot{q}_v}{\rho c} \quad (2.19)$$

unde r [m] este raza, φ este latitudinea, iar ψ este azimutul punctului considerat.

În cazul în care conducția termică are loc în regim permanent (staționar), variația în timp a temperaturii este nulă, adică: $\frac{\partial t}{\partial \tau} = 0$.

b) Condiții de determinare univocă

Pentru rezolvarea unor probleme de transfer termic, alături de ecuația diferențială aferentă procesului, trebuie să se cunoască și elementele descriptive specifice procesului analizat, care constituie așa numitele *condiții de determinare univocă* sau *condiții de univocitate*. Acestea se referă la condițiile generale, geometrice, fizice, inițiale și la limită, care permit rezolvarea ecuațiilor prin diferite metode.

- *Condițiile generale* de desfășurare a proceselor termice conductive privesc următoarele aspecte:
 - materialul este omogen sau eterogen,
 - materialul este izotrop sau anizotrop,
 - materialul conține sau nu surse interne de căldură,
 - regimul termic este permanent sau tranzitoriu,
 - propagarea căldurii are loc uni, bi, sau tridirecțional (tridimensional).
- *Condițiile geometrice* precizează forma geometrică și dimensiunile corpului în care are loc procesul termic conductiv.
- *Condițiile fizice* stabilesc valorile proprietăților termo-fizice ale corpului:
 - conductivitatea termică λ [W/(m·grd)],
 - căldura specifică c [J/(kg·grd)],
 - densitatea ρ [kg/m³],
 - difuzivitatea termică a [m²/s],

precum și variația acestora în spațiu sau în funcție de un alt parametru (temperatura, de exemplu). De asemenea, aceste condiții evidențiază variația în timp și spațiu a surselor interne de căldură.

- *Condițiile inițiale* (condiții la limită de timp, sau temporale) se referă la distribuția temperaturilor în interiorul corpului analizat la momentul inițial ($\tau = 0$). În cazul general, această distribuție poate fi exprimată analitic sub forma: $t|_{\tau=0} = f(x, y, z)$ [°C]. Cazul cel

mai simplu îl reprezintă distribuția uniformă a temperaturii în mediul considerat: $t|_{\tau=0} = t_0 = \text{const.}$ Ecuația diferențială a conducție termice în regim tranzitoriu fiind de ordinul unu în variabila «timp», rezolvarea ei necesită o singură condiție inițială.

- **Condițiile de suprafață** (condiții la limită spațiale sau de contur). Acestea definesc interacțiunile termice dintre suprafețele corpului considerat și mediul înconjurător (mediul de încălzire sau de răcire). Există trei feluri de condiții de suprafață:
 - *de ordinul I* (sau de tip *Dirichlet*) – precizează distribuția temperaturii la suprafața corpului în fiecare moment τ . Ea are forma $t_s = f(x, y, z, \tau)$, unde t_s este temperatura suprafeței, iar (x, y, z) sunt coordonatele suprafeței respective. În cazul particular, în care temperatura suprafeței rămâne constantă pe durata procesului de transfer termic, ecuația precedentă se scrie sub forma $t_s = \text{ct.}$
 - *de ordinul II* (sau de tip *Neumann*) – stabilește valoarea fluxului termic unitar schimbat cu mediul ambiant pe o porțiune din suprafața corpului, în orice moment τ : $\dot{q}_s = f(x, y, z, \tau)$. În cazul cel mai simplu, densitatea fluxului termic rămâne constantă pe toată suprafața considerată: $\dot{q}_s = \text{ct.}$ Cazuri speciale sunt reprezentate de suprafețe perfect izolate (adiabate) și suprafețe plasate într-un plan de simetrie, pentru care $\dot{q}_s = 0$ (gradient de temperatură nul la suprafață)
 - *de ordinul III* (sau de tip *Cauchy* sau *Robin*) – presupune corpul în contact cu un fluid a cărui temperatură este cunoscută, t_f . Procesul de transfer termic între un fluid și un solid este de tip convectiv și este precizat de legea lui Newton. Conservarea energiei la fiecare unitate de suprafață a corpului analizat impune egalitatea fluxului termic conductiv ce părăsește corpul cu fluxul termic convectiv preluat de fluidul aflat în contact, considerând radiația termică neglijabilă. Deci, condiția la limită de ordinul al III-lea se reduce la egalitatea: $-\lambda \frac{\partial t}{\partial n} \Big|_s = \alpha_c (t_f - t_s)$ [W/m^2], unde α_c este coeficientul de transfer termic prin convecție, considerat cunoscut.
 - *de ordinul IV* – definește conservarea fluxului termic conductiv la suprafața de separare dintre corpul (mediul) considerat, având temperatura t_1 și conductivitatea termică λ_1 și un alt corp (mediu) având temperatura t_2 și conductivitatea termică λ_2 : $-\lambda_1 \frac{\partial t_1}{\partial n} \Big|_s = -\lambda_2 \frac{\partial t_2}{\partial n} \Big|_s$, unde n este normala pe suprafața de separare (de contact) [7].

c) Metode de rezolvare a ecuației generale a conducției termice

Există, în principiu, trei metode de rezolvare a modelului matematic format din ecuația diferențială generală a conducției termice împreună cu condițiile inițiale și spațiale:

- **Metoda analitică.** Rezolvarea analitică a ecuației diferențiale generale a conducției termice tridimensionale, în cazul regimurilor tranzitorii, este o sarcină foarte dificilă și greu de realizat. De aceea, pentru rezolvarea analitică se recurge la folosirea unor condiții de univocitate, luând în considerare corpuri cu geometrie simplă. Literatura de specialitate prezintă soluțiile unor probleme strict particulare, cum ar fi: cazul conducției unidimensionale (perete plan sau placă de lățime și lungime infinite și grosime finită, cilindru de lungime infinită, sferă, corpuri semiinfinite, etc.), cazul conducției bidimensionale (placă cu două dimensiuni finite și una infinită, etc.). Cazul complicat al conducției tridimensionale poate fi rezolvat analitic, spre exemplu, prin metoda

superpoziției, care constă în determinarea soluției ecuației tridimensionale prin suprapunerea soluțiilor ecuației conducției pentru cazurile bidimensionale, sau folosind metoda tipică a seriilor Fourier [8].

- *Metoda numerică.* În numeroase aplicații practice, corpurile au forme geometrice și/sau condiții la limită complexe (de exemplu, condițiile la limită pot varia cu timpul), ceea ce face imposibilă soluționarea analitică a problemelor de conducție tranzitorie. De asemenea, în timpul transmiterii căldurii se poate modifica temperatura mediului ambiant, coeficientul de transfer termic, sau proprietățile termo-fizice ale materialului ca urmare a modificării temperaturii. Temperatura inițială în corp poate să nu fie uniformă, ci să aibă o distribuție oarecare ce nu permite integrarea analitică a ecuației lui Fourier. Corpul poate avea și surse interne de căldură, constante sau variabile în timp, uniform sau neuniform distribuite. Situații atât de complexe sunt imposibil de rezolvat analitic. În aceste cazuri, problemele sunt tratate prin tehnici numerice de calcul, ca de exemplu metoda cu diferențe finite, metoda cu element finit, etc. Aceste metode numerice constau în înlocuirea sistemului de ecuații diferențiale cu derivate parțiale ce descrie fenomenul particular, cu un set de ecuații liniare ce pot fi manipulate de un calculator electronic.
- *Metoda criterială.* Este o metodă folosită pe scară largă în practica tratamentelor termice și termochimice, unde soluțiile trebuie să se obțină ușor, rapid și să aibă un sens fizic. Metodele criteriale (asocierea de variabile, în condițiile de univocitate precizate mai sus, sub forma unor relații de calcul aproximativ) permit o simplificare importantă a modului de operare cu acestea. Folosirea relațiilor criteriale este justificată doar dacă aproximările respective nu conduc la valori ale parametrilor regimului termic net diferite de cele care asigură eficiență ridicată procesului respectiv. Soluțiile criteriale ale ecuației diferențiale a conducției termice prin corpuri solide permit determinarea temperaturii momentane, temperaturii medii și gradientului termic în funcție de timp, coordonată, de temperatura inițială și de cea a mediului de încălzire (răcire) și de mărimile termo-fizice (ale materialului corpului și mediului) ce intră în condițiile de univocitate. Criteriile de similitudine folosite în tratarea problemelor de conducție termică sunt:
 - *criteriul Biot* (Bi) sau dimensiune caracteristică (de calcul),
 - *criteriul Fourier* (Fo) sau timp relativ,
 - *criteriul Boussinesq* (Bq),
 - *criteriul de poziție* sau coordonată relativă.

d) Mărimi termo-fizice

Ca urmare a modificării temperaturii, în metalul încălzit au loc următoarele fenomene:

- modificarea proprietăților fizice și mecanice,
- modificări volumetrice, legate de dilatarea termică,
- modificări structurale și transformări fazice,
- fenomene fizico-chimice, cum ar fi oxidarea și decarburarea.

În cazul particular al soluționării analitice a problemei conducției termice, mărimile termo-fizice care apar în calcule, pentru a simplifica rezolvarea, se consideră invariabile cu temperatura; realitatea este însă cu totul alta.

Conductivitatea termică (λ). În cazul oțelurilor, conductivitatea termică depinde de compoziția chimică, de temperatură, de gradul de impurificare, precum și de condițiile de prelucrare. Odată cu creșterea conținutului de carbon, conductivitatea termică scade. Literatura de specialitate pune la dispoziție o serie de relații de calcul al conductivității termice a oțelurilor funcție de conținutul de carbon și elemente de aliere. De asemenea, pentru diferite tipuri de oțel,

există relații de calcul a conductivității termice la o anumită temperatură, funcție de valoarea acesteia la temperatura de 0 °C.

În urma analizării tuturor datelor existente în literatură, se pot trage următoarele concluzii:

- ⇒ conductivitatea termică scade cu majorarea conținutului de incluziuni nemetalice în oțel. Pentru oțelurile aliate, aceasta este de regulă mai mică decât a oțelurilor carbon, iar oțelurile puternic aliate au conductivitate termică mai scăzută decât cele slab aliate. De asemenea, conductivitatea termică se micșorează odată cu ridicarea conținutului de carbon;
- ⇒ conductivitatea termică a oțelurilor carbon se micșorează cu atât mai mult la ridicarea temperaturii, cu cât este mai ridicată valoarea acesteia la temperatura mediului ambiant;
- ⇒ conductivitatea termică a oțelurilor aliate nu se modifică aproape deloc, dacă la temperatura mediului ambiant $\lambda = 23,3...48,84 \text{ W/(m}\cdot\text{grd)}$, și crește dacă $\lambda < 23,3 \text{ W/(m}\cdot\text{grd)}$ [6].

Căldura specifică (c). În cazul oțelului, la fel ca și pentru orice aliaj, aceasta depinde de compoziția lui chimică, de temperatură și de structură. Fierul face parte din grupa metalelor la care variația căldurii specifice în raport cu temperatura nu este supusă unor legi precise pentru fiecare interval de temperatură. Există și în acest caz o serie de relații în literatura de specialitate, cu ajutorul cărora se poate determina valoarea căldurii specifice a oțelurilor, funcție de temperatură și compoziția chimică. Analizându-se aceste date se pot concluziona următoarele:

- ⇒ compoziția chimică a oțelului influențează într-o mică măsură asupra căldurii specifice;
- ⇒ la ridicarea temperaturii, căldura specifică, atât a oțelurilor carbon cât și a oțelurilor aliate, crește considerabil;

Difuzivitatea termică (a), fiind o mărime care depinde, conform relației de definiție, de valorile lui λ , c , și ρ , este și ea influențată de temperatură și compoziția chimică.

2.3.3. Problema convecției la încălzire

Schimbul de căldură prin convecție caracterizează o parte din fenomenele exterioare ale încălzirii, el determinând cantitatea de căldură transmisă în spațiul de lucru al cuptorului spre materialul de încălzit, purtătorii de energie termică fiind particulele în mișcare ale unui fluid. Acest mod de schimb de căldură devine preponderent la cuptoarele de încălzire de joasă temperatură (sub 600...700 °C) în care gazele au o viteză considerabilă (peste 5 m/s), sau în cazul cuptoarelor cu funcționare în regim variabil la care, în perioada de început a încălzirii, temperatura medie nu depășește 700 °C.

Legea lui Newton (2.20) exprimă densitatea fluxului termic, $\dot{q}_{f-s} [\text{W/m}^2]$, transmis prin convecție, între un fluid și o suprafață, sub forma unui produs între o proprietate a sistemului (α_c) și forța care generează procesul (diferența de temperatură, numită și diferență de potențial termic).

$$\dot{q}_{f-s} = \alpha_c (t_f - t_s) [\text{W/m}^2] \quad (2.20)$$

Este evident că temperaturile fluidului (t_f) și suprafeței (t_s) trebuie să fie diferite pentru a exista schimbul de căldură între suprafața considerată și fluidul înconjurător.

Definirea în acest mod simplu a unui mecanism de transfer în esență complicat, conferă coeficientului de transfer de căldură prin convecție, α , proprietatea de a îngloba toți factorii care influențează procesul de convecție. Aceștia sunt, în esență, următorii:

- ⇒ natura fluidului;
- ⇒ caracterul curgerii fluidului;

- ⇒ viteza curgerii fluidului;
- ⇒ forma geometrică a suprafeței;
- ⇒ poziția și așezarea suprafeței în spațiu;
- ⇒ starea și proprietățile fizice ale fluidului:
 - temperatura;
 - presiunea;
 - conductivitatea termică;
 - vâscozitatea dinamică sau cinematică;
 - căldura specifică;
 - densitatea.

Datorită numărului mare al acestor factori ce influențează mărimea coeficientului α_C , nu se poate pune la dispoziție o formulă generală pentru determinarea lui. De cele mai multe ori, valoarea lui α_C se determină cu ajutorul ecuațiilor criteriale obținute prin metoda similitudinii, prin combinarea ecuațiilor matematice cu rezultatele experimentelor practice efectuate pe modele.

Similitudinea este una din metodele de lucru ale fizicii moderne cu rezultate fructuoase, ea constând în determinarea unor criterii de asemănare între fenomene astfel ca, prin cunoașterea unui anumit fenomen (denumit generic *original*) să se poată trage concluzii juste asupra desfășurării fenomenelor asemenea cu acesta (denumite *modele*).

Primul care a utilizat similitudinea în cazul convecției termice a fost Ernst Kraft Wilhelm Nusselt (1882-1957). Odată cu folosirea acestei metode de lucru s-a făcut ordine în multitudinea de rezultate experimentale existente pentru convecție, rezultate care erau însă valabile numai pentru cazul concret cercetat experimental.

Nusselt a aplicat la studiul convecției termice cea mai riguroasă metodă, plecând de la ecuațiile diferențiale care descriu fenomenul în orice punct al câmpului și anume:

- ecuația de continuitate;
- ecuația de mișcare a fluidelor vâscoase (Navier-Stokes);
- ecuația conservării energiei;
- ecuația de contur a convecției.

Acestea sunt cele patru ecuații diferențiale prin care fenomenul de transmitere a căldurii prin convecție este descris complet. Scrise pentru *model*, aceste ecuații diferențiale ale *originalului* vor avea alte valori ale mărimilor ce intră în relații. Pentru ca fenomenele descrise de cele două sisteme de ecuații (aferele *originalului* și *modelului*) să fie similare, trebuie ca raportându-le între ele, rezultatul să fie egal cu unitatea. În urma acestui raționament se deduc o serie de invarianți sau criterii de similitudine cu ajutorul cărora pot fi soluționate probleme particulare ale fenomenului de transfer termic prin convecție. Cele mai relevante criterii adimensionale utilizate în studiul transferului de căldură în camera de lucru a cuptoarelor sunt:

- ⇒ *criteriul Nusselt* - poate fi interpretat ca raport între fluxul termic convectiv și fluxul termic conductiv asociate cu transferul termic printr-un strat de fluid;
- ⇒ *criteriul Grashof* – poate fi interpretat ca raport între forțele ascensionale (arhimedice) și forțele vâscoase, reprezentând pentru convecția naturală (sau liberă) ceea ce criteriul Reynolds reprezintă pentru convecția forțată;
- ⇒ *criteriul Prandtl* – este exclusiv o proprietate a fluidului și reprezintă raportul dintre difuzivitatea moleculară a impulsului și difuzivitatea moleculară a energiei termice, în stratul limită dinamic, respectiv termic;
- ⇒ *criteriul Reynolds* – reprezintă cea mai importantă caracteristică a curgerii fluidelor prin conducte sau a mișcării solidelor imersate într-un fluid. El reprezintă raportul dintre forțele de inerție și cele de frecare dintr-un fluid vâscos.

Scopul cercetărilor experimentale asupra propagării căldurii prin convecție este determinarea coeficientului de transfer de căldură prin convecție, α_C . În acest scop, pentru cazul mișcării fluidului

642.440
369 E

și transmiterea căldurii prin convecție și conducție în stratul limită, se pot determina funcții criteriale de forma:

$$\text{Nu} = f(\text{Re}, \text{Pr}) \quad (2.21)$$

Din relația (2.21) se vede că se poate determina imediat valoarea lui α_c cu condiția să se determine, teoretic sau experimental, funcția care leagă între ele criteriile Nu, Re și Pr.

Dacă este vorba de o mișcare relativ lentă a fluidului, cum este cea cauzată de convecția liberă, criteriul Reynolds își pierde importanța, având valoare foarte mică, în schimb apare criteriul Grashof:

$$\text{Nu} = f(\text{Gr}, \text{Pr}) \quad (2.22)$$

În cazul în care, pentru studiul unui fenomen de convecție particular, nu se cunosc ecuațiile care descriu complet fenomenul, pentru a stabili relația criterială între valoarea α_c și factorii care o influențează, se recurge la metoda analizei dimensionale. În cadrul acestei metode, relațiile cantitative între mărimea studiată și factorii influenți se scriu de obicei sub formă de funcții exponențiale (ecuații criteriale). Exponenții care apar în ecuațiile criteriale se determină experimental [6].

2.3.4. Problema radiației la încălzire

Schimbul de căldură prin radiație reprezintă principalul mod de transmitere a căldurii în cazul cuptoarelor având temperaturi peste 800 °C. La aceste cuptoare, circa 85...95% din cantitatea totală de căldură se transmite la suprafața corpurilor încălzite prin radiație și numai 5...15% prin convecție. Cu cât temperatura cuptorului este mai ridicată, cu atât schimbul de căldură prin radiație va deține o pondere mai însemnată în totalul fluxului termic transmis.

După cum s-a arătat în subcapitolul 2.3.1, în spațiul util al cuptoarelor cu flacără, unde piesele sunt încălzite direct de gaze, suprafața liberă a materialului supus încălzirii primește un flux de căldură radiativ format din două componente: una provenită din radiația gazelor din incintă și alta din radiația de suprafață a zidăriei interioare a cuptorului.

A. Schack a atras primul atenția, în 1924, asupra aportului mare al radiației gazelor în spațiul de încălzire tehnologică. Tot calculele sale sunt primele baze de calcul. De asemenea, Schack denumesc pereții înconjurători care transferă căldură prin radiație spre piesă, suprafețe indirecte [5].

Relativ la schimbul de căldură prin radiație între suprafețele interioare ale zidăriei cuptorului și suprafața liberă a materialului supus încălzirii putem preciza faptul că, metoda de calcul se poate stabili în modul cel mai simplu, considerând că schimbul de căldură are loc între două plane paralele, cu suprafețe infinite, deși un asemenea caz nu se întâlnește în practică.

Pentru a ne apropia mai mult de fenomenul real, este necesară introducerea unor date suplimentare cu privire la poziția relativă a suprafețelor între care are loc schimbul de căldură prin radiație. În funcție de aceasta, se definește o mărime numită *coeficient unghiular* [9] (factor de formă, sau coeficient de iradiere), definit ca fiind fracțiunea din radiație unei suprafețe îndreptată spre o altă suprafață și este funcție doar de mărimi geometrice. Acest coeficient unghiular se notează cu $\phi_{i,j}$, unde indicii i și j precizează suprafețele radiantă și respectiv absorbantă.

Fracțiunea din radiația unei suprafețe concave, care se îndreaptă asupra aceleași suprafețe, se notează prin coeficientul unghiular, având un indice format din două cifre sau litere egale, corespunzând suprafeței date, spre exemplu ϕ_{11} sau ϕ_{22} (după cum se face referire la suprafața notată S_1 sau suprafața notată S_2). Dacă suprafața este plană, ea nu poate emite radiații asupra ei

însăși. Astfel, dacă S_1 este o suprafață plană, atunci $\phi_{11} = 0$. Valorile numerice ale coeficienților unghiulari propriu-ziși variază în intervalul de la 0 la 1.

Pentru simplificarea deducțiilor, în studiul schimbului de căldură prin radiație în spațiul de lucru închis al cuptorului, în prezența unui mediu radiant și absorbant, se admite că această capacitate a mediului gazos de a absorbi energia radiantă se referă la orice lungime de undă și gazul nu are capacitate de reflexie, ci posedă o anumită transparență pentru toate lungimile de undă.

2.4. Tensiuni interne la încălzire

În procesul de prelucrare la cald prin presiune, încălzirea materialului metalic are o influență hotărâtoare asupra calității produselor obținute. La alegerea metodei de încălzire trebuie să se țină seama de proprietățile materialului metalic și de felul prelucrării lui ulterioare.

Astfel, încălzirea lingourilor cere o atenție și prudență mai mare decât încălzirea semifabricatelor, ceea ce se explică prin particularitățile structurii și însușirilor materialului metalic turnat mai predispus la formarea defectelor în timpul procesului de prelucrare la cald.

Factorii principali care condiționează influența încălzirii asupra calității sunt: temperatura și uniformitatea încălzirii, viteza de încălzire și starea atmosferei cuptorului.

Însumând toate acestea, este necesar să observăm că temperatura de încălzire și uniformitatea încălzirii sunt factorii cei mai importanți și poate chiar mai importanți decât viteza sau durata încălzirii [10].

Problema influenței regimului de încălzire asupra calității capătă o importanță deosebită pentru oțelurile cu conținut ridicat de carbon și cele aliate. Din cauza diversității compoziției, tehnologia încălzirii acestor oțeluri este mai complicată și mai de răspundere decât pentru oțelurile moi și pentru cele cu conținut mediu de carbon.

Într-adevăr, oțelurile cu conținut ridicat de carbon și oțelurile aliate posedă o serie de particularități care îngreunează și întârzie procesele de încălzire și anume:

- au coeficienți mai mici de conductivitate termică;
- sunt mai sensibile la factorii care provoacă defectarea materialului metalic: la supraîncălzire și ardere, la formarea tensiunilor termice și a crăpăturilor, etc.;
- în timpul prelucrării sunt mai rezistente la deformare și în consecință au limitele temperaturilor de prelucrare la cald mai strânse, necesitând o uniformitate mai mare a încălzirii în grosime, pe perimetru și în lungime [11].

Datorită diferenței de temperatură pe secțiunea piesei în curs de încălzire, dilatarea acesteia este neuniformă. Dilatarea dintre straturile vecine, rigid legate între ele, fiind reciproc frânată, dilatățile posibile dar nerealizate se transformă în deformații elastice locale, cărora le sunt asociate tensiuni interne, a căror mărime poate fi calculată cu aproximație din relația simplificată:

$$\sigma_1 + \sigma_2 + \sigma_3 = \frac{2\beta E}{1-\nu} \Delta T \quad [\text{N/mm}^2] \quad (2.23)$$

unde: $\sigma_1 > \sigma_2 > \sigma_3$ - sunt tensiunile principale ce acționează pe laturile unui element de volum cubic din secțiunea piesei; E [N/mm^2] - este modulul de elasticitate longitudinal; ν [-] - coeficientul lui Poisson, caracteristic materialului din care este executată piesa; β [$1/\text{K}$] - coeficientul de dilatare liniară; ΔT [K] - diferența dintre temperatura elementului de volum considerat și temperatura medie a piesei [12].

Admițând că tensiunea principală maximă este dată de relația:

$$\sigma_{\max} = \frac{\sigma_1 + \sigma_2 + \sigma_3}{2} \quad [\text{N/mm}^2] \quad (2.24)$$

rezultă că tensiunea maximă ce acționează asupra elementului de volum dat va fi:

$$\sigma_{\max} = \frac{\beta E}{1 - \nu} \Delta T \quad [\text{N/mm}^2] \quad (2.25)$$

Relația (2.25) arată că tensiunile interne provocate de neuniformitatea încălzirii pe secțiune cresc direct proporțional cu ΔT și depind de caracteristicile fizico- mecanice ale materialului supus încălzirii (β , E , ν).

Pentru a calcula ΔT se determină, mai întâi, temperatura medie pe întreaga secțiune a corpului, cu ajutorul relațiilor:

$$\Rightarrow \text{pentru placă:} \quad T_{\text{med}_p} = T_c + \frac{1}{3}(T_s - T_c) \quad [\text{K}] \quad (2.26)$$

$$\Rightarrow \text{pentru cilindru:} \quad T_{\text{med}_c} = T_c + \frac{1}{2}(T_s - T_c) \quad [\text{K}] \quad (2.27)$$

cea ce presupune cunoașterea temperaturilor T_s [K] (temperatura suprafeței) și T_c [K] (temperatura centrului) la momentul de timp pentru care se face calculul.

Cunoscând și temperatura punctului considerat T_x [K] se determină:

$$\Delta T = T_x - T_{\text{med}} \quad [\text{K}] \quad (2.28)$$

De exemplu, pentru suprafață și centru:

$$\Rightarrow \text{placă:} \quad \begin{cases} T_{\text{SP}} = T_{\text{SP}} - T_{\text{med}_p} = \frac{2}{3}(T_s - T_c) \quad [\text{K}] \\ T_{\text{CP}} = T_{\text{CP}} - T_{\text{med}_p} = -\frac{1}{3}(T_s - T_c) \quad [\text{K}] \end{cases} \quad (2.29)$$

$$\Rightarrow \text{cilindru:} \quad \begin{cases} T_{\text{SC}} = T_{\text{SC}} - T_{\text{med}_c} = \frac{1}{2}(T_s - T_c) \quad [\text{K}] \\ T_{\text{CC}} = T_{\text{CC}} - T_{\text{med}_c} = -\frac{1}{2}(T_s - T_c) \quad [\text{K}] \end{cases} \quad (2.30)$$

Semnele diferite pentru tensiunile din suprafață și din centru arată că acestea variază și ca sens, nu numai ca mărime, pe secțiune. Astfel, la începutul încălzirii suprafața se dilată mai mult, dar dilatarea fiind frânată de miezul mai rece, în suprafață apar tensiuni de compresiune, iar în centru de întindere. În straturile intermediare, tensiunile scad ca valoare, existând un anumit punct pentru care $\Delta T = 0$ și tensiunile sunt nule, aici producându-se și schimbarea semnului tensiunilor (figura 2.2.c). Cu trecerea timpului, ΔT crește până la un anumit moment, care va corespunde realizării tensiunilor maxime pe întreaga perioadă de încălzire, după care începe să scadă, tensiunile tinzând spre anulare (figura 2.2.c).

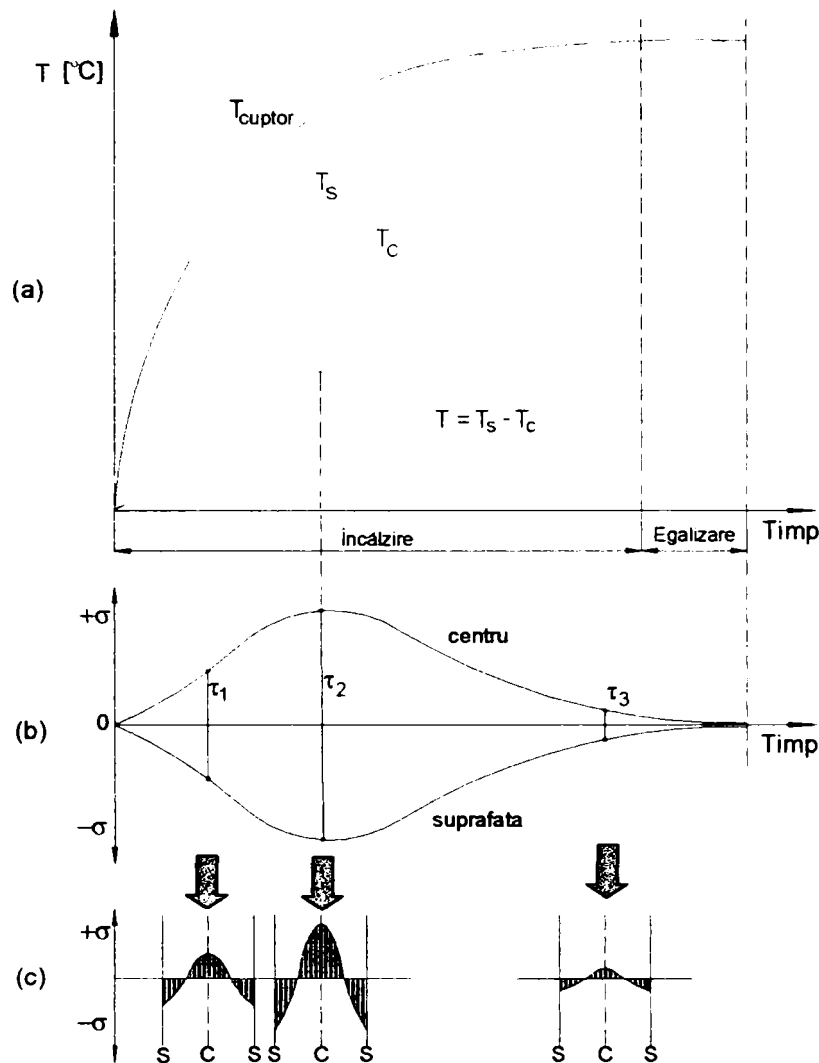


Fig.2.2. Legătura dintre variația temperaturii pe secțiunea unei piese în curs de încălzire (a), variația tensiunilor termice (b) și epura tensiunilor axiale la momentele de timp τ_1 , τ_2 și τ_3 (c)

Din relația (2.25) mai rezultă că tensiunile cresc cu creșterea coeficientului de dilatare β și a modulului de elasticitate E și cu scăderea coeficientului lui Poisson ν . Aceste mărimi trebuie luate în calcul cu valoarea corespunzătoare la temperatura punctului considerat. Pentru oțeluri, până la temperatura de circa $500\text{ }^\circ\text{C}$, produsul $\beta \cdot E \approx 0,25$, iar $\nu \approx 0,3$, ceea ce permite utilizarea relațiilor simplificate:

$$\Rightarrow \text{pe suprafața plăcii:} \quad \sigma_{\max} \approx 0,25 \cdot (T_s - T_c) \text{ [N/mm}^2\text{]} \quad (2.31)$$

$$\Rightarrow \text{în centrul și pe suprafața cilindrului:} \quad \sigma_{\max} \approx 0,18 \cdot (T_s - T_c) \text{ [N/mm}^2\text{]} \quad (2.32)$$

Efectele tensiunilor termice depind de starea inițială a materialului și de raportul dintre mărimea tensiunilor calculate cu relațiile de mai înainte și proprietățile de rezistență ale materialului.

Pentru materialele metalice tenace (metale pure, soluții solide omogene, materiale cu structuri bifazice cu aspect uniform globular), capabile să suporte o deformare plastică avansată înainte de rupere, efectul este de deformare, dacă se îndeplinește condiția:

$$\sigma_r' \geq \sigma_{\max} \geq \sigma_c' \quad (2.33)$$

sau de fisurare, dacă tensiunile de întindere depășesc rezistența la rupere.

Pentru materialele dure și fragile (aliaje turnate, fonte, oțeluri călite), care nu au o limită de curgere sesizabilă și se rup fără, sau cu o neînsemnată deformare plastică anterioară, efectul este de fisurare sub acțiunea tensiunilor de întindere, la îndeplinirea condiției:

$$\sigma_{\max} \geq \sigma'_p \quad (2.34)$$

În ambele cazuri trebuie să se ia în calcul valorile σ'_c , σ'_r și σ'_p la temperaturile pentru care se face calculul tensiunilor termice. Semnificația acestor mărimi este următoarea: σ_c (sau R_e) [N/mm^2] reprezintă limita de curgere aparentă, σ_r (sau R_m) [N/mm^2] este rezistența la rupere și σ_p [N/mm^2] este limita de proporționalitate. Indicele „prim” se referă la faptul că mărimile respective se consideră la temperaturi ridicate.

Pentru înțelegerea corectă a problemei, trebuie reținut faptul foarte important că, însăși valoarea tensiunilor termice este mai mare la materialele fragile (E mai mare, ν mai mic) decât la cele tenace, în aceleași condiții de încălzire.

Dacă se ține seama de neuniformitățile structurale și de prezența anumitor concentratori de tensiuni în piesă (incluziuni nemetalice, constituenți duri «colțuroși», grafitul lamelar în fontele cenușii, treceri bruște de secțiune, canale, găuri), condițiile (2.33) și (2.34) trebuie completate cu luarea în considerare a unui coeficient de siguranță, astfel încât:

$$\sigma_{\max} \leq 0,7 \cdot \sigma'_c \quad (2.35)$$

$$\sigma_{\max} \leq 0,7 \cdot \sigma'_p \quad (2.36)$$

După cum se observă, tensiunile periculoase sunt cele de întindere, capabile să producă fisurarea pieselor. Pericolul crește atunci când piesele sunt deja «încărcate» cu tensiuni remanente de întindere produse de prelucrările tehnologice anterioare (turnare, forjare, sudare, ecrusare). În acest caz, tensiunile termice se însumează cu cele preexistente.

Pentru a se evita efectele nedorite ale tensiunilor termice este necesar să se reducă mărimea acestora, lucru posibil dacă se micșorează neuniformitatea încălzirii, respectiv ΔT în relația (2.25). În acest scop se recurge la reducerea temperaturii cuptorului, efectuând încălzirea în 2-3 trepte succesive cu 1-2 preîncălziri și o încălzire finală, sau se face încălzirea cu viteză constantă și mai mică decât aceea care rezultă din încălzirea în cuptorul cu temperatură constantă ridicată [12].

Capitolul 3

Metode numerice în tratarea problemelor de transfer termic

- 3.1. Metode pentru rezolvarea ecuațiilor diferențiale caracteristice proceselor termogazodinamice
- 3.2. Metoda diferențelor finite
 - 3.2.1. Aproximarea prin diferențe ascendente
 - 3.2.2. Aproximarea prin diferențe descendente
 - 3.2.3. Aproximarea prin diferențe centrate
- 3.3. Metoda elementelor finite
 - 3.3.1. Tipuri de probleme rezolvabile prin metoda elementelor finite
 - 3.3.2. Ipoteze de lucru
 - 3.3.3. Stabilitatea domeniului de analiză
 - 3.3.4. Discretizarea domeniului de analiză
 - 3.3.5. Obținerea modelului numeric cu elemente finite

Capitolul 3

Metode numerice în tratarea problemelor de transfer termic

3.1. Metode pentru rezolvarea ecuațiilor diferențiale caracteristice proceselor termogazodinamice

Cele mai multe procese fizice și, în particular, procesele termogazodinamice sunt descrise de ecuații diferențiale cu derivate parțiale de ordin superior, neliniare. După alcătuirea modelului matematic care descrie un anumit proces se pune problema modului în care modelul respectiv poate fi rezolvat. Există două tipuri de metode de rezolvare:

- ⇒ analitice;
- ⇒ numerice.

Prin metode *analitice* se înțeleg acele metode care exprimă soluția prin formule algebrice, serii de puteri, funcții transcendente și altele. Aceste soluții sunt obținute prin operații matematice complexe cum ar fi: separarea variabilelor, transformări de coordonate, superpoziții de soluții, dezvoltări în serie etc. Soluțiile analitice se exprimă de obicei sub formă generală, particularizarea lor făcându-se prin introducerea condițiilor spațiale (pe frontieră) și a celor temporale (inițiale).

Soluțiile analitice obținute pentru cele mai multe procese termogazodinamice presupun însă o limitare destul de severă și anume liniaritatea ecuațiilor diferențiale; de exemplu proprietățile de material sunt presupuse independente de temperatură. De asemenea aceste soluții impun de obicei o geometrie regulată.

Spre deosebire de soluțiile analitice, soluțiile *numerice* sunt obținute prin calcule aritmetice, sub forma valorilor numerice, pentru fiecare aplicație în parte [13].

Datorită faptului că ecuațiile diferențiale care descriu procese termogazodinamice sunt neliniare, se preferă soluțiile numerice (lipsite de restricția de liniaritate) chiar în cazul în care ar putea fi utilizate tehnici analitice.

S-au dezvoltat extrem de multe metode numerice pentru rezolvarea ecuațiilor diferențiale. Esența acestor metode constă în obținerea unui sistem de ecuații algebrice, de obicei liniare, care aproximează soluția ecuațiilor diferențiale inițiale. Necunoscutele sistemului sunt valorile funcției căutate în nodurile unei rețele de discretizare, care aproximează domeniul de integrare.

În funcție de complexitatea geometriei și a condițiilor de unicitate, numărul ecuațiilor rezultate poate fi foarte mare astfel că rezolvarea sistemului nu se poate face decât cu ajutorul calculatorului electronic, recurgându-se la o tehnică numerică suplimentară, pentru rezolvarea sistemului de ecuații algebrice înlocuitor.

Observațiile de mai sus conduc la concluzia că soluția care se poate obține este aproximativă fiind afectată de erorile de reprezentare internă (în calculator) a numerelor și de erorile de metodă (atât cea de integrare a ecuațiilor diferențiale cât și cea de rezolvare a sistemului rezultat). Cu toate acestea, rezultatele pot fi mai apropiate de realitate decât soluțiile analitice obținute pe seama liniarizării ecuațiilor diferențiale.

Sistemul de ecuații diferențiale poate fi rezolvat în principiu pe două căi:

a) Prin obținerea unei ecuații cu diferențe finite. Aceasta se poate face prin:

- utilizarea unor dezvoltări în serie Taylor;
- integrarea pe un volum finit de control;
- aplicarea unui principiu variațional.

b) Metoda elementelor finite care se bazează pe un model integral al fenomenului studiat.

La alegerea unei anumite metode numerice trebuie să se țină cont de:

- sistemul de ecuații diferențiale care descriu fenomenul analizat și de tipul acestor ecuații;
- geometria corpului și sistemul de coordonate. De obicei este folosit sistemul de coordonate cartezian. Simetria problemei analizate poate însă dicta utilizarea unui sistem cilindric, sferic sau chiar polar;
- metoda de integrare. De cele mai multe ori specificul problemei analizate restrânge substanțial posibilitățile de integrare. Metodele numerice de integrare prezintă instabilități legate de pasul rețelei de discretizare, de condițiile de limită și așa mai departe, care trebuie cunoscute atunci când se alege o anumită metodă numerică pentru rezolvarea unei probleme concrete [14].

3.2. Metoda diferențelor finite

Prima problemă care se pune în legătură cu găsirea soluțiilor ecuațiilor care descriu fenomenele de transfer de căldură este aceea de a le pune sub o formă care să permită rezolvarea numerică, adică o formă care să poată fi manipulată de un calculator electronic. Pentru aceasta, domeniul în care se analizează desfășurarea fenomenului nu se consideră ca un mediu continuu, ci ca o mulțime finită de puncte, formând nodurile unei rețele.

Procedeul de a transforma ecuațiile cu derivate parțiale într-o formă cu diferențe finite constă în dezvoltarea lor în serie Taylor, în jurul punctelor care formează nodurile rețelei, trunchiată în funcție de ordinul derivatei care trebuie exprimată [14].

Cu alte cuvinte, sistemul de ecuații diferențiale cu derivate parțiale, valabil pentru orice punct al domeniului de analiză, se transformă într-un sistem de ecuații algebrice, valabile numai pentru anumite puncte ale domeniului, puncte ce definesc rețeaua de discretizare a modelului [15].

În figura 3.1. este prezentată o rețea care acoperă un domeniu tridimensional. Liniile acestei rețele se numerotează în ordine crescătoare, în cele trei direcții, poziția fiecărui punct în rețea fiind precizată de o anumită combinație (i, j, k) reprezentând numerele liniilor la intersecția cărora se găsește nodul respectiv. Pasul rețelei într-o anumită direcție poate fi constant sau variabil. De asemenea, în raport cu cele trei direcții, rețeaua poate avea același pas sau pași diferiți.

În general, pentru simplitate, se consideră pași egali ai rețelei în toate direcțiile. Acolo unde apar gradienti mari ai câmpurilor de parametri (gradienti mari de temperatură) rețeaua trebuie să aibă pasul mult mai fin decât în restul domeniului. Cu cât pasul unei rețele este mai mic precizia calculului crește dar, în același timp, se mărește și necesarul de memorie, motiv pentru care trebuie căutat un optim între cheltuielile legate de utilizarea calculatorului și precizia obținută.

În rezolvarea problemelor de transfer de căldură prin conducție se preferă rețelele cu pas constant și egal în toate direcțiile. Alegerea rețelei este foarte importantă pentru a se realiza o discretizare care să conducă la o precizie acceptabilă. În afara recomandărilor generale menționate, nu pot fi date, însă, "rețete", alegerea convenabilă a rețelei depinzând de experiența celui care utilizează această metodă.

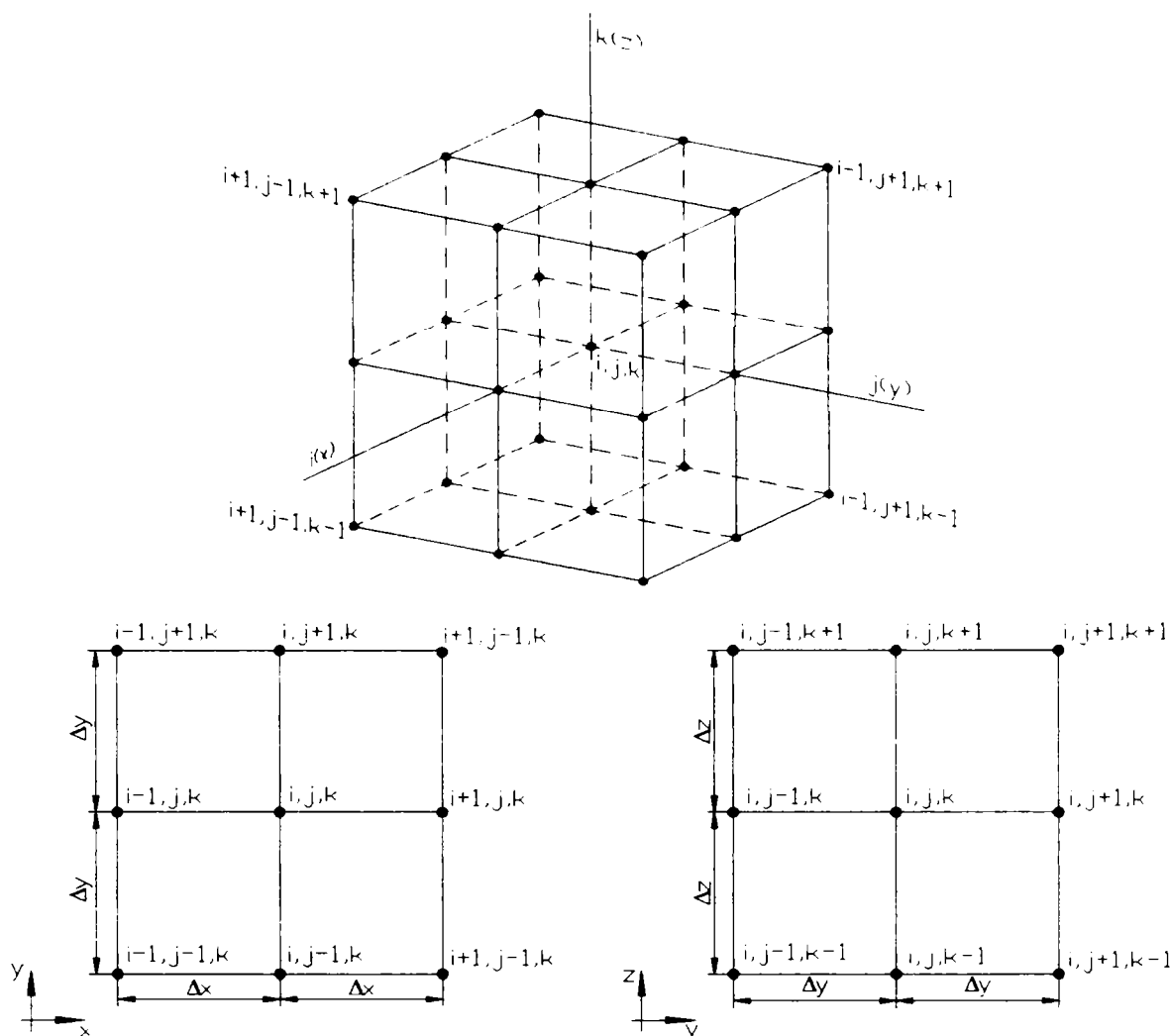


Fig.3.1. Rețeaua tridimensională

O funcție continuă și derivabilă, $\varphi(x,y,z)$ se poate dezvolta în serie Taylor în vecinătatea unui punct, pe una din direcțiile x,y , sau z . Considerând direcția x , formulele dezvoltării în serie sunt:

$$\varphi(x + \Delta x, y, z) = \varphi(x, y, z) + \frac{\partial \varphi(x, y, z)}{\partial x} \Delta x + \frac{1}{2!} \frac{\partial^2 \varphi(x, y, z)}{\partial x^2} \Delta x^2 + \frac{1}{3!} \frac{\partial^3 \varphi(x, y, z)}{\partial x^3} \Delta x^3 + \dots \quad (3.1)$$

$$\varphi(x - \Delta x, y, z) = \varphi(x, y, z) - \frac{\partial \varphi(x, y, z)}{\partial x} \Delta x + \frac{1}{2!} \frac{\partial^2 \varphi(x, y, z)}{\partial x^2} \Delta x^2 - \frac{1}{3!} \frac{\partial^3 \varphi(x, y, z)}{\partial x^3} \Delta x^3 + \dots \quad (3.2)$$

după cum dezvoltarea se face ascendent (în sensul axei) - (3.1) sau descendent (în sens invers axei) - (3.2).

În mod similar se pot scrie expresiile dezvoltării în serie și după celelalte direcții. Prin trunchierea corespunzătoare a acestor serii, derivatele se pot exprima în raport cu valorile funcției considerate în punctele (x,y,z) și $(x \pm \Delta x, y \pm \Delta y, z \pm \Delta z)$.

3.2.1. Aproximarea prin diferențe ascendente

Dacă din ecuația (3.1) se înlătură termenii începând de la ordinul doi, se obține în jurul nodului (i,j,k) din rețea, în direcția $i(x)$, expresia:

$$\Phi_{i+1,j,k} = \Phi_{i,j,k} + \left(\frac{\partial \Phi}{\partial x} \right)_{i,j,k} (x_{i+1} - x_i) = \Phi_{i,j,k} + \left(\frac{\partial \Phi}{\partial x} \right)_{i,j,k} \Delta x \quad (3.3)$$

de unde, derivata $\frac{\partial \Phi}{\partial x}$ se explicitează sub forma:

$$\left(\frac{\partial \Phi}{\partial x} \right)_{i,j,k} = \frac{\Phi_{i+1,j,k} - \Phi_{i,j,k}}{x_{i+1} - x_i} = \frac{\Phi_{i+1,j,k} - \Phi_{i,j,k}}{\Delta x} \quad (3.4)$$

În mod similar, în raport cu celelalte direcții, derivatele de ordinul întâi se exprimă astfel:

$$\left(\frac{\partial \Phi}{\partial y} \right)_{i,j,k} = \frac{\Phi_{i,j+1,k} - \Phi_{i,j,k}}{\Delta y} ; \quad \left(\frac{\partial \Phi}{\partial z} \right)_{i,j,k} = \frac{\Phi_{i,j,k+1} - \Phi_{i,j,k}}{\Delta z} \quad (3.5)$$

Erorile de trunchiere sunt de ordinul întâi și au ordinul de mărime Δx , Δy , respectiv Δz .

3.2.2. Aproximarea prin diferențe descendente

Dacă din ecuația (3.2) se înlătură termenii începând de la ordinul doi, se obține în jurul nodului (i,j,k) din rețea, în direcția $i(x)$, expresia:

$$\Phi_{i-1,j,k} = \Phi_{i,j,k} - \left(\frac{\partial \Phi}{\partial x} \right)_{i,j,k} (x_i - x_{i-1}) = \Phi_{i,j,k} - \left(\frac{\partial \Phi}{\partial x} \right)_{i,j,k} \Delta x \quad (3.6)$$

de unde derivata $\frac{\partial \Phi}{\partial x}$ se explicitează sub forma:

$$\left(\frac{\partial \Phi}{\partial x} \right)_{i,j,k} = \frac{\Phi_{i,j,k} - \Phi_{i-1,j,k}}{x_i - x_{i-1}} = \frac{\Phi_{i,j,k} - \Phi_{i-1,j,k}}{\Delta x} \quad (3.7)$$

În mod similar, în raport cu celelalte direcții, derivatele de ordinul întâi se exprimă:

$$\left(\frac{\partial \Phi}{\partial y} \right)_{i,j,k} = \frac{\Phi_{i,j,k} - \Phi_{i,j-1,k}}{\Delta y} ; \quad \left(\frac{\partial \Phi}{\partial z} \right)_{i,j,k} = \frac{\Phi_{i,j,k} - \Phi_{i,j,k-1}}{\Delta z} \quad (3.8)$$

Erorile de trunchiere sunt și aici de ordinul de mărime Δx , Δy , respectiv Δz .

3.2.3. Aproximarea prin diferențe centrate

Aproximarea prin diferențe ascendente sau descendente permite explicitarea derivatelor de ordinul întâi în raport cu valorile funcției în nodurile rețelei situate fie de o parte, fie de cealaltă a nodului considerat (i,j,k) . Se pot exprima și derivatele de ordinul doi, dar într-o formă neconvenabilă și nesatisfăcătoare în ceea ce privește precizia.

Aproximarea prin diferențe centrate permite stabilirea expresiilor derivatelor parțiale de ordinul întâi și doi în funcție de valorile situate de o parte și de alta a nodului considerat, cu o precizie superioară aproximării prin diferențe ascendente sau descendente.

Prin scăderea expresiilor (3.1) și (3.2) se obține, pentru direcția x:

$$\varphi(x + \Delta x, y, z) - \varphi(x - \Delta x, y, z) = 2 \left[\frac{\partial \varphi(x, y, z)}{\partial x} \Delta x + \frac{1}{3!} \frac{\partial^3 \varphi(x, y, z)}{\partial x^3} \Delta x^3 + \dots \right] \quad (3.9)$$

Înlăturând termenii de ordinul trei în sus, se obține în jurul nodului (i,j,k) din rețea expresia:

$$\varphi_{i+1,j,k} - \varphi_{i-1,j,k} = 2 \Delta x \left(\frac{\partial \varphi}{\partial x} \right)_{i,j,k} \quad (3.10)$$

de unde derivata $\frac{\partial \varphi}{\partial x}$ se explicită sub forma:

$$\left(\frac{\partial \varphi}{\partial x} \right)_{i,j,k} = \frac{\varphi_{i+1,j,k} - \varphi_{i-1,j,k}}{2 \Delta x} \quad (3.11)$$

În mod similar, în raport cu celelalte direcții, derivatele de ordinul întâi se exprimă:

$$\left(\frac{\partial \varphi}{\partial y} \right)_{i,j,k} = \frac{\varphi_{i,j+1,k} - \varphi_{i,j-1,k}}{2 \Delta y} ; \quad \left(\frac{\partial \varphi}{\partial z} \right)_{i,j,k} = \frac{\varphi_{i,j,k+1} - \varphi_{i,j,k-1}}{2 \Delta z} \quad (3.12)$$

Prin adunarea expresiilor (3.1) și (3.2) se obține, pentru direcția x:

$$\begin{aligned} & \varphi(x + \Delta x, y, z) + \varphi(x - \Delta x, y, z) = \\ & = 2 \left[\varphi(x, y, z) + \frac{1}{2!} \frac{\partial^2 \varphi(x, y, z)}{\partial x^2} \Delta x^2 + \frac{1}{4!} \frac{\partial^4 \varphi(x, y, z)}{\partial x^4} \Delta x^4 + \dots \right] \end{aligned} \quad (3.13)$$

Înlăturând termenii de la ordinul patru în sus, se obține:

$$\varphi_{i+1,j,k} + \varphi_{i-1,j,k} = 2 \varphi_{i,j,k} + \left(\frac{\partial^2 \varphi}{\partial x^2} \right)_{i,j,k} \Delta x^2 \quad (3.14)$$

de unde derivata $\frac{\partial^2 \varphi}{\partial x^2}$ se explicită sub forma:

$$\left(\frac{\partial^2 \varphi}{\partial x^2} \right)_{i,j,k} = \frac{\varphi_{i+1,j,k} - 2 \varphi_{i,j,k} + \varphi_{i-1,j,k}}{\Delta x^2} \quad (3.15)$$

În mod similar pentru celelalte direcții:

$$\left(\frac{\partial^2 \varphi}{\partial y^2}\right)_{i,j,k} = \frac{\varphi_{i,j+1,k} - 2\varphi_{i,j,k} + \varphi_{i,j-1,k}}{\Delta y^2} \quad (3.16)$$

$$\left(\frac{\partial^2 \varphi}{\partial z^2}\right)_{i,j,k} = \frac{\varphi_{i,j,k+1} - 2\varphi_{i,j,k} + \varphi_{i,j,k-1}}{\Delta z^2}$$

Dezvoltând funcția φ în serie Taylor, simultan după două direcții, în intervalele $x \pm \Delta x$ și $y \pm \Delta y$, $y \pm \Delta y$ și $z \pm \Delta z$, $x \pm \Delta x$ și $z \pm \Delta z$ se pot stabili expresiile cu diferențe finite și pentru derivatele mixte de ordinul doi $\frac{\partial^2 \varphi}{\partial x \partial y}$, $\frac{\partial^2 \varphi}{\partial y \partial z}$ și $\frac{\partial^2 \varphi}{\partial x \partial z}$:

$$\begin{aligned} \left(\frac{\partial^2 \varphi}{\partial x \partial y}\right)_{i,j,k} &= \frac{\varphi_{i+1,j+1,k} - \varphi_{i+1,j-1,k} + \varphi_{i-1,j-1,k} - \varphi_{i-1,j+1,k}}{4 \Delta x \Delta y} \\ \left(\frac{\partial^2 \varphi}{\partial y \partial z}\right)_{i,j,k} &= \frac{\varphi_{i,j+1,k+1} - \varphi_{i,j+1,k-1} + \varphi_{i,j-1,k-1} - \varphi_{i,j-1,k+1}}{4 \Delta y \Delta z} \\ \left(\frac{\partial^2 \varphi}{\partial x \partial z}\right)_{i,j,k} &= \frac{\varphi_{i+1,j,k+1} - \varphi_{i+1,j,k} + \varphi_{i-1,j,k-1} - \varphi_{i-1,j,k+1}}{4 \Delta x \Delta z} \end{aligned} \quad (3.17)$$

Aproximarea prin diferențe centrate constituie modul uzual de discretizare, deoarece permite exprimarea tuturor derivatelor parțiale de ordinul întâi și doi, în raport cu valorile funcției din nodurile situate în jurul celui considerat (i,j,k) , cu o eroare de discretizare mai mică decât $(\Delta x)^2$.

Dezavantajul principal al metodei cu *diferențe finite* îl constituie utilizarea unei *rețele rectangulare* de discretizare a domeniului de analiză. Deci folosirea ei pe domenii cu contururi sau suprafețe curbe introduce o serie de dificultăți și de artificii de calcul. Totodată apar numeroase probleme de stabilitate și de convergență a soluțiilor, fapt ce impune determinarea condițiilor specifice - pentru fiecare clasă de probleme - de apariție și, respectiv, de evitare a lor [15].

3.3. Metoda elementelor finite

În metoda *elementelor finite* ca punct de plecare se utilizează un *model integral* al fenomenului de studiat. Acest model poate fi obținut în mod direct prin calcul, sau poate fi derivat din modelul diferențial corespunzător cu ajutorul calculului variațional sau al metodei reziduurilor ponderate. Spre deosebire de metoda diferențelor finite, această metodă se bazează pe aproximarea *locală pe porțiuni* sau subdomenii a variabilelor de câmp. Datorită folosirii unui model integral, ca bază de plecare, și a unor seturi de funcții continue pe porțiuni, metoda elementelor finite nu mai este condiționată de existența unei rețele rectangulare. Cu ajutorul ei se pot discretiza practic corpuri geometrice oarecare (cu frontieră nu neapărat rectilinie în R^2 sau plană în R^3).

Datorită performanțelor sale ridicate, metoda elementelor finite a devenit aproape o metodă standard de analiză și proiectare în ingineria construcțiilor, ingineria aerospațială și cea nucleară. Mai mult decât atât, prin posibilitatea de a determina distribuții spațiale ale parametrilor studiați în diferitele corpuri, precum și variația acestora în timp, modelarea numerică cu elemente finite contribuie și la formarea unui nou mod de gândire a analizei inginerești, aceasta datorită, în principal, posibilității de *a vedea* în spațiu prin intermediul calculatorului [15].

În general se apreciază că metoda elementelor finite este capabilă să reflecte mai bine comportarea structurilor, pieselor și subansamblurilor decât alte metode numerice cum este, de exemplu, metoda *diferențelor finite*. Principalele *avantaje* ale metodei elementelor finite față de această metodă sunt următoarele:

- Posibilitatea de a modela forme neregulate prin folosirea de elemente finite cu forme și dimensiuni diferite, adecvate configurației geometrice a corpului studiat;
- Posibilitatea de a trata fără nici o dificultate probleme în care proprietățile fizice ale corpului variază, cum este cazul materialelor neomogene, anizotrope, compozite, stratificate, etc.;
- Posibilitatea adaptării dimensiunilor elementelor finite la mărimea gradientului funcției studiate;
- Posibilitatea considerării oricăror condiții la limită pentru problema studiată;
- Posibilitatea elaborării unor algoritmi și programe cu grad mare de generalitate, apte să rezolve o gamă largă de probleme dintr-un anumit domeniu de specialitate sau chiar din mai multe domenii;

Metoda elementelor finite are și unele *dezavantaje*, din care mai importante sunt următoarele:

- Datele de intrare sunt, în general, numeroase, astfel că este necesar un efort relativ mare pentru pregătirea și introducerea lor. Pentru ușurarea muncii de introducere a datelor de intrare au fost elaborate programe preprocesoare;
- Rezultatele se obțin sub forma unei ample colecții de valori numerice ale funcției sau funcțiilor studiate, într-un număr relativ mare de noduri, încât analiza comportării structurii sau corpului impune studiul atent al acestei mulțimi de numere. Pentru facilitarea analizării rezultatelor au fost elaborate programe postprocesoare, care, în general, exploatează facilitățile grafice ale sistemelor moderne de calcul, furnizând rezultatele nu numai numeric ci și grafic, sub formă de curbe, diagrame, etc.;
- Calitatea rezultatelor depinde de experiența și abilitatea analistului de a elabora un model cu elemente finite pentru problema studiată [16].

3.3.1. Tipuri de probleme rezolvabile prin metoda elementelor finite

În inginerie, aplicațiile metodei elementelor finite pot fi grupate în trei clase de probleme:

- *Probleme de echilibru sau de regim staționar*, în care funcția sau funcțiile necunoscute nu depind de timp. Aici se încadrează, de exemplu, studiul comportării elastice a corpurilor în regim static, problemele de transfer de căldură în regim staționar, studiul regimurilor staționare de curgere a fluidelor, etc.;
- *Probleme de valori proprii*, în care parametri sunt, de asemenea, independenți de timp și în care se determină anumite valori critice ale parametrilor în condițiile respectării configurației de echilibru;
- *Probleme de propagare* sau de regim tranzitoriu, în care funcțiile necunoscute sunt dependente de timp. Un exemplu de astfel de probleme este transferul căldurii în regim tranzitoriu [14].

Considerând cazul particular al problemelor de transfer de căldură și luând în considerare un sistem termic format dintr-unul sau mai multe corpuri solide, care se află în interacțiune energetică cu mediul ambiant, putem sintetiza următoarele tipuri de probleme care apar:

- *Probleme de stare*. Se cunosc date privind structura și proprietățile fizice ale corpurilor analizate, precum și modul de interacțiune al sistemului cu mediul ambiant. Se urmărește determinarea distribuției de temperatură, deci starea sistemului termic considerat;
- *Probleme de interacțiune*. Se cunosc date privind structura, proprietățile fizice ale corpurilor analizate și distribuția de temperatură din sistem. Se urmărește determinarea

interacțiunii energetice a sistemului cu mediul ambiant, respectiv transferul de căldură de la / spre acesta din urmă;

- *Probleme mixte.* Se cunosc parțial date atât despre starea sistemului termic, cât și despre interacțiunea lui cu mediul ambiant. Se urmărește obținerea celorlalte date care să completeze informațiile privind starea sistemului și interacțiunea lui cu mediul ambiant. Problemele reale cad, de obicei, în această categorie. Pentru rezolvarea lor se urmărește mai întâi o reformulare, astfel încât să poată aparține primului sau celui de-al doilea tip de probleme prezentate mai înainte. Dacă nu dispunem de informații suficiente pentru reformularea problemei, acestea se vor presupune pe baza unei experiențe anterioare. Cu ajutorul soluțiilor obținute se vor testa datele presupuse anticipat, la nevoie reluându-se calculul cu noi date[15].

3.3.2. Ipoteze de lucru

□ Ipotezele de lucru se referă în principal la geometria corpului analizat, proprietățile materialelor folosite, domeniul de variație al principalelor mărimi de câmp investigate și, în sfârșit, la regimul de funcționare al sistemului dat.

□ Uneori, corpurile de analizat au geometrii complicate, care sunt greu de descris matematic și de discretizat în elemente finite, mai ales atunci când se fac analize tridimensionale. În aceste cazuri se pune problema introducerii unor simplificări în geometria lor, care să nu afecteze în mod sensibil rezultatele finale ale cercetărilor. Prin introducerea unor simplificări geometrice se obține o idealizare a corpului analizat, care urmează apoi să fie luată ca bază pentru obținerea domeniului de analiză.

□ Ipotezele de material se referă la structura și proprietățile fizice ale acestuia, conducând, în general, la următoarea tipologie: materiale elastice liniare, materiale elastice neliniare, materiale vâscoelastice și materiale plastice. Pentru fiecare dintre aceste tipuri de materiale există legi constitutive specifice, care descriu comportarea lor sub acțiunea sarcinilor externe. Stabilirea ipotezelor de material înseamnă deci precizarea legilor constitutive care vor fi luate în considerație la elaborarea modelului analitic.

□ O ipoteză de material foarte importantă este aceea de a considera corpul analizat ca pe un mediu continuu, omogen local, în sensul că admiterea unor neomogenități de material se poate face numai până la nivelul unui element finit. Deci unele elemente finite pot avea proprietăți fizice diferite de elementele vecine. În cazul materialelor termoizolante, care au de obicei incluziuni de aer, se consideră că domeniul de analiză este constituit dintr-un singur material, omogen, cu proprietăți fizice medii obținute din cele ale elementelor constituente.

3.3.3. Stabilitatea domeniului de analiză

De obicei, în literatura de specialitate se trece peste prezentarea acestei etape, considerându-se domeniul de analiză al problemei respective ca fiind ceva dat, mai ales atunci când se fac prezentări de exemple simple de calcul pe domenii de analiză de tip dreptunghiular, paralelipipedic sau cilindric. În astfel de situații se urmărește numai claritatea expunerii și deci simplitatea modelului prezentat.

În realitate, corpul a cărui comportare o modelăm cu ajutorul metodei elementelor finite și domeniul de analiză al experimentului numeric propriu-zis sunt două lucruri complet distincte; chiar și atunci când ele au aceeași configurație geometrică sunt diferite, datorită proprietăților fizice atașate materialului sau a condițiilor la limită implementate în modelul numeric.

Domeniul de analiză reprezintă deci un model al corpului de studiat, sau numai al unei regiuni semnificative aparținând acesteia. El se obține printr-un proces de idealizare a geometriei corpului original și uneori de selecție a regiunii de analiză. Idealizarea este necesară pentru a reduce din complexitatea configurației geometrice a corpului original, pentru a micșora efortul de calcul al problemei. În același scop se face și selecția unei regiuni de analiză, atunci când anumite proprietăți de simetrie și omogenitate permit acest lucru. De asemenea, selecția apare necesară atunci când se lucrează cu corpuri de dimensiuni foarte mari, sau atunci când o problemă tridimensională se transformă într-una plană sau unidimensională. În cazul în care domeniul de analiză reprezintă numai o parte din corpul original de studiat, o atenție deosebită trebuie acordată adaptării condițiilor la limită, astfel încât să nu schimbăm o dată cu geometria și structura câmpului de variabile analizat.

3.3.4. Discretizarea domeniului de analiză

1) Procesul de discretizare

Conceptul de *discretizare* are la bază ideea de descompunere unei structuri în părțile ei componente. Aceste elemente fizice componente, prin modelare matematică, devin *elemente finite*. Procesul de discretizare al domeniului de analiză are ca suport fizic posibilitatea descompunerii corpului analizat în elementele sale componente.

Spre deosebire de structuri, corpurile solide sau fluide considerate ca medii continue nu mai dispun de elemente componente distincte. O descompunere fizică a lor ar fi deci arbitrară, putându-se realiza în diferite moduri. Aspectul intuitiv al descompunerii corpului în elemente componente dispăre, iar discretizarea devine un proces pur matematic. Realizarea ei se face pe baza unor legi de descompunere astfel ca elemente finite rezultate să constituie subdomenii adiacente care să poată reconstitui integral corpul original de analiză. Într-o primă fază, procesul de discretizare conduce deci la transformarea domeniului de analiză într-un ansamblu de elemente finite. Urmează apoi numerotarea elemente finite, a nodurilor și stabilirea matricelor de conexiuni.

Atunci când corpurile au geometrii complexe, faza de discretizare este precedată de o împărțire a domeniului de analiză pe subdomenii sau subcorpuri. Procedeu este folosit în mod curent la generarea automată a rețelelor de elemente finite, în cadrul sistemelor de propagare destinate cercetării și proiectării asistate de calculator.

2) Factori care condiționează discretizarea

Spre deosebire de alte etape ale metodei cu elemente finite, discretizarea domeniului de analiză nu dispune încă de o teorie generală cu caracter metodologic. Realizarea ei se face pe baza unor considerente practice rezultate din experiența și arta cercetătorilor în acest domeniu. Procesul de discretizare se învață în mod progresiv de fiecare, prin încercări și erori de obținere a unui optim între precizia soluției numerice și efortul de calcul corespunzător. În acest sens, consider utilă o succintă prezentare a principalilor factori care influențează procesul de discretizare.

Tipul elementelor finite. De cele mai multe ori, tipul elementelor finite care urmează a fi folosite derivă din tipul problemei de rezolvat și structura domeniului de analiză a acesteia.

Într-un sens mai larg, elementul finit apare ca un model de aproximare cu proprietăți fizice, geometrice și funcționale. Din punct de vedere geometric, elementul finit reproduce într-o formă idealizată părți dintr-un corp real supus analizei. El are dimensiuni și i se pot atașa proprietăți fizice cum sunt densitatea, vâscozitatea, elasticitatea, conductibilitatea termică, etc., în funcție de scopul urmărit. Din punct de vedere funcțional, elementul finit aproximează una sau mai multe variabile ale problemei în spațiul ocupat de corpul real pe care îl modelează.

Elementele finite *nodale* constituie cele mai larg folosite tipuri de elemente finite. Acestea au proprietatea că pot reprezenta variația unui câmp oarecare utilizând în mod exclusiv:

- o mulțime de valori nodale ale funcției de aproximat sau ale funcției și derivatelor sale până la un anumit ordin;
- o mulțime de funcții asociate fiecărui tip de element finit în parte.

Pentru alte tipuri de elemente finite, cum sunt de exemplu cele hibride și mixte, această structură nodală devine insuficientă. Ea trebuie completată și cu o mulțime de parametri independenți, definiți în interiorul elementului finit, pentru satisfacerea unor condiții suplimentare.

În figura 3.2. sunt prezentate câteva tipuri reprezentative de elemente finite.

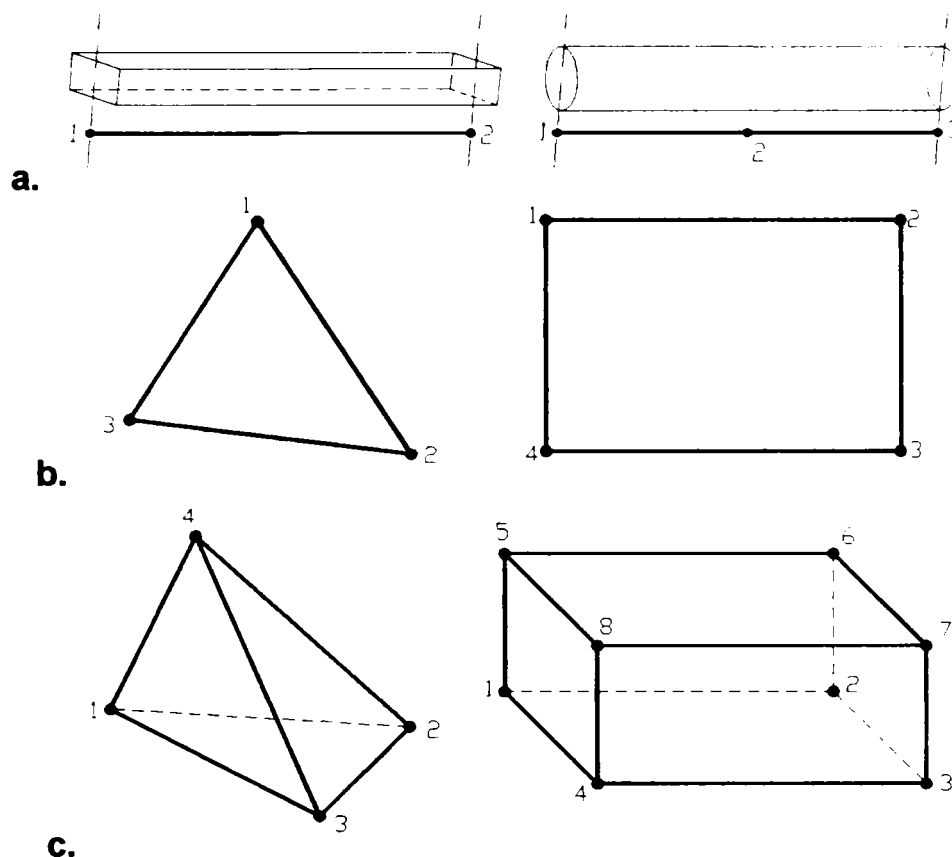


Fig.3.2. Tipuri de elemente finite
a) unidimensionale, b) bidimensionale, c) tridimensionale

Deși se recomandă folosirea aceluiași tip de elemente finite pentru discretizarea unui domeniu de analiză, uneori se pot folosi mai multe tipuri de elemente finite. În acest caz o atenție deosebită trebuie acordată zonei de lipire a două tipuri diferite de elemente finite pentru a asigura compatibilitatea dintre ele și respectiv continuitatea variabilelor de câmp.

Mărimea elementelor finite influențează în mod direct convergența soluției și deci trebuie aleasă cu mult discernământ. Dacă mărimea elementelor este mică, se anticipează o soluție finală mai bună, dar și un timp de calcul mai mare. Uneori, pentru discretizarea domeniului de analiză se folosesc elemente finite de mărimi diferite. Ca regulă generală, elementele mai mici se aleg pentru zonele cu variații mari ale mărimilor de câmp.

Poziționarea nodurilor. Dacă domeniul de analiză nu prezintă discontinuități de geometrie, proprietăți de material sau de sarcini externe, poziționarea nodurilor se face în mod uniform, în strânsă corelație cu tipul și mărimea elementelor finite. Astfel, punctele de discontinuitate vor deveni noduri pentru rețeaua de elemente finite, pentru manipularea mai ușoară a lor. De reținut este și faptul că pe suprafața de frontieră a domeniului de analiză trecerea de la un segment de suprafață

cu anumite condiții la limită la un altul cu condiții la limită diferite se face tot printr-o demarcație nodală.

Numărul de elemente finite este strâns legat de alegerea domeniului de analiză, tipul și mărimea elementelor, precum și de numărul de variabile asociate fiecărui nod. Ca regulă generală, creșterea numărului de elemente conduce la îmbunătățirea soluției problemei. În figura 3.3. se observă că soluția numerică tinde asimptotic spre soluția exactă, fapt pentru care creșterea numărului de elemente peste o anumită limită N_0 nu mai conduce la o îmbunătățire semnificativă a soluției numerice, deși efortul de calcul poate crește foarte mult.

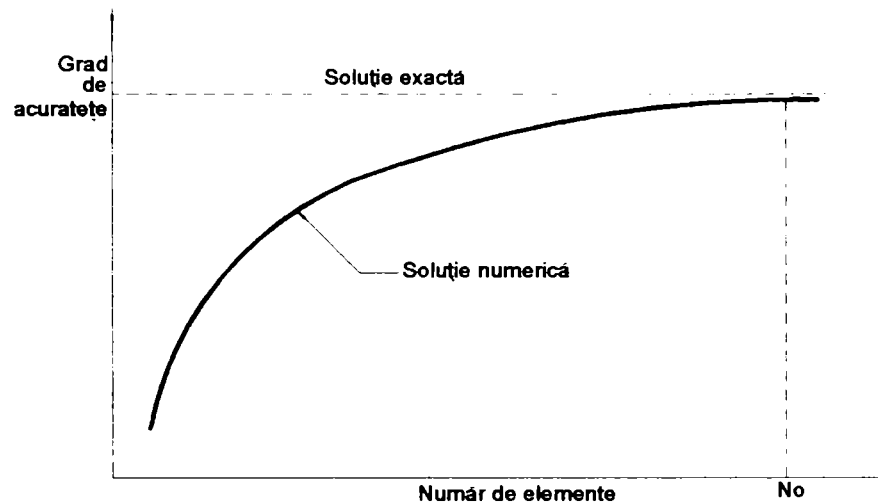


Fig.3.3. Efectul creșterii numărului de elemente

Gradul de uniformitate a rețelei de elemente finite. Unele tipuri de probleme sau unele modele numerice sunt sensibile la gradul de uniformitate al discretizării. Ca regulă generală, în condiții nerestricțive, o discretizare este cu atât mai adecvată cu cât este mai uniformă. O astfel de discretizare este preferată și de către cei care folosesc programe specializate pentru discretizarea domeniilor de analiză. Din punct de vedere practic, ca urmare a necesității micșorării efortului de calcul, se preferă o rețea neuniformă de elemente finite, așa cum am arătat anterior despre mărimea elementelor. Indiferent de mărimea elementelor finite, un grad bun de uniformitate al discretizării se obține atunci, când elementele finite au raportul laturilor cât mai apropiat de unitate. În acest sens, se recomandă evitarea folosirii elementelor finite "alungite" sau a celor puternic distorsionate, pentru a nu se crea direcții preferențiale de variație a mărimilor de câmp.

Stabilirea zonelor de frontieră. Acest aspect este important în cazul analizelor bi și tridimensionale. Zonele de frontieră cu diferitele condiții la limită care apar se stabilesc în faza de formulare a problemei. Deoarece în cadrul problemelor multicâmp aceeași suprafață de frontieră trebuie segmentată în mod diferit pentru diferitele tipuri de încercări care apar (presiuni, tensiuni, temperaturi, etc.), procesul de discretizare trebuie corelat în mod direct cu cerințele impuse de implementarea condițiilor la limită.

3.3.5. Obținerea modelului numeric cu elemente finite

Metoda elementelor finite folosește de cele mai multe ori un principiu variațional. Potrivit calculului variațional, rezolvarea unei ecuații diferențiale într-un anumit domeniu A și în anumite condiții (inițiale și la limită) este echivalentă cu extremizarea în acel domeniu a unei mărimi scalare, dependentă de una sau mai multe funcții necunoscute și care poartă numele de *funcțională*. Prin urmare, problema se reduce la determinarea aproximativă a funcției u , astfel încât funcționala:

$$J = \int_A f\left(\{u\}, \frac{\partial}{\partial x}\{u\}, \dots\right) dA + \int_{\Gamma} g\left(\{u\}, \frac{\partial}{\partial x}\{u\}, \dots\right) d\Gamma \quad (3.18)$$

definită pe domeniul A și pe conturul său Γ să atingă un extrem. Acoladele care încadrează funcția u arată că aceasta poate avea forma unui vector coloană, adică:

$$\{u\} = \begin{Bmatrix} u_1 \\ u_2 \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix} \quad (3.19)$$

Caracteristic metodei este faptul că în scopul extremizării aproximative a funcționalei J , domeniul A se divide în subdomenii, de dimensiuni finite, numite *elemente finite*, contribuția fiecărui element la constituirea, prin însumare, a funcționalei estimându-se pe baza unui model simplificat de reprezentare a funcției necunoscute u . Valoarea aproximativă a funcționalei J este dată deci de relația:

$$J = \sum J^{(e)} \quad (3.20)$$

unde indicele superior (e) semnifică raportarea la un anumit element de discretizare.

Funcția necunoscută u , se exprimă în mod aproximativ pentru fiecare element, printr-o relație de forma:

$$\{u\}^{(e)} = [N]\{U\}^{(e)} \quad (3.21)$$

în care vectorul coloană $\{U\}^{(e)}$ poate să conțină fie valorile funcției u în punctele caracteristice ale elementului, numite *noduri*, fie anumite caracteristici ale elementului.

Matricea $[N]$ se numește *funcție de formă* și depinde de tipul elementului finit utilizat și de coordonatele punctului din interiorul elementului pentru care se calculează (estimează) valoarea funcției u . Funcția de formă are valori nenule numai în limitele elementului respectiv.

Din cele prezentate rezultă că în aplicarea metodei elementelor finite se dispune de posibilitatea *parcelării* domeniului A în diferite moduri, putând să se aleagă numărul și tipul de elemente. Această alegere are consecințe directe asupra dimensiunii sistemului de ecuații care rezultă, a timpului de calcul necesar precum și asupra preciziei soluției.

Prin tipul elementului finit se înțelege atât forma sa geometrică (figura 3.2), cât și modul de definire a funcției necunoscute în limitele elementului. Pentru domeniile bidimensionale se cunosc diferite elemente de formă triunghiulară sau dreptunghiulară, având laturile rectilinii sau curbilinii. Nodurile unui element triunghiular, de exemplu, pot să fie cele trei vârfuri ale triunghiului, dar există și elemente care folosesc și puncte suplimentare (plasate în centrul de greutate, la mijlocul laturilor, etc.). Definirea funcției de formă este de asemenea proprie fiecărui tip de element. Există condiții speciale care trebuiesc îndeplinite de orice funcție de formă pentru a se asigura convergența calculului prin metoda elementelor finite, către soluția exactă a problemei. De exemplu, dacă funcționala care trebuie minimizată în domeniul A conține derivate de ordinul m , convergența nu este asigurată dacă primele $(m-1)$ derivate ale acesteia nu sunt continue la granița dintre două elemente vecine.

Deoarece la nivelul fiecărui element, funcția necunoscută u este definită conform (3.21), printr-un număr finit de parametri (elementele vectorului $\{U\}^{(e)}$), rezultă că funcționala J va depinde de toți parametri $\{U\}$ din domeniul A . Condiția de extremizare a funcționalei (3.18) va avea forma sistemului de ecuații:

$$\frac{\partial J}{\partial \{U\}} = \begin{Bmatrix} \frac{\partial J}{\partial U_1} \\ \frac{\partial J}{\partial U_2} \\ \vdots \\ \vdots \end{Bmatrix} = 0 \quad (3.22)$$

Înlocuind (3.20) în (3.22) se obține regula de asamblare a sistemului global de ecuații, din ecuațiile furnizate de fiecare element în parte:

$$\frac{\partial J}{\partial \{U\}} = \sum \frac{\partial J}{\partial \{U\}^{(e)}} = 0 \quad (3.23)$$

însurarea fiind extinsă la toate elementele domeniului A .

Soluția sistemului (3.23) constă din valorile parametrilor U din toate elementele finite în care a fost împărțit domeniul A , valorile funcție necunoscute u rezultând din (3.21). Modul de rezolvare numerică a sistemului depinde însă de forma concretă a expresiei (3.21) care trebuie introdusă în (3.18) în vederea formării sistemului de ecuații (3.22).

În cazul particular al obținerii unei funcționale J pătratice în raport cu necunoscuta u și cu derivatele acesteia, sistemul (3.22) capătă forma:

$$\frac{\partial J}{\partial \{U\}} = [K]\{U\} + \{F\} = 0 \quad (3.24)$$

în care matricile $[K]$ și $\{F\}$ sunt constante și se obțin prin "asamblarea" omoloagelor acestora corespunzătoare fiecărui element finit.

$$\begin{aligned} [K] &= \sum [k]^{(e)} \\ \{F\} &= \sum \{f\}^{(e)} \end{aligned} \quad (3.25)$$

în care însurarea se face după toate elementele din domeniu.

În aplicațiile practice, metoda elementelor finite conduce la sisteme cu un număr foarte mare de ecuații (mergând până la dimensiuni de ordinul 10 000). Este evident că pentru rezolvarea unor asemenea sisteme este nevoie, pe de o parte de calculatoare electronice cu o capacitate corespunzătoare, iar pe de altă parte de metode numerice speciale.

Deși se pot realiza programe proprii de analiză cu elemente finite, este de dorit ca să se utilizeze programe de firmă eficiente (cum sunt ALGOR, SAP, ANSYS, COSMOS, TERMIC-2N, etc.) care au avantajul utilizării unor:

- tipuri de elemente eficiente (în funcție de specificul problemei analizate);
- metode de programare eficiente și exploatarea cu maximum de eficiență a posibilităților oferite de echipamentele de calcul;
- tehnici numerice adecvate.

partea a doua

- **Studii și cercetări experimentale**

2

Capitolul 4

Studii și cercetări experimentale

- 4.1. Prezentarea instalației experimentale pilot
- 4.2. Modul de achiziție al datelor
- 4.3. Metodologia de experimentare
- 4.4. Concluzii

Capitolul 4

Studii și cercetări experimentale

4.1. Prezentarea instalației experimentale pilot

Realizarea unor condiții de desfășurare a fenomenelor termice complexe care au loc în incintele cuptoarelor cu flacără a condus la ideea conceperii unei instalații experimentale, capabilă să furnizeze toate datele necesare dezvoltării și validării modelelor numerice implementate ulterior (figura 4.1) [17].

Instalația experimentală, sub forma unei incinte asemănătoare spațiului de lucru al unui cuptor tip cameră, permite efectuarea unor măsurători directe de temperatură, în anumite puncte situate în interiorul spațiului util, la suprafața interioară a vetrei și pereților laterali și în special, în diferite puncte din interiorul materialului supus încălzirii (figura 4.2).

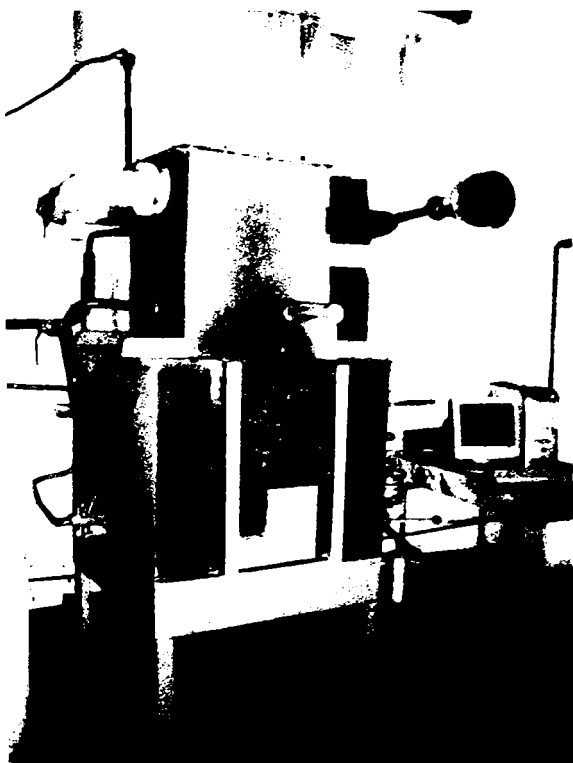


Fig.4.1. Instalația experimentală pilot.
Vedere de ansamblu

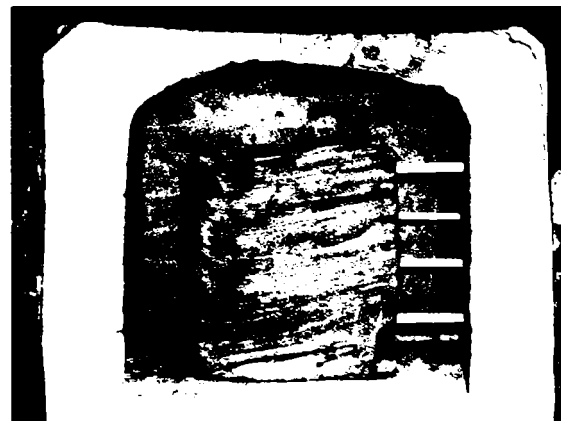


Fig.4.2. Instalația experimentală pilot
Detaliu privind interiorul spațiului de lucru

Efectul termic este obținut prin arderea gazului metan, provenit de la rețeaua locală, într-un arzător cu un debit maxim de $9 \text{ m}^3_{\text{N}}/\text{h}$. Evacuarea gazelor arse se face pe la partea superioară a incintei de lucru. În scopul ridicării temperaturii maxime de încălzire, aerul pentru combustie,

insuflat cu ajutorul unui ventilator, este preîncălzit într-un preîncălzitor de tip recuperativ, cu serpentină, situat deasupra camerei de încălzire. Vehicularea gazelor arse se realizează cu ajutorul unui ejector care insuflă aer în conducta de evacuare a acestora, creând astfel o oarecare depresiune în interiorul incintei de lucru.

Zidăria camerei de încălzire este realizată din materiale ceramice refractare folosite uzual în construcția cuptoarelor industriale. Protecția la exterior a acesteia este asigurată de o manta metalică executată din tablă de oțel.

Blumul supus încălzirii (figura 4.3) reprezintă un semifabricat turnat continuu la S.C. „Siderurgica” S.A. Hunedoara. Materialul este un oțel marca S 235 JRG2 [18] sau OL 37-2k [19].

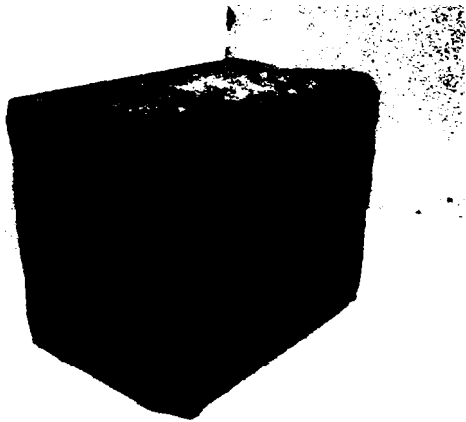


Fig.4.3. Blumul supus încălzirii

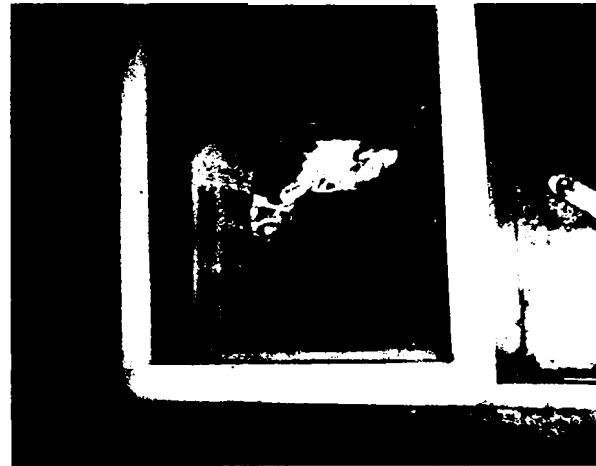


Fig.4.4. Detaliu privind modul de încălzire asimetric

Încălzirea blumului se face intenționat asimetric, arzătorul fiind poziționat pe unul din pereții laterali ai camerei (figura 4.4). Mai mult, ușa camerei de lucru a instalației experimentale a fost închisă parțial, lăsând în partea inferioară a incintei o fantă (figura 4.5) care să simuleze o situație reală din practică, în care apare o infiltrație de aer rece fals care diluează atmosfera gazoasă din interior și care modifică distribuția câmpului termic, scopul efectuării acestor experimentări nefiind realizarea unei încălziri uniforme a blumului, ci studiul fenomenelor termice care au loc și realizarea unor modele care să permită simularea numerică a acestor fenomene, în condiții cât mai concrete.



Fig.4.5. Închiderea parțială a ușii

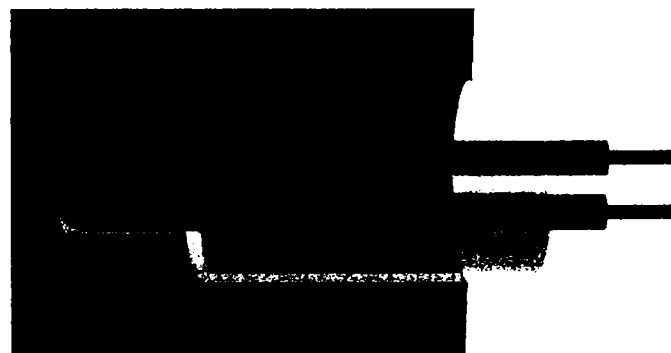


Fig.4.6. Poziționarea termocuplului în punctul de măsurare

Pentru realizarea acestui deziderat au fost achiziționate date referitoare la valorile temperaturilor a opt puncte (cinci în interiorul blumului, unul la suprafața interioară a zidăriei, unul în vatră și unul în interiorul incintei de lucru) considerate ca fiind semnificative în soluționarea problemei propuse.

Măsurătorile de temperatură au fost efectuate cu ajutorul unor termocupluri Cr-Al. Modul de poziționare al acestora în punctele de măsurare este prezentat în figura 4.6.

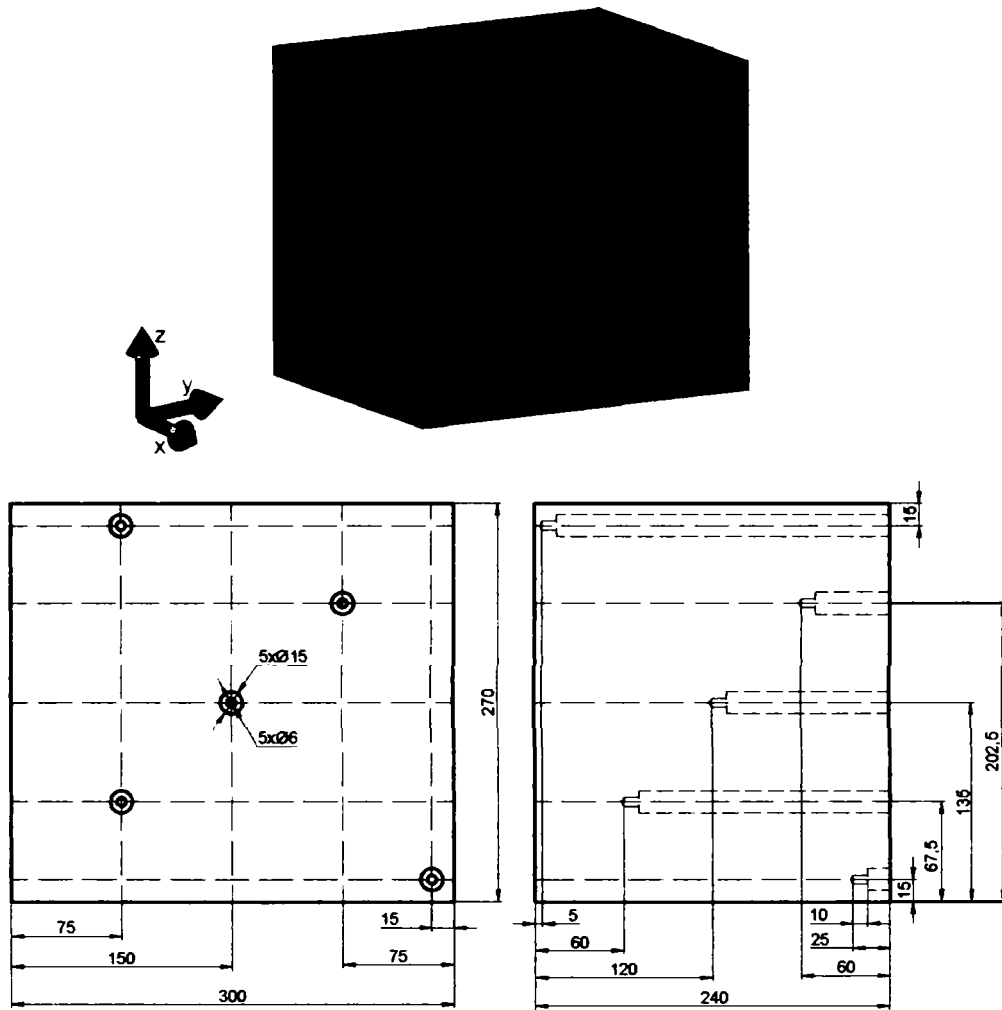


Fig.4.7. Geometria punctelor de măsurare

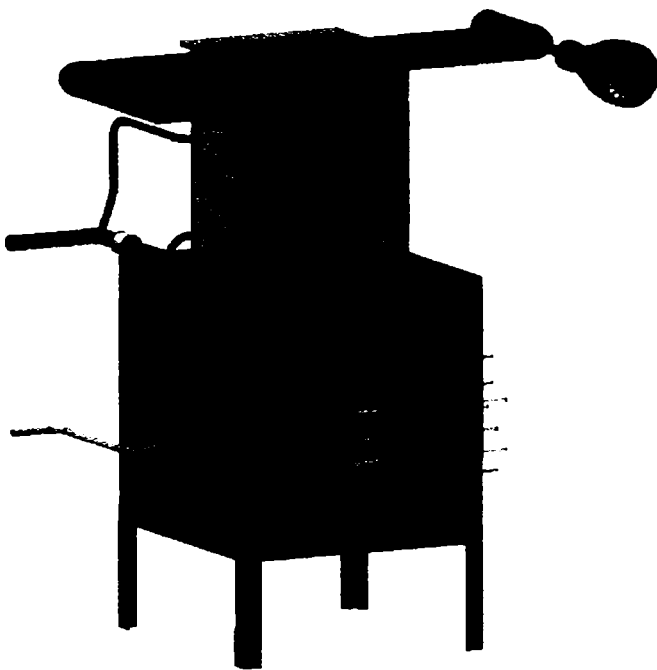


Fig.4.8. Modelul geometric tridimensional al instalației experimentale pilot

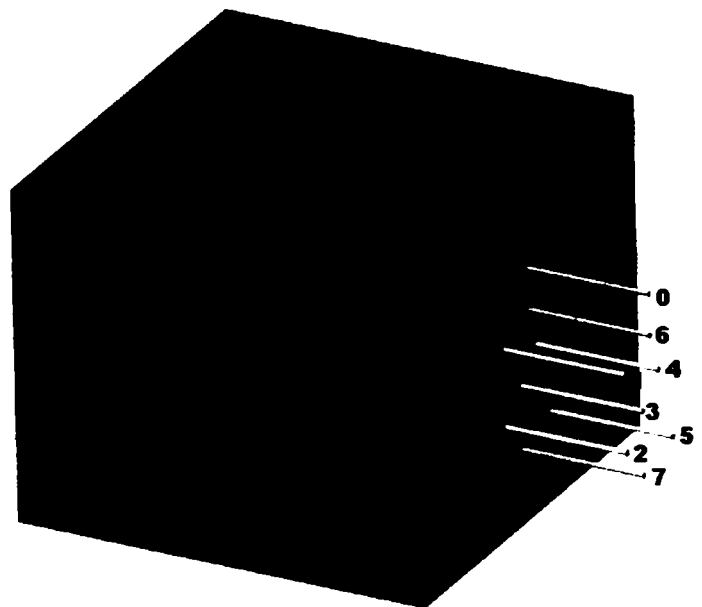


Fig.4.9. Detaliu privind poziționarea termocuplurilor. Model geometric tridimensional

În scopul construirii modelelor de simulare și, nu în ultimul rând, pentru o înțelegere mai bună a modului de poziționare a dispozitivelor de măsurare, au fost realizate modelele geometrice ale semifabricatului și instalației experimentale, cu ajutorul pachetelor software Mechanical Desktop și Autodesk Inventor (figurile 4.7, 4.8 și 4.9).

Dimensiunile geometrice ale blumului, precum și coordonatele punctelor de măsurare (pozițiile orificiilor pentru introducerea termocupurilor) sunt prezentate de asemenea în figura 4.7.

4.2. Modul de achiziție al datelor

După cum am mai spus, măsurarea temperaturilor în punctele caracteristice s-a făcut cu ajutorul unor termocupuri Cr-Al, produse de S.C. „CAOM” S.A. Pașcani, având următoarele caracteristici [20] [21] [22]:

Caracteristicile termocupului Cr-Al

Tabelul 4.1

Denumire	Codul	Temperatura limită de utilizare [°C]			t.t.e.m. maximă [mV]
		minimă	maximă		
			La funcționare continuă	La funcționare intermitentă	
Cromel-Alumel	K	-200	1000	1300	52,598

Determinarea precisă a dependenței tensiune termo-electromotoare (*t.t.e.m.*) – temperatură, pentru necesitățile măsurării temperaturii, s-a făcut empiric prin etalonare. Pentru termocupurile cele mai frecvent utilizate, această dependență este prezentată tabelar [23], considerându-se temperatura punctului rece având valoarea de 0 °C. Pentru temperaturi diferite de aceasta ale sudurii reci, valorile *t.t.e.m.* se corelează cu date din tabele complementare. Erorile tolerate ale termocupurilor Cr-Al sunt indicate în tabelul de mai jos:

Eroarea tolerată a termocupului Cr-Al(joncțiunea de referință 0 °C)

Tabelul 4.2

Temp.[°C]	Eroarea tolerată		Temp.[°C]	Eroarea tolerată	
	[mV]	[°C]		[mV]	[°C]
-200	0,0800	5,00	600	0,2310	5,50
-100	0,1240	4,00	700	0,2620	6,25
0	0,1560	4,00	800	0,2870	7,00
100	0,1680	4,00	900	0,3100	7,75
200	0,1600	4,00	1000	0,3315	8,50
300	0,1680	4,00	1100	0,3515	9,25
400	0,1720	4,00	1200	0,3700	10,00
500	0,2040	4,75	1300	0,3762	10,75

Semnalele de tensiune generate de termocupuri au fost captate de un modul de achiziție analogic pe 8 canale de tipul ADAM- 4018, prezentat în figura 4.10.

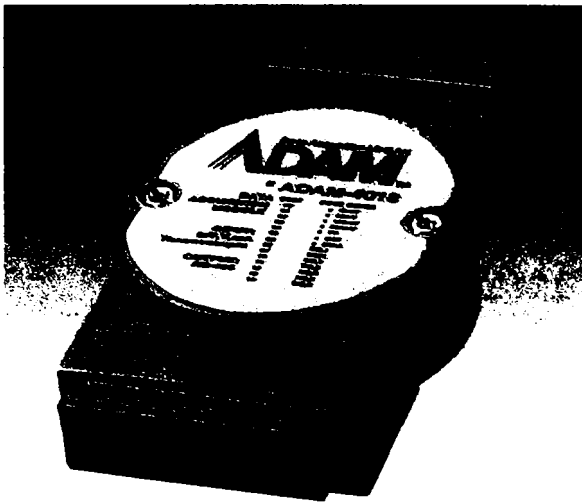


Fig.4.10. Modulul de achiziție ADAM-4018



Fig.4.11. Convertorul ADAM-4520

Caracteristicile tehnice ale acestui modul sunt următoarele:

ADAM-4018 – Caracteristici tehnice

Tabelul 4.3

Rezoluție efectivă	16 – bit
Canale	8 – diferențiale
Tipul intrărilor	Termocupluri, mV, V, mA
Plaja de intrare	$\pm 15 \text{ mV}$, $\pm 50 \text{ mV}$, $\pm 100 \text{ mV}$, $\pm 500 \text{ mV}$, $\pm 1 \text{ V}$, $\pm 2,5 \text{ V}$, $\pm 20 \text{ mA}$
Tipul termocuplurilor și domeniul de temperaturi	J (0 - 760) °C K (0 - 1370) °C T (-100 - 400) °C E (0 - 1000) °C R (500 - 1750) °C S (500 - 1750) °C B (500 - 1800) °C
Supratensiune maximă	$\pm 35 \text{ V}$
Frecvența de achiziție	10 eșantioane / sec. (în total)
Impedanța de intrare	20 M Ω
Banda de frecvență	13,1 Hz @ 50 Hz 15,72 Hz @ 60 Hz
Precizie	$\pm 0,1 \%$ pentru intrări de tensiune
Conține controler de frecvență	
Tensiunea de alimentare	+ 10 ÷ +30 V _{CC}
Putere consumată	0,8 W

Datele prelevate au fost transmise spre un calculator de tip PC prin intermediul unui convertor din portul serial tip RS-485 în RS-232 de tipul ADAM-4520, prezentat în figura 4.11. Cutia de conexiuni reprezentând ansamblul blocului de achiziție format din cele două dispozitive (modulul de achiziție și convertorul de semnal), împreună cu modulul de alimentare la rețeaua electrică, este prezentat în figura 4.12. Schema electrică de conexiune a dispozitivelor de măsurare este prezentată în figura 4.13.

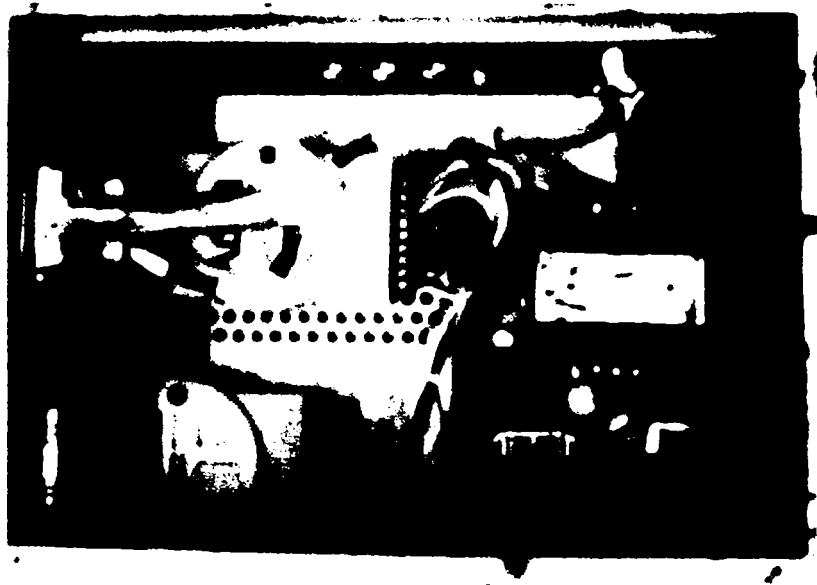
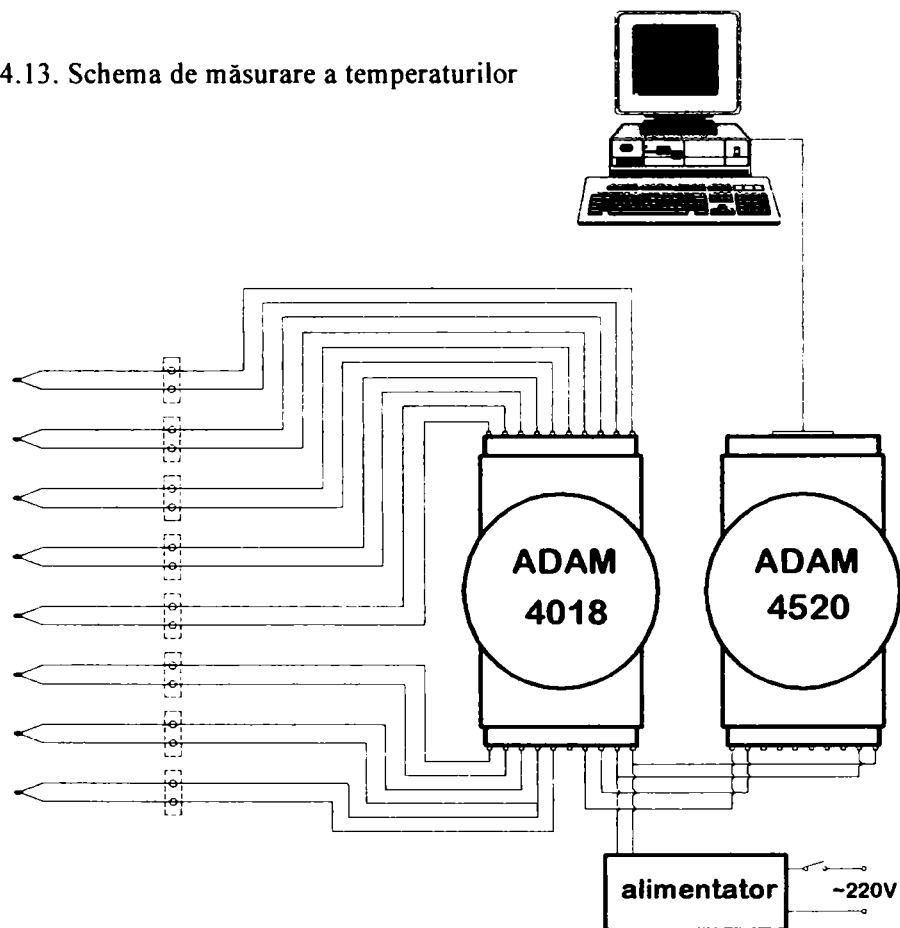


Fig.4.12. Blocul de achiziție a datelor

Fig.4.13. Schema de măsurare a temperaturilor



4.3. Metodologia de experimentare

Stabilirea pe cale experimentală a distribuției câmpului termic în interiorul blumului, de-a lungul perioadei de încălzire, presupune alegerea unor puncte de măsurare a temperaturilor în așa fel încât poziția acestora să furnizeze informații cât mai complete asupra evoluției fenomenului studiat. Astfel, în interiorul blumului au fost practicate cinci orificii (numerotate cu 1,2,3,4 și

respectiv 5) pentru introducerea termocupurilor, la diferite nivele și adâncimi, după cum se poate vedea în figura 4.7. Cifra din dreptul fiecărui orificiu identifică termocupul corespunzător (figura 4.9).

Urmărirea variației temperaturii în camera de încălzire și evoluția câmpului termic stabilit la suprafața interioară a zidăriei impune stabilirea unor puncte suplimentare de măsurare. Astfel, în peretele lateral, opus arzătorului, este practicat un orificiu (numerotat cu 6 în figura 4.9) care permite introducerea unui termocup ce măsoară temperatura în imediata vecinătate a suprafeței interioare a peretelui. De asemenea, un orificiu practicat la nivelul vetrei (numerotat cu 7) permite introducerea unui termocup ce preia temperatura într-un punct situat în centrul bazei de sprijin a blumului și la o adâncime de 5 mm sub nivelul suprafeței interioare a vetrei.

În fine, printr-un orificiu ce străpunge peretele lateral opus arzătorului se introduce un termocup (numerotat cu 0 în figura 4.9) în vederea monitorizării evoluției temperaturii atmosferei gazoase din interiorul incintei de încălzire.

Semnalele de tensiune provenite de la cele opt termocupuri au fost preluate de modulul ADAM-4018 și prin intermediul convertorului de semnal tip ADAM-4520 au fost introduse în calculator. Un software dedicat transformă semnalele primite în valori de temperatură, măsurate în grade Celsius.

Înregistrarea semnalelor provenite de la cele opt canale s-a făcut la intervale de timp de 5 s. Momentul începerii înregistrării temperaturilor coincide cu momentul de început al procesului de încălzire. Cu alte cuvinte, la $\tau = 0$ temperatura camerei de încălzire este identică cu cea a mediului ambiant. Timpul total de încălzire este de 10 h, iar în acest interval debitele de gaz metan și aer necesar combustiei au fost menținute constante, reglajul arderii făcându-se la început procesului. Altfel spus, încălzirea semifabricatului s-a făcut prin intermediul unui flux de căldură aproximativ constant pe parcursul întregii perioade, dar neuniform în spațiul de lucru al instalației.

Pentru a nu introduce erori în procesul de măsurare, „sudura rece” a termocupurilor a fost menținută la o temperatură constantă (19 °C) pe toată perioada achiziționării datelor. Cum această temperatură este diferită de 0 °C, s-au impus o serie de corecții conform tabelului 4.2, făcute automat de software-ul instalat pe calculatorul folosit.

Experimentul s-a repetat de trei ori, respectând cu strictețe aceleași condiții de desfășurare. Datele preluate din cele trei măsurători au fost prelucrate statistic, rezultatul obținut reprezentând media valorilor de temperatură măsurate pentru fiecare punct caracteristic, la intervale egale de 5 s, pe toată durata celor 10 h de încălzire.

4.4. Concluzii

În figura 4.14 sunt prezentate rezultatele măsurătorilor experimentale, cele opt curbe reprezentând variația temperaturilor în punctele de măsurare, pe durata încălzirii. Semnificația acestora este următoarea: curba 0 reprezintă variația temperaturii în timp a atmosferei gazoase din interiorul incintei de lucru a instalației. Curbele numerotate cu cifre de la 1 la 5 reprezintă variații ale temperaturilor punctelor situate în interiorul blumului, conform figurii 4.7. Curba 6 reprezintă variația temperaturii suprafeței interioare a peretelui opus arzătorului, iar curba 7 înregistrează variația temperaturii porțiunii vetrei care servește ca bază de sprijin pentru semifabricat. După cum se poate constata, etichetele curbelor prezentate sunt identice cu numerele de identificare a termocupurilor.

□ Referitor la evoluția în timp a temperaturii atmosferei gazoase din cameră, putem observa mici abateri de la forma normală de variație, abateri puse pe seama căderilor temporare ale presiunii gazului în arzător, motivele fiind independente de instalația experimentală.

□ Urmărind evoluția curbelor 6 și 7 este sesizabil faptul că, la începutul perioadei de încălzire, vatra are tendința de a se încălzi mai repede decât peretele lateral. Pe măsura trecerii timpului, temperatura peretelui devine mai mare decât a vetrei, încălzirea decurgând în continuare cu o diferență oarecum nesemnificativă de cca. 25...35 °C.

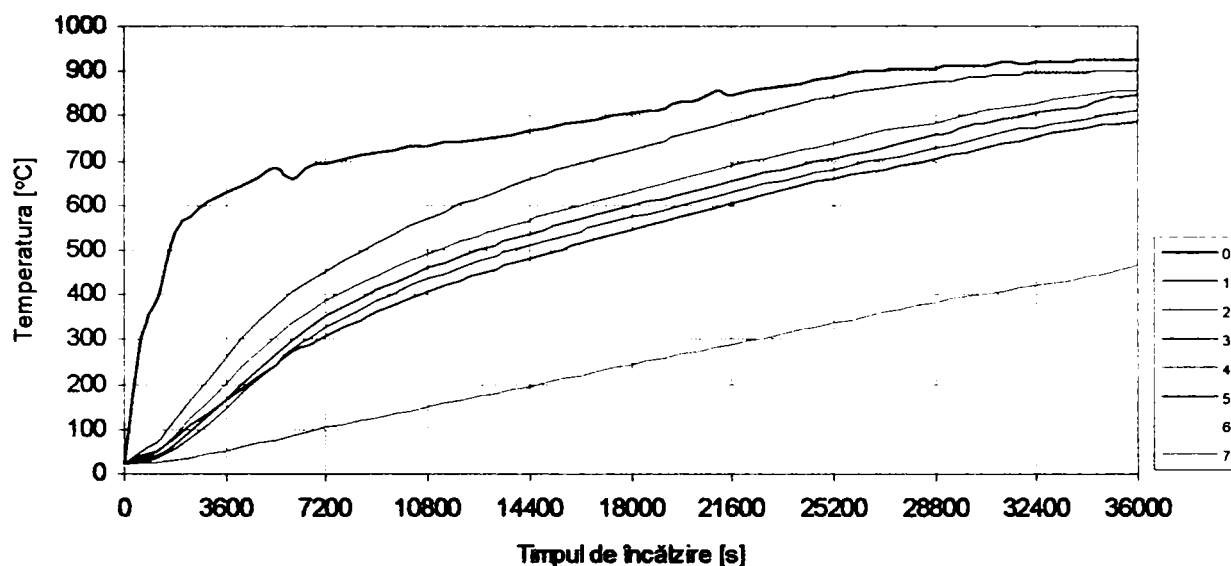


Fig.4.14. Variația în timp a temperaturilor măsurate în punctele caracteristice

□ Temperaturile celor 5 puncte din interiorul blumului cresc continuu și uniform pe întreaga durată a încălzirii. Valorile cele mai ridicate sunt înregistrate de termocuplul nr.1, lucru evident datorită poziției acestuia (în imediata vecinătate a suprafeței verticale exterioare a blumului care este radiată direct de flacără).

□ Închiderea parțială a ușii instalației și realizarea fantei prin care se infiltrează aer rece fals reprezintă motivul pentru care valorile indicate de termocuplul cu nr.2 sunt atât de mici relativ la cele indicate de termocuplul nr.1, având în vedere și poziția relativă a acestora.

□ Cele mai mici valori sunt înregistrate de termocuplul nr.5, explicația constând în poziția acestuia în blum, pe de o parte el situându-se într-o zonă a blumului în care curentul de aer fals își face puternic simțită prezența, iar pe de altă parte datorită faptului că această zonă este diametral opusă celei radiate direct de flacără.

partea a treia

- **Modelul analitic al transferului termic în interiorul blumului**
- **Modelul numeric cu diferențe finite**
- **Modelul numeric cu elemente finite**



Capitolul 5

Modelul analitic al transferului termic în interiorul blumului

- 5.1. Aparatul matematic
- 5.2. Soluția analitică. Rezultate
- 5.3. Concluzii

Capitolul 5

Modelul analitic al transferului termic în interiorul blumului

5.1. Aparatul matematic

După cum am precizat și în capitolul 2, rezolvarea analitică a ecuației diferențiale cu derivate parțiale a conducției termice poate fi o problemă relativ simplă sau extrem de complicată, funcție de situația particulară considerată. Încercând rezolvarea cazului nestaționar, multidimensional, avem la dispoziție următoarele variante:

- găsirea unei soluții analitice, disponibilă deja în literatura de specialitate și care să corespundă cazului studiat;
- rezolvarea analitică efectivă;
- obținerea soluției prin metoda grafică, în cazul în care avem de-a face cu o problemă bidimensională.

În cele ce urmează se propune rezolvarea efectivă a problemei, abordând metoda clasică a seriilor Fourier. Ca rezultat al metodei propuse se obține o soluție analitică aproximativă a ecuației în cauză. În multe cazuri din practica inginerescă, soluția analitică, chiar și aproximativă, este mai apreciată decât soluția numerică a aceleiași probleme[24]. Găsirea soluției pentru ecuația conducției termice este asociată de obicei cu dificultăți majore, în special datorită neliniarității (proprietăți de material dependente de temperatură) și a condițiilor de contur cu privire la fenomenul de radiație. De aceea este inevitabilă luarea în considerare a unor ipoteze care să ducă, într-o oarecare măsură la idealizarea modelului analitic, comparativ cu situația reală.

În acest capitol se dorește obținerea unei soluții analitice a ecuației conducției termice stabilite în interiorul blumului încălzit în instalația experimentală prezentată în capitolul 4. Această situație impune stabilirea următoarelor ipoteze simplificatoare:

- în primul rând, proprietățile termo-fizice ale materialului supus încălzirii și anume: densitatea - ρ [kg/m^3], căldura specifică - c [$\text{J}/(\text{kg}\cdot\text{grd})$] și conductivitatea termică λ [$\text{W}/(\text{m}\cdot\text{grd})$] sunt considerate mărimi constante în raport cu temperatura. Acestea vor fi introduse în calcul ca și valori medii pe întreg intervalul de încălzire de 10 h;
- temperatura incintei de lucru este aceeași la un moment dat în toate punctele acesteia (nu variază în spațiu);
- încălzirea se realizează uniform, variația în timp a temperaturii incintei de lucru a instalației experimentale fiind fidel urmărită de suprafețele exterioare ale semifabricatului, ceea ce presupune o valoare foarte mare (de ordinul sutelor până la mii de [$\text{W}/(\text{m}^2\cdot\text{K})$]) a coeficientului global de transfer termic de la gazele arse la suprafața liberă a metalului;
- semifabricatul este suspendat în atmosfera gazoasă, neexistând nici un fel de contact direct între suprafața de sprijin a acestuia și vatră;
- nu este luat în considerare efectul radiației termice al gazelor și al pereților interiori ai zidăriei instalației;

- se consideră că în interiorul materialului supus încălzirii nu apar surse generatoare de căldură;
- distribuția temperaturii în material este uniformă la începutul încălzirii, toate punctele din interiorul blumului și de pe suprafețele exterioare având la momentul inițial temperatura mediului ambiant, adică 20 °C.

Ținând cont de cele precizate mai sus, realizarea modelului analitic al problemei presupune parcurgerea următorilor pași:

⇒ **pasul 1:** alegerea sistemului de coordonate care se potrivește cel mai bine problemei și identificarea variabilelor independente care determină funcția de temperatură.

Fiind vorba despre un domeniu tridimensional, sub forma unui paralelipiped, cu toate cele trei dimensiuni finite, se va alege un sistem de coordonate cartezian. Temperatura variază după toate cele trei direcții x , y și z . Regimul de încălzire este unul nestaționar și, în consecință, temperatura în interiorul blumului variază în timp. Așadar, funcția de temperatură este de forma: $T = T(x, y, z, \tau)$.

Deoarece încălzirea este uniformă, există o simetrie față de centrul blumului, punct în care se va plasa originea sistemului de coordonate. În această situație este suficientă studierea unui sfert din domeniul de analiză, conform figurii 5.1.

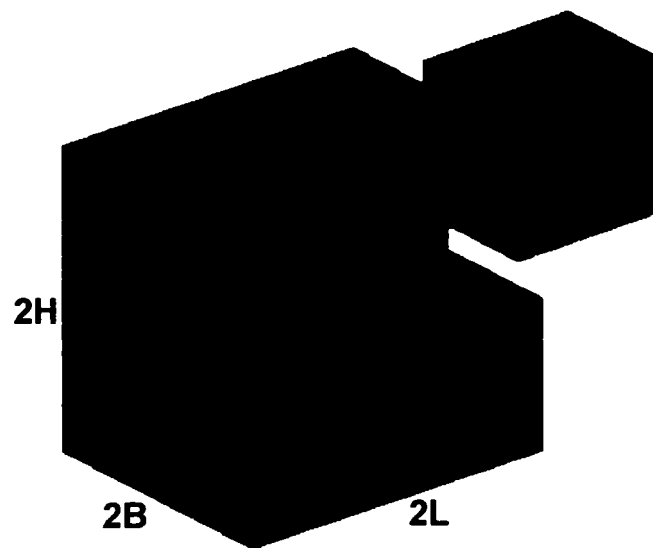


Fig.5.1. Stabilirea domeniului de analiză

⇒ **pasul 2:** Scrierea ecuației diferențiale cu derivate parțiale corespunzătoare, pornind de la forma generală a acesteia, conform relației (2.13) și ținând cont de ipotezele simplificatoare prezentate mai sus [25]:

$$\frac{\partial t}{\partial \tau} = \frac{\lambda}{\rho c} \left(\frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} + \frac{\partial^2 t}{\partial z^2} \right) \quad (5.1)$$

⇒ **pasul 3:** Obținerea unei soluții generale pentru ecuația (5.1). În acest sens soluția se caută de forma următoare [26]:

$$t = v(\tau) + u(x, y, z, \tau) \quad (5.2)$$

După cum se vede, funcția de temperatură căutată am separat-o în două componente, o funcție $v(\tau)$ variabilă doar în timp și o funcție $u(x,y,z,\tau)$ variabilă atât în spațiu cât și în timp. Punerea problemei în acest fel, de obicei nu implică dificultăți majore.

⇒ **pasul 4:** Stabilirea condițiilor de univocitate reprezentate de condițiile inițială și cele de suprafață (sau condiții de contur).

Condiția inițială stabilește câmpul termic în interiorul blumului în momentul începerii încălzirii. Această condiție este exprimată prin relația de mai jos [26]:

$$t|_{\tau=0} = F(x, y, z) \quad (5.3)$$

unde $F(x,y,z)$ reprezintă funcția care precizează distribuția temperaturii în blum la momentul inițial. După cum am arătat mai sus, se consideră că în momentul inițial temperatura în interiorul semifabricatului este aceeași în fiecare punct și egală cu cea a mediului ambiant. Prin urmare:

$$F(x,y,z) = t_0 \approx 20 \text{ }^\circ\text{C} \quad (5.4)$$

Pentru cazul particular în discuție, condițiile de suprafață impuse vor fi cele de ordinul I (sau de tip *Dirichlet*) care precizează distribuția temperaturii la suprafața corpului în fiecare moment. Luând în considerare ipoteza încălzirii uniforme a blumului în incinta de lucru a instalației, variația temperaturii în timp pentru toate cele șase suprafețe libere ale paralelipipedului va fi aceeași. Acest lucru se traduce matematic în următoarele expresii [26]:

$$\begin{aligned} t|_{x=\pm B} &= \varphi(\tau) \\ t|_{y=\pm L} &= \varphi(\tau) \\ t|_{z=\pm H} &= \varphi(\tau) \end{aligned} \quad (5.5)$$

unde $\varphi(\tau)$ este funcția de variație a temperaturii în timp pentru toate punctele aflate pe întreaga suprafață laterală a blumului. Conform ipotezei de mai sus această funcție va reprezenta variația temperaturii incintei instalației experimentale, ca o medie între valorile corespunzătoare diferitelor puncte din spațiul de lucru și considerată într-o variantă simplificată, conform figurii 5.2.

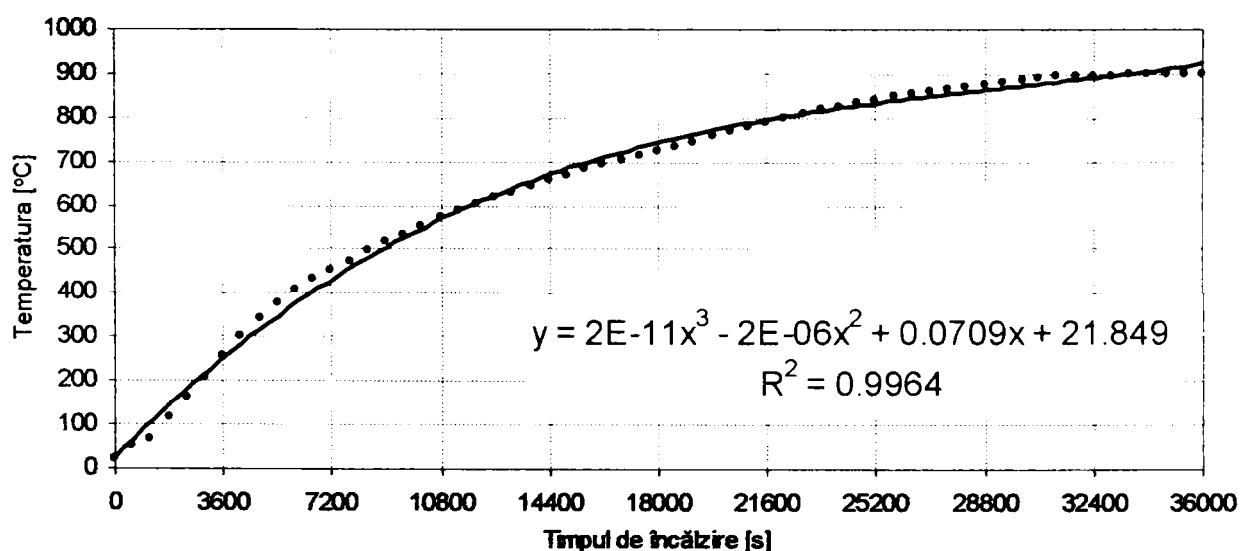


Fig.5.2. Variația în timp a temperaturii suprafețelor exterioare ale blumului (funcția $\varphi(\tau)$)

Funcția $\varphi(\tau)$ s-a obținut prin interpolare polinomială de ordinul 3 a variației temperaturii incintei de lucru obținută ca o medie a valorilor măsurate experimental [27]. După cum se observă și din graficul din figura 5.2, expresia acestei funcții este:

$$\varphi(\tau) = 2 \cdot 10^{-11} \cdot \tau^3 - 2 \cdot 10^{-6} \cdot \tau^2 + 0,0709 \cdot \tau + 21,849 \quad (5.6)$$

Coeficientul de corelație $R^2 = 0,9964$ indică faptul că funcția $\varphi(\tau)$ aproximează fidel variația reală.

Observație: În relația (5.5) s-a notat cu L , B , H [m] semi-lungimea, semi-lățimea și respectiv semi-înălțimea blumului. Valorile acestor dimensiuni sunt precizate în figura 4.7.

⇒ **Pasul 5:** Înlocuirea soluției generale în ecuația principală cu respectarea condițiilor inițiale și la limită spațiale și determinarea constantelor care intervin în calcul.

După substituirea ecuației (5.2) în (5.1) se obține:

$$\frac{du}{d\tau} = \frac{\lambda}{\rho c} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} \right) - \varphi'(\tau) \quad (5.7)$$

Trebuie specificat faptul că, pentru cazul de față vom considera $v(\tau) = \varphi(\tau)$

Funcția necunoscută u trebuie să satisfacă ecuația (5.7) cu condițiile inițiale și la limită:

$$u|_{\tau=0} = F(x, y, z) - \varphi(0) \quad (5.8)$$

$$\begin{cases} u|_{x=\pm B} = 0 \\ u|_{x=\pm L} = 0 \\ u|_{x=\pm H} = 0 \end{cases} \quad (5.9)$$

Soluția ecuației (5.7) se va căuta de forma unei serii triple Fourier [28]:

$$u = \sum_{l=1}^{\infty} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} A_{l,m,n}(\tau) \cos \delta_l \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} \quad (5.10)$$

unde:

$$\begin{cases} \delta_l = \frac{2l-1}{2} \pi, & l = 1, 2, 3, \dots \\ \delta_m = \frac{2m-1}{2} \pi, & m = 1, 2, 3, \dots \\ \delta_n = \frac{2n-1}{2} \pi, & n = 1, 2, 3, \dots \end{cases} \quad (5.11)$$

Se dezvoltă termenul liber ale ecuației (5.7) în serie Fourier:

$$-\varphi'(\tau) = \sum_{l=1}^{\infty} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \alpha_{l,m,n}(\tau) \cos \delta_l \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} \quad (5.12)$$

de unde:

$$\alpha_{l,m,n} = \frac{\int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} [-\varphi'(\tau)] \cos \delta_1 \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} dx dy dz}{\int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} \cos^2 \delta_1 \frac{x}{B} \cos^2 \delta_m \frac{y}{L} \cos^2 \delta_n dx dy dz} \quad (5.13)$$

adică:

$$\alpha_{l,m,n} = \frac{8(-1)^{l+m+n} \varphi'(\tau)}{\delta_1 \delta_m \delta_n} \quad (5.14)$$

După înlocuirea formulelor (5.10) și (5.12) în ecuația (5.7) și după efectuarea calculelor obținem:

$$A'_{l,m,n}(\tau) + \frac{\lambda}{\rho c} \left(\frac{\delta_1^2}{B^2} + \frac{\delta_m^2}{L^2} + \frac{\delta_n^2}{H^2} \right) A_{l,m,n}(\tau) = \alpha_{l,m,n} \quad (5.15)$$

Prin rezolvarea ecuației (5.15) se obține:

$$A_{l,m,n}(\tau) = C_{l,m,n} e^{-\beta_3 \tau} + e^{-\beta_3 \tau} \int_0^{\tau} e^{\beta_3 \tau} \alpha_{l,m,n} d\tau \quad (5.16)$$

unde am notat:

$$a = \frac{\lambda}{\rho c} \text{ [m}^2/\text{s]} \quad \text{și} \quad \beta_3 = \frac{\delta_1^2}{B^2} + \frac{\delta_m^2}{L^2} + \frac{\delta_n^2}{H^2}.$$

Constantele $C_{l,m,n}$ se determină din condițiile inițiale și rezultă de forma:

$$C_{l,m,n} = \frac{\int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} [F(x,y,z) - \varphi(0)] \cos \delta_1 \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} dx dy dz}{\int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} \cos^2 \delta_1 \frac{x}{B} \cos^2 \delta_m \frac{y}{L} \cos^2 \delta_n \frac{z}{H} dx dy dz} \quad (5.17)$$

sau:

$$C_{l,m,n} = \frac{1}{BLH} \int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} \int_{-B-L-H}^{+B+L+H} [F(x,y,z) - \varphi(0)] \cos \delta_1 \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} dx dy dz \quad (5.18)$$

⇒ **Pasul 6:** Introducerea constantelor determinate anterior în soluția generală pentru a obține soluția particulară a problemei [26].

În urma înlocuirii ecuațiilor (5.14) și (5.18) în ecuația (5.16), iar mai apoi ecuația (5.16) în (5.10) și ecuația (5.10) în (5.2) se va obține soluția ecuației conducției termice nestaționare tridimensionale sub forma:

$$\begin{aligned}
 t = \varphi(\tau) + \frac{1}{BLH} \sum_{l=1}^{\infty} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \cos \delta_l \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} e^{-\beta_3 a \tau} \cdot \\
 \cdot \int_{-B-L-H}^{+B+L+H} [F(x, y, z) - \varphi(0)] \cos \delta_l \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} dx dy dz - \\
 - \sum_{l=1}^{\infty} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{8(-1)^{l+m+n+1}}{\delta_l \delta_m \delta_n} \cos \delta_l \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} e^{-\beta_3 a \tau} \int_0^{\tau} e^{\beta_3 a \tau} \varphi'(\tau) d\tau
 \end{aligned} \quad (5.19)$$

Conform relației (5.6) expresia funcției $\varphi(\tau)$ se poate scrie astfel:

$$\varphi(\tau) = M\tau^3 + N\tau^2 + P\tau + Q \quad (5.20)$$

unde am făcut notațiile următoare: $M = 2 \cdot 10^{-11}$, $N = -2 \cdot 10^{-6}$, $P = 0,0709$ și $Q = 21,849$
 Prin derivarea expresiei (5.20) se obține:

$$\varphi'(\tau) = 3M\tau^2 + 2N\tau + P \quad (5.21)$$

Cu expresia (5.21) se calculează în continuare integrala de la sfârșitul termenului al III-lea din soluția (5.19).

$$\begin{aligned}
 \int_0^{\tau} e^{\beta_3 a \tau} \varphi'(\tau) d\tau &= \int_0^{\tau} e^{\beta_3 a \tau} (3M\tau^2 + 2N\tau + P) d\tau = \\
 &= 3M \int_0^{\tau} e^{\beta_3 a \tau} \tau^2 d\tau + 2N \int_0^{\tau} e^{\beta_3 a \tau} \tau d\tau + P \int_0^{\tau} e^{\beta_3 a \tau} d\tau
 \end{aligned} \quad (5.22)$$

Se calculează pe rând cele trei integrale din (5.22):

$$I_1 = \int_0^{\tau} e^{\beta_3 a \tau} d\tau = \frac{1}{\beta_3 a} e^{\beta_3 a \tau} \Big|_0^{\tau} = \frac{1}{\beta_3 a} (e^{\beta_3 a \tau} - 1) \quad (5.23)$$

$$I_2 = \int_0^{\tau} e^{\beta_3 a \tau} \tau d\tau = \frac{\tau}{\beta_3 a} e^{\beta_3 a \tau} \Big|_0^{\tau} - \frac{1}{(\beta_3 a)^2} e^{\beta_3 a \tau} \Big|_0^{\tau} = \frac{\tau}{\beta_3 a} e^{\beta_3 a \tau} - \frac{1}{(\beta_3 a)^2} (e^{\beta_3 a \tau} - 1) \quad (5.24)$$

Integrala I_2 s-a rezolvat folosind formula de integrare prin părți unde părțile au fost alese astfel:

$$\begin{aligned}
 f &= \tau & f' &= 1 \\
 g' &= e^{\beta_3 a \tau} & g &= \frac{1}{\beta_3 a} e^{\beta_3 a \tau}
 \end{aligned}$$

$$I_3 = \int_0^{\tau} e^{\beta_3 a \tau} \tau^2 d\tau = \frac{\tau^2}{\beta_3 a} e^{\beta_3 a \tau} \Big|_0^{\tau} - \frac{2}{\beta_3 a} \int_0^{\tau} e^{\beta_3 a \tau} \tau d\tau \quad (5.25)$$

sau

$$I_3 = \int_0^{\tau} e^{\beta_3 a \tau} \tau^2 d\tau = \frac{\tau^2}{\beta_3 a} e^{\beta_3 a \tau} \Big|_0^{\tau} - \frac{2}{\beta_3 a} I_2 = \frac{\tau^2}{\beta_3 a} e^{\beta_3 a \tau} - \frac{2}{\beta_3 a} I_2 \quad (5.26)$$

Integrala I_3 s-a rezolvat aplicând aceeași formulă de integrare prin părți.

Înlocuind relațiile (5.23), (5.24) și (5.26) în (5.22) se obține:

$$\int_0^{\tau} e^{\beta_3 a \tau} \varphi'(\tau) d\tau = 3M \left(\frac{\tau^2}{\beta_3 a} e^{\beta_3 a \tau} - \frac{2}{\beta_3 a} I_2 \right) + 2NI_2 + PI_1 \quad (5.27)$$

5.2. Soluția analitică. Rezultate

Soluția analitică particulară a problemei studiate se obține înlocuind expresia (5.27) în (5.19):

$$\begin{aligned} t = & \varphi(\tau) + \frac{1}{BLH} \sum_{l=1}^{\infty} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \cos \delta_l \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} e^{-\beta_3 a \tau} \cdot \\ & \cdot \int_{-B-L-H}^{+B+L+H} [F(x, y, z) - \varphi(0)] \cos \delta_l \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} dx dy dz - \\ & - \sum_{l=1}^{\infty} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{8(-1)^{l+m+n+1}}{\delta_l \delta_m \delta_n} \cos \delta_l \frac{x}{B} \cos \delta_m \frac{y}{L} \cos \delta_n \frac{z}{H} e^{-\beta_3 a \tau} \cdot \\ & \cdot 3M \left(\frac{\tau^2}{\beta_3 a} e^{\beta_3 a \tau} - \frac{2}{\beta_3 a} I_2 \right) + 2NI_2 + PI_1 \end{aligned} \quad (5.28)$$

unde funcția $\varphi(\tau)$ este dată de expresia (5.6).

Programul sursă MATLAB [29] pentru efectuarea calculelor este prezentat în continuare:

```

% PROGRAM pentru determinarea distribuției 3D a temperaturii
clc, clear, format compact; format bank
h=figure;
for i=1:h,
    close(i),
end,
a=7.4*10^(-6);
F=20;
B=0.12;
L=0.15;
H=0.135;
A=2*10^(-11);
B1=-2*10^(-6);
D=0.0709;
E=21;
x=[0 B/2 47*B/48 B]; %coordonata x a punctelor în care se calc.temp.
%x=[0 B*.1 B*.2 B*.3 B*.4 B*.5 B*.6 B*.7 B*.8 B*.9 B];
    
```



```


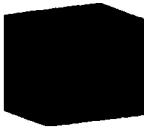





y=[0 L/2 9*L/10 L]; %coordonata y a punctelor în care se calc.temp.
x=[0 L*.1 L*.2 L*.3 L*.4 L*.5 L*.6 L*.7 L*.8 L*.9 L];

z=0.12; %coordonata z a punctelor în care se calc.temp.
v=1:60;
timp(v)=600+(v-1)*600 %pasul de timp este de 600s
ta=timp;
nrc=4, %nrc = număr cicluri de însumare
f(ta)=A*(ta.^3)+B1*ta.^2+D*ta+E;
for k=1:length(ta)
    tau=ta(k);
    g=zeros(length(y),length(x));
    for m=1:length(x)
        for n=1:length(y)
            s1=0;
            s2=0;
            for i=1:nrc
                for j=1:nrc
                    for k=1:nrc
                        di=(2*i-1)*pi/2;
                        dj=(2*j-1)*pi/2;
                        dk=(2*k-1)*pi/2;
                        bet=(di/B)^2+(dj/L)^2+(dk/H)^2;
                        ter1=1/(di*dj*dk)*cos(di*x(m)/B)*cos(dj*y(n)/L)*cos(dk*z/H);
                        ter1=ter1*exp(-bet*a*tau);
                        ter1=ter1*sin(di)*sin(dj)*sin(dk);
                        ter1=ter1*6*(B/di)*(L/dj)*(H/dk);
                        ter2=((8*(-1)^(i+j+k+1))/(di*dj*dk));
                        ter2=ter2*cos(di*x(m)/B)*cos(dj*y(n)/L)*cos(dk*z/H);
                        ter2=ter2*exp(-bet*a*tau);
                        int1=tau*exp(bet*a*tau)/(bet*a)-(exp(bet*a*tau)-1)/(bet*a)^2;
                        int=3*A*((tau^2)*exp(bet*a*tau)/(bet*a)-
                            -2*int1/(bet*a))+2*B1*int1+D*(exp(bet*a*tau)-1)/(bet*a);
                        ter2=ter2*int;
                        s1=s1+ter1;
                        s2=s2+ter2;
                    end
                end
            end
            g(n,m)=f(tau)+(F-f(0))*s1-s2;
        end
    end
end
vec(1)=g(3,2)
disp(g)
h=figure;
meshc(x,y,g)
meshc(g)
surf(g)
end % sfârșit program

```

În urma rulării se pot pune la dispoziție valorile temperaturilor punctelor corespunzătoare pozițiilor în care au fost făcute măsurătorile experimentale. Corespondența coordonatelor punctelor în care s-a efectuat calculul analitic cu punctele de măsurare în blumul experimental sunt prezentate în tabelul următor.

Tabelul 5.1

Coordonatele punctelor în care s-a calculat analitic temperatura	Punctele de măsurare din interiorul blumului
	
$x = 47B/48, y = L/2, z = 0,12$	 1
$x = B/2, y = L/2, z = 0,0675$	 2
$x = 0, y = 0, z = 0$	 3
$x = B/2, y = L/2, z = 0,0675$	 4
$x = 47B/48, y = 9L/10, z = 0,12$	 5

Legătura din tabelul 5.1 ține seama de ipoteza simetriei încălzirii, luată în considerare la stabilirea modelului analitic al problemei (vezi figura 5.1).

Variația în timp a temperaturii, calculată din 600 în 600 s, pentru întreg intervalul de încălzire de 10 h și pentru fiecare din cele 5 puncte, a căror coordonate sunt prezentate în tabelul 5.1, este arătată în figura următoare.

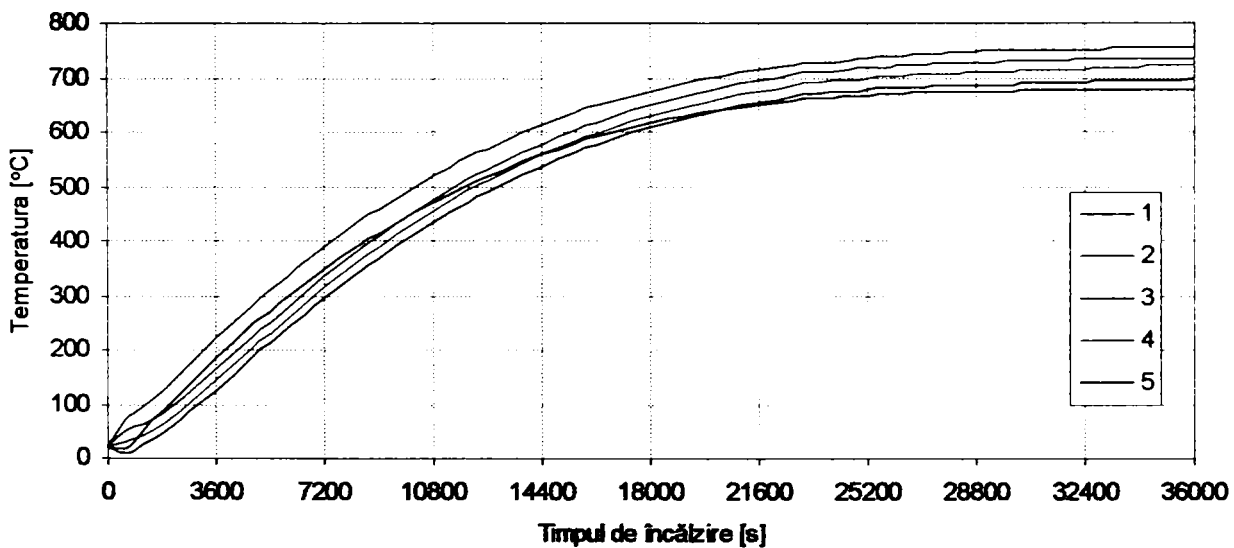


Fig.5.2. Variația în timp a temperaturii calculată analitic pentru cele 5 puncte

5.3. Concluzii

Având la dispoziție curbele experimentale de variație a temperaturii în cele 5 puncte de măsurare din interiorul semifabricatului (vezi figura 4.7), considerate de referință, în continuare se vor face aprecieri comparative la adresa celor obținute pe cale analitică, în punctele analoge conform tabelului 5.1.

În figura 5.3 s-au reprezentat comparativ variațiile în timp ale temperaturii obținute prin măsurători experimentale și calculate analitic, pentru punctul de măsurare nr.1 (vezi figura 4.7). Semnificația graficelor din figurile 5.4, 5.5, 5.6 și 5.7 este aceeași, dar pentru punctele caracteristice 2, 3, 4, și respectiv 5.

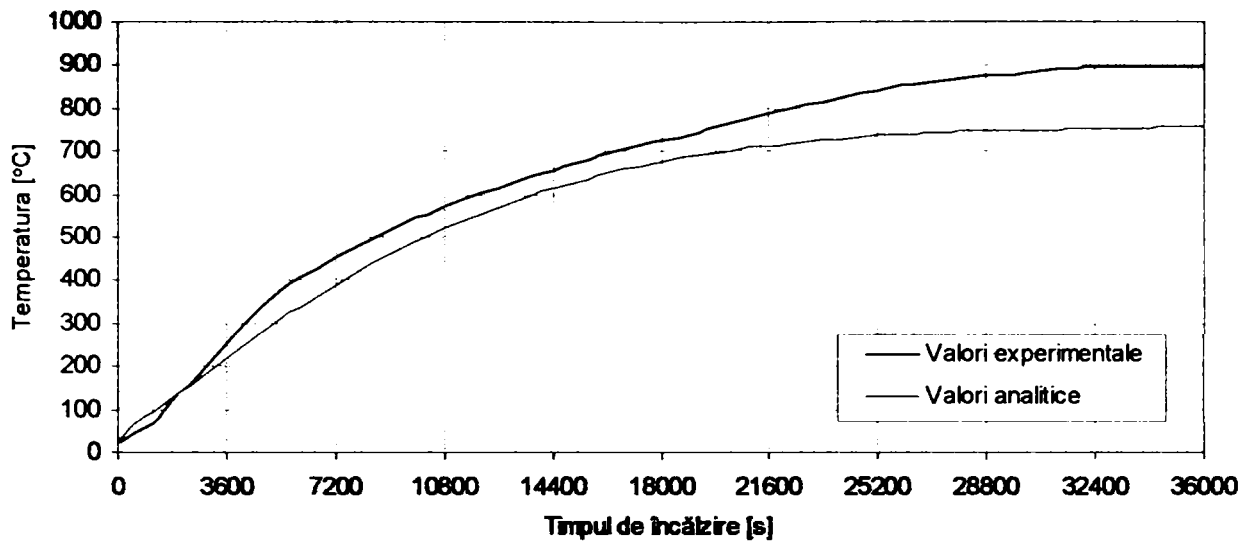


Fig.5.3. Rezultate comparative corespunzătoare punctului 1 de măsurare

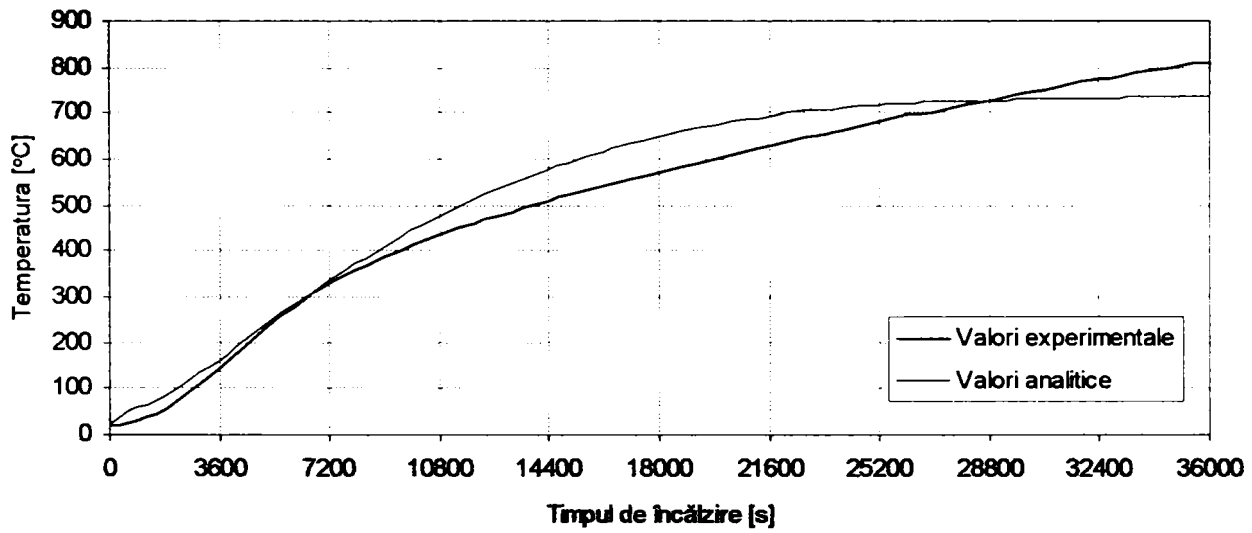


Fig.5.4. Rezultate comparative corespunzătoare punctului 2 de măsurare

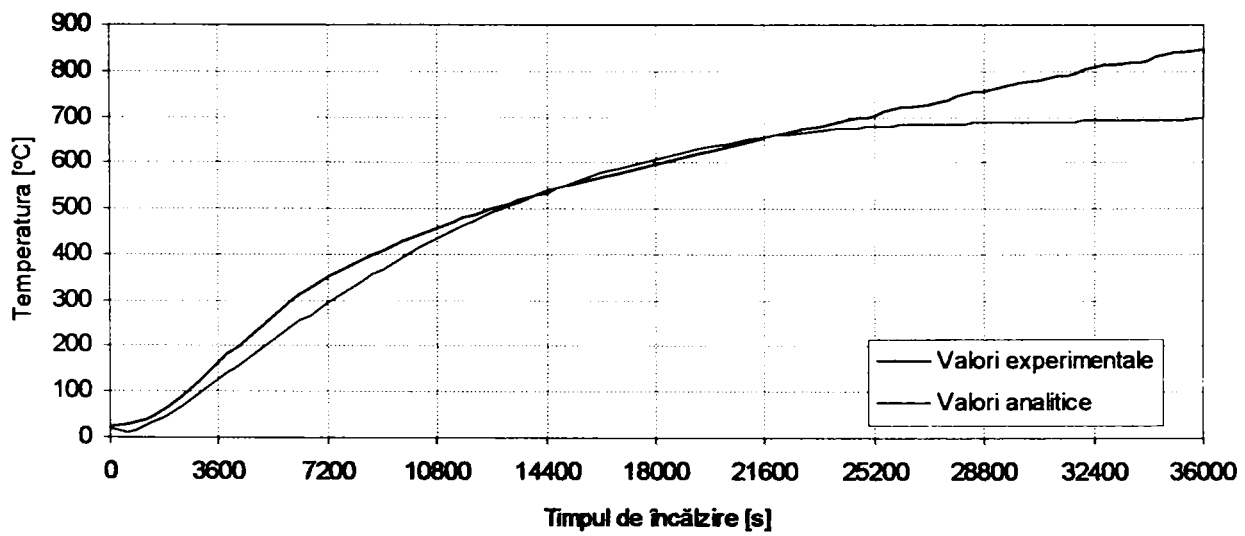


Fig.5.5. Rezultate comparative corespunzătoare punctului 3 de măsurare

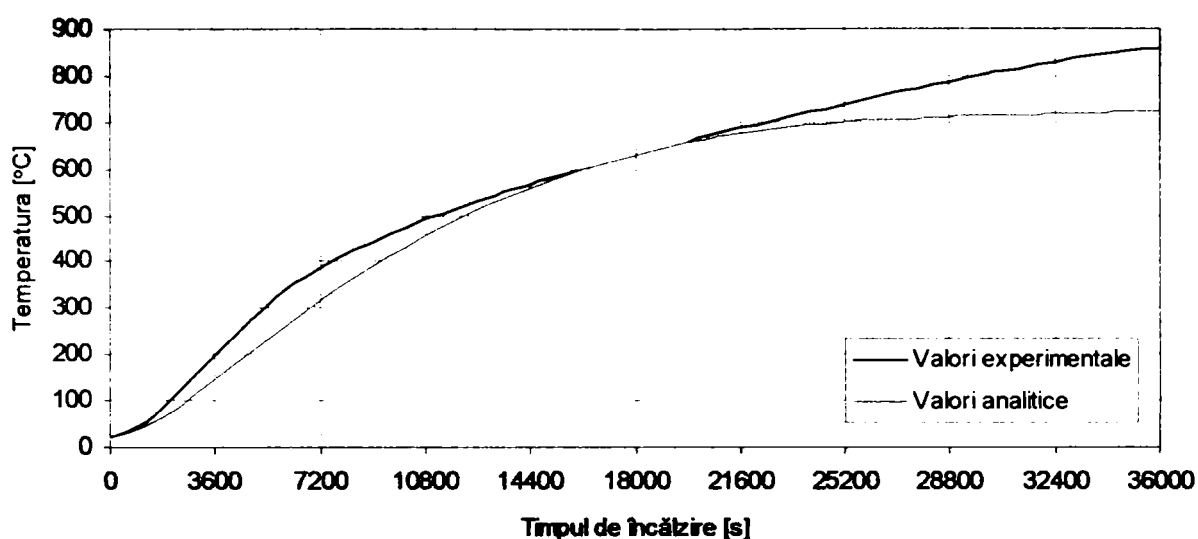


Fig.5.6. Rezultate comparative corespunzătoare punctului 4 de măsurare

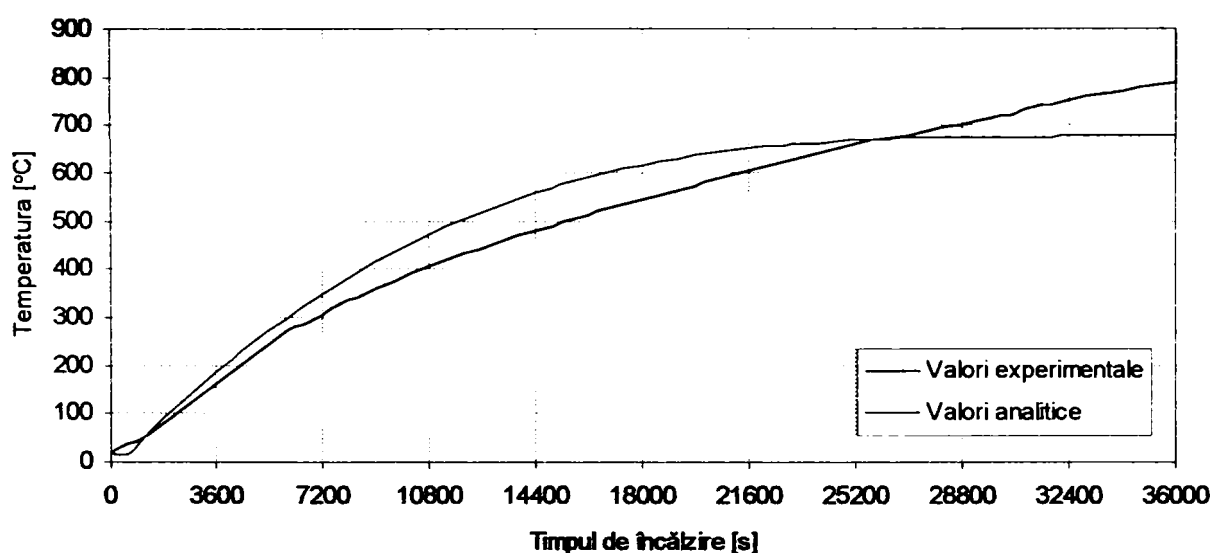


Fig.5.7. Rezultate comparative corespunzătoare punctului 5 de măsurare

În figura 5.8 s-a reprezentat, pentru întreg intervalul de încălzire, diferența efectivă între valorile măsurate experimental, considerate etalon și cele obținute analitic, pentru toate cele cinci puncte caracteristice. În plus, figura 5.9 prezintă eroarea relativă a valorilor calculate analitic față de cele măsurate experimental, pentru aceleași cinci puncte. Modul de calcul al acestor valori este următorul:

$$\Rightarrow \text{Pentru diferența efectivă: } \Delta t = t_{\text{exp}} - t_{\text{analitic}} \text{ [}^{\circ}\text{C]}$$

$$\Rightarrow \text{Pentru eroarea relativă: } \varepsilon_r = \frac{t_{\text{exp}} - t_{\text{analitic}}}{t_{\text{exp}}} \cdot 100 \text{ [\%]}$$

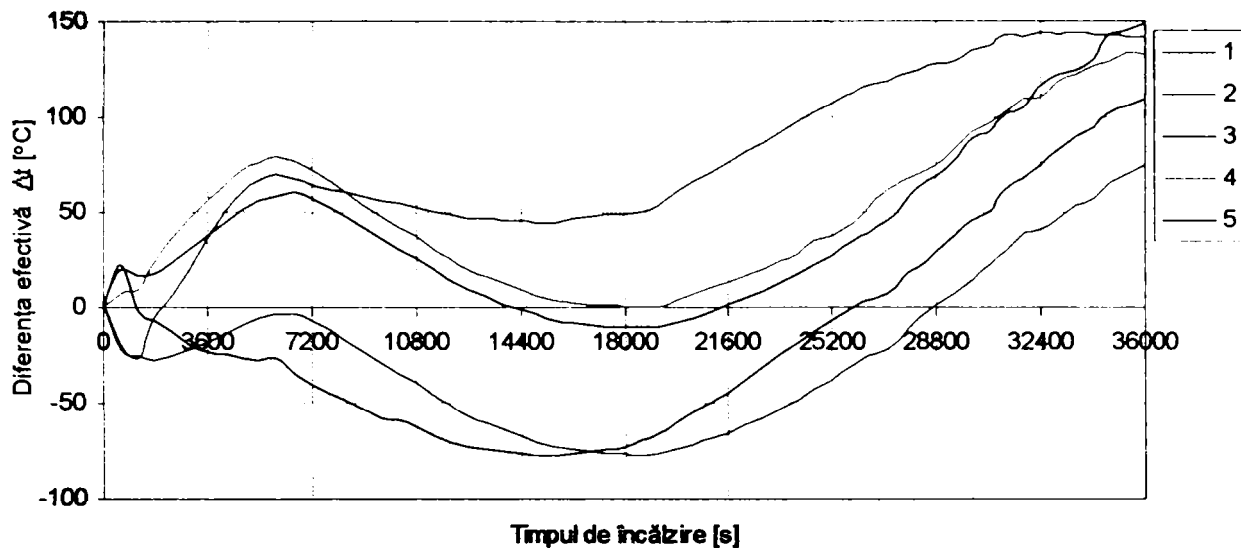


Fig.5.8. Diferența efectivă între valorile măsurate experimental și cele obținute prin calcul analitic

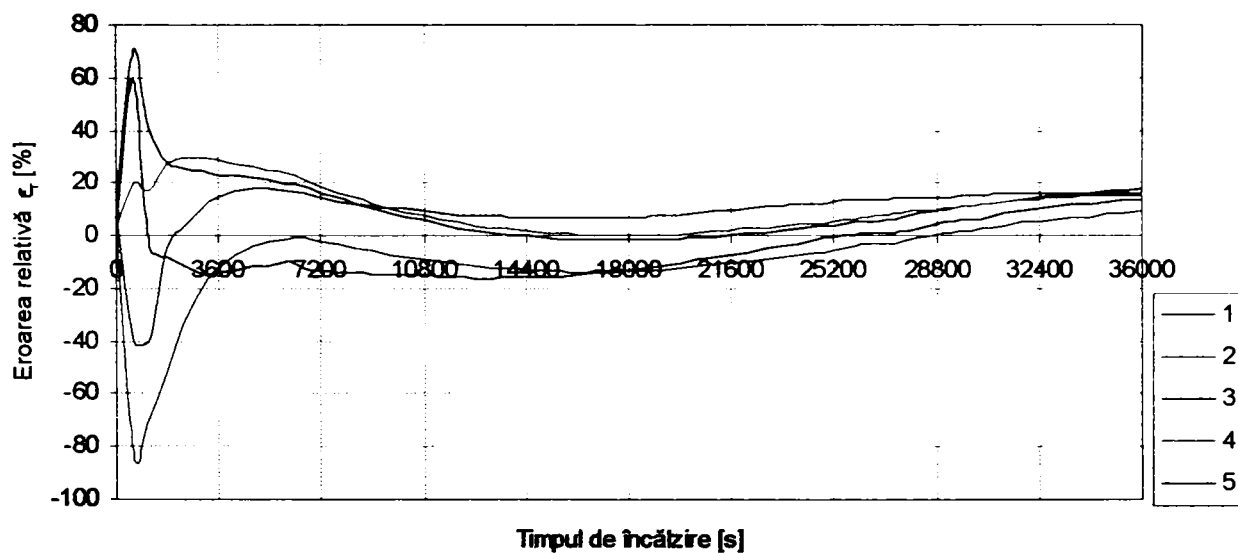


Fig.5.9. Eroarea relativă ale valorile obținute prin calcul analitic față de cele măsurate experimental

□ Privind în ansamblu dinamica evoluției temperaturii măsurate experimental în cele cinci puncte, putem spune că valorile calculate analitic în punctele corespunzătoare urmăresc această variație cu o abatere a cărei mărime variază funcție de perioada de încălzire.

□ Pentru întreg intervalul de încălzire, diferențele efective minime între valorile obținute prin calcul analitic și cele măsurate experimental se înregistrează pentru punctul caracteristic nr.3, în rest aceste diferențe fiind destul de semnificative. Acest aspect se poate observa și în graficul din figura 5.8. Explicația acestui fapt constă în ipoteza simetriei încălzirii care stă la baza elaborării modelului analitic, în cazul real fiind vorba despre o încălzire asimetrică. Această ipoteză simplificatoare afectează în mod deosebit calculul temperaturii punctelor de la periferia domeniului studiat și mai puțin centrul acestuia (punctul caracteristic nr.3).

□ Urmărind în continuare figura 5.8 se poate constata faptul că diferențele între valorile măsurate și cele calculate analitic sunt mai mari în perioadele de început și respectiv de sfârșit ale intervalului de încălzire. Este normal să se întâmple așa deoarece, în elaborarea modelului analitic, proprietățile termo-fizice de material au fost considerate constante cu temperatura și având valori medii pe întreg intervalul de temperatură. Introducerea acestor valori medii pentru fiecare moment

de timp face ca modelul analitic să se apropie de realitate doar pentru temperaturi corespunzătoare mijlocului intervalului de încălzire.

□ Spre sfârșitul perioadei de încălzire diferențele valorilor calculate față de cele măsurate devin semnificative, ajungând până la 150 °C. Această abatere exagerată este pusă pe seama unor neajunsuri legate de aparatul matematic, în mod special fiind vorba aici de convergența discutabilă a seriilor Fourier pentru timpi foarte mari.

□ În altă ordine de idei, legat de erorile relative ale valorilor calculate analitic față de cele măsurate experimental se poate vedea în figura 5.9 că acestea sunt foarte mari, ajungând până la 70%, în intervalul corespunzător primei ore de încălzire. Acestea scad apoi încadrându-se într-o plajă de $\pm 20\%$, urmând ca spre sfârșitul încălzirii să înregistreze din nou o ușoară creștere.

□ În final, putem concluziona că modelul analitic construit este în măsură să pună la dispoziția utilizatorului rezultate relativ exacte cu privire la evoluția câmpului de temperaturi în interiorul unui domeniu de analiză ales, cu condiția ca datele inițiale furnizate să se apropie cât mai mult de cele reale existente în practică, iar condițiile la limită, inițiale și spațiale, să fie cât mai puțin simplificate astfel încât, modelul analitic să simuleze cât mai fidel fenomenul real [30] [31].

Capitolul 6

Modelul numeric cu diferențe finite

- 6.1. Considerații teoretice
 - 6.1.1. Cazul regimului nestaționar tridimensional
 - 6.1.2. Criteriul de stabilitate a ecuațiilor cu diferențe finite
- 6.2. Implementarea modelului numeric cu diferențe finite
 - 6.2.1. Modelul tridimensional al încălzirii blumului
 - 6.2.2. Conductivitatea termică
 - 6.2.3. Temperatura redusă
 - 6.2.4. Entalpia
 - 6.2.5. Ipoteze simplificatoare
- 6.3. Descrierea și funcționarea programului sursă pentru simularea câmpului termic
- 6.4. Prezentarea rezultatelor
 - 6.4.1. Câmpul termic
 - 6.4.2. Evoluția temperaturii punctelor caracteristice
- 6.5. Concluzii

Capitolul 6

Modelul numeric cu diferențe finite

6.1. Considerații teoretice

6.1.1. Cazul regimului nestaționar tridimensional

După cum am arătat și în capitolul 3, rezolvarea prin metoda diferențelor finite a ecuației conducției termice are la bază transformarea acestei ecuații diferențiale cu derivate parțiale într-un sistem de ecuații liniare cu diferențe finite.

Pentru cazul particular al conducției nestaționare tridimensionale, fără surse interne de căldură, ecuația (2.15) ia următoarea formă:

$$\frac{\partial t}{\partial \tau} = a \left(\frac{\partial^2 t}{\partial x^2} + \frac{\partial^2 t}{\partial y^2} + \frac{\partial^2 t}{\partial z^2} \right) \quad (6.1)$$

Având în vedere dependența conductivității termice λ de temperatură, se introduce noțiunea de *temperatură redusă* care include această variație [32]:

$$\Phi = \int_{t_0}^t \frac{\lambda}{\lambda_0} dt \quad (6.2)$$

unde λ și λ_0 sunt conductivitățile termice la temperatura t și la o temperatură arbitrară t_0 . Se obține:

$$\frac{\partial \Phi}{\partial \tau} = a(\Phi) \left(\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} \right) \quad (6.3)$$

$$\frac{\partial \Phi}{\partial \tau} \frac{\partial t}{\partial \tau} \frac{\rho c}{\lambda} = \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} \quad (6.4)$$

$$\frac{\lambda}{\lambda_0} \frac{\partial t}{\partial \tau} \frac{\partial H}{\partial t} \frac{\rho}{\lambda} = \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} \quad (6.5)$$

$$\frac{\partial H}{\partial \tau} = \frac{\lambda_0}{\rho} \left(\frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} + \frac{\partial^2 \Phi}{\partial z^2} \right) \quad (6.6)$$

unde în relațiile (6.5) și (6.6) am notat cu H [kJ/kg] entalpia materialului supus încălzirii. Ultima relație este avantajoasă pentru că face posibil calculul direct al conținutului de căldură (entalpia H). Pentru căldura specifică a materialului s-a folosit expresia :

$$c = \frac{\partial H}{\partial \tau} \quad (6.7)$$

În urma unei discretizări a domeniului de analiză conform figurii 3.1 se va exprima temperatura unui punct al acestuia (i,j,k) funcție de temperatura punctelor vecine [33]. Pentru început se consideră că acest punct face parte din interiorul domeniului (nu se află pe frontieră).

$$\Phi_{i-1,j,k} = \Phi_{i,j,k} - \frac{x_1}{1!} \frac{\partial \Phi_{i,j,k}}{\partial x} + \frac{x_1^2}{2!} \frac{\partial^2 \Phi_{i,j,k}}{\partial x^2} \quad (6.8)$$

$$\Phi_{i+1,j,k} = \Phi_{i,j,k} + \frac{x_2}{1!} \frac{\partial \Phi_{i,j,k}}{\partial x} + \frac{x_2^2}{2!} \frac{\partial^2 \Phi_{i,j,k}}{\partial x^2} \quad (6.9)$$

unde am notat cu x_1 și respectiv x_2 pașii inegali de discretizare. Rezultă astfel:

$$x_2 \Phi_{i-1,j,k} + x_1 \Phi_{i+1,j,k} = (x_1 + x_2) \Phi_{i,j,k} + \frac{x_2 x_1^2 + x_1 x_2^2}{2} \frac{\partial^2 \Phi_{i,j,k}}{\partial x^2} \quad (6.10)$$

$$\frac{\partial^2 \Phi_{i,j,k}}{\partial x^2} = \frac{2}{x_1 x_2 (x_1 + x_2)} \left[x_2 \Phi_{i-1,j,k} + x_1 \Phi_{i+1,j,k} - (x_1 + x_2) \Phi_{i,j,k} \right] \quad (6.11)$$

Pentru axele y și z se procedează identic. Variația de entalpie exprimată sub formă de diferențe finite are următoarea formă:

$$\frac{\partial H}{\partial \tau} \cong \frac{H_{i,j,k,n+1} - H_{i,j,k,n}}{\tau_{n+1} - \tau_n} = \frac{\Delta H_{i,j,k}}{\Delta \tau} \quad (6.12)$$

Indicele n din (6.12) exprimă momentul de timp.

$$\begin{aligned} \frac{\Delta H_{i,j,k}}{\Delta \tau} = & \frac{2\lambda_0}{\rho x_1 x_2 (x_1 + x_2)} \left[x_2 \Phi_{i-1,j,k} + x_1 \Phi_{i+1,j,k} - (x_1 + x_2) \Phi_{i,j,k} \right] + \\ & + \frac{2\lambda_0}{\rho y_1 y_2 (y_1 + y_2)} \left[y_2 \Phi_{i,j-1,k} + y_1 \Phi_{i,j+1,k} - (y_1 + y_2) \Phi_{i,j,k} \right] + \\ & + \frac{2\lambda_0}{\rho z_1 z_2 (z_1 + z_2)} \left[z_2 \Phi_{i,j,k-1} + z_1 \Phi_{i,j,k+1} - (z_1 + z_2) \Phi_{i,j,k} \right] \end{aligned} \quad (6.13)$$

În cazul în care distribuția punctelor de discretizare este omogenă de-a lungul celor trei axe (caz frecvent utilizat) și notând $x_1 = x_2 = \Delta x$, $y_1 = y_2 = \Delta y$, $z_1 = z_2 = \Delta z$ se obține:

$$\begin{aligned} \frac{\Delta H_{i,j,k}}{\Delta \tau} = & \frac{\lambda_0}{\rho \Delta x^2} [\Phi_{i-1,j,k} + \Phi_{i+1,j,k} - 2\Phi_{i,j,k}] + \\ & + \frac{\lambda_0}{\rho \Delta y^2} [\Phi_{i,j-1,k} + \Phi_{i,j+1,k} - 2\Phi_{i,j,k}] + \\ & + \frac{\lambda_0}{\rho \Delta z^2} [\Phi_{i,j,k-1} + \Phi_{i,j,k+1} - 2\Phi_{i,j,k}] \end{aligned} \quad (6.14)$$

Ecuția cu diferențe finite face posibilă determinarea variației de entalpie într-un interval de timp $\tau_{n+1} - \tau_n$, în funcție de temperatura punctelor vecine. Dacă se cunoaște distribuția inițială de temperaturi $\Phi_{i,j,k,0}$ (sau entalpii $H_{i,j,k,0}$) se poate determina pe baza relațiilor (6.13) sau (6.14) distribuția de temperaturi după un interval de timp dat, rezultând $\Phi_{i,j,k,1}$. Pe baza acestui rezultat se poate determina distribuția de temperaturi $\Phi_{i,j,k,2}$ etc. Deci, printr-o metodă iterativă se poate obține evoluția distribuției de temperaturi în cursul încălzirii semifabricatului.

Ecuțiile (6.13) și (6.14) sunt valabile pentru un punct din interiorul materialului. Pentru un punct situat pe o suprafață limită (frontieră), ecuațiile respective se modifică. Pentru punctele de pe frontieră se scrie ecuația fluxului de căldură la suprafață:

$$\dot{Q} = -\lambda \frac{\partial \Phi_{i,j,k}}{\partial x} \quad (6.15)$$

unde $\Phi_{i,j,k}$ este temperatura la suprafață (limita considerată paralelă cu axa Oy). Expresia $\lambda \frac{\partial \Phi_{i,j,k}}{\partial x}$ se poate scrie sub formă de diferență finită în funcție de o temperatură $\Phi_{i+1,j,k}$ a unui punct imaginar situat la distanța $x = x_1$ (pentru simplificare) de suprafață. Fluxul de căldură prin suprafață trebuie să fie:

$$\dot{Q} = \frac{\lambda_0}{2x} (\Phi_{i-1,j,k} - \Phi_{i+1,j,k}) \quad (6.16)$$

de unde:

$$\Phi_{i+1,j,k} = \Phi_{i-1,j,k} - \frac{2x}{\lambda_0} \dot{Q} \quad (6.17)$$

Relația (6.11) devine:

$$\frac{\partial^2 \Phi_{i,j,k}}{\partial x^2} = \frac{2}{x^2} \left[\Phi_{i-1,j,k} - \frac{2x}{\lambda_0} \dot{Q} - \Phi_{i,j,k} \right] \quad (6.18)$$

iar (6.13) primește forma:

$$\begin{aligned} \frac{H_{i,j,k+1} - H_{i,j,k}}{\tau_{k+1} - \tau_k} = & \frac{2\lambda_0}{\rho x^2} \left[\Phi_{i-1,j,k} - \frac{x}{\lambda_0} \dot{Q} - \Phi_{i,j,k} \right] + \\ & + \frac{2\lambda_0}{\rho y_1 y_2 (y_1 + y_2)} [y_2 \Phi_{i,j-1,k} + y_1 \Phi_{i,j+1,k} - (y_1 + y_2) \Phi_{i,j,k}] \end{aligned} \quad (6.19)$$

Pentru o suprafață paralelă cu axa Ox se obține (în mod identic):

$$\frac{H_{i,j,k+1} - H_{i,j,k}}{\tau_{k+1} - \tau_k} = \frac{2\lambda_0}{\rho x_1 x_2 (x_1 + x_2)} \left[x_2 \Phi_{i-1,j,k} + x_1 \Phi_{i+1,j,k} - (x_1 + x_2) \Phi_{i,j,k} \right] + \frac{2\lambda_0}{\rho y^2} \left[\Phi_{i,j-1,k} - \frac{y}{\lambda_0} \dot{Q} - \Phi_{i,j,k} \right] \quad (6.20)$$

Pentru un punct de colț rezultă relația:

$$\frac{H_{i,j,k+1} - H_{i,j,k}}{\tau_{k+1} - \tau_k} = \frac{2\lambda_0}{\rho x^2} \left[\Phi_{i-1,j,k} - \frac{x}{\lambda_0} \dot{Q} - \Phi_{i,j,k} \right] + \frac{2\lambda_0}{\rho y^2} \left[\Phi_{i,j-1,k} - \frac{y}{\lambda_0} \dot{Q} - \Phi_{i,j,k} \right] \quad (6.21)$$

6.1.2. Criteriul de stabilitate a ecuațiilor cu diferențe finite

Una din problemele critice ale aplicării prin iterație a ecuației cu diferențe finite este asigurarea stabilității și acurateței soluției în cursul integrării. Pentru a asigura stabilitatea soluției, adică pentru a împiedica oscilația soluției în cursul integrării, intervalul de timp între iterații și dimensiunile rețelei trebuie ales în conformitate cu criteriile de stabilitate deduse. Acuratețea soluției, adică concordanța cât mai bună cu soluția analitică este de asemenea de dorit. Aceasta depinde de forma ecuației cu diferențe finite și de desimea rețelei.

În concluzie, desimea rețelei se stabilește pornind de la necesitatea găsirii unei rezolvări a următoarei contradicții: pe de o parte folosirea unei rețele mai dese mărește precizia modelului (eroarea introdusă prin ipoteza că suprafața adiacentă fiecărui nod are aceeași temperatură cu a nodului scade odată cu aria nodului rețelei); pe de altă parte o rețea deasă mărește durata de efectuare a calculului atât datorită creșterii numărului de noduri, cât și reducerii intervalelor de timp între iterații dictate de condițiile de stabilitate a soluției.

Din analiza stabilității ecuației cu diferențe finite se deduc următoarele criterii:

$$\text{în interior} \quad \Delta\tau \leq \frac{1}{2a \left[\frac{1}{x_1 x_2} + \frac{1}{y_1 y_2} \right]} \quad (6.22)$$

$$\text{pe margine} \quad \Delta\tau \leq \frac{1}{2a \left[\frac{1}{x_1 x_2} + \frac{1}{y^2} + \frac{\dot{Q}}{\lambda y} \right]} \quad (6.23)$$

$$\text{pe colț} \quad \Delta\tau \leq \frac{1}{2a \left[\frac{1}{x^2} + \frac{1}{y^2} + \frac{\dot{Q}}{\lambda x} + \frac{\dot{Q}}{\lambda y} \right]} \quad (6.24)$$

Intervalul de timp ales reprezintă de fapt timpul în care procesul nestaționar al transmiterii căldurii este aproximat cu un proces staționar. Din această cauză, cu cât caracteristicile procesului real se abat mai puternic de la cele ale unui proces staționar, cu atât trebuie să fie mai mică durata iterației.

În relațiile (6.13), (6.14), (6.19), (6.20) și (6.21) se observă că membrul drept depinde numai de mărimi cunoscute la momentul iterației k . Dacă notăm aceste expresii cu $G_{i,j,k}$ se obține:

$$\frac{H_{i,j,k+1} - H_{i,j,k}}{\tau_{k+1} - \tau_k} = G_{i,j,k} \quad (6.25)$$

Mărimea $G_{i,j,k}$ reprezintă de fapt viteza de variație a entalpiei punctului (i,j,k) în procesul considerat staționar care începe la momentul τ_k . Cu cât această mărime este mai mare în valoare absolută, cu atât procesul este mai nestaționar, iar intervalul de timp trebuie ales mai mic.

Soluția optimă constă în folosirea unui interval de timp variabil și limitarea variației entalpiei la o valoare fixată ΔH_{\max} . Astfel se obține:

$$\Delta\tau_k = \tau_{k+1} - \tau_k = \frac{\Delta H_{\max}}{\max_{(i,j) \in D} |G_{i,j,k}|} \quad (6.26)$$

unde D este domeniul valorilor (i,j,k) .

Astfel se obține un interval de timp mic atunci când variația entalpiei este mare și un interval mare atunci când aproximarea cu un proces staționar este mai bună.

6.2. Implementarea modelului numeric cu diferențe finite

6.2.1. Modelul tridimensional al încălzirii blumului

Se propune realizarea unui model de simulare numerică, folosind metoda diferențelor finite, a fenomenului de încălzire a blumului în instalația experimentală prezentată în capitolul 4. În acest scop domeniul de analiză, reprezentat de un mediu continuu sub forma unui paralelipiped cu dimensiunile din figura 4.7, este transformat într-un mediu discontinuu (discret), format dintr-o rețea de puncte (rețea de discretizare) a cărei desime este dată de pașii aleși pentru fiecare axă de coordonate.

Realizarea modelului numeric impune stabilirea unor date inițiale și anume:

- valoarea temperaturii inițiale a blumului, considerată uniformă în toată masa acestuia: $t_0 = 20$ °C;
- valoarea temperaturii mediului ambiant: $t_a = 20$ °C;
- dimensiunile geometrice ale blumului: $0,24 \times 0,30 \times 0,27$ m³;
- numărul nodurilor din interiorul domeniului de analiză, după cele trei axe: câte 20 pentru fiecare;
- valorile conductivității termice a materialului blumului funcție de temperatură – figura 6.1;
- valorile entalpiei materialului blumului funcție de temperatură – figura 6.4;
- coeficientul de convecție între suprafața blumului și gazele arse – figura 6.8;
- durata totală de încălzire: $\tau_{\text{total}} = 36000$ s (10 h);
- maximul variației entalpiei la o iterație – figura 6.8.

Relația (6.13) permite calculul entalpiei unui nod în funcție de entalpia aceluiași nod la momentul precedent și temperaturile reduse ale nodului respectiv și a celor adiacente. Pentru a continua iterarea, avem nevoie să deducem temperatura din entalpie, apoi temperatura redusă din temperatură. La o iterație, sunt necesare următoarele transformări:

$$\dots \rightarrow \Phi \rightarrow H \rightarrow t \rightarrow \Phi \rightarrow \dots \quad (6.27)$$

Pentru a efectua calculele, trebuie cunoscute dependențele funcționale $\lambda(t)$, $\Phi(t)$, $t(\Phi)$, $H(t)$ și $t(H)$. Acestea trebuie să fie suficient de precise, dar și să fie executate rapid de către calculator. De exemplu, dacă avem o rețea de 50×50 noduri și un număr de 150.000 iterații (situație obișnuită), aceste funcții vor fi calculate de 375.000.000 ori. Rezultă de aici că funcțiile respective sunt critice din punct de vedere al timpului de execuție.

Graficele și tabelele următoare corespund mărcii de oțel OL 37- 2k.

6.2.2. Conductivitatea termică

Conductivitatea termică a oțelului este dată în literatură sub formă de tabele sau grafice. Reprezentând grafic o astfel de dependență, se observă ca ea poate fi aproximată foarte bine printr-o funcție pe două intervale: un arc de parabolă până la o temperatură t_{cr} ($t_{cr} = 768 \text{ }^\circ\text{C}$, este temperatura punctului Curie) și o valoare constantă peste această temperatură. Funcția are forma:

$$\lambda(t) = \begin{cases} at^2 + bt + c, & t \leq t_{cr} \\ \lambda_{cr}, & t > t_{cr} \end{cases} \quad (6.28)$$

unde, pentru a asigura continuitatea funcției în punctul $t = t_{cr}$ se calculează λ_{cr} cu:

$$\lambda_{cr} = at_{cr}^2 + bt_{cr} + c \quad (6.29)$$

Parabola este un polinom de regresie de ordinul 2, ai cărui coeficienți sunt calculați prin metoda celor mai mici pătrate. Valorile obținute sunt:

$$\begin{aligned} a &= -3,325347756276 \cdot 10^{-5} \\ b &= -8,628845527225 \cdot 10^{-3} \\ c &= 5,0928618576856 \end{aligned}$$

În tabelul 6.1 se prezintă valorile experimentale [34] [35] și cele approximate prin această metodă, iar în figura 6.1 aceleași date sub formă grafică.

Tabelul 6.1

$t \text{ [}^\circ\text{C]}$	$\lambda \text{ [W/m}^\circ\text{C]}$ experimental	$\lambda \text{ [W/m}^\circ\text{C]}$ aproximat
20	50,66	50,743
200	48,13	47,873
400	41,85	42,157
600	33,95	33,780
800	24,65	24,688
1000	24,65	24,688

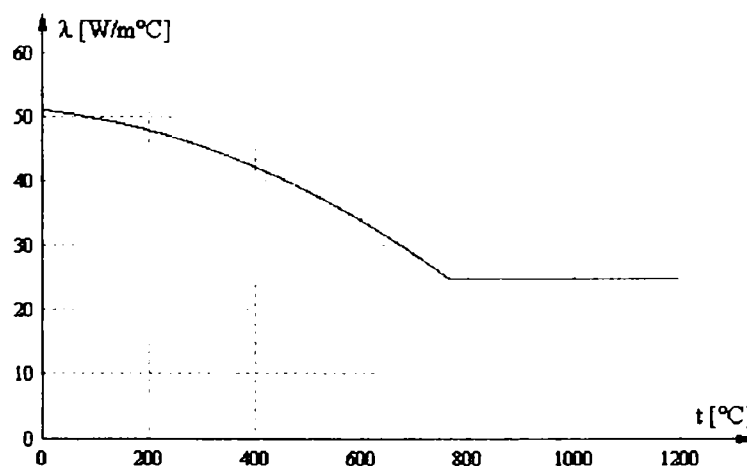


Fig.6.1. Variația aproximată a conductivității termice λ cu temperatura

6.2.3. Temperatura redusă

Faptul că $\lambda(t)$ are forma dată de (6.28) ușurează mult calculul integralei:

$$\Phi(t) = \int_{t_0}^t \frac{\lambda}{\lambda_0} dt \tag{6.30}$$

deoarece se poate aplica o metodă analitică. Pentru simplificare (amintim că t_0 este ales arbitrar, iar $\lambda_0 = \lambda(t_0)$), alegem $t_0 = 0$ °C. Deoarece $\lambda_0 = \lambda(0) = c$, relația (6.30) devine:

$$\Phi(t) = \frac{1}{c} \int_{t_0}^t \lambda(t) dt = \begin{cases} \frac{a}{3c} t^3 + \frac{b}{2c} t^2 + t, & t \leq t_{cr} \\ \frac{a}{3c} t_{cr}^3 + \frac{b}{2c} t_{cr}^2 + t_{cr} + (t - t_{cr}) \frac{\lambda_{cr}}{c}, & t > t_{cr} \end{cases} \tag{6.31}$$

sau:

$$\Phi(t) = \begin{cases} t \left(\frac{a}{3c} t^2 + \frac{b}{2c} t + 1 \right), & t \leq t_{cr} \\ t \frac{\lambda_{cr}}{c} + t_{cr} \left(\frac{a}{3c} t_{cr}^2 + \frac{b}{2c} t_{cr} + 1 - \frac{\lambda_{cr}}{c} \right), & t > t_{cr} \end{cases} \tag{6.32}$$

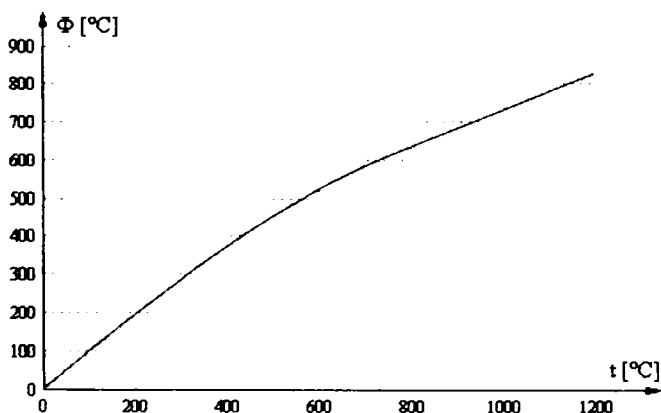


Fig.6.2. Dependența $\Phi(t)$

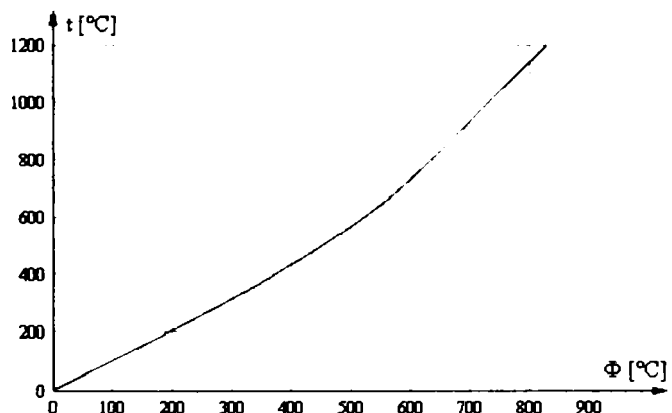


Fig.6.3. Dependența $t(\Phi)$

Pentru calculul $t(\Phi)$ se rezolvă ecuația de gradul I dacă $t < t_{cr}$, iar în cazul $t \geq t_{cr}$ se folosește metoda aproximațiilor succesive (metoda lui Newton).

6.2.4. Entalpia

Variația entalpiei materialului cu temperatura este dată în literatura de specialitate sub formă tabelară sau grafică [36]. În tabelul 6.2 sunt date valorile pentru H în funcție de temperatură. Ultimele trei linii (cele scrise cursiv) corespund temperaturilor superioare punctului *solidus*. Funcțiile $H(t)$ și $t(H)$ se obțin prin interpolare liniară.

Tabelul 6.2

Nr. crt.	t [°C]	H [kcal/kg]	H [kJ/kg]
0	100	11,2	46,88
1	200	22,6	94,59
2	300	36,2	151,52
3	400	49,9	208,86
4	500	64,3	269,13
5	600	82,0	343,21
6	700	101,0	422,74
7	800	130,8	547,46
8	900	148,1	619,87
9	1000	164,6	688,93
10	1100	181,7	760,51
11	1200	198,6	831,24
12	1300	207,5	868,49
13	1400	220,0	920,64
14	1500	284,0	1188,51
15	1600	310,6	1300,27

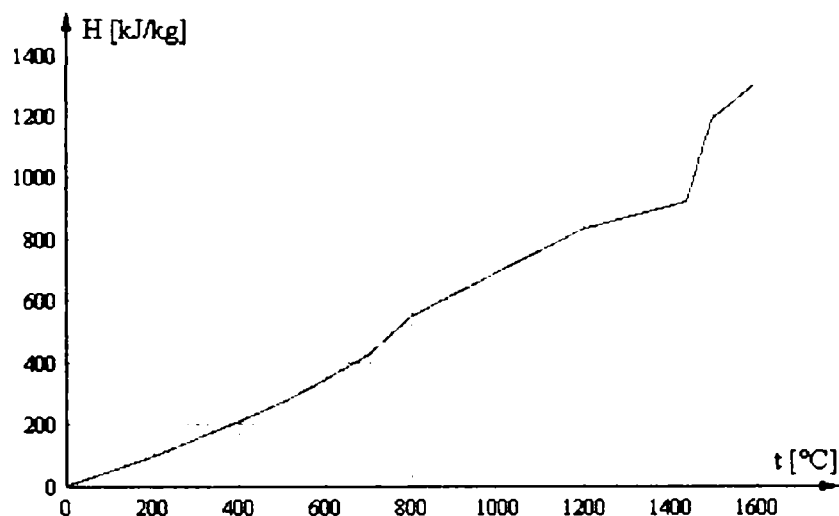


Fig.6.4. Dependenta H(t)

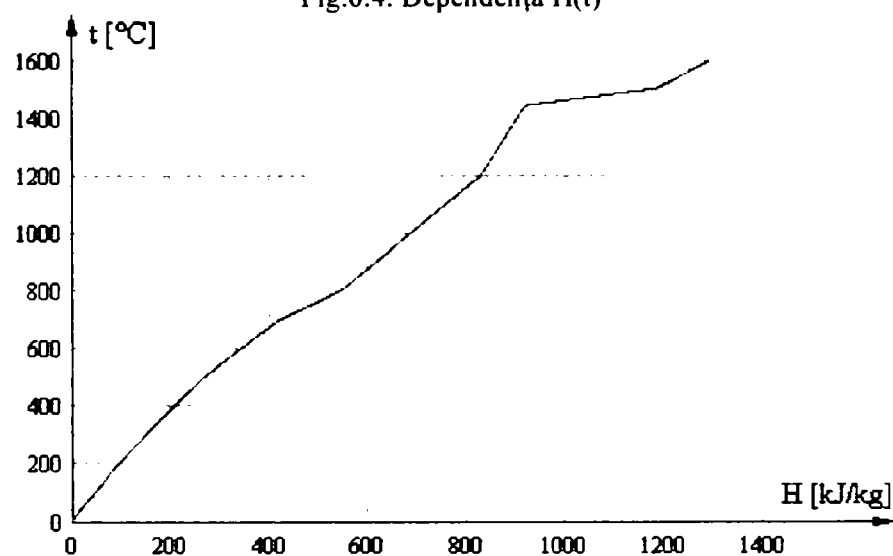


Fig.6.5. Dependenta t(H)

Figurile 6.4 și 6.5 prezintă funcțiile $H(t)$ și $t(H)$ sub formă grafică.

6.2.5. Ipoteze simplificatoare

După cum s-a precizat deja, simularea vizează determinarea câmpului de temperaturi stabilit în interiorul blumului încălzit în instalația experimentală. Realizarea modelului numeric cu diferențe finite implică luarea în considerare a unor ipoteze simplificatoare care se referă în principal la stabilirea modului de transfer de căldură de la atmosfera incintei de încălzire la suprafața materialului. În acest sens:

□ s-a neglijat efectul radiației gazelor arse și a suprafeței interioare a zidăriei, schimbul de căldură virtual realizându-se doar prin convecție, cu un coeficient α_C constant în timp și același pentru fiecare din suprafețele libere ale blumului.

□ valoarea mare a acestuia ($1000 \text{ W}/(\text{m}^2\text{K})$) sugerează faptul că temperaturile suprafețelor libere ale blumului urmăresc îndeaproape temperatura gazelor arse din imediata vecinătate a lor.

□ pentru suprafața de sprijin a semifabricatului valoarea coeficientului α_C a primit o valoare exagerată ($10^5 \text{ W}/(\text{m}^2\text{K})$) în timp ce temperatura exterioră suprafeței respective devine de fapt temperatura vetrei. În acest mod se simulează transferul termic prin contact între vatră și baza blumului.

6.3. Descrierea și funcționarea programului sursă pentru simularea câmpului termic

Programul de calcul este realizat în limbajul C++ și funcționează sub platforme Win32 (adică Windows 95, 98, Me, NT4, 2000, XP – cu procesor Intel). Alegerea limbajului este motivată de faptul că este bine cunoscut de autor, iar programele compilate în C/C++ sunt cele mai rapide (cu excepția limbajului de asamblare, care este însă foarte greu de folosit) [37] [38].

Pentru interfața grafică, programul folosește MFC (Microsoft Foundation Classes), o bibliotecă de clase care încapsulează funcționalitatea interfeței standard de programare Windows (API – Application Program Interface). Funcțiile C standard (cum sunt `fopen()` sau `exit()`), care nu sunt parte din Windows API necesită o altă bibliotecă: MSVCRT (Microsoft Visual C Runtime). Graficele tridimensionale sunt realizate cu implementarea Windows a specificației OpenGL (Open Graphics Library). Aceste biblioteci dinamice (MFC42.DLL, MSVCRT.DLL, OPENGL32.DLL și GLU32.DLL) sunt instalate implicit cu orice sistem de operare din cele menționate, cu excepția Windows 95 – unde trebuie instalate explicit.

Sursa programului are o construcție modulară și orientată pe obiecte. Un modul C++ constă în general dintr-o pereche de fișiere: unul cu extensia .H (de la *header* – antet) care conține declarații de funcții și/sau clase, fiind folosit la interfațarea cu celelalte module, și unul cu extensia .CPP (de la C++) care conține definiții (implementări ale funcțiilor și claselor declarate în header). Programul de față conține 28 astfel de module, însă numai 4 ## privesc în mod direct calculele și vor fi prezentate în anexă. Celelalte sunt auxiliare acestora sau implementează grafice, ferestre, dialoguri etc. [39] [40] [41] [42] [43] [44] [45].

Cele mai importante clase C++ implicate în procesul de simulare sunt:

– **Material** – conține descrierea unui material (oțel sau fontă). Această clasă implementează dependențele funcționale $\lambda(t)$, $\Phi(t)$, $t(\Phi)$ prin metode analitice și $H(t)$ și $t(H)$ prin interpolare liniară, așa cum s-a arătat în subcapitolele 6.2.1, 6.2.2 și 6.2.3. Clasa dispune de metode pentru serializare (poate fi scrisă într-un fișier sau citită dintr-unul).

– **HeatFlowSequence** – această clasă realizează calculele corespunzătoare unei singure iterații. Constructorul acestei clase are ca argumente două referințe: una către clasa `MetalDataFile` (descrisă mai jos) care a creat-o și una către o altă instanță a `HeatFlowSequence` care reprezintă momentul de timp precedent. Dacă această referință este invalidă (NULL), atunci este vorba de prima iterație, iar acțiunea va fi de a aplica valorile inițiale pentru t , Φ și H . Pentru a reduce la minim necesitatea conversiilor între t , Φ și H , clasa conține trei matrici distincte corespunzătoare acestor mărimi în nodurile rețelei.

Toate calculele sunt realizate în virgulă mobilă dublă precizie, pentru a reduce erorile propagate. Acestea se datorează faptului că, deși precizia este destul de mare atât pentru simplă precizie (`float` – 32 biți, 8 zecimale) cât și în dublă precizie (`double` – 64 biți, 17 zecimale), simularea conduce la un șir foarte lung de valori care sunt calculate unele din altele.

După terminarea calculelor, numai temperaturile vor fi salvate în fișierul gestionat de `MetalDataFile` în simplă precizie – suficientă pentru construirea de grafice.

– **SequenceInterpolator** – deoarece clasele care implementează grafice au nevoie de funcții continue, iar temperatura unui nod este considerată media temperaturii suprafețelor adiacente, este necesară interpolarea valorilor obținute prin calcul pentru a obține o funcție continuă. Această sarcină îi revine clasei `SequenceInterpolator`. Interpolarea este liniară și se realizează mai întâi după axa X și apoi după Y. Contrar aparențelor, rezultatul nu este o suprafață plană ci un hiperboloid mărginit de patru segmente de dreaptă pe care le atinge.

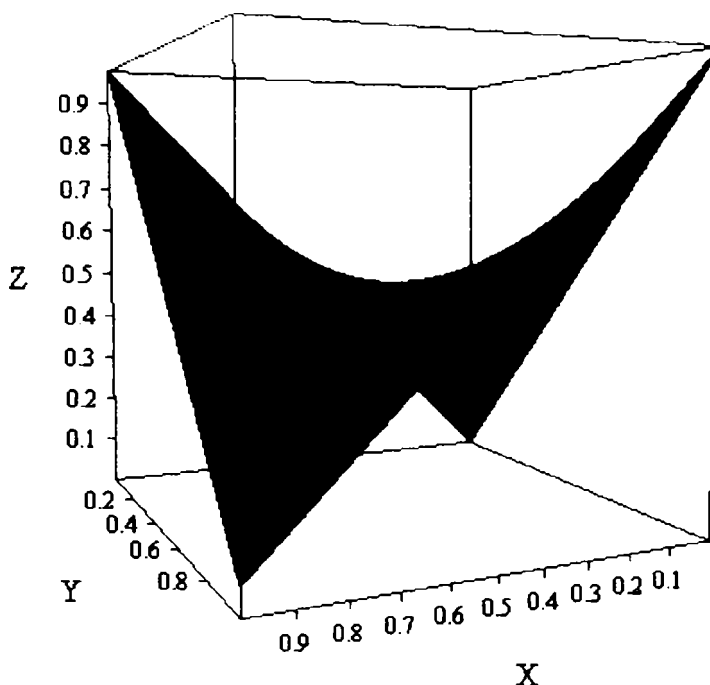


Fig.6.6

Clasa `MetalDataFile`, atunci când este în starea de furnizor de date, păstrează în memorie două instanțe ale `SequenceInterpolator`. Acestea sunt folosite pentru încă o interpolare liniară, de data aceasta în timp.

– **CProgressDialog** – este o derivată a clasei MFC **CDialog** și implementează un dialog modal (care nu permite accesul la fereastra principală). Acesta se deschide atunci când se pornește procesul de calcul – pe care îl controlează indirect – și se închide la terminarea calculelor fie prin comandă manuală, fie la atingerea condiției de oprire.

La inițializarea dialogului se pornește un fir de execuție (*thread*) separat de cel principal, cu care va rula în paralel. Acest *thread* efectuează de fapt calculele apelând funcția `calculate()` a clasei `MetalDataFile`. La rândul ei, aceasta apelează funcția `callback()` a clasei `CProgressDialog` la fiecare iterație, ceea ce permite actualizarea datelor afișate. Deoarece este vorba de calcule intensive și îndelungate *thread*-ul are nivel de prioritate scăzut (-2), ceea ce înseamnă că va ceda timp altor programe dacă este cazul (se poate lucra bine în același timp cu altceva). În figura 6.7. este prezentată fereastra de dialog care se deschide în momentul inițializării programului, iar în figura 6.8, cea prin intermediul cărei sunt introduse datele necesare rulării programului.

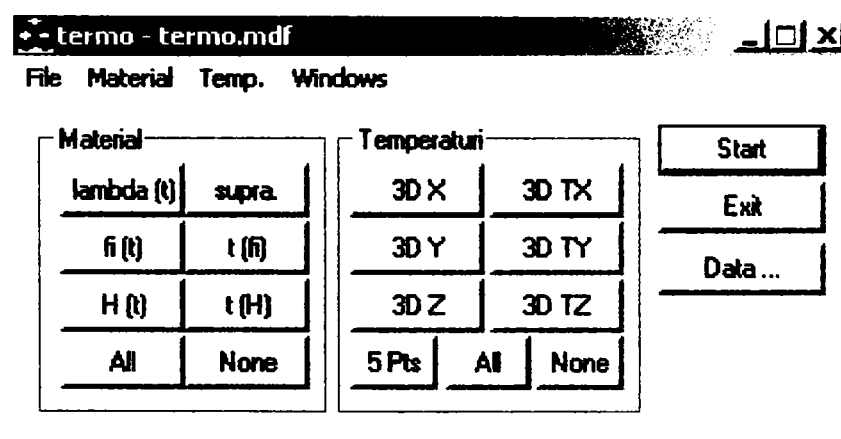


Fig.6.7. Fereastra principală a programului

Data			
Nr. puncte discretizare	20	x	20
Geometria (mm)	240	x	300
		x	270
Temperatura inițială [°C]	20		
Valorile alfa (1-6)	1000	1000	1000
	1000	1000	100000
Variația maximă a entalpiei / iteratie	100		
Durata minimă a iteratiei [s]	100		
Durata totală a simulării	36000		
Intervalul minim între 2 salvări pe disc	30		
<input type="button" value="OK"/> <input type="button" value="Cancel"/> <input type="button" value="Reset"/>			

Fig.6.8. Fereastra de introducere a datelor

– **MetalDataFile** – această clasă gestionează un fișier temporar, în care sunt scrise datele în procesul de calcul și din care sunt citite pentru construcția de grafice. S-a folosit un fișier și nu memoria pentru că dimensiunea datelor rezultate în urma calculelor poate fi foarte mare și este în plus și imprevizibilă. Fișierul este șters de către destructorul acestei clase (atunci când se termină programul), dar poate fi copiat în alt loc și utilizat mai târziu prin comenzi din meniu.

Salvarea datelor în fișier nu are loc la fiecare iterație – nu ar avea rost decât dacă intervalul de la ultima salvare este de cel puțin 5 secunde în timp simulat (timpul lingoului virtual). Astfel se asigură o dimensiune rezonabilă a fișierului fără să se piardă rezoluția simulării.

Această clasă conține bucla principală de calcul și instanțe sau referințe către instanțe ale claselor precedente. Există o singură instanță globală a **MetalDataFile**.

6.4. Prezentarea rezultatelor

6.4.1. Câmpul termic

Programul realizat permite vizualizarea calitativă a câmpului termic, pentru diferite momente de timp, în secțiuni plane ale blumului, perpendiculare pe una din axele de coordonate ale sistemului cartezian, secțiuni situate la diferite înălțimi față de planul de bază cu care acestea sunt paralele. Spre exemplu, pentru vizualizarea distribuției temperaturii într-o secțiune paralelă cu planul xOy situată la distanța $z = 0$ față de acesta (în acest caz secțiunea este chiar planul xOy), după o oră de la începutul încălzirii, se va activa fereastra grafică prin selectarea butonului inscripționat „3DZ” aflat în fereastra principală din figura 6.7, după care se deschide fereastra intitulată „Dialog” din figura 6.9 și se va muta cursorul, corespunzător stabilirii procentuale a distanței z , în poziția 0, iar cursorul, corespunzător stabilirii momentului de timp, în poziția 60. Câmpul termic în acest caz este prezentat în figura 6.10.

Mutând cursorul corespunzător stabilirii momentului de timp în poziția de sfârșit a cursei sale (la sfârșitul celor 10 h de încălzire) se obține câmpul termic în același plan dar în momentul de

sfârșit al încălzirii (figura 6.11). În continuare, păstrând notațiile din figura 5.1, dimensiunea pe Ox a blumului este (2B), pe Oy este (2L), iar de-a lungul axei Oz este (2H).

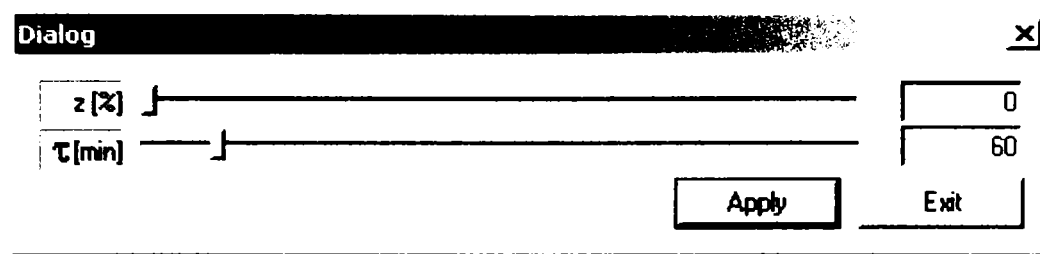


Fig.6.9. Fereastra " Dialog "

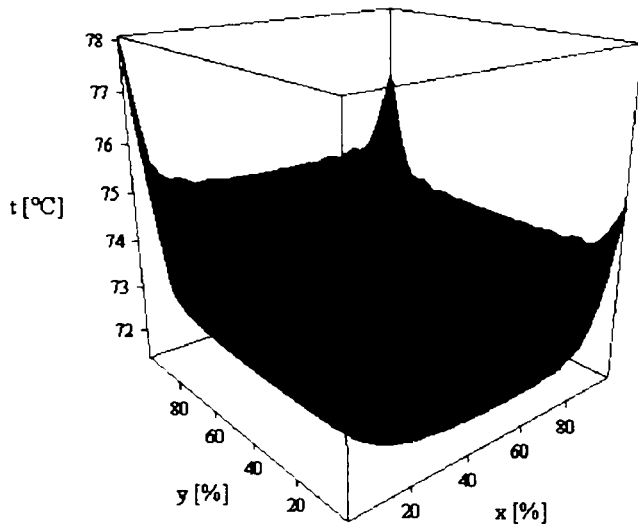


Fig.6.10. Câmpul termic în suprafața de sprijin a blumului după o oră de la începutul încălzirii

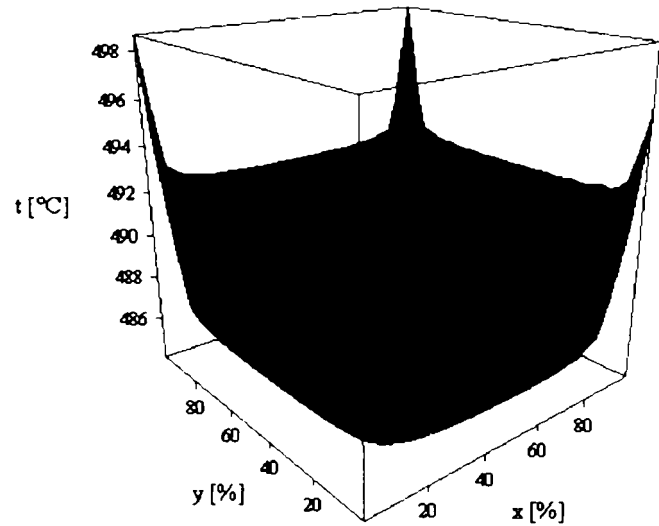


Fig.6.11. Câmpul termic în suprafața de sprijin a blumului la sfârșitul încălzirii (după 10 h)

În mod similar se pot fi vizualizate distribuțiile de temperaturi în diferite secțiuni și la diferite momente de timp. Câteva exemple sunt prezentate în continuare.

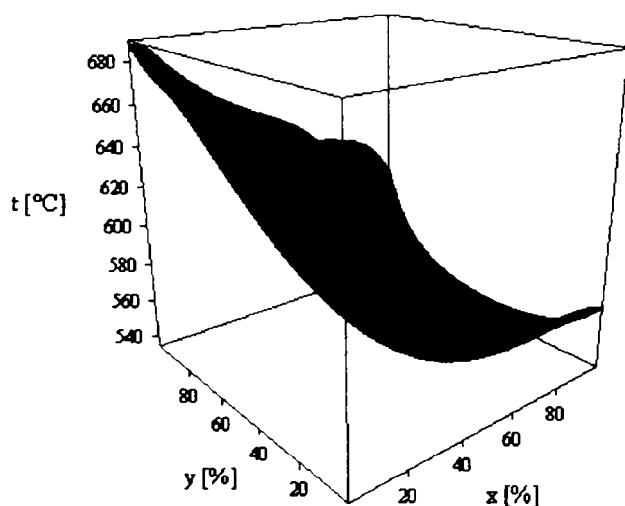


Fig.6.12. Câmpul termic într-un plan paralel cu xOy, situat la $z = \frac{1}{2} (2H)$, după 5 h de încălzire

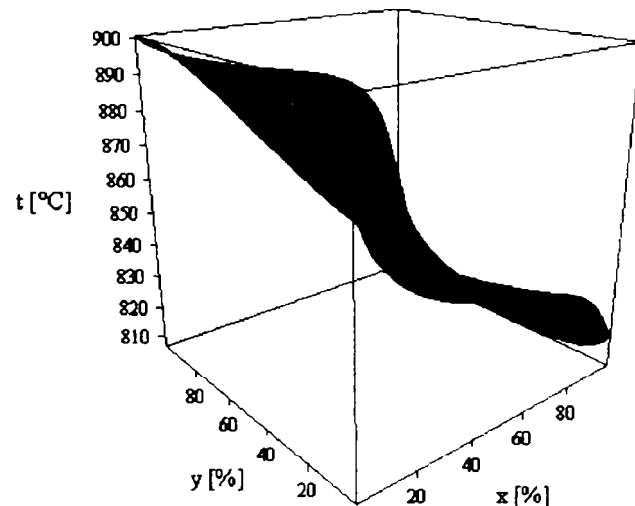


Fig.6.13. Câmpul termic într-un plan paralel cu xOy, situat la $z = \frac{3}{4} (2H)$, la sfârșitul încălzirii (după 10 h)

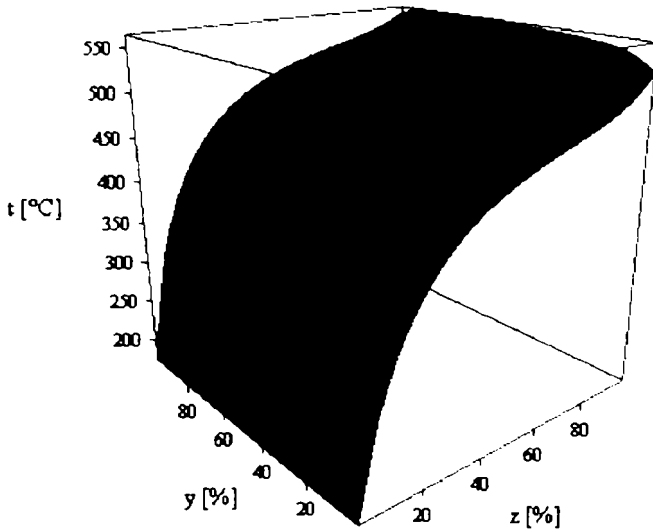


Fig.6.14. Câmpul termic într-un plan paralel cu yOz, situat la $x = \frac{1}{4}$ (2B), după 3 h de încălzire

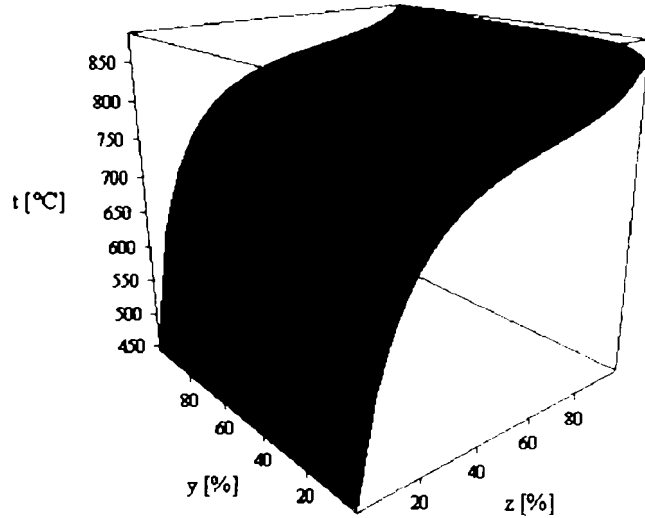


Fig.6.15. Câmpul termic într-un plan paralel cu yOz, situat la $x = \frac{1}{4}$ (2B), după 9 h de încălzire

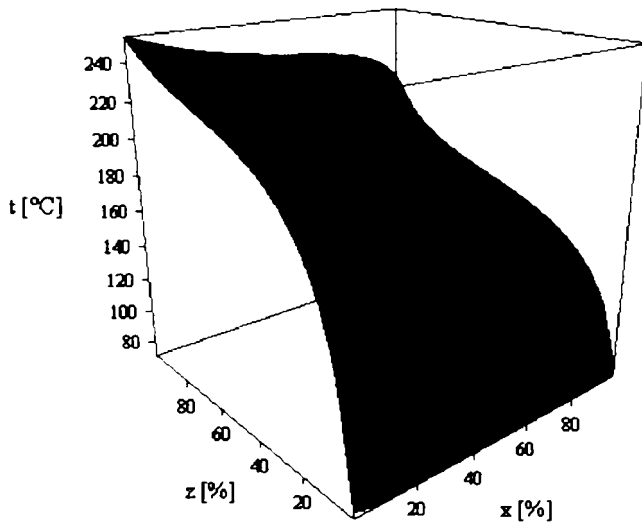


Fig.6.16. Câmpul termic într-un plan paralel cu xOz, situat la $y = \frac{1}{4}$ (2L), după o oră de la începutul încălzirii

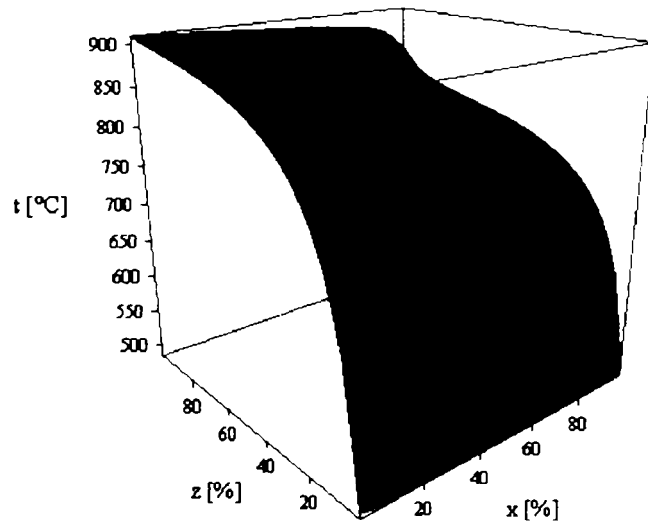


Fig.6.17. Câmpul termic într-un plan paralel cu xOz, situat la $y = \frac{1}{2}$ (2L), la sfârșitul încălzirii (după 10 h)

Pot fi prezentate, de asemenea, grafice care ilustrează variația în timp a temperaturii blumului după o axă a sistemului de coordonate. De această dată se va activa fereastra grafică dorită prin selectarea unui buton din cele trei inscripționate „3DTX”, „3DTY” și respectiv „3DTZ”, aflate în fereastra principală din figura 6.7, după cum se dorește ca variația să fie după axa Ox, Oy sau Oz. Se deschide apoi fereastra intitulată „Dialog” din figura 6.18, urmând a se stabili celelalte două coordonate carteziene.

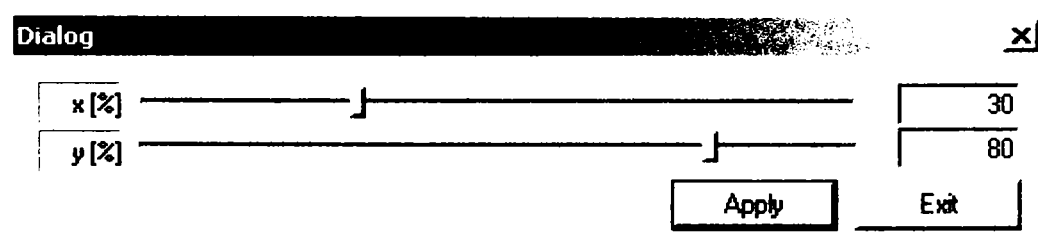


Fig.6.18. Fereastra „ Dialog ”

Observație: Fereastra din figura 6.18 corespunde unui grafic de variație a temperaturii blumului în timp după axa Oz și stabilește procentual, prin intermediul cursorilor, coordonatele x și respectiv y .

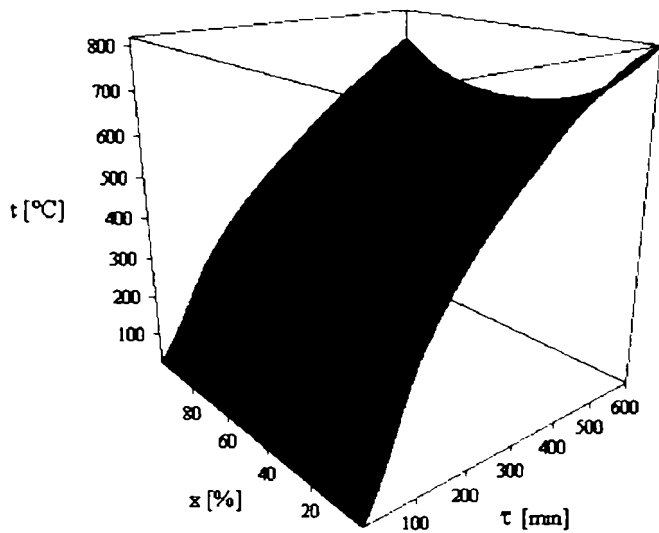


Fig.6.19. Variația temperaturii în timp după axa Ox, pentru $y = \frac{1}{4}$ (2L) și $z = \frac{1}{4}$ (2H)

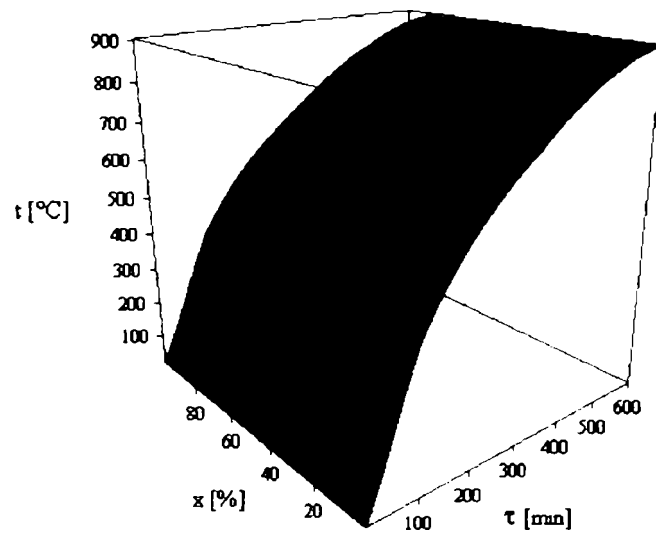


Fig.6.20. Variația temperaturii în timp după axa Ox, pentru $y = (2L)$ și $z = (2H)$

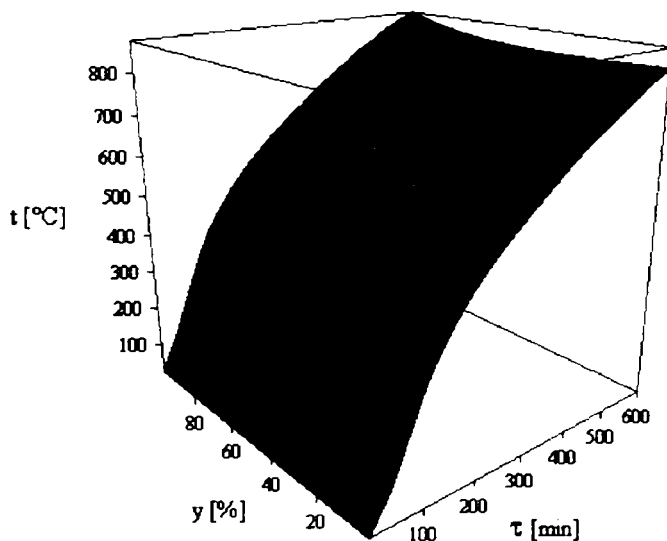


Fig.6.21. Variația temperaturii în timp după axa Oy, pentru $x = \frac{3}{4}$ (2B) și $z = \frac{3}{4}$ (2H)

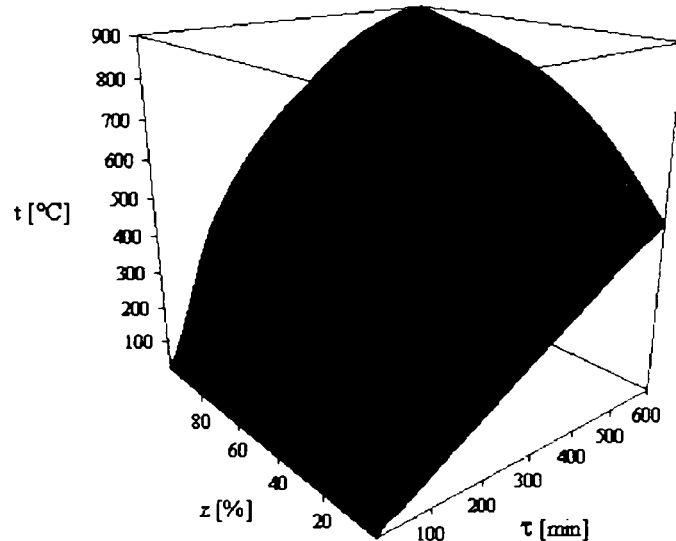


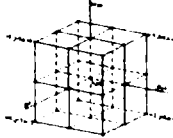






Fig.6.22. Variația temperaturii în timp după axa Oz, pentru $x = \frac{1}{4}$ (2B) și $y = \frac{1}{4}$ (2L)

6.4.2. Evoluția temperaturii punctelor caracteristice

Prin noțiunea de *puncte caracteristice* am definit acele puncte ale rețelei de discretizare din domeniul modelului numeric cu diferențe finite, care corespund punctelor din domeniul real (blumul) în care au fost introduse termocupurile pentru măsurare temperaturii în instalația experimentală pilot.

Corespondența coordonatelor punctelor caracteristice cu punctele de măsurare în blumul experimental sunt prezentate în tabelul următor.

Tabelul 6.3

Coordonatele punctelor caracteristice	Punctele de măsurare din interiorul blumului
	
$x = (2B)/48, y = (2L)/4, z = 17(2B)/18$	 1
$x = (2B)/4, y = (2L)/4, z = (2H)/4$	 2
$x = (2B)/2, y = (2L)/2, z = (2H)/2$	 3
$x = 3(2B)/4, y = 3(2L)/4, z = 3(2H)/4$	 4
$x = 43(2B)/48, y = 19(2L)/20, z = (2H)/18$	 5

Evoluția în timp a temperaturii nodurilor caracteristice poate fi urmărită cu ajutorul ferestrei de dialog MetalGraphFive, prezentată în figura de mai jos.

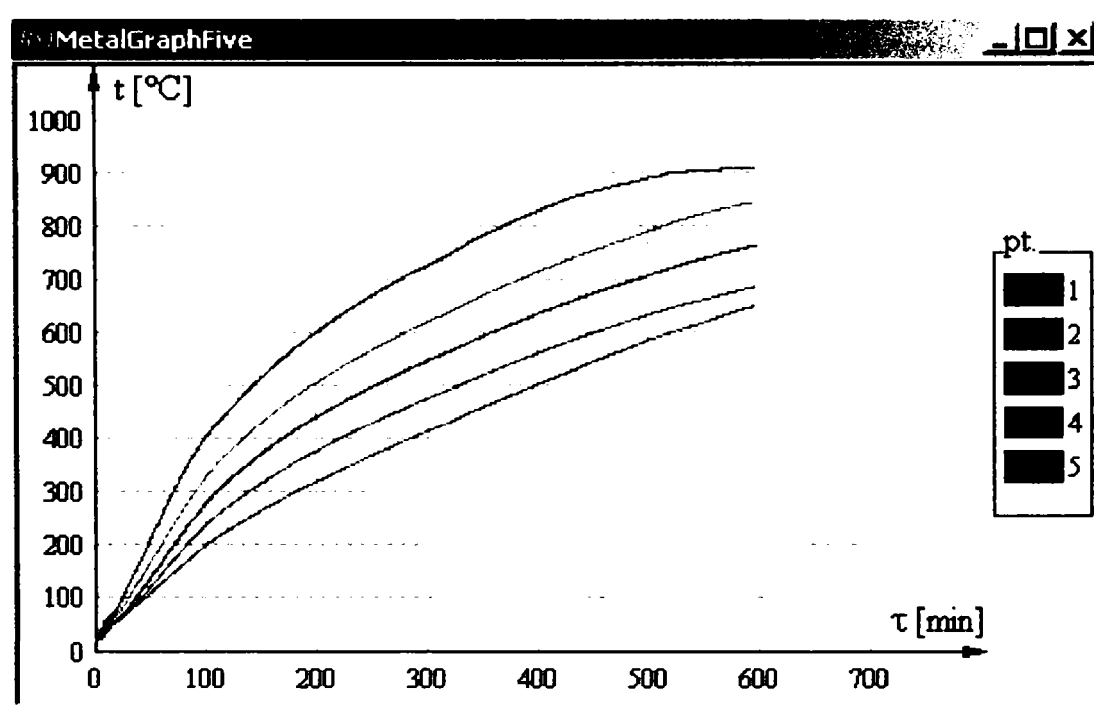


Fig.6.23. Prezentarea grafică a evoluției temperaturii punctelor caracteristice

Datele obținute pot fi exportate în scopul comparării lor cu valorile obținute prin măsurători experimentale.

6.5. Concluzii

După cum am arătat încă din introducere, unul din obiectivele prezentei teze de doctorat îl reprezintă validarea metodologiei numerice folosite și, în consecință, a rezultatelor numerice, prin rezultatele experimentale.

Această validare, în cazul metodei cu diferențe finite, presupune o comparație între valorile temperaturilor obținute prin rezolvarea modelului numeric construit cu ajutorul mediului de programare C++ și valorile determinate experimental.

Având la dispoziție doar curbele experimentale de variație a temperaturii în punctele de măsurare, conform figurii 4.7, compararea rezultatelor se va face doar pentru valorile corespunzătoare acestor puncte.

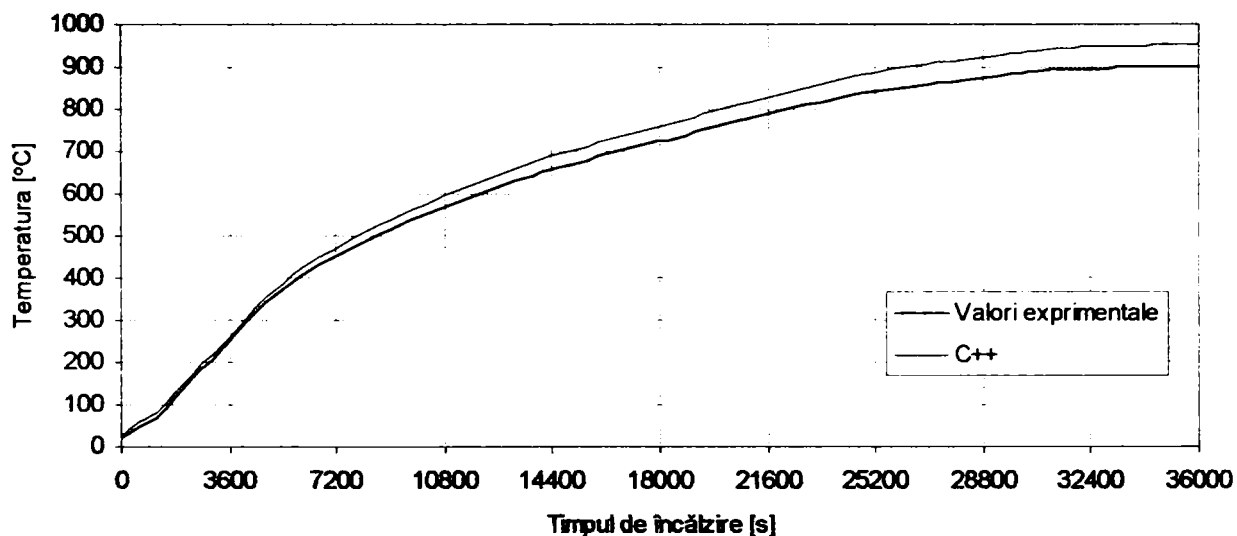


Fig.6.24. Rezultate comparative corespunzătoare punctului 1 de măsurare

În figura 6.24 s-au reprezentat comparativ variațiile în timp ale temperaturii obținute prin măsurători experimentale și calculate numeric cu C++, pentru punctul caracteristic nr.1. Semnificația graficelor din figurile 6.25, 6.26, 6.27 și 6.28 este aceeași, dar pentru punctele caracteristice 2, 3, 4, și respectiv 5.

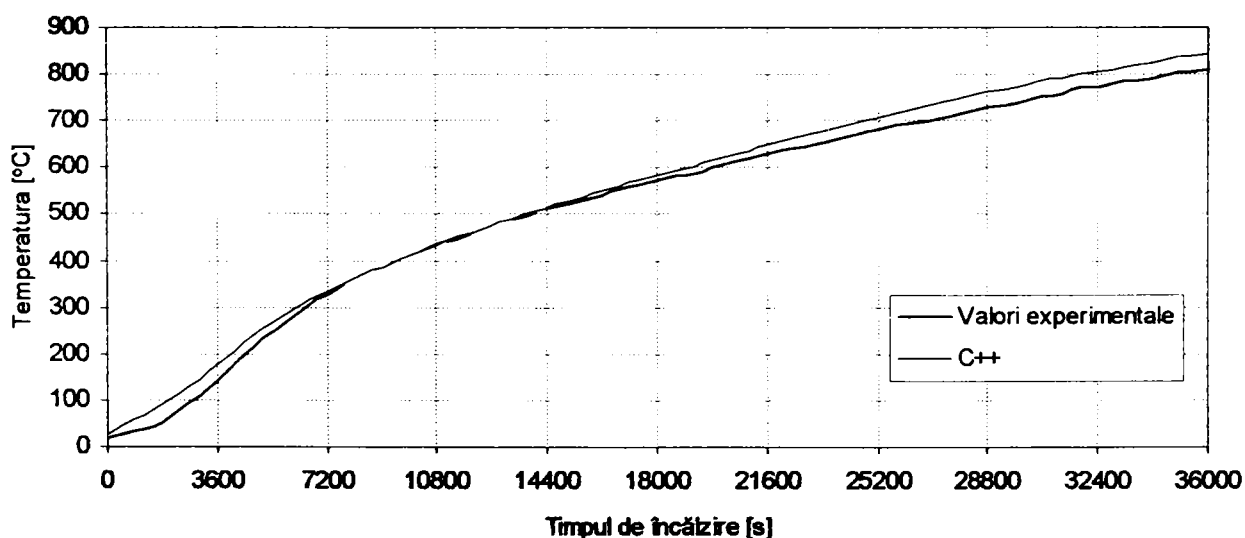


Fig.6.25. Rezultate comparative corespunzătoare punctului 2 de măsurare

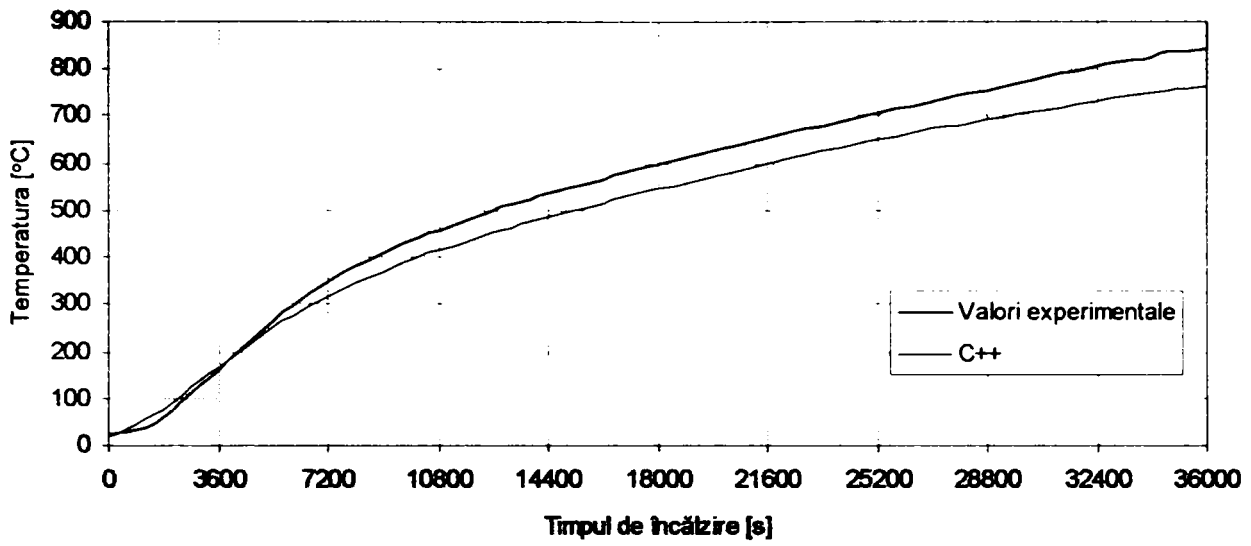


Fig.6.26. Rezultate comparative corespunzătoare punctului 3 de măsurare

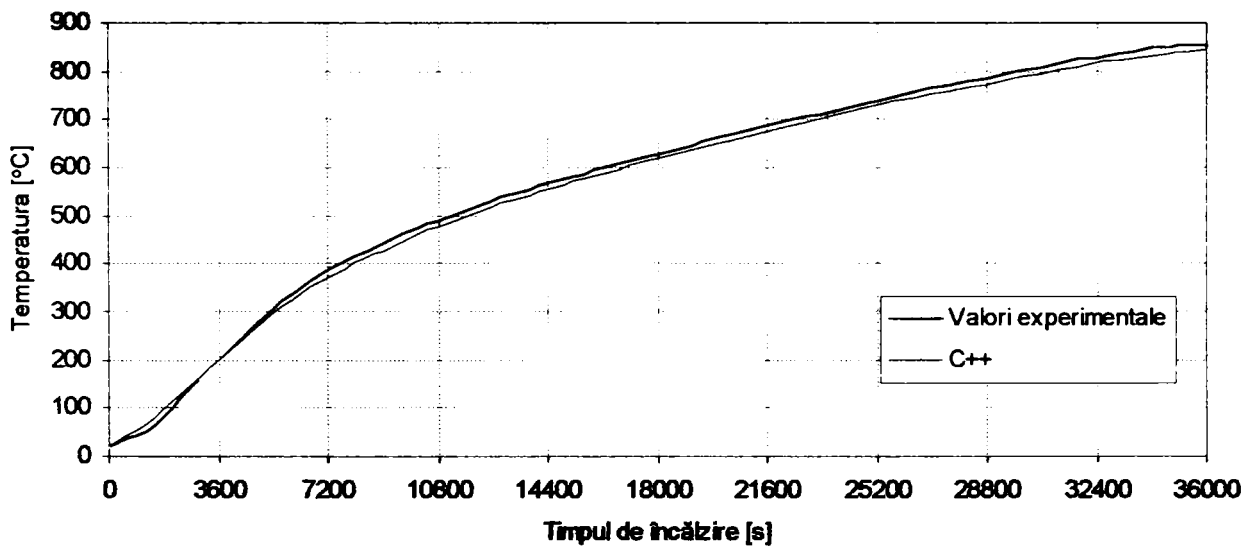


Fig.6.27. Rezultate comparative corespunzătoare punctului 4 de măsurare

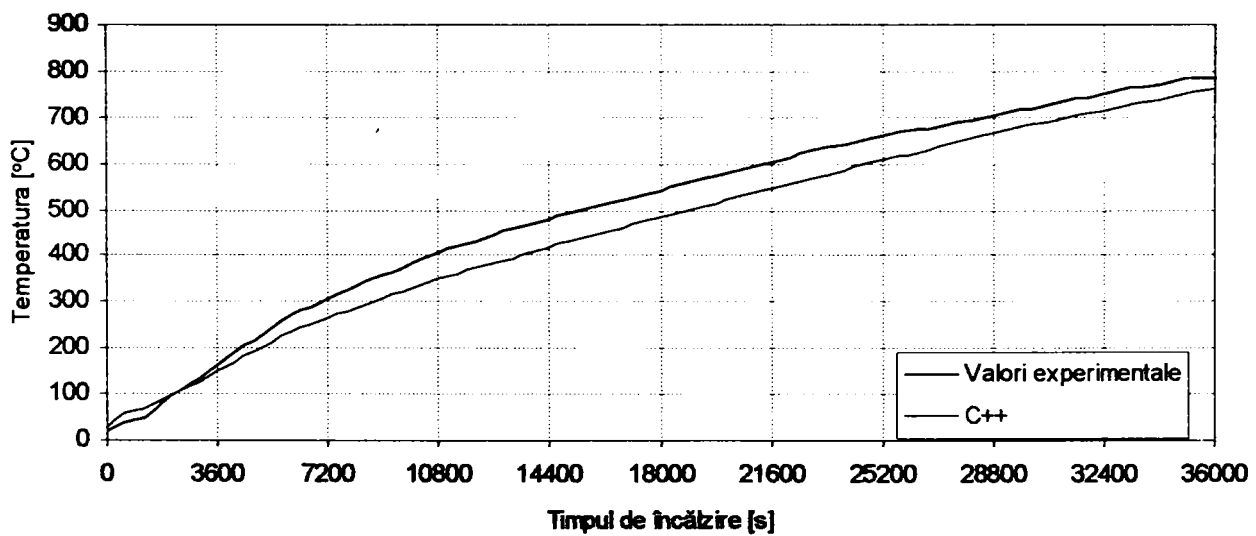


Fig.6.28. Rezultate comparative corespunzătoare punctului 5 de măsurare

În figura 6.29 s-a reprezentat, pentru întreg intervalul de încălzire, diferența efectivă între valorile măsurate experimental, considerate etalon și cele obținute numeric cu C++, pentru toate cele cinci puncte de măsurare (respectiv puncte caracteristice). În plus, figura 6.30 prezintă eroarea relativă a valorilor calculate numeric cu C++ față de cele măsurate experimental, pentru aceleași cinci puncte. Modul de calcul al acestor valori este următorul:

⇒ Pentru diferența efectivă: $\Delta t = t_{exp} - t_{C++} \text{ [}^\circ\text{C]}$

⇒ Pentru eroarea relativă: $\epsilon_r = \frac{t_{exp} - t_{C++}}{t_{exp}} \cdot 100 \text{ [%]}$

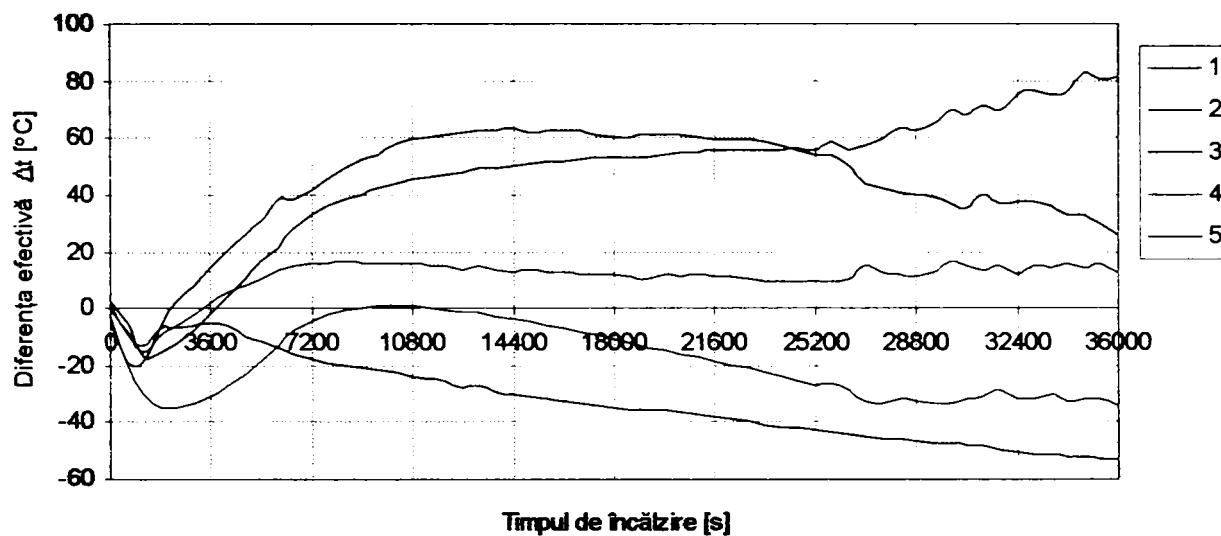


Fig.6.29. Diferența efectivă între valorile măsurate experimental și cele obținute cu C++

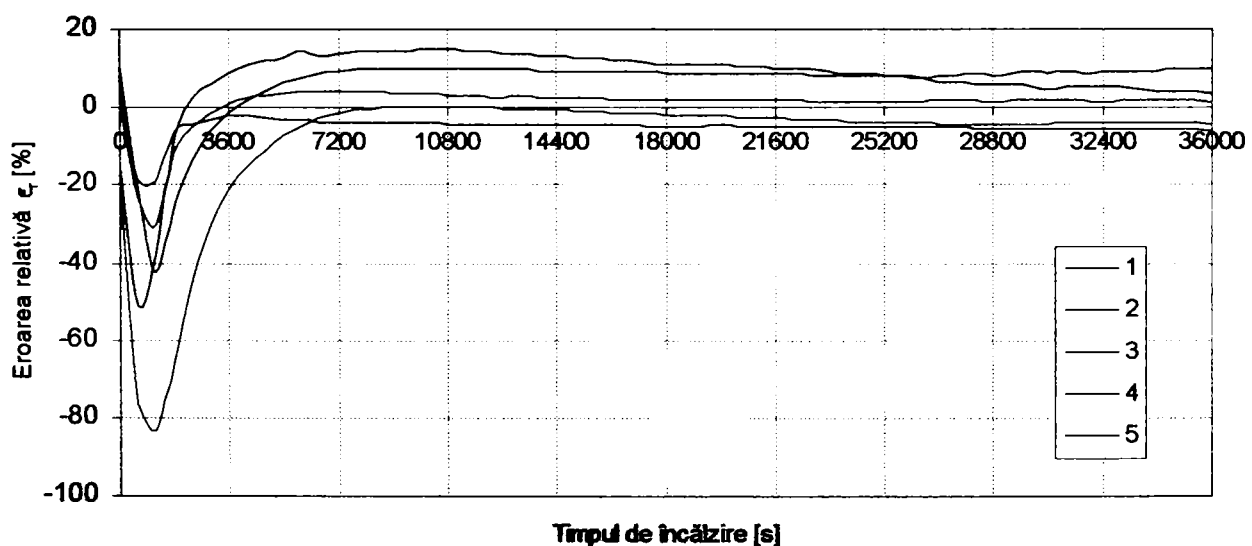


Fig.6.30. Erorile relative ale valorile obținute cu C++ față de cele măsurate experimental

□ Relativ la dinamica variației temperaturii măsurate experimental în cele cinci puncte, putem spune că valorile calculate cu programul C++ în toate cele cinci noduri caracteristice, corespunzătoare punctelor de măsurare, urmăresc fidel această variație.

□ Per ansamblu, diferențele, în valoare absolută (figura 6.29), între valorile experimentale și cele furnizate de C++ sunt mai mari pentru punctele caracteristice corespunzătoare 3 și 5. Precizia cea mai bună s-a obținut pentru punctul caracteristic 4.

□ La începutul perioadei de încălzire valorile calculate cu C++ sunt mai mari decât cele măsurate experimental, urmând ca după această perioadă situația să se inverseze, excepție făcând punctul 1, pentru care valorile calculate sunt mai mari pe toată perioada de încălzire. Explicația constă în faptul că modelul numeric nu ia în calcul efectul radiației termice la încălzirea virtuală a blumului, în schimb coeficienții de convecție adoptați prezintă valori neconforme cu realitatea în virtutea simplificării construirii modelului numeric.

□ Conform figurii 6.30 se poate spune că, în prima oră de încălzire, erorile relative ale modelului numeric sunt destul de mari, atingând chiar procente de 80 %, urmând ca în următoarele etape ale încălzirii acestea să se micșoreze semnificativ, încadrându-se într-o plajă acceptabilă de 10%.

□ Ca și o concluzie generală relativ la valorile de temperatură obținute cu modelul numeric implementat în C++, se poate spune că acesta furnizează date corecte, comparativ cu cele măsurate experimental și considerate de referință. În perioada de început a încălzirii însă, modelul este deficitar, valorile calculate în acest interval fiind eronate. Pentru remedierea acestui neajuns ar trebui pusă în discuție considerarea unui coeficient global de schimb de căldură variabil cu temperatura. Pentru cea mai mare parte a domeniului de temperaturi considerat, simularea numerică se dovedește a fi o unealtă eficientă în studiul fenomenului de transfer termic.

Capitolul 7

Modelul numeric cu elemente finite

- 7.1. Considerații teoretice
- 7.2. Implementarea modelului numeric cu elemente finite
 - 7.2.1. Etapele obținerii modelului numeric
 - 7.2.2. Preprocesarea
 - 7.2.3. Procesarea
 - 7.2.4. Postprocesarea
- 7.3. Prezentarea rezultatelor
 - 7.3.1. Câmpul termic
 - 7.3.2. Tensiunile interne datorate gradientilor termici
 - 7.3.3. Deformațiile produse datorită încălzirii
 - 7.3.4. Evoluția temperaturii nodurilor caracteristice
- 7.4. Concluzii

Capitolul 7

Modelul numeric cu elemente finite

7.1. Considerații teoretice

Pentru început să considerăm cazul cel mai general: transferul de căldură *tridimensional*, printr-un mediu *neomogen* și *anizotrop*, în regim *nestaționar*. Ecuația care descrie un astfel de fenomen este (2.13) și anume:

$$\frac{\partial t}{\partial \tau} = \frac{1}{\rho c} \left[\frac{\partial}{\partial x} \left(\lambda_x \frac{\partial t}{\partial x} \right) + \frac{\partial}{\partial y} \left(\lambda_y \frac{\partial t}{\partial y} \right) + \frac{\partial}{\partial z} \left(\lambda_z \frac{\partial t}{\partial z} \right) \right] + \frac{\dot{q}_v}{\rho c} \quad (7.1)$$

cu precizarea că $\lambda_x, \lambda_y, \lambda_z$ sunt conductivitățile termice după direcțiile x, y și respectiv z , în $[W/(m \cdot \text{grad})]$, celelalte mărimi care intervin, fiind deja prezentate.

Ecuația Fourier (7.1) descrie fenomenul conducției căldurii în forma cea mai generală. Pentru particularizarea unor cazuri concrete se impune adăugarea *condițiilor la limită* (condiții de unicitate) [46].

Condițiile la limită de timp (temporale) stabilesc distribuția de temperatură la începutul procesului considerat și se mai numesc *condiții inițiale*. Ele se scriu sub forma:

$$t = t_0, \text{ pentru } \tau = 0 \quad (7.2)$$

unde $t(x,y,z,\tau)$ este funcția de temperatură necunoscută, iar t_0 este temperatura inițială a domeniului de analiză.

Condițiile la limită spațiale (figura 7.1) stabilesc parametrii schimbului de căldură pe frontierele domeniului studiat. Notăm cu V volumul acestuia și cu S suprafața exterioară.

Condițiile la limită spațiale de tipul întâi: impunerea unei temperaturi (cunoscută) pe suprafața S_T a domeniului:

$$t = g(x, y, z), \forall M(x, y, z) \in S_T \quad (7.3)$$

unde $g(x,y,z)$ reprezintă distribuția spațială a temperaturii pe frontiera S_T .

Condițiile la limită spațiale de tipul al doilea: impunerea unei densități de flux termic \dot{q} schimbat de suprafața S_q a domeniului cu mediul ambiant:

$$\lambda_x \frac{\partial t}{\partial x} n_x + \lambda_y \frac{\partial t}{\partial y} n_y + \lambda_z \frac{\partial t}{\partial z} n_z + \dot{q} = 0, \forall N(x, y, z) \in S_q \quad (7.4)$$

în care n_x, n_y, n_z sunt cosinusurile directoare ale normalei la suprafața S_q .

Condițiile la limită spațiale de tipul al treilea: impunerea coeficientului de convecție α_c între peretele solid și un fluid cu temperatura T_α , care spală suprafața S_α :

$$\lambda_x \frac{\partial t}{\partial x} n_x + \lambda_y \frac{\partial t}{\partial y} n_y + \lambda_z \frac{\partial t}{\partial z} n_z + \alpha_c (t - t_\alpha) = 0, \forall P(x, y, z) \in S_\alpha \tag{7.5}$$

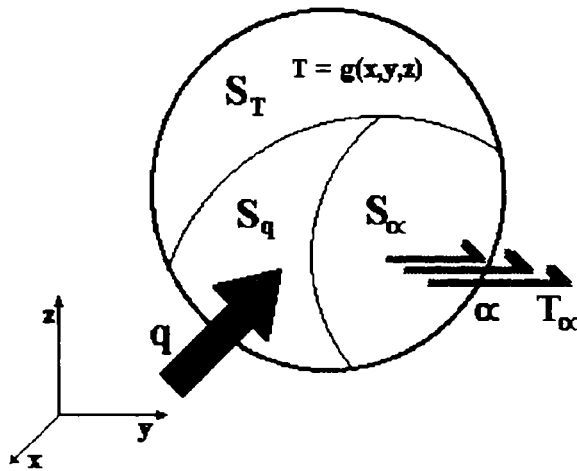


Fig.7.1. Condițiile la limită spațiale

De precizat că:

$$S_T \cup S_q \cup S_\alpha = S \tag{7.6}$$

astfel încât:

$$S_T \cap S_q \cap S_\alpha = \emptyset \tag{7.7}$$

Integrarea ecuației diferențiale generale a conducerii termice (7.1) în condițiile la limită corespunzătoare, arătate mai sus, permite determinarea temperaturii în orice punct al corpului și în orice moment. Integrarea este posibilă în cazul problemelor simple, ca de exemplu conducția prin pereți plani, cilindrici, sferici, simpli și compuși. Operația este dificilă în cazul corpurilor cu forme geometrice complicate, din materiale neomogene și anizotrope. În cele mai multe probleme practice, concrete, integrarea analitică este dificilă sau chiar imposibilă [15].

Soluționarea acestor probleme se face pe cale numerică. în abordarea variantei cu elemente finite se recurge la calculul variațional.

La modul general, calculul variațional își propune găsirea unei funcții $y(x)$ pentru care integrala unei alte funcții $f(x, y, y')$ să aibă o valoare maximă sau minimă. Funcția $f(x, y, y')$ se numește *funcție argument*, iar integrala:

$$J = \int_{x_1}^{x_2} f(x, y, y') dx \tag{7.8}$$

se numește *funcțională*. Funcția căutată $y(x)$ se numește *extremală*. Funcția argument depinde de variabila independentă x , de funcția pe care o căutăm $y(x)$ și de derivata acesteia $y'(x)$ [47].

Revenind la ecuația diferențială a conducerii (7.1), integrarea acesteia, în condițiile la limită (7.2), (7.3), (7.4) și (7.5) este echivalentă cu minimizarea funcționalei:

$$J = \int_V \frac{1}{2} \left[\lambda_x \left(\frac{\partial t}{\partial x} \right)^2 + \lambda_y \left(\frac{\partial t}{\partial y} \right)^2 + \lambda_z \left(\frac{\partial t}{\partial z} \right)^2 \right] dV + \int_V \left(\rho c \frac{\partial t}{\partial \tau} - \dot{q}_v \right) t dV + \int_{S_q} \dot{q} t dS + \int_{S_\alpha} \alpha_c t \left(\frac{1}{2} T - T_\alpha \right) dS \tag{7.9}$$

Toate mărimile care intervin în relația (7.9) au fost definite. Convenția de semne pentru \dot{q} și \dot{q}_v este următoarea: \dot{q} este pozitiv când pătrunde în corp dinspre mediul ambiant și negativ când iese din corp spre mediul exterior; \dot{q}_v este pozitiv dacă sursele interioare debitează căldură și negativ dacă absorb căldură.

Deoarece funcția convențională prin care se aproximează temperatura nu este continuă pe toată structura, ci numai pe cuprinsul elementelor finite, integrala din relația (7.9) trebuie efectuată pe elemente finite[12].

Pentru ca expunerea să fie cât mai clară, se prezintă cazul unui câmp bidimensional, staționar. Pentru o astfel de problemă, funcționala (7.9) devine:

$$J = \int_V \frac{1}{2} \left[\lambda_x \left(\frac{\partial t}{\partial x} \right)^2 + \lambda_y \left(\frac{\partial t}{\partial y} \right)^2 \right] dV - \int_V \dot{q}_v t dV + \int_{S_q} \dot{q} t dS + \int_{S_a} \alpha_c t \left(\frac{1}{2} t - t_a \right) dS \quad (7.10)$$

Dacă se consideră domeniul plan de analiză raportat la un sistem cartezian de axe de coordonate, discretizat în triunghiuri, având vârfurile notate $i(x_i, y_i)$, $j(x_j, y_j)$, $m(x_m, y_m)$, atunci ecuația matricială (3.21) se va scrie:

$$\{ \tilde{t}(x, y) \}^{(e)} = \{ N_i(x, y) N_j(x, y) N_m(x, y) \} \begin{Bmatrix} t_i \\ t_j \\ t_m \end{Bmatrix} = [N] \{ t \}_{(e)} \quad (7.11)$$

unde t_i , t_j , t_m sunt temperaturile în vârfurile triunghiurilor de discretizare, iar $\tilde{t}(x, y)$ este valoarea aproximată a temperaturii în câmpul elementului finit (e).

Dacă se adoptă o lege liniară de variație a temperaturii în câmpul elementului finit (e), atunci funcțiile de formă vor avea expresiile:

$$\begin{aligned} N_i(x, y) &= (a_i + b_i x + c_i y) / 2V^{(e)} \\ N_j(x, y) &= (a_j + b_j x + c_j y) / 2V^{(e)} \\ N_m(x, y) &= (a_m + b_m x + c_m y) / 2V^{(e)} \end{aligned} \quad (7.12)$$

unde:

$$\begin{aligned} a_i &= x_j y_m - x_m y_j ; \quad b_i = y_j - y_m ; \quad c_i = x_m - x_j \\ a_j &= x_m y_i - x_i y_m ; \quad b_j = y_m - y_i ; \quad c_j = x_i - x_m \\ a_m &= x_i y_j - x_j y_i ; \quad b_m = y_i - y_j ; \quad c_m = x_j - x_i \end{aligned} \quad (7.13)$$

Se va scrie acum funcționala J din (7.10) pe elementul finit curent (e).

$$J^{(e)} = \int_{V^{(e)}} \frac{1}{2} \left[\lambda_x \left(\frac{\partial \tilde{t}}{\partial x} \right)^2 + \lambda_y \left(\frac{\partial \tilde{t}}{\partial y} \right)^2 \right] dV - \int_{V^{(e)}} \dot{q}_v \tilde{t} dV + \int_{S_q^{(e)}} \dot{q} \tilde{t} dS + \int_{S_\alpha^{(e)}} \alpha_c \left(\frac{\tilde{t}^2}{2} - \tilde{t} \cdot t_\alpha \right) dS \quad (7.14)$$

În (7.14) intervin și derivatele parțiale ale temperaturii. Derivând (7.11) în raport cu x și y se obține relația:

$$\{p\} = \begin{Bmatrix} \frac{\partial \tilde{t}}{\partial x} \\ \frac{\partial \tilde{t}}{\partial y} \end{Bmatrix} = \begin{bmatrix} \frac{\partial N_i}{\partial x} & \frac{\partial N_j}{\partial x} & \frac{\partial N_m}{\partial x} \\ \frac{\partial N_i}{\partial y} & \frac{\partial N_j}{\partial y} & \frac{\partial N_m}{\partial y} \end{bmatrix} \begin{Bmatrix} t_i \\ t_j \\ t_m \end{Bmatrix} = [B] \{t\}_{(e)} \quad (7.15)$$

Matrice [B] conține derivatele funcțiilor de formă în raport cu coordonatele. Dacă se notează matricea coeficienților de conductivitate termică

$$[D] = \begin{bmatrix} \lambda_x & 0 \\ 0 & \lambda_y \end{bmatrix} \quad (7.16)$$

funcționala (7.14) se scrie sub forma:

$$J^{(e)} = \int_{V^{(e)}} \frac{1}{2} ([B] \{t\}_{(e)})^T [D] [B] \{t\}_{(e)} dV - \int_{V^{(e)}} \dot{q}_v [N] \{t\}_{(e)} dV + \int_{S_q^{(e)}} \dot{q} [N] \{t\}_{(e)} dS + \int_{S_\alpha^{(e)}} \frac{\alpha_c}{2} ([N] \{t\}_{(e)})^2 dS - \int_{S_\alpha^{(e)}} \alpha_c t_\alpha [N] \{t\}_{(e)} dS \quad (7.17)$$

Deoarece:

$$([B] \{t\}_{(e)})^T = \{t\}_{(e)}^T [B]^T \text{ și} \\ ([N] \{t\}_{(e)})^2 = ([N] \{t\}_{(e)})^T ([N] \{t\}_{(e)}) = \{t\}_{(e)}^T [N]^T [N] \{t\}_{(e)} \quad (7.18)$$

ecuația (7.17) se scrie astfel:

$$J^{(e)} = \int_{V^{(e)}} \frac{1}{2} \{t\}_{(e)}^T [B]^T [D] [B] \{t\}_{(e)} dV - \int_{V^{(e)}} \dot{q}_v [N] \{t\}_{(e)} dV + \int_{S_q^{(e)}} \dot{q} [N] \{t\}_{(e)} dS + \int_{S_\alpha^{(e)}} \frac{\alpha_c}{2} \{t\}_{(e)}^T [B]^T [D] [B] \{t\}_{(e)} dS - \int_{S_\alpha^{(e)}} \alpha_c t_\alpha [N] \{t\}_{(e)} dS \quad (7.19)$$

Aplicând regulile de derivare a expresiilor matriciale și ținând seama că $dV = h dA$ și $dS = h dl$, unde h este grosimea elementului finit, dA elementul diferențial de arie și dl elementul diferențial de lungime, rezultă:

$$\frac{\partial J^{(e)}}{\partial \{t\}_{(e)}} = h \left(\int_{A^{(e)}} [B]^T [D] [B] dA + \int_{l_q^{(e)}} \alpha_c [N]^T [N] dl \right) \{t\}_{(e)} - h \int_{A^{(e)}} \dot{q}_v [N]^T dA +$$

$$+ h \int_{l_q^{(e)}} \dot{q} [N]^T dl - h \int_{l_\alpha^{(e)}} \alpha_c t_\alpha [N]^T dl \quad (7.20)$$

Grosimea h a elementului finit s-a considerat constantă. La problemele plane ea se ia egală cu unitatea.

Concentrat, (7.20) se scrie:

$$\frac{\partial J^{(e)}}{\partial \{t\}_{(e)}} = [k] \{t\}_{(e)} - \{f\} \quad (7.21)$$

în această relație:

$$[k] = h \int_{A^{(e)}} [B]^T [D] [B] dA + h \int_{l_q^{(e)}} \alpha_c [N]^T [N] dl \quad (7.22)$$

este matricea de caracteristici fizice a elementului finit, în care intervin mărimi fizice ale transferului de căldură prin conducție în interiorul elementului finit și prin convecție pe conturul acestuia. Prima integrală corespunde conducției în corp, iar a doua corespunde convecției pe latura $l_\alpha^{(e)}$ a suprafeței $S_\alpha^{(e)}$ a elementului finit considerat (figura 7.2). Dacă nu există schimb de căldură prin convecție pe contur, $\alpha_c = 0$ și a doua integrală nu există. Această matrice se mai numește și *matrice de rigiditate* prin analogie cu denumirea folosită la probleme de elasticitate.

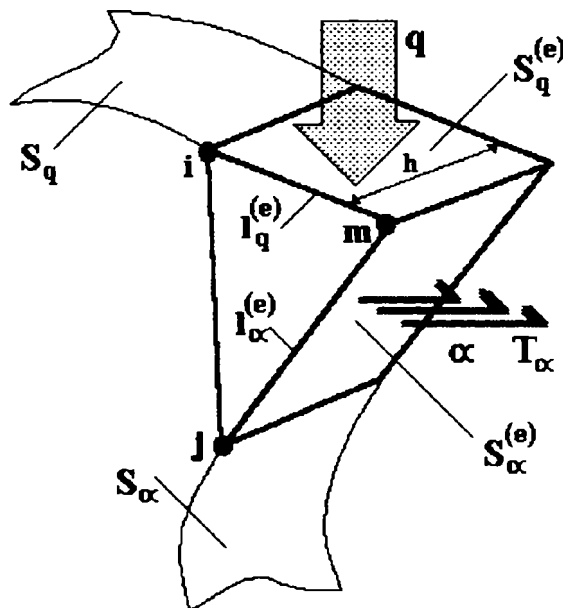


Fig.7.2. Elementul finit cu schimb de căldură prin flux termic și convecție

De asemenea, în (7.21) vectorul $\{f\}$:

$$\{f\} = h \int_{A^{(e)}} q_v [N]^T dA - h \int_{l_q^{(e)}} q [N]^T dl + h \int_{l_\alpha^{(e)}} \alpha_c t_\alpha [N]^T dl \quad (7.23)$$

reprezintă *vectorul încărcărilor*, în cazul de față încărcări termice.

- Prima integrală din (7.23) reprezintă contribuția surselor interioare de căldură \dot{q}_v pe elementul finit de volum $V^{(e)} = hA^{(e)}$.
- A doua integrală din (7.23) reprezintă contribuția fluxului termic unitar pe suprafața elementului finit $S_q^{(e)} = h_l A^{(e)}$.
- A treia integrală din (7.23) reprezintă contribuția convecției termice cu parametrii α_c și t_α pe suprafața $S_\alpha^{(e)}$.

Minimizarea funcționalei J se obține anulând derivata în raport cu temperaturile din toate nodurile structurii. ținând seama de (3.23) se obține:

$$\frac{\partial J}{\partial \{t\}} = \frac{\partial}{\partial \{t\}} \sum J^{(e)} = \sum \frac{\partial J^{(e)}}{\partial \{t\}_{(e)}} = 0 \quad (7.24)$$

Din (7.21) și (7.24) se obține sistemul de ecuații al structurii:

$$\begin{aligned} \sum ([k]\{t\}_{(e)} - \{f\}) &= 0 \quad \text{sau} \\ (\sum [k])\{t\}_{(e)} &= \sum \{f\} \quad \text{sau} \\ [K]\{t\} &= \{F\} \end{aligned} \quad (7.25)$$

în care:

$[K] = \sum [k]$ - reprezintă *matricea de rigiditate (conductivitate)* a întregului domeniu,

$\{F\} = \sum \{f\}$ - este *vectorul încărcărilor termice* pentru toată structura,

$\{t\}$ - este *vectorul temperaturilor necunoscute* pentru toată structura.

Ecuația (7.25) reprezintă forma cu elemente finite a ecuației diferențiale a conducției termice. Reprezintă un sistem de atâtea ecuații câte noduri cu temperatură necunoscută are secțiunea analizată, prin a cărei rezolvare se determină aceste temperaturi.

Matricea $[K]$ și vectorul $\{F\}$ din (7.25) se constituie prin asamblarea matricelor $[k]$ și a vectorilor $\{f\}$ corespunzătoare elementelor finite, pe măsura alcătuirii lor.

Expresiile matricei de conductivitate și a vectorului încărcării termice se pot determina în funcție de tipul elementului finit ales.

Dacă ne propunem studierea regimurilor nestaționare, comparând funcționala (7.9), valabilă pentru cazul general de transfer de căldură, cu funcționala (7.10) din cazul regimului staționar, se observă că, deosebirea dintre cele două funcționale (menținând analiza în cadrul problemelor plane) constă în faptul că, în locul termenului: $-\int_V \dot{q}_v \, tdV$ de la regimul staționar (7.10), apare termenul:

$$\int_V \left(\rho c \frac{\partial t}{\partial \tau} - \dot{q}_v \right) tdV = -\int_V \dot{q}_v \, tdV + \int_V \rho c \frac{\partial t}{\partial \tau} \, tdV \quad (7.26)$$

la regimul nestaționar. Deci, pentru trecerea de la regimul staționar la cel tranzitoriu trebuie adăugată contribuția rezultată prin minimizarea termenului:

$$j = \int_V \rho c \frac{\partial t}{\partial \tau} t dV \quad (7.27)$$

Deoarece minimizarea se face pe elemente finite, pentru un element finit vom avea:

$$j^{(e)} = \int_{V^{(e)}} (\rho c)^{(e)} \frac{\partial \tilde{t}}{\partial \tau} \tilde{t} dV \quad (7.28)$$

Exprimând temperatura \tilde{t} prin intermediul funcțiilor de formă conform (7.11), în care $[N]$ este independentă de timpul τ , se obține:

$$\frac{\partial \tilde{t}}{\partial \tau} = [N] \frac{\partial \{t\}_{(e)}}{\partial \tau} \quad (7.29)$$

Introducând (7.11) și (7.29) în (7.28) rezultă:

$$j^{(e)} = \int_{V^{(e)}} (\rho c)^{(e)} [N]^T \{t\}_{(e)} [N] dV \frac{\partial \{t\}_{(e)}}{\partial \tau} \quad (7.30)$$

Pentru a minimiza pe $j^{(e)}$ este necesar ca $\frac{\partial j^{(e)}}{\partial \{t\}_{(e)}} = 0$, ceea ce conduce la:

$$\left(\int_{V^{(e)}} (\rho c)^{(e)} [N]^T [N] dV \right) \frac{\partial \{t\}_{(e)}}{\partial \tau} = 0 \quad (7.31)$$

Matricea:

$$[C] = \int_{V^{(e)}} (\rho c)^{(e)} [N]^T [N] dV \quad (7.32)$$

se numește *matricea capacităților calorice* sau a *căldurilor specifice* pentru un element finit [15].

Deci, pentru trecerea de la regimul staționar la cel tranzitoriu, ecuația elementului finit (7.21)

trebuie completată cu termenul $[C] \frac{\partial \{t\}_{(e)}}{\partial \tau}$ devenind:

$$[C] \frac{\partial \{t\}_{(e)}}{\partial \tau} + [k] \{t\}_{(e)} - \{f\} = 0 \quad (7.33)$$

În locul unui sistem de ecuații liniare se obține un sistem de ecuații diferențiale liniare de ordinul întâi în raport cu timpul. Pentru integrarea în raport cu timpul a sistemului de ecuații diferențiale (7.33) și determinarea câmpului de temperatură tranzitoriu, se pot folosi, în principiu, trei metode: metoda diferențelor finite, metoda reziduurilor (Galerkin) și metoda elementelor finite, în care elementele au o dimensiune în plus pentru timp.

7.2. Implementarea modelului numeric cu elemente finite

7.2.1. Etapele obținerii modelului numeric

În rezolvarea numerică a problemelor de transfer termic prin metoda elementului finit se parcurg anumite etape bine determinate. Descrierea amănunțită a acestor etape a făcut obiectul subcapitolelor 3.3 și 7.1. Etapele specifice în rezolvarea problemei concrete privind determinarea câmpului termic stabilit în interiorul semifabricatului sub formă de blum, la încălzirea în incinta de lucru a instalației experimentale pilot (prezentată în capitolul 4), vor fi descrise în cele ce urmează:

- ⇒ În primul rând trebuie stabilit clar tipul problemei de transfer termic abordat. Fiind vorba despre un studiu al încălzirii în timp a semifabricatului în incinta instalației experimentale, avem de-a face cu o problemă de transfer termic nestaționar. În altă ordine de idei, luarea în considerare, pe de o parte, a unor proprietăți termo-fizice de material (conductivitate termică, căldură specifică) dependente de temperatură, iar pe de altă parte considerarea în calcule a fenomenului complex de radiație, impun o analiză neliniară a fenomenului de transfer termic;
- ⇒ Al doilea pas în construirea modelului numeric îl constituie stabilirea domeniului de analiză. Ne propunem să determinăm distribuția câmpului de temperaturi într-un corp care, din punct de vedere geometric, reprezintă un paralelipiped, cu toate cele trei dimensiuni finite. Cu alte cuvinte, domeniul de analiză este unul tridimensional și prin urmare, avem de-a face cu o distribuție tridimensională a câmpului termic studiat;
- ⇒ Al treilea pas constă în generarea geometriei domeniului de analiză studiat;
- ⇒ Pasul următor include discretizarea domeniului de analiză în elemente finite, ceea ce presupune alegerea tipului de element finit, a constantelor caracteristice, stabilirea numărului de noduri pentru fiecare element și generarea rețelei de discretizare;
- ⇒ Urmează stabilirea proprietăților termo-fizice de material, ținând cont de marca de oțel din care a fost turnat semifabricatul experimental.
- ⇒ Ultimul pas cuprinde implementarea condițiilor la limită spațiale și a celor la limită temporale (sau condiții inițiale);

Pașii enumerați mai sus fac parte din etapa de preprocesare a modelului numeric și furnizează datele de intrare pentru programul folosit.

Etapa finală, numită postprocesare include lansarea în execuție a programului și afișarea rezultatelor obținute sub formă calitativă și cantitativă [48] [49].

7.2.2. Preprocesarea

Generarea geometriei domeniului de analiză s-a realizat cu ajutorul pachetului **Mechanical Desktop 6**. Paralelipipedul, modelat ca și solid (figura 7.3) are lungimea (dimensiunea după axa Oy) de 0,30 m, lățimea (dimensiunea după axa Ox) de 0,24 m și înălțimea (dimensiunea după axa Oz) având valoarea de 0,27 m (vezi figura 4.7).

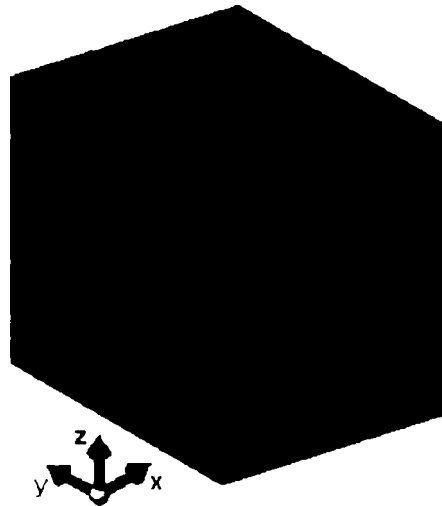


Fig.7.3. Geometria domeniului de analiză

Odată creat fișierul care conține geometria domeniului de analiză, în pasul următor acesta este exportat ca și fișier cu extensia *.sat* urmând a fi importat în modulul *Superdraw* al pachetului *ALGOR Multiphysics*. În această etapă are loc asimilarea geometriei și discretizarea domeniului de analiză (figura 7.4).

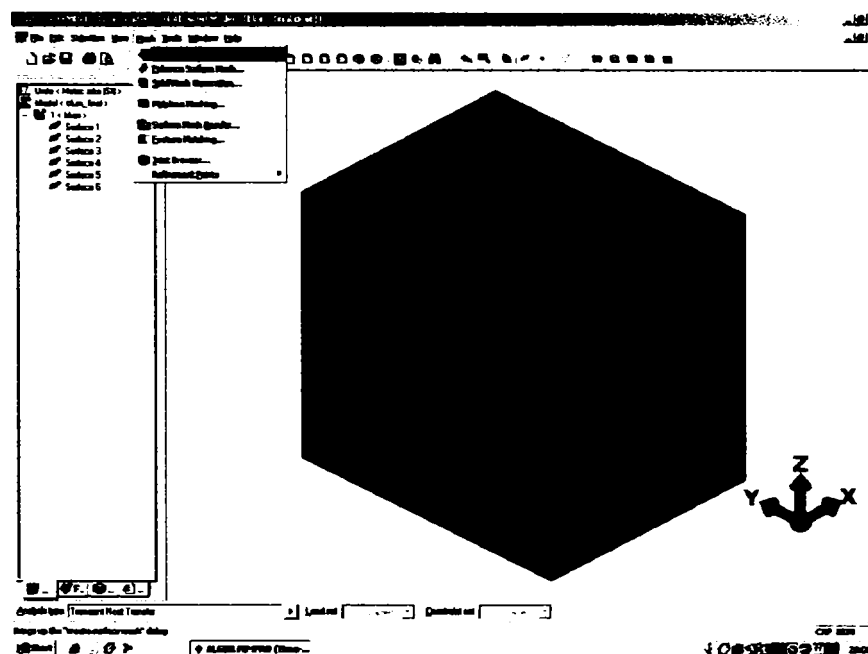


Fig.7.4. Discretizarea domeniului de analiză

După cum am mai spus, procesul de discretizare al domeniului de analiză are ca suport fizic posibilitatea descompunerii corpului analizat în elementele sale componente. Aceste elemente fizice componente, prin modelare matematică, devin *elemente finite*. În concluzie, elementele finite, definite în cadrul procesului de discretizare, apar ca rezultat al descompunerii domeniului de studiu în mai multe subdomenii compatibile și cu interior disjunct. Conexiunea acestor subdomenii se face prin intermediul nodurilor, care nu sunt altceva decât puncte selectate în domeniul considerat, la care se specifică variabilele studiate (temperaturile) sau derivate ale acestora.

Înainte de a trece la pasul următor, programul cere utilizatorului specificarea sistemului de unități de măsură utilizat în continuare. Pentru selectarea sistemului internațional de unități de măsură (SI) se deschide fereastra de dialog prezentată în figura 7.5.

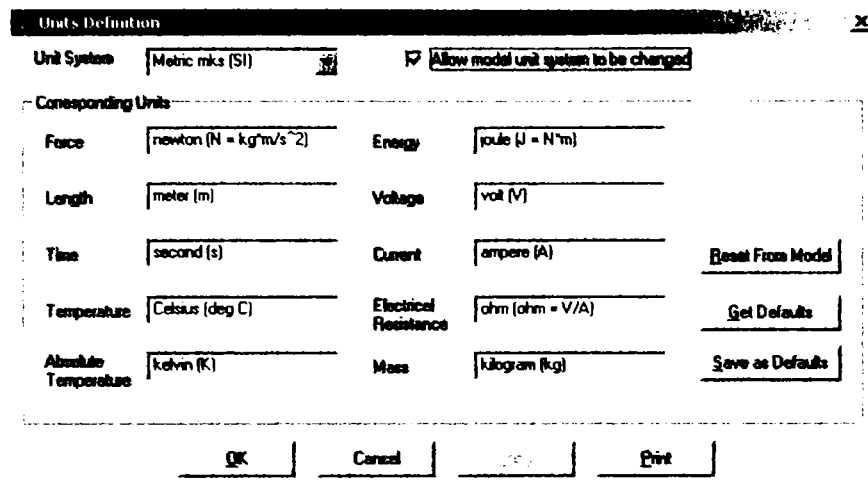


Fig.7.5. Stabilirea sistemului de unități de măsură

Alegerea tipului de element finit potrivit analizei în cauză se face ținând cont atât de cele prezentate în subcapitolul 3.3.4, cât și de lista de elemente disponibile în programul ALGOR. De cele mai multe ori, tipul elementelor finite care urmează a fi folosite derivă din tipul problemei de rezolvat și structura domeniului de analiză a acesteia. Având în vedere analiza termică tridimensională a blumului, se va alege ca tip de element finit unul tridimensional, adecvat unei analize termice neliniare, care să permită luarea în considerare a proprietăților termo-fizice de material (conductivitate termică și căldură specifică) variabile cu temperatura. Acest lucru este posibil prin alegerea opțiunii *orthotropic* din fereastra de dialog corespunzătoare definirii elementului finit de tipul „*Thermal Brick*” din figura 7.6.

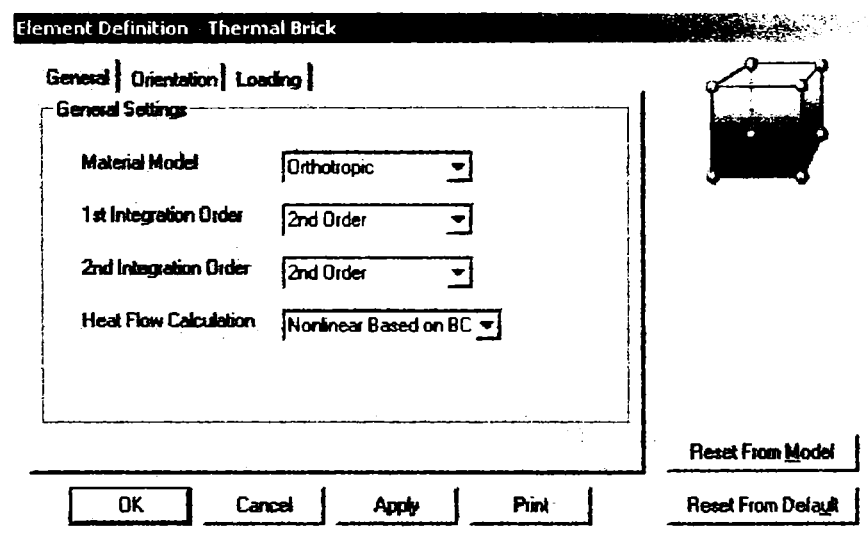


Fig.7.6. Definirea tipului de element finit

În pasul următor se asociază elementului finit stabilit anterior un tip de material. Programul pune la dispoziția utilizatorului o listă cu cele mai uzuale materiale folosite în inginerie, alături de o bibliotecă în care sunt incluse toate proprietățile fizice ale acestora. Alegerea din listă a unui tip de material presupune acceptarea ipotezei conform căreia proprietățile termo-fizice asociate sunt invariabile cu temperatura, programul utilizând valori medii pentru calcule. Această situație nu corespunde tipului de problemă ales pentru studiu și, ca urmare, va trebui definit explicit, de către utilizator, un tip de material care să permită analiza neliniară. Programul facilitează această operație prin fereastra de dialog prezentată în figura 7.7.

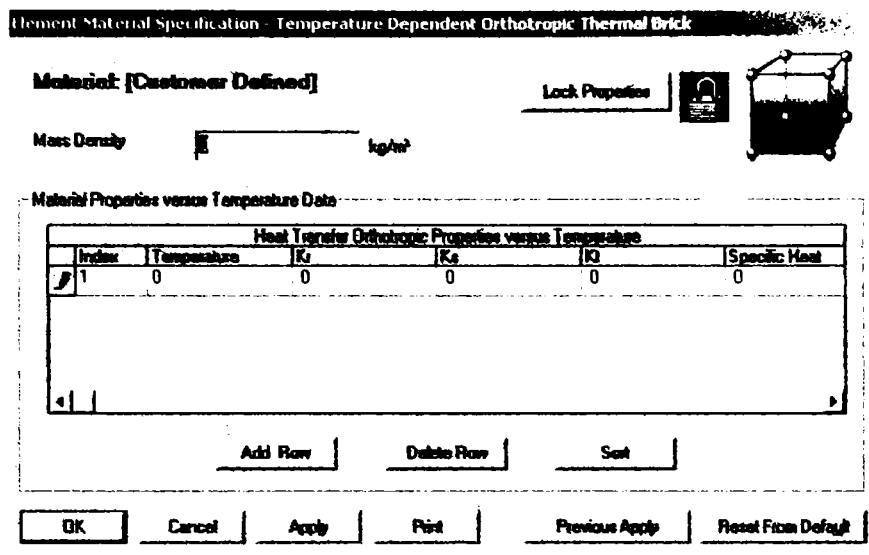


Fig.7.7. Definirea materialului asociat elementului finit și proprietățile termo-fizice funcție de temperatură

După cum se poate observa, tot în fereastra de dialog din figura 7.7 sunt definite funcțiile care descriu variațiile conductivității termice (după cele trei axe de coordonate – notate în program cu k_x , k_y , k_z) și a căldurii specifice cu temperatura. Aceste funcții de temperatură sunt reprezentate în figurile 7.8 și respectiv 7.9 [50] [51].

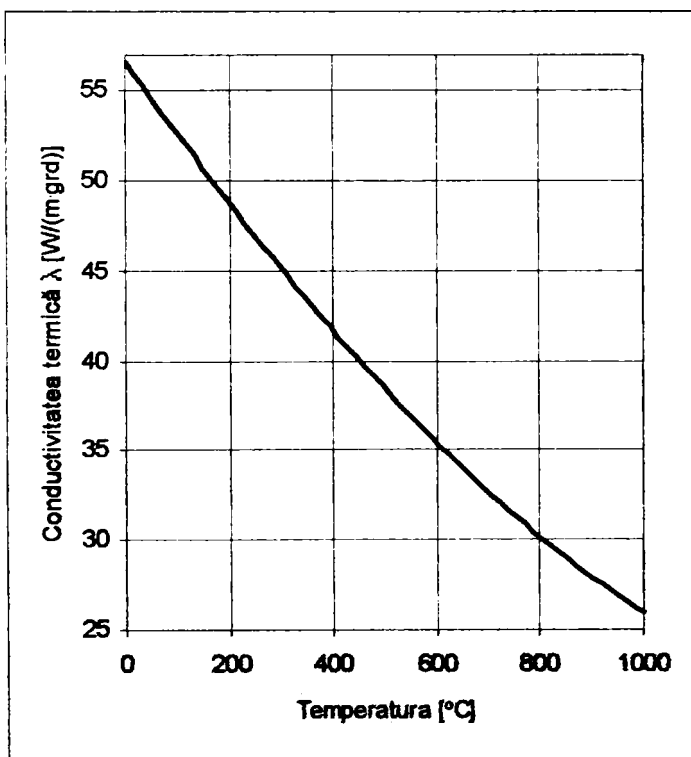


Fig.7.8. Conductivitatea termică funcție de temperatură

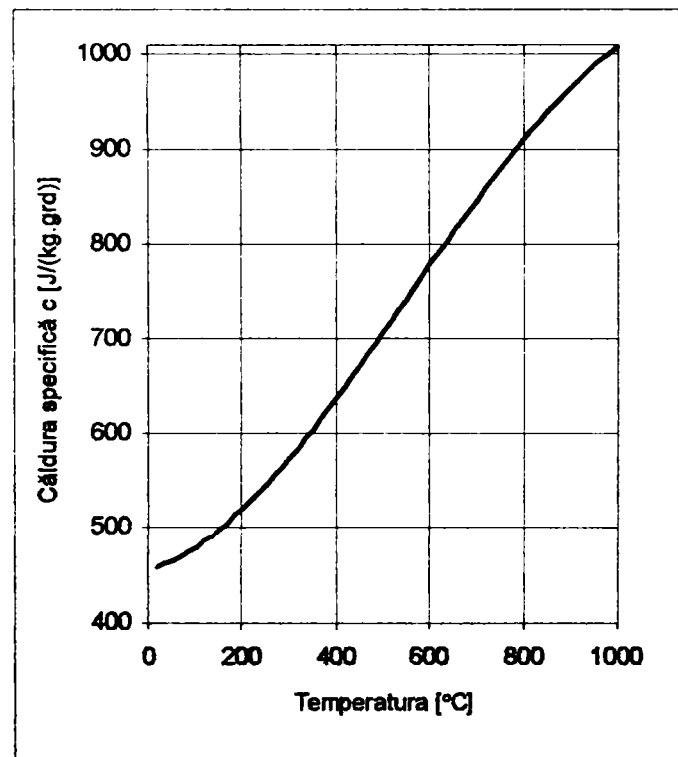


Fig.7.9. Căldura specifică funcție de temperatură

Pasul următor constă în stabilirea condițiilor la limită spațiale (de frontieră). Frontiera domeniului de analiză este alcătuită, în acest caz, din cele șase suprafețe laterale ale paralelipipedului ce reprezintă semifabricatul sub formă de blum. Deoarece acesta se găsește poziționat în interiorul spațiului de lucru al instalației experimentale, sprijinit pe vatra acesteia,

interacțiunea cu mediul gazos ce-l înconjoară se realizează doar prin intermediul a cinci suprafețe. A șasea, reprezentând baza blumului, este în contact direct cu vatra. Mai mult, cele cinci suprafețe ale blumului aflate în contact direct cu gazele arse sunt expuse în mod diferit fluxurilor termice din incintă, ceea ce presupune o încărcare termică diferită a acestora.

Astfel, suprafețele blumului aflate sub incidența directă a flăcării schimbă căldură cu mediul înconjurător predominant prin fenomenele de convecție și radiație termică a gazelor. Spre deosebire de acestea, suprafețele aflate în «umbră» primesc un flux termic datorat convecției și, spre sfârșitul perioadei de încălzire, când temperatura incintei crește semnificativ, radiației pereților camerei de lucru.

Suprafața exterioară a blumului, considerată bază de sprijin, fiind în contact nemijlocit cu materialul vetrei primește căldură de la acesta prin fenomenul de conducție termică. Într-o aproximare acceptabilă, temperatura bazei de sprijin poate urmări, relativ fidel, variația de temperatură a vetrei.

Implementarea condițiilor de frontieră se face cu ajutorul interfeței grafice FEMPRO (Finite Element Modeling, Results Evaluation and Presentation Interface) pusă la dispoziție de pachetul ALGOR (figura 7.10).

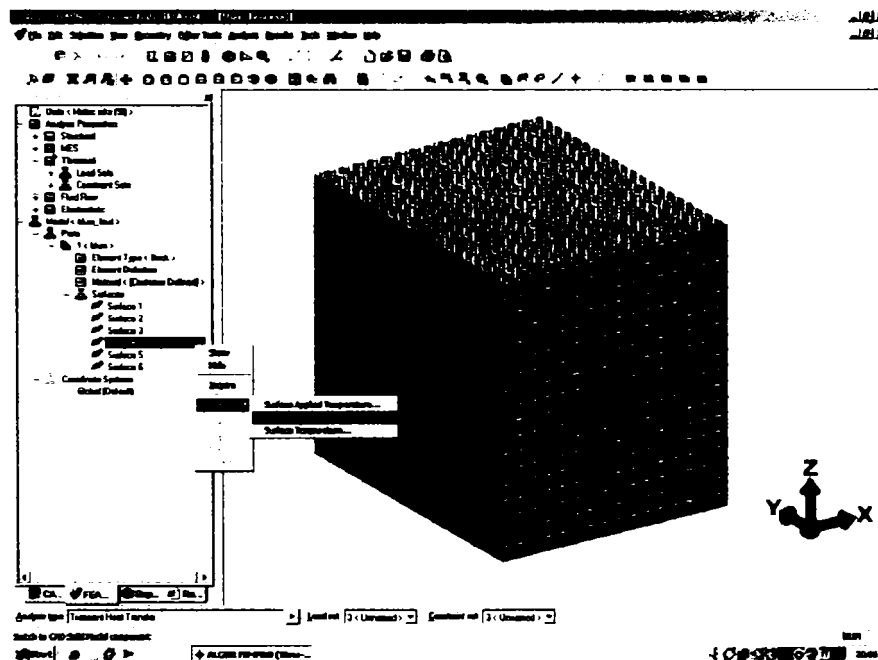


Fig.7.10. Implementarea condițiilor de frontieră cu interfața grafică FEMPRO

Încărcările termice corespunzătoare suprafețelor de frontieră ale domeniului de analiză presupun stabilirea, pe de o parte a coeficientului de transfer termic global α , între suprafața solidă și fluidul care o spală, iar pe de altă parte a modului de transfer termic prin radiație de la pereții interiori ai incintei de lucru la suprafață. Prima problemă este rezolvată direct de către program, acesta fiind capabil să calculeze automat coeficientul α pentru fiecare pas de timp. Problema radiației provenită de la pereți a fost rezolvată prin intermediul facilității oferite de programul ALGOR denumită sugestiv «body to body radiation» prin deschiderea ferestrei de dialog din figura 7.12.

Fiind vorba despre un fenomen tranzitoriu, este obligatorie introducerea curbelor care specifică modul de variație în timp a parametrilor caracteristici. Această operație este facilitată de fereastra de dialog prezentată în figura 7.11.

În final, etapa de preprocesare se încheie cu stabilirea condițiilor inițiale sau la limită de timp. Cum încălzirea blumului s-a făcut pe durata a zece ore, ne propunem ca și simularea numerică să se desfășoare tot pe acest interval.

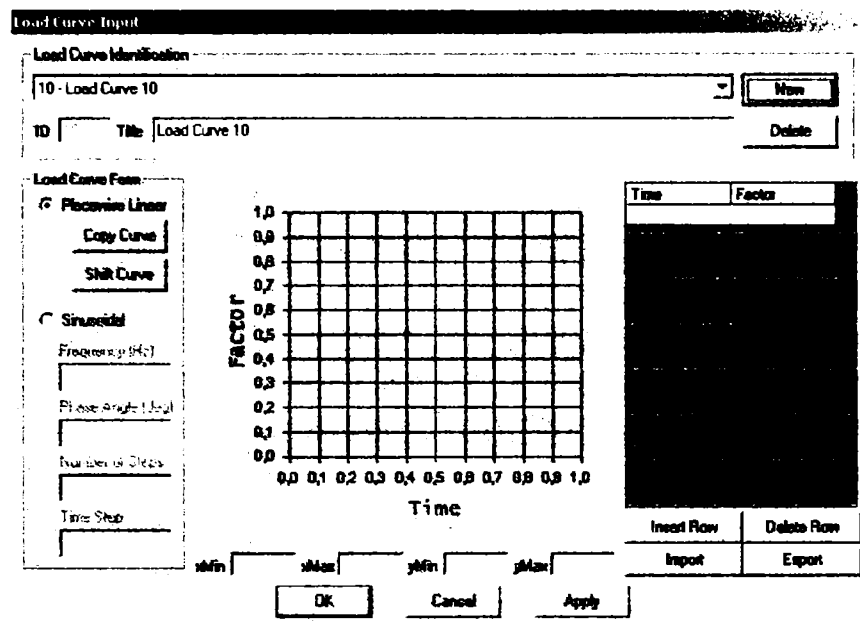


Fig.7.11. Setarea curbelor de variație în timp a parametrilor caracteristici

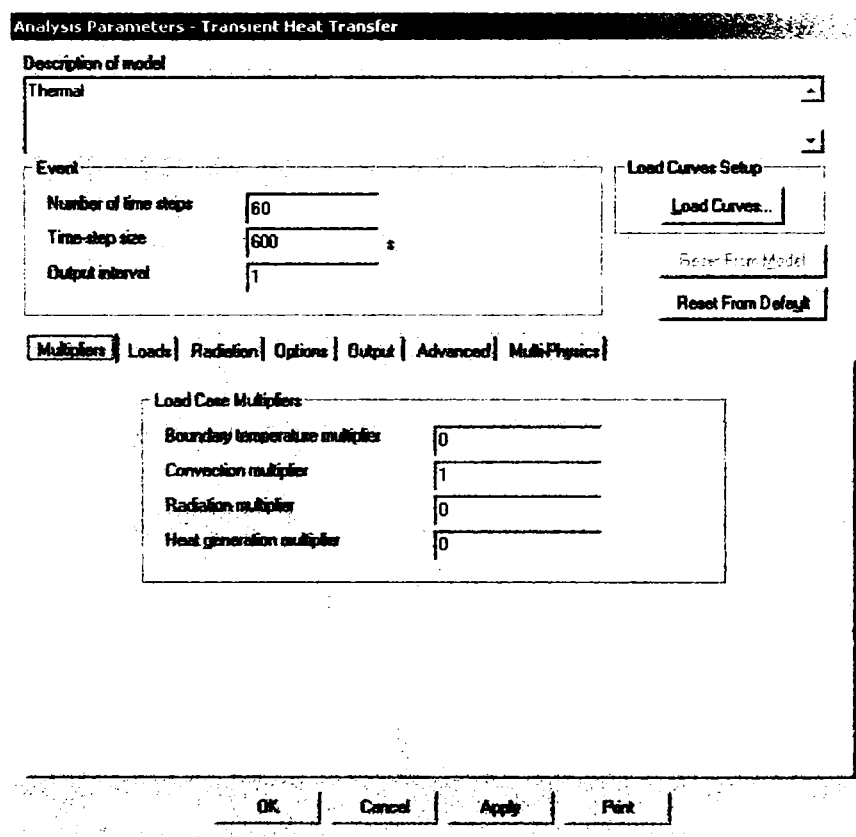


Fig.7.12. Stabilirea condițiilor de desfășurare în timp a fenomenului

După cum se poate observa și în figura 7.12, numărul pașilor de timp a fost ales ca fiind 60, rezultând, pentru un timp total de 10 h, o mărime a intervalelor de timp la care programul calculează valorile temperaturilor în nodurile rețelei de discretizare de 600 s.

7.2.3. Procesarea

În cazul în care proprietățile termo-fizice ale materialului sunt considerate variabile cu temperatura, sau când se ține cont de fenomenul de radiație în rezolvarea problemei, analiza devine una neliniară. Cu alte cuvinte, soluția nu poate fi obținută direct deoarece numărul necunoscutelor este mai mare decât cel al ecuațiilor sau datorită neliniarității ecuațiilor.

În aceste ipoteze, rezolvarea unei probleme staționare de transfer termic impune realizarea unor iterații multiple. Spre exemplu, fără a cunoaște valorile temperaturilor, conductivitatea termică nu poate fi determinată dar, pe de altă parte, temperaturile nodurilor rețelei nu pot fi calculate fără a cunoaște valorile conductivității termice. Din această cauză este necesară o abordare iterativă a problemei.

Spre deosebire de transferul termic staționar, care abordează problema iterativ, în analiza nestaționară se consideră că proprietățile termo-fizice și fluxul radiativ sunt constante pentru fiecare pas de timp. Calculul acestora se bazează pe valorile temperaturilor determinate la pasul anterior. Astfel, soluția corespunzătoare următorului pas de timp este liniară. Cu alte cuvinte, conductivitatea termică, căldura specifică și fluxul de căldură schimbat prin radiație sunt deci actualizate pentru fiecare secvență, pe baza valorilor de temperatură calculate anterior.

Procesul descris mai sus este prezentat schematic în figura 7.13 [52].

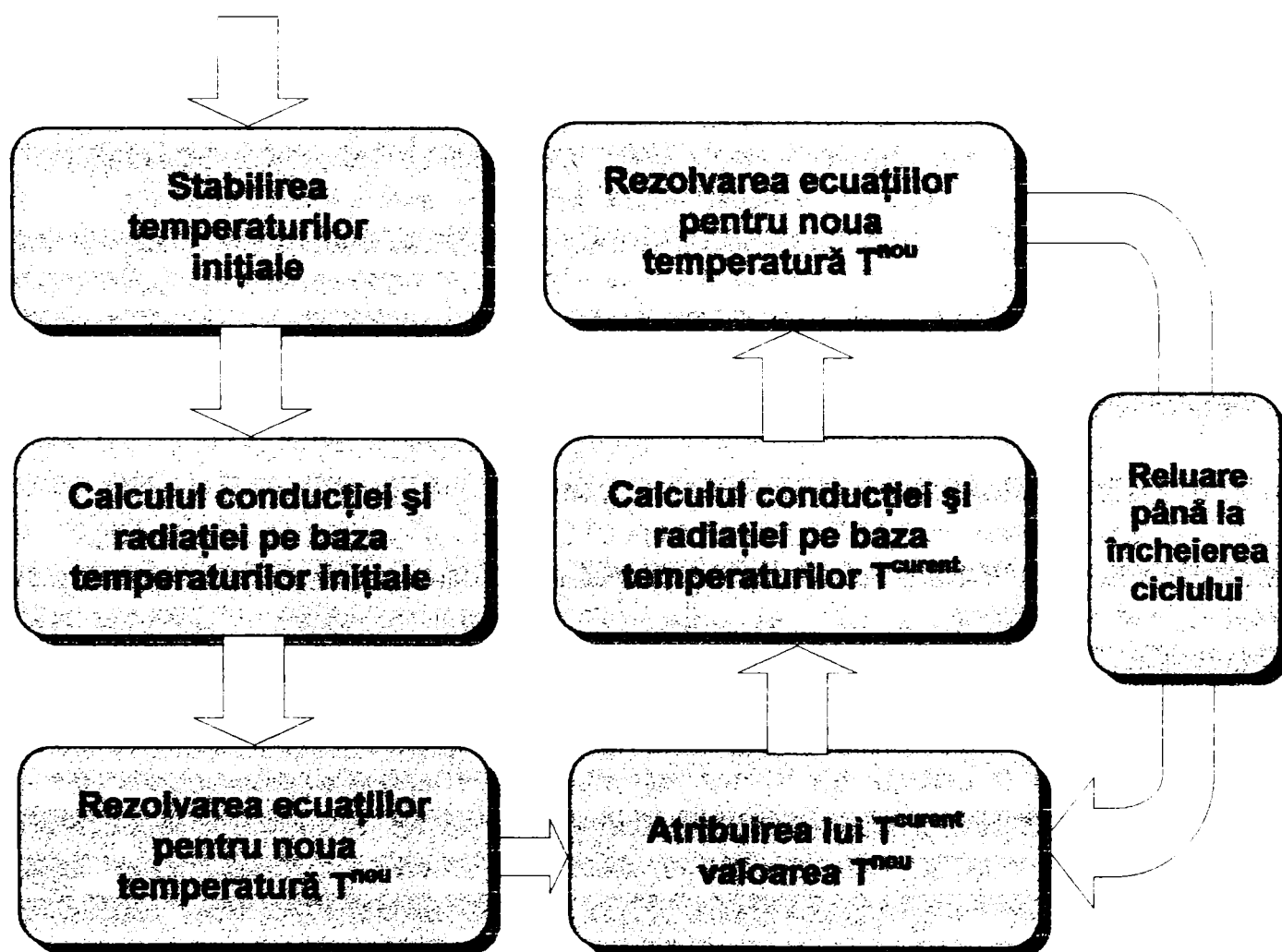


Fig.7.13. Procesul de soluționare a unei probleme neliniare de transfer termic nestaționar

7.2.4. Postprocesarea

În acest moment, au fost introduse toate datele cerute de program pentru rezolvarea câmpului de temperaturi corespunzător domeniului de analiză stabilit. ALGOR pune la dispoziția utilizatorului o statistică a datelor introduse, prin intermediul ferestrei de dialog din figura 7.14.

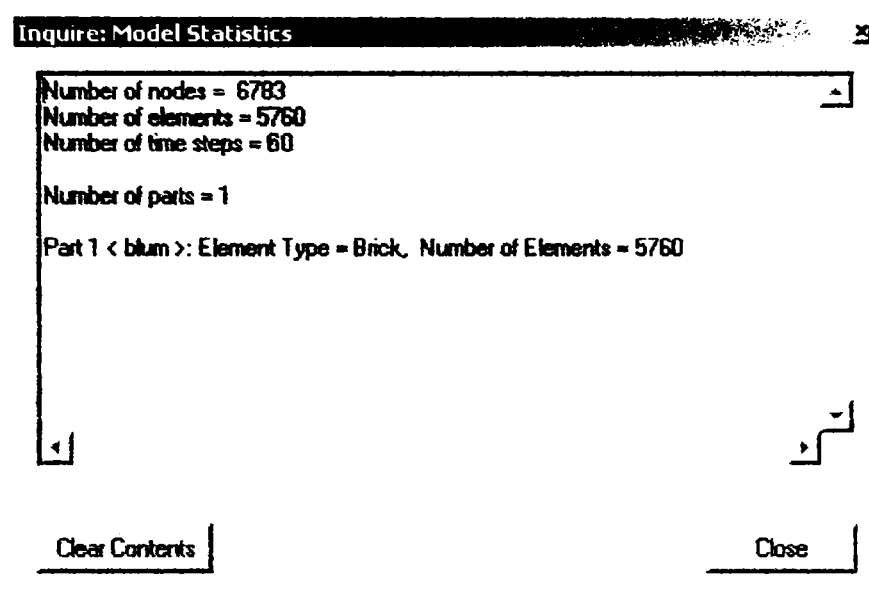


Fig.7.14. Statistica modelului în ALGOR

Discretizarea domeniului de analiză s-a făcut într-un număr de 5760 de elemente finite de tipul „Thermal Brick”, rezultând 6783 de noduri. În fiecare din aceste noduri programul calculează valorile temperaturilor pentru un număr de 60 de pași de timp, pe întreaga durată de 10 h a încălzirii.

Având stabilită varianta de rezolvare (tipul *solver*-ului) se poate trece la rularea programului. În timpul procesării se poate alege opțiunea de a deschide o fereastră de vizualizare a pașilor executați pe parcursul operațiunilor de calcul. În cazul în care nu a intervenit nici un fel de eroare în timpul procesului de introducere a datelor, programul rulează un anumit interval de timp (necesar rezolvării sistemului de ecuații creat) după care inițiază un dialog prin care operatorul este anunțat că procesarea datelor s-a încheiat cu succes.

În ceea ce privește prezentarea rezultatelor finale, putem preciza faptul că pachetul ALGOR permite vizualizarea lor în două moduri: calitativ și cantitativ.

Modalitatea de vizualizare *calitativă* a rezultatelor implică o reprezentare în hărți de culori și/sau linii de egală temperatură (izoterme) a câmpului termic studiat, permițând o viziune globală asupra acestuia la un moment dat al încălzirii sau asupra evoluției în timp a temperaturilor în întreg domeniul de analiză.

Modalitatea de vizualizare *cantitativă* presupune furnizarea datelor, sub formă tabelară (în cazul nostru valorile temperaturilor), în fiecare nod al rețelei de discretizare sau în anumite noduri specificate anterior. Această modalitate permite trasarea unor grafice de variație a mărimilor studiate și compararea rezultatelor cu cele obținute pe cale experimentală. Se pot stabili astfel, cu mare exactitate, erorile care apar în modelul numeric față de valorile obținute experimental și proveniența acestor erori (datorită ipotezelor simplificatoare, a modelului numeric, a rețelei de discretizare, etc.). Această modalitate este cea care permite validarea “în profunzime” a metodologiei alese.

7.3. Prezentarea rezultatelor

Pachetul ALGOR, prin intermediul interfeței SUPERVIEW facilitează vizualizarea calitativă a rezultatelor obținute în urma procesării datelor introduse.

După cum am arătat, rezolvarea câmpului termic nestaționar în interiorul blumului s-a făcut pe un interval de 10 h, cu un pas de timp de 600 s. Vizualizarea calitativă, din considerente de spațiu, se va face la pașii de timp 2, 6, 12, 18, 24, 30, 36, 42, 48, 54 și 60. Tot din considerente de spațiu, imaginile secționate ale câmpului termic în interiorul semifabricatului vor fi prezentate doar pentru pașii 2, 42 și respectiv 60.

Suplimentar, au fost calculate, cu ajutorul facilităților oferite de program, tensiunile și deformațiile de natură termică induse în interiorul masei blumului datorită gradientilor termici.

7.3.1. Câmpul termic

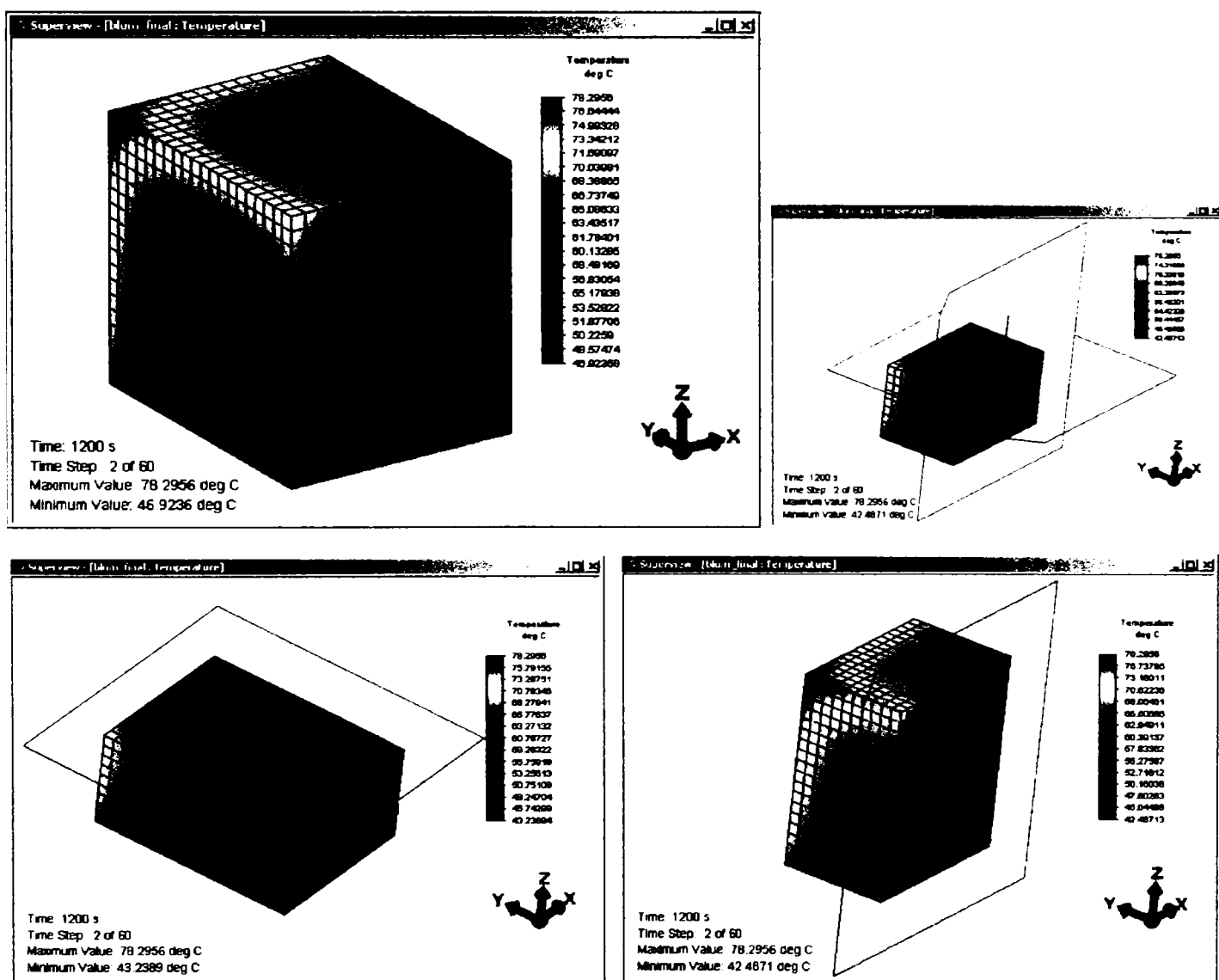


Fig.7.15. Câmpul termic în interiorul blumului la pasul de timp nr.2

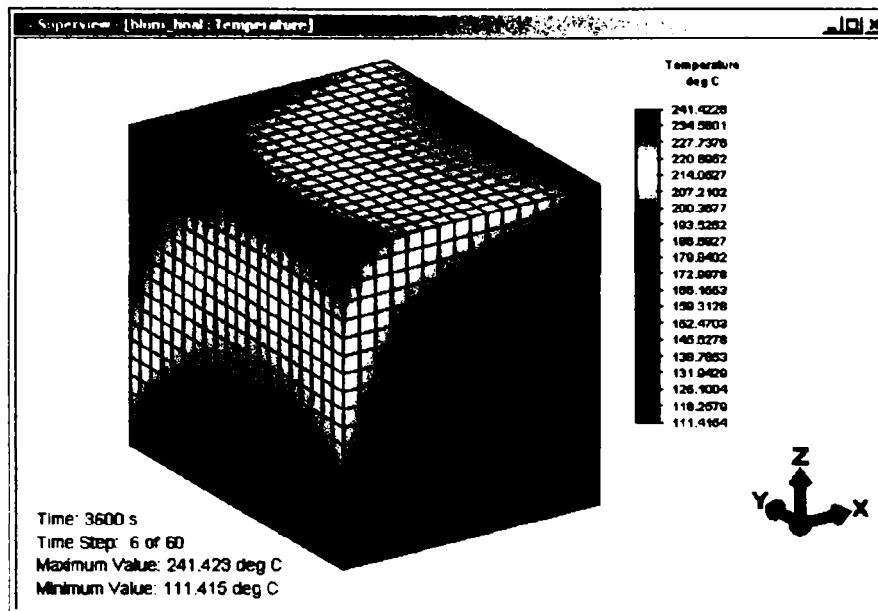


Fig.7.16. Câmpul termic în interiorul blumului la pasul de timp nr.6

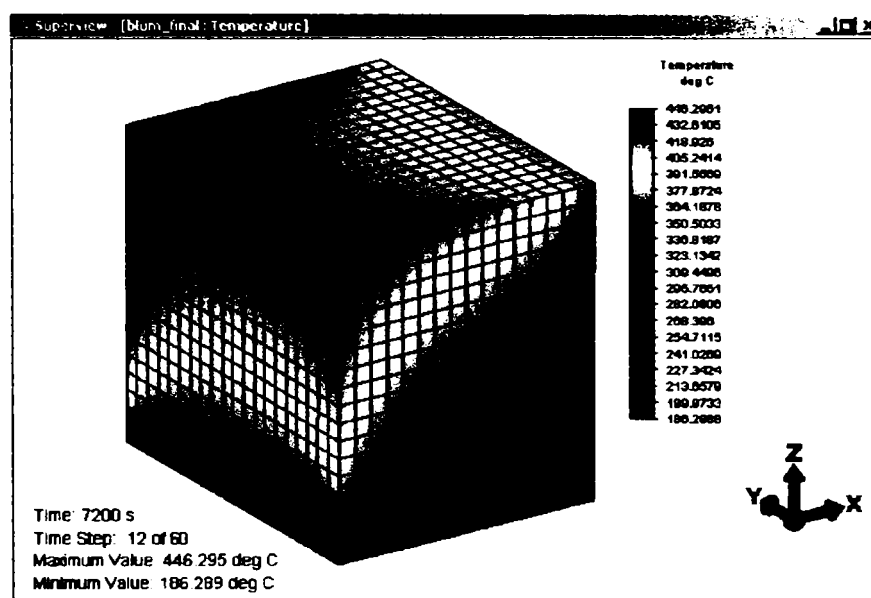


Fig.7.17. Câmpul termic în interiorul blumului la pasul de timp nr.12

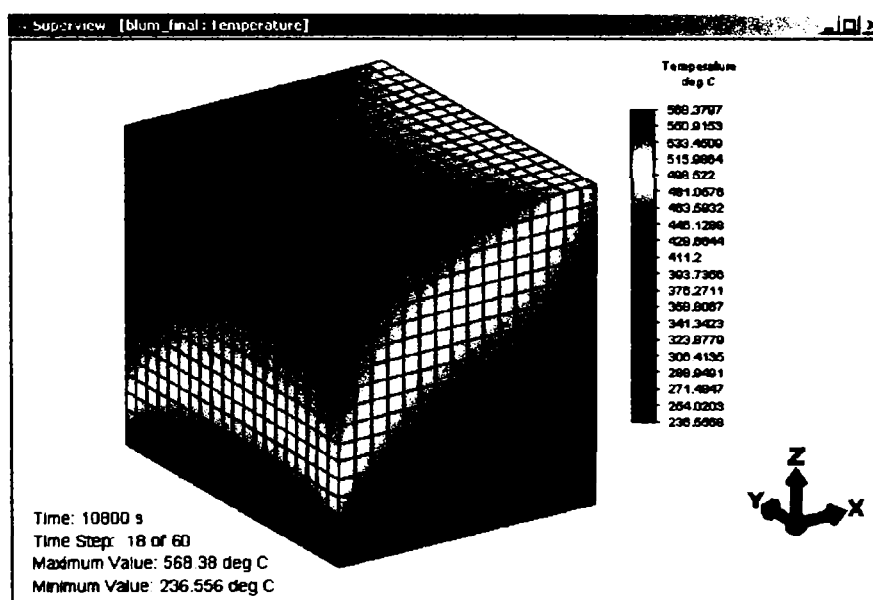


Fig.7.18. Câmpul termic în interiorul blumului la pasul de timp nr.18

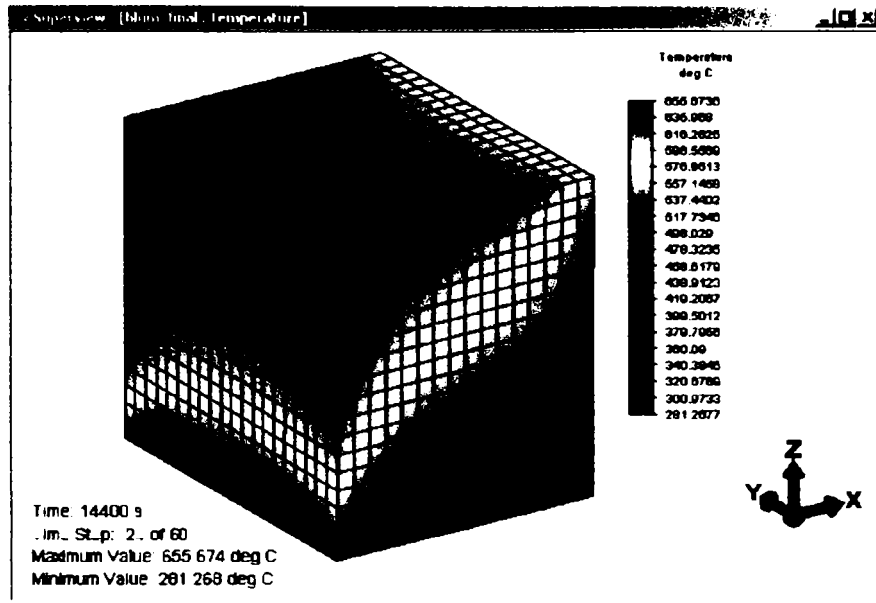


Fig.7.19. Câmpul termic în interiorul blumului la pasul de timp nr.24

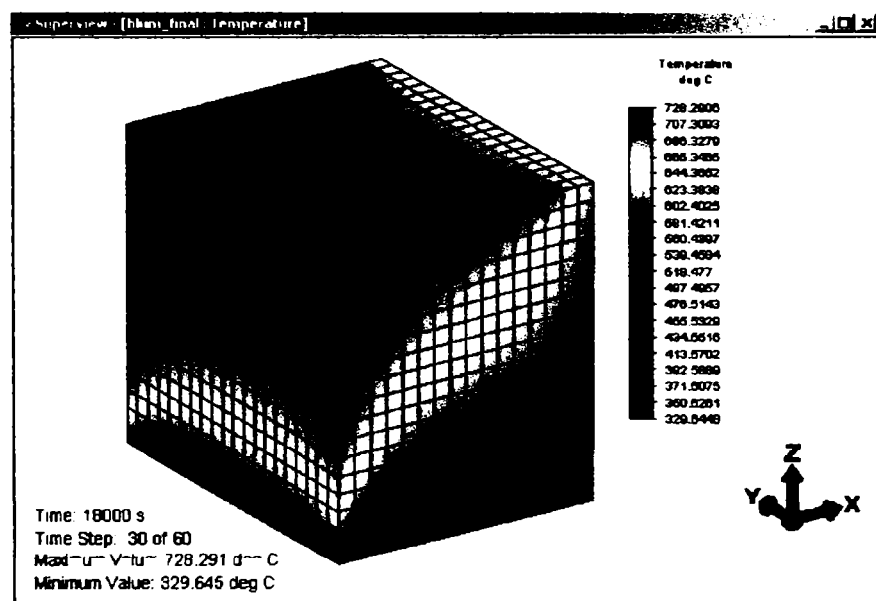


Fig.7.20. Câmpul termic în interiorul blumului la pasul de timp nr.30

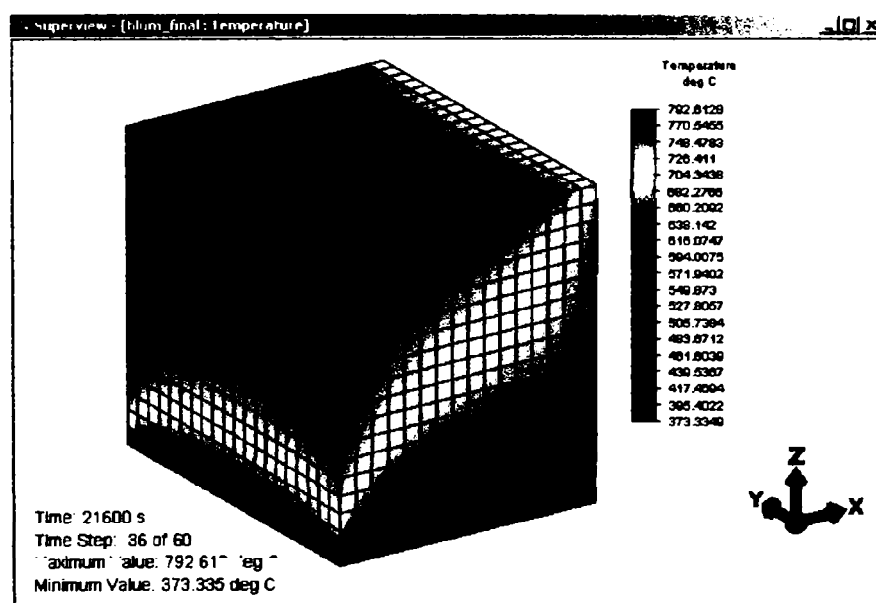


Fig.7.21. Câmpul termic în interiorul blumului la pasul de timp nr.36

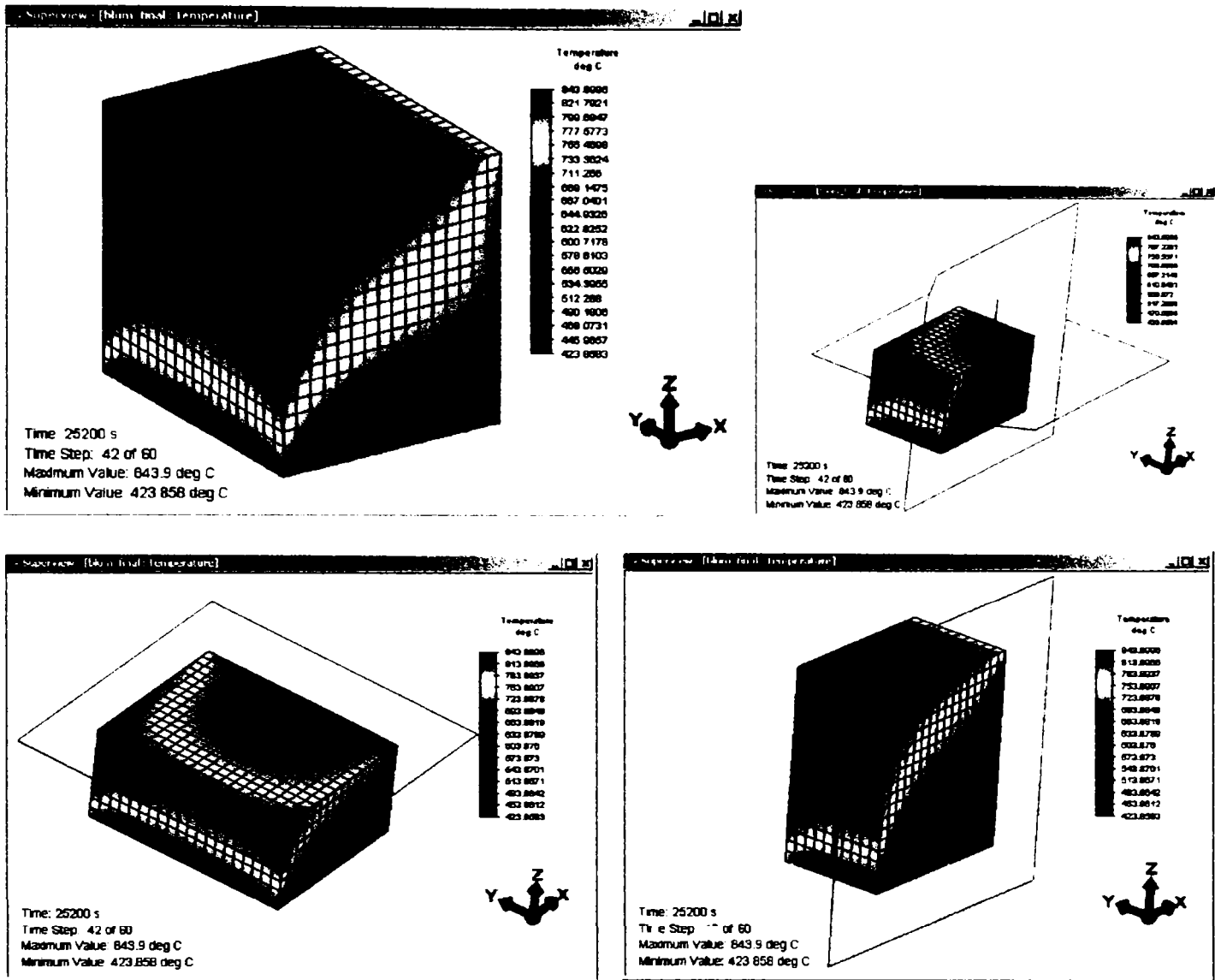


Fig.7.22. Câmpul termic în interiorul blumului la pasul de timp nr.42

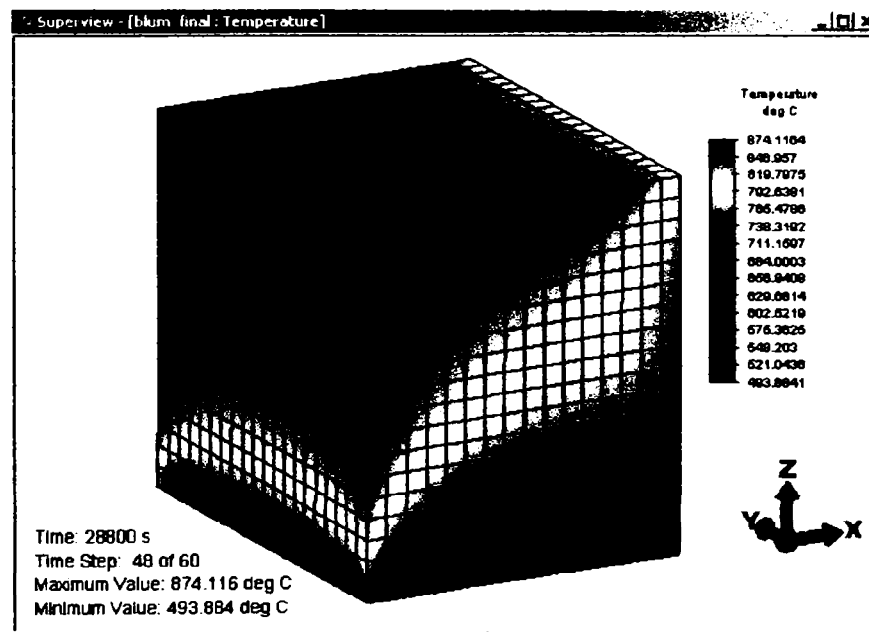


Fig.7.23. Câmpul termic în interiorul blumului la pasul de timp nr.48

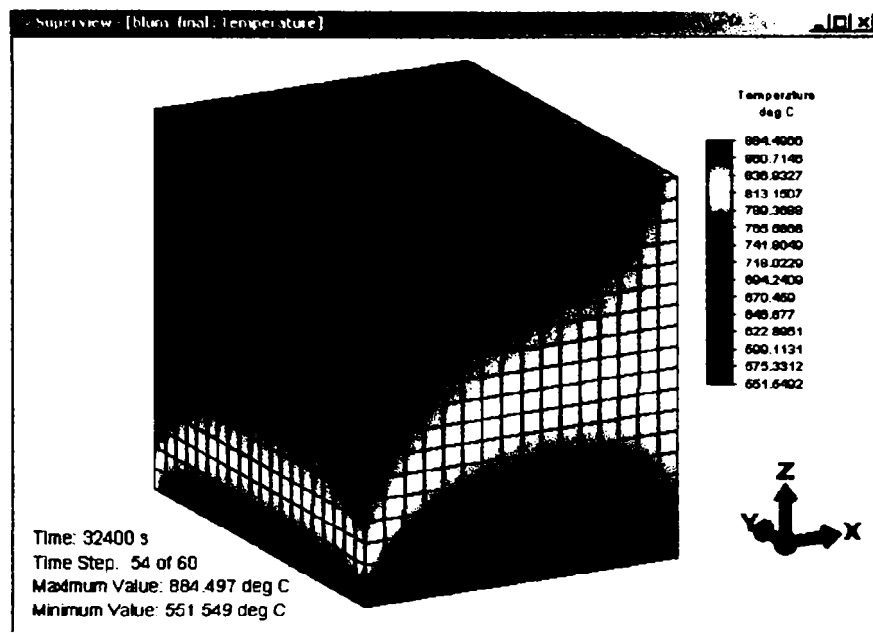


Fig.7.24. Câmpul termic în interiorul blumului la pasul de timp nr.54

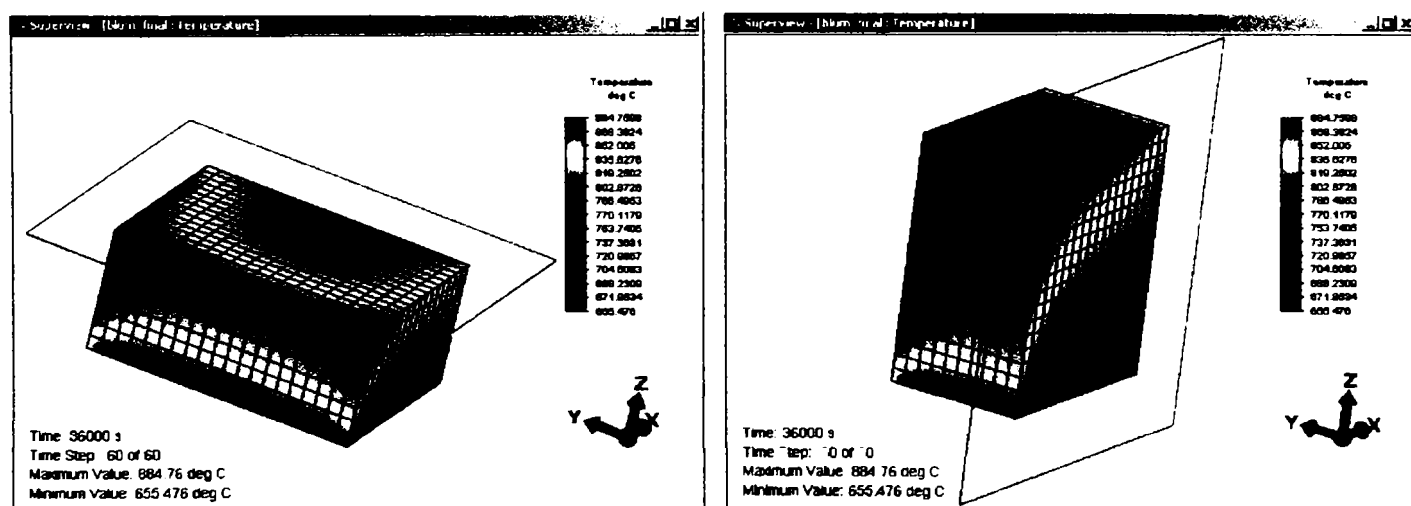
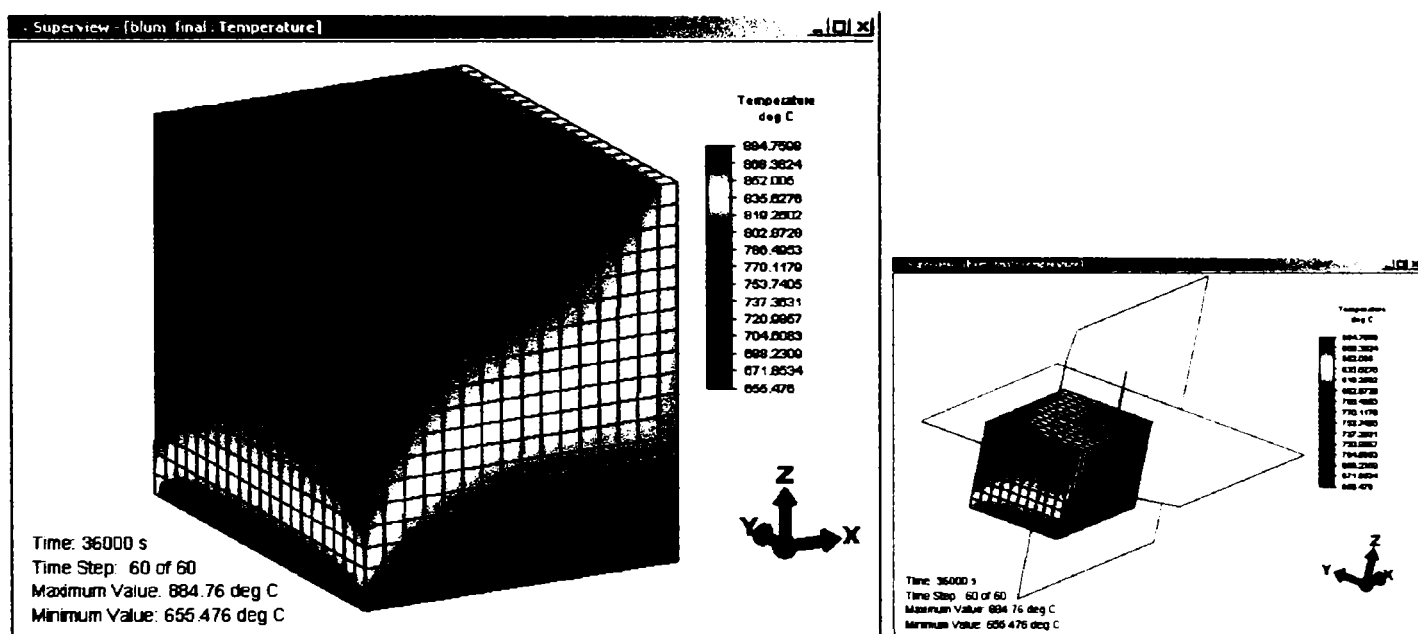


Fig.7.25. Câmpul termic în interiorul blumului la pasul de timp nr.60

7.3.2. Tensiunile interne datorate gradientilor termici

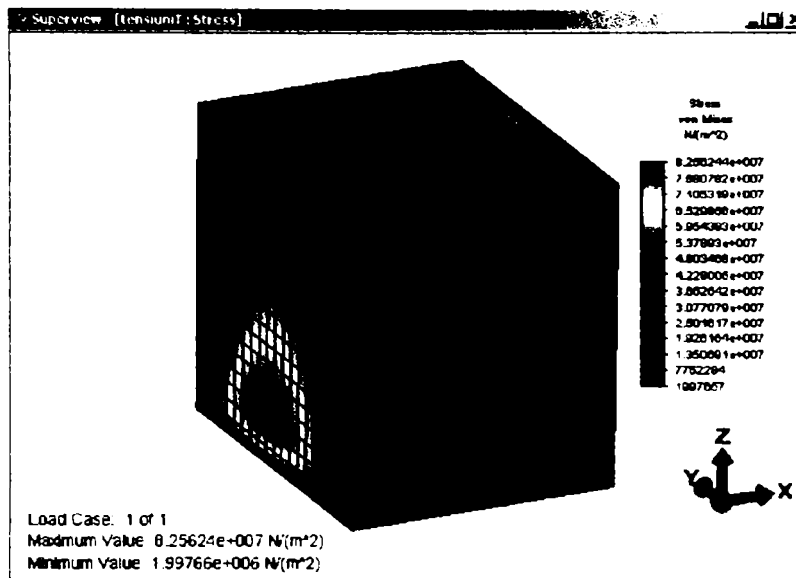


Fig.7.26. Tensiunile termice în interiorul blumului la pasul de timp nr.6

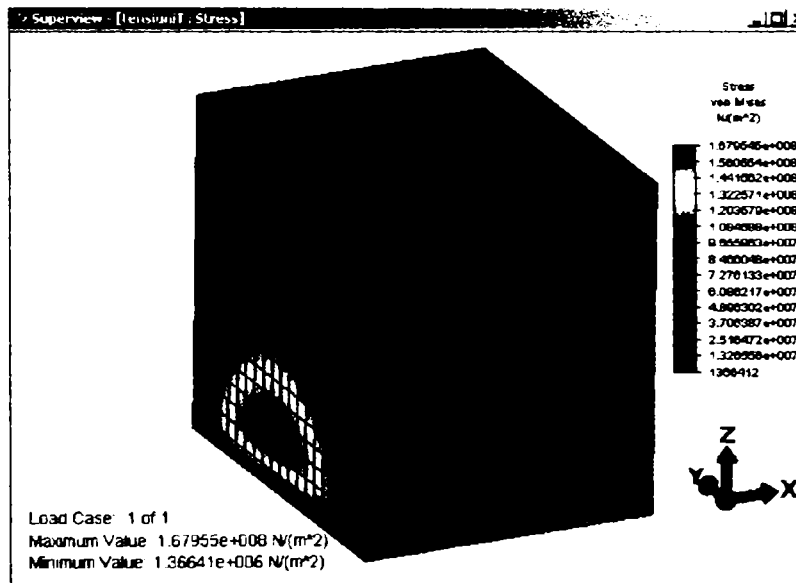


Fig.7.27. Tensiunile termice în interiorul blumului la pasul de timp nr.12

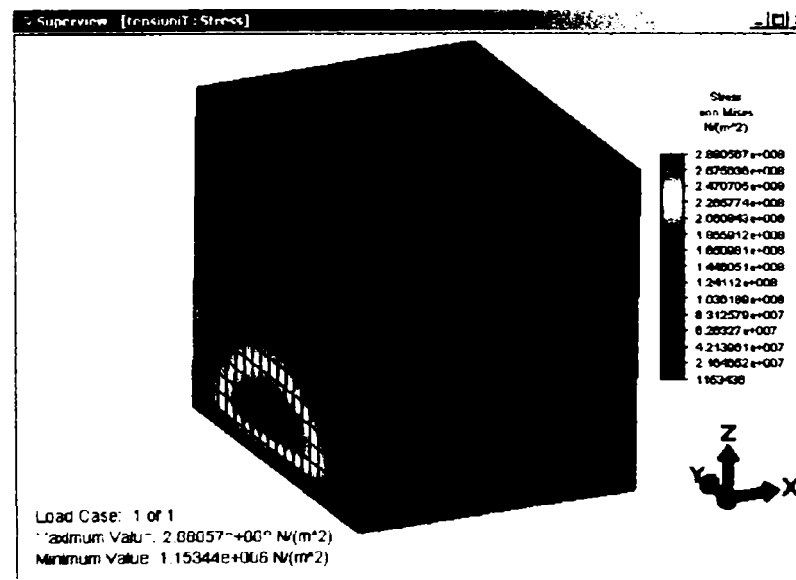


Fig.7.28. Tensiunile termice în interiorul blumului la pasul de timp nr.18

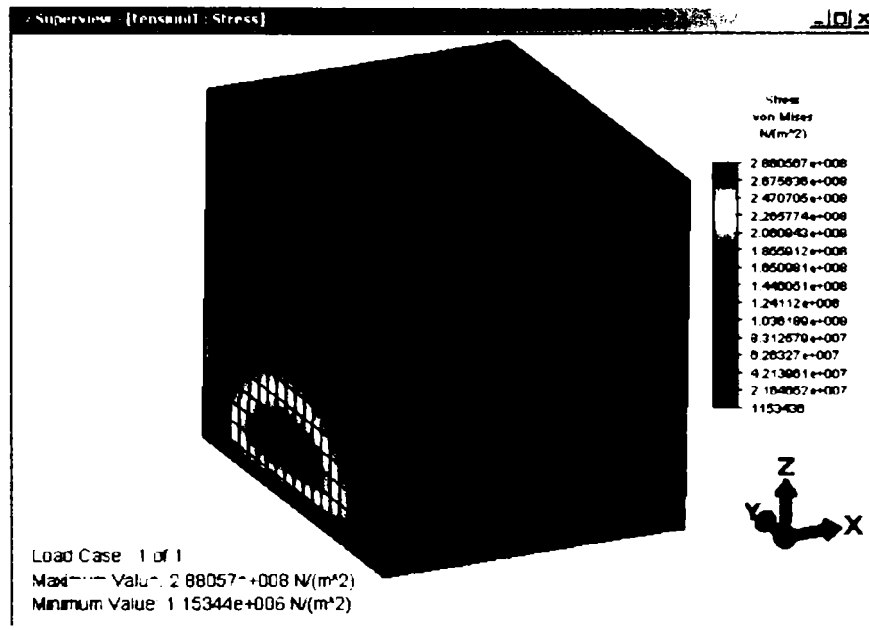


Fig.7.29. Tensiunile termice în interiorul blumului la pasul de timp nr.24

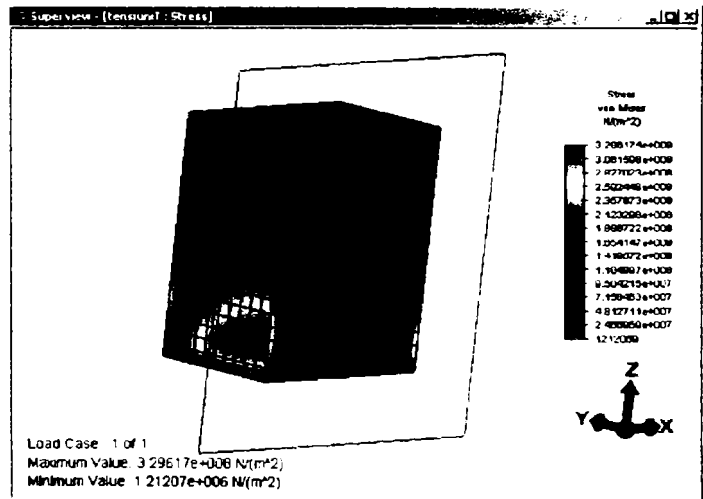
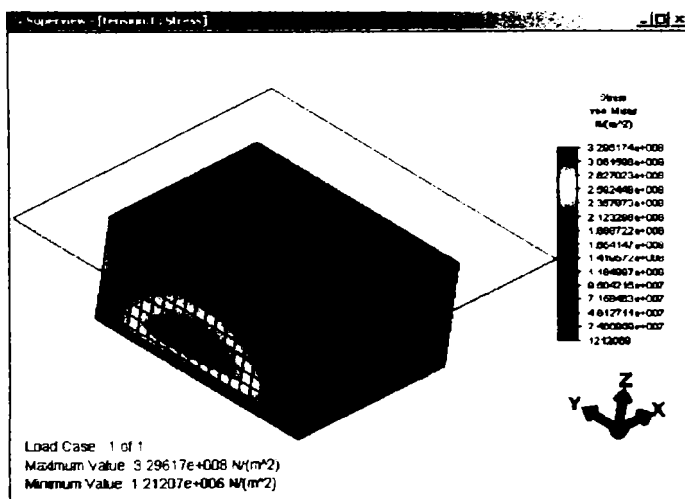
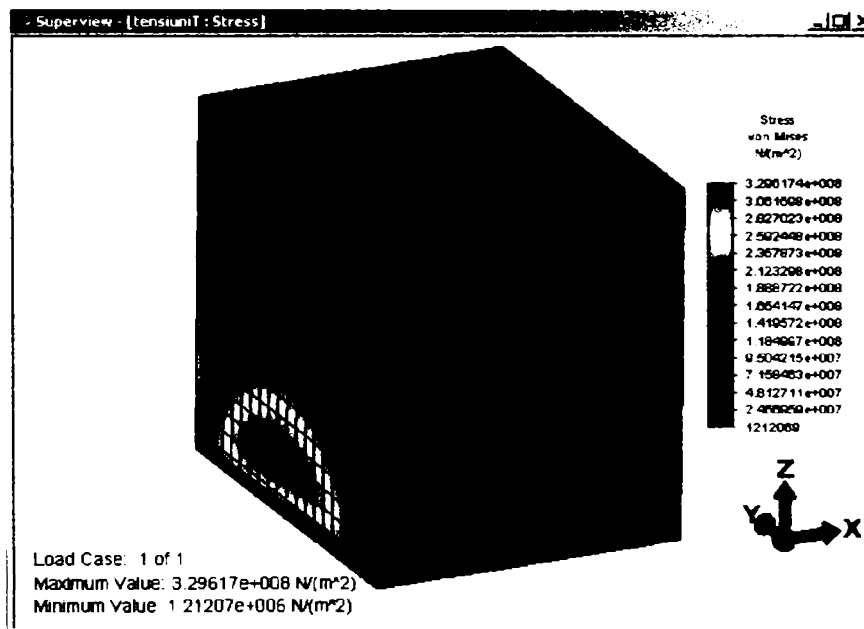


Fig.7.30. Tensiunile termice în interiorul blumului la pasul de timp nr.30

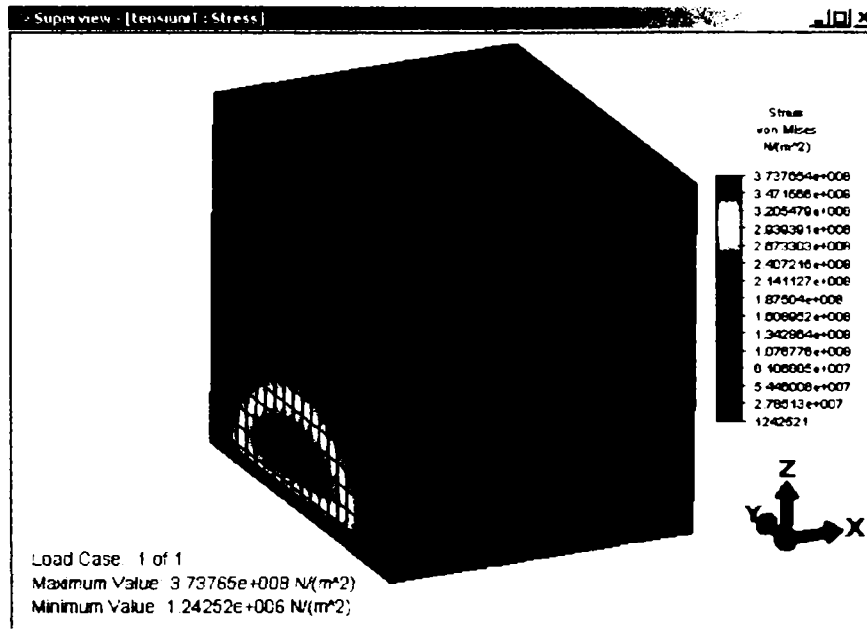


Fig.7.31. Tensiunile termice în interiorul blumului la pasul de timp nr.36

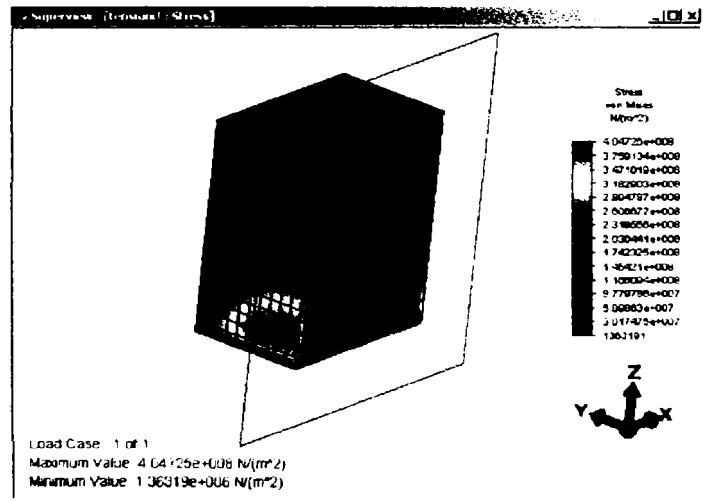
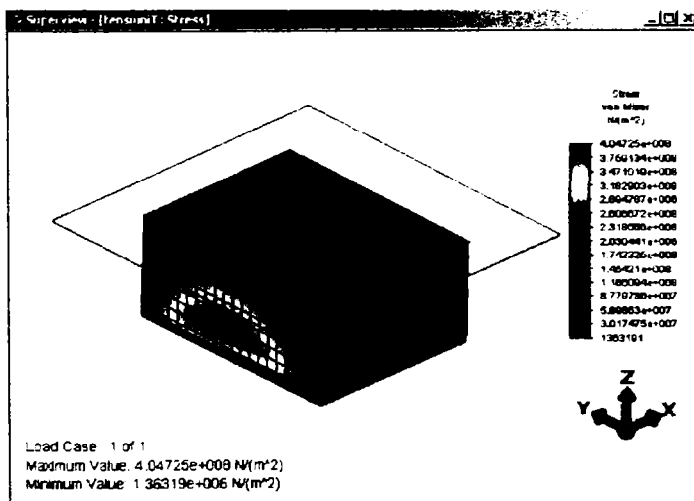
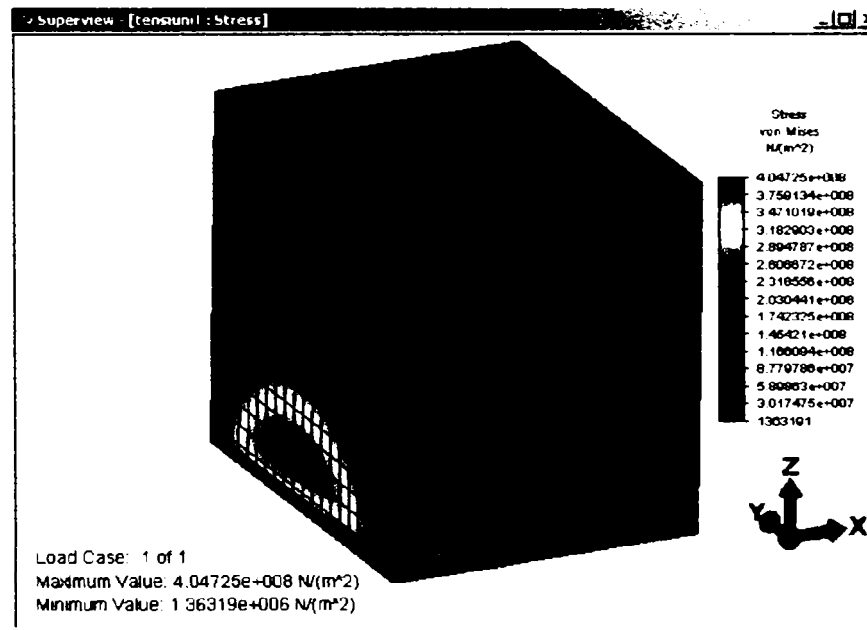


Fig.7.32. Tensiunile termice în interiorul blumului la pasul de timp nr.42

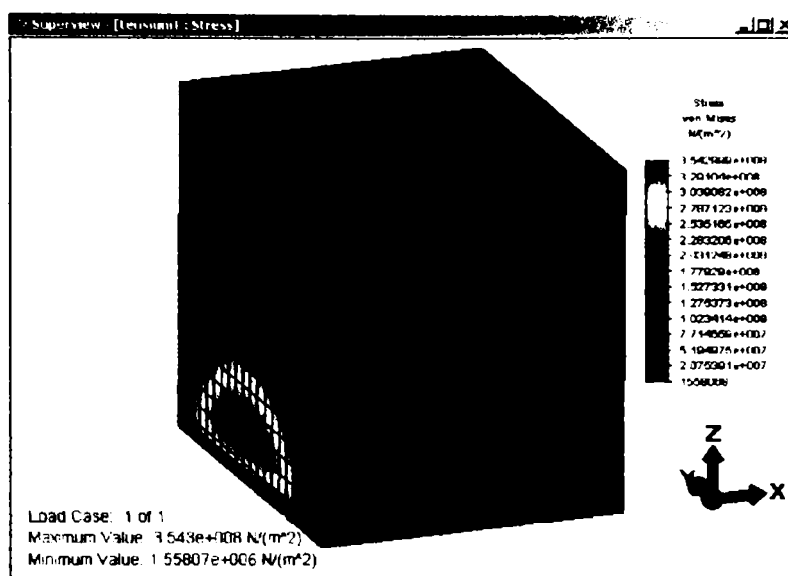


Fig.7.33. Tensiunile termice în interiorul blumului la pasul de timp nr.48

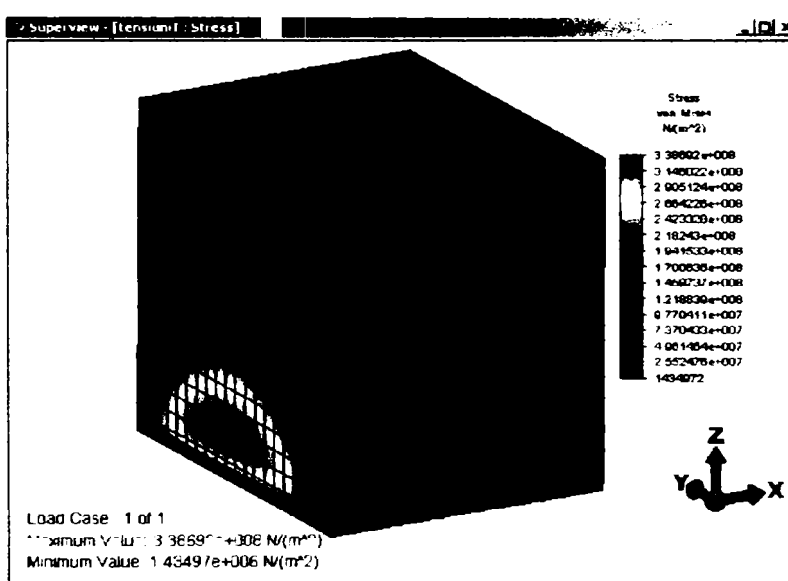


Fig.7.34. Tensiunile termice în interiorul blumului la pasul de timp nr.54

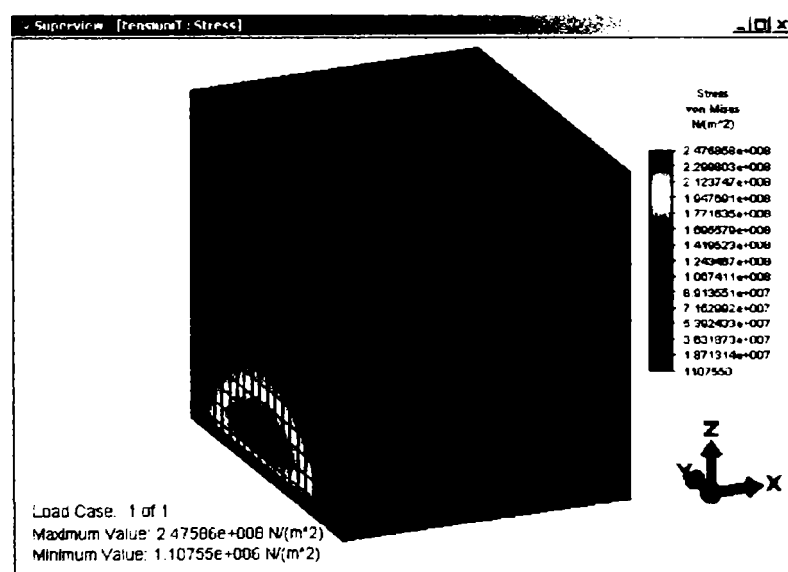


Fig.7.35. Tensiunile termice în interiorul blumului la pasul de timp nr.60

7.3.3. Deformațiile produse datorită încălzirii

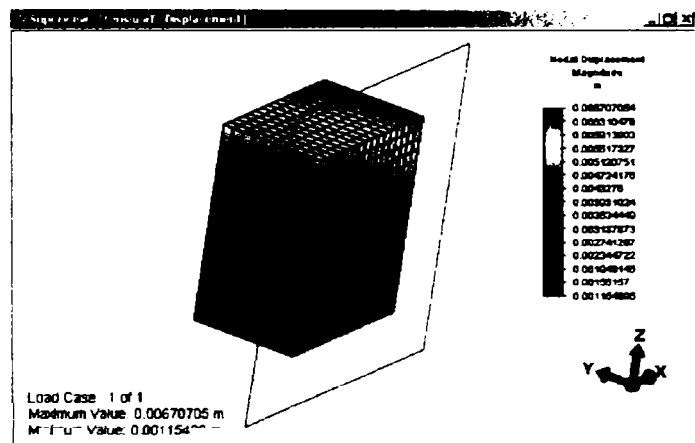
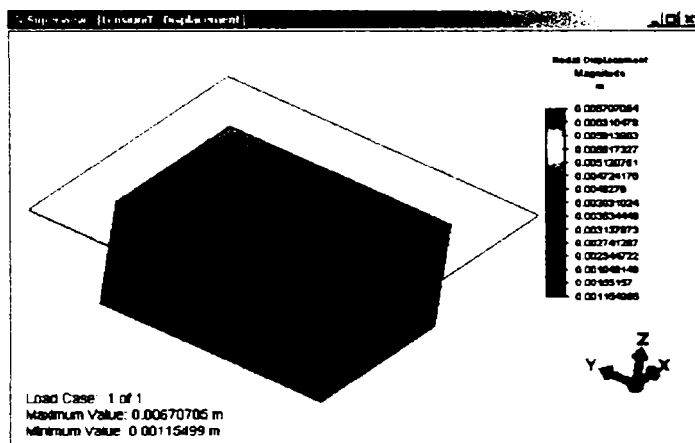
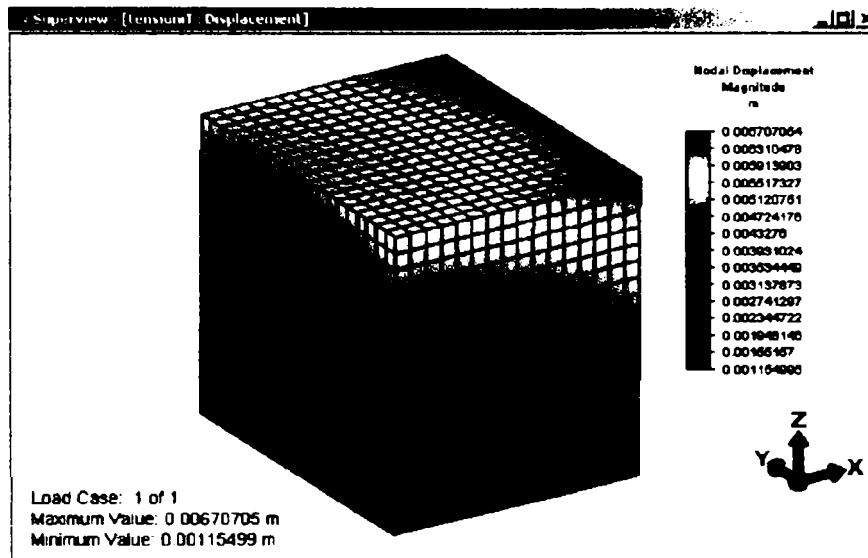


Fig.7.36. Deplasările nodale în interiorul blumului la mijlocul perioadei de încălzire

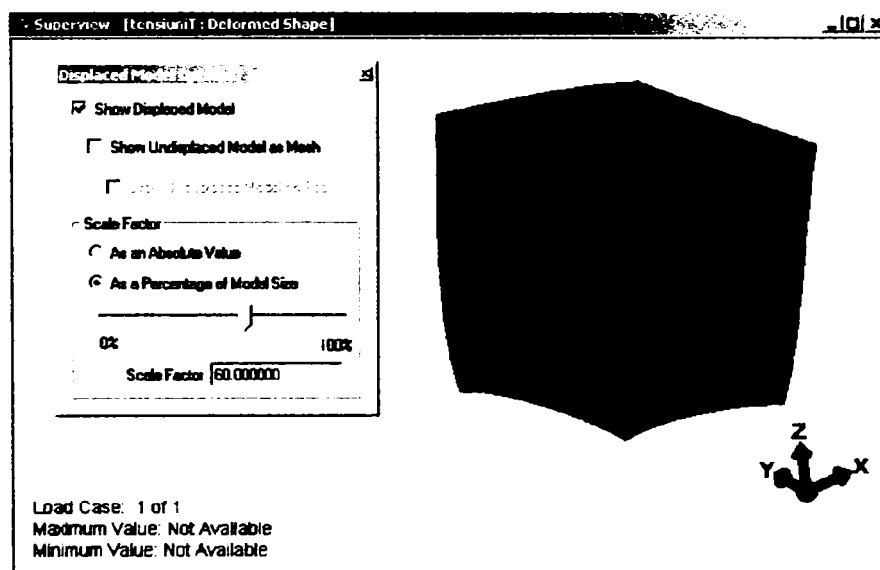


Fig.7.36. Vizualizarea structurii deformate a modelului la mijlocul perioadei de încălzire

7.3.4. Evoluția temperaturii nodurilor caracteristice

Prin noțiunea de *noduri caracteristice* am definit acele noduri ale rețelei de discretizare care corespund punctelor în care au fost introduse termocupurile pentru măsurarea variației temperaturii blumului în instalația experimentală pilot. Identificarea de către utilizator a nodurilor respective, situate în interiorul domeniului de analiză al modelului implementat în ALGOR, s-a realizat prin secționarea domeniului tridimensional cu plane paralele cu planele xOy , xOz și respectiv yOz . Intersecția acestor plane determină nodurile căutate. Această facilitate este pusă la dispoziție de către programul ALGOR prin intermediul interfeței Superview (figura 7.37).

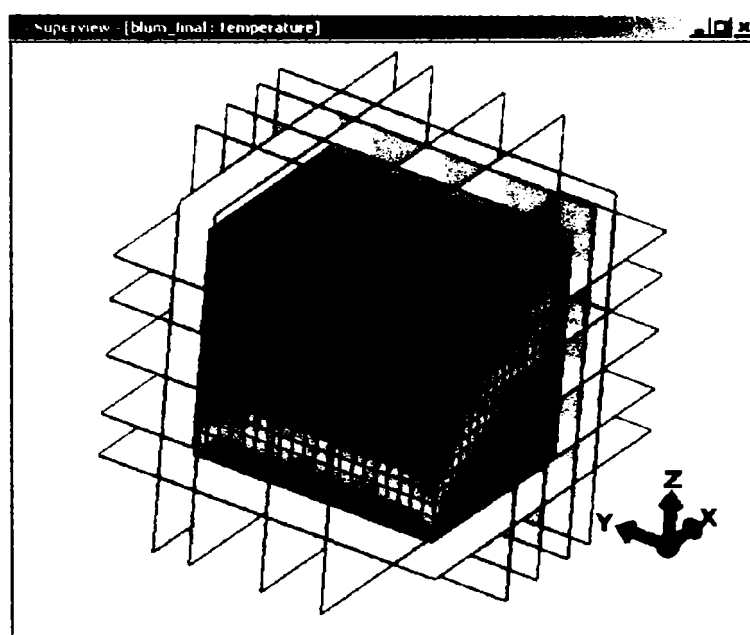


Fig.7.37. Vizualizarea planelor la intersecția cărora se află nodurile caracteristice

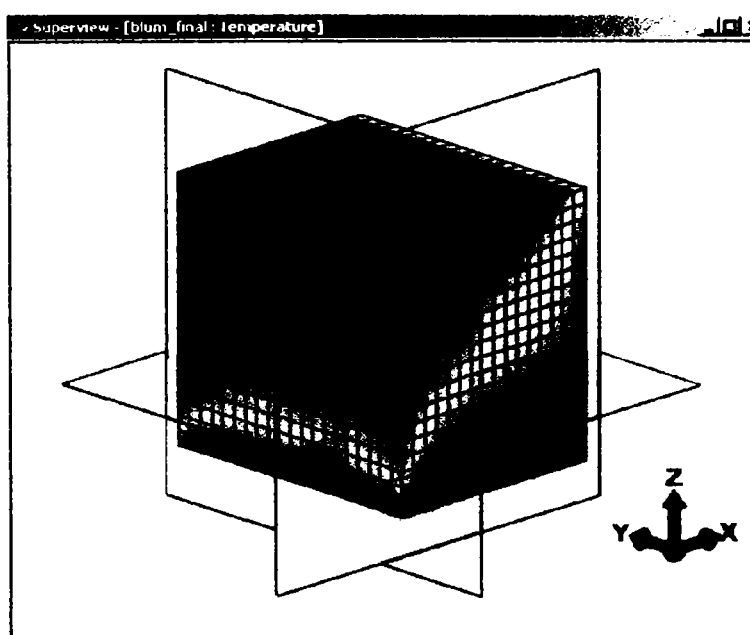
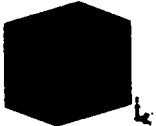
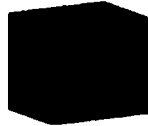

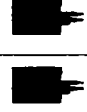

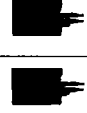

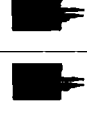




Fig.7.38. Localizarea nodului caracteristic în domeniul de analiză al modelului numeric corespunzător punctului 2 de măsurare

Spre exemplu, pentru punctul caracteristic notat cu 2 (vezi figura 4.7), localizarea nodului corespunzător din modelul numeric s-a făcut conform figurii 7.38.

Correspondența nodurilor caracteristice, aparținând domeniului de analiză al modelului numeric cu elemente finite implementat în ALGOR, cu punctele din interiorul blumului, în care s-au realizat măsurătorile de temperatură, este prezentată în tabelul de mai jos.

Tabelul 7.1

Nodurile caracteristice din domeniul de analiză al modelului numeric	Punctele de măsurare din interiorul blumului
 6156	 1
 1518	 2
 3392	 3
 5266	 4
 695	 5

Evoluția în timp a temperaturii nodurilor caracteristice poate fi urmărită cu ajutorul ferestrei de dialog intitulată Monitor, prezentată în figura de mai jos.

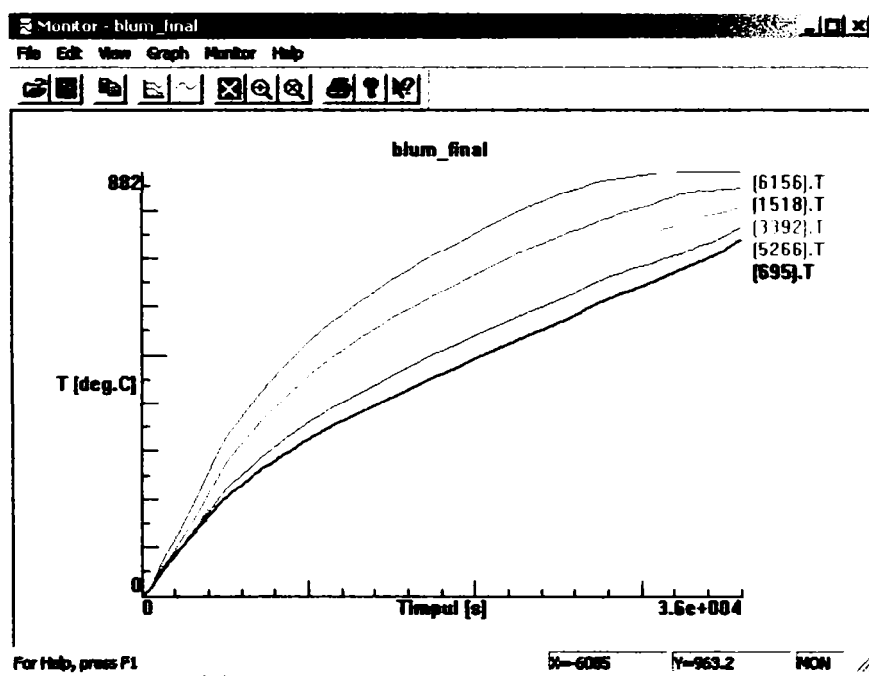


Fig.7.39. Prezentarea grafică a evoluției temperaturii nodurilor caracteristice

Datele obținute pot fi exportate ca și fișiere cu extensia .CSV (comma separate values) în scopul comparării lor cu valorile obținute prin măsurători experimentale.

7.4. Concluzii

După cum am arătat încă din introducere, unul din obiectivele prezentei teze de doctorat îl reprezintă validarea metodologiei numerice folosite și, în consecință, a rezultatelor numerice, prin rezultatele experimentale.

Această validare, în cazul metodei cu elemente finite, presupune o comparație între valorile temperaturilor obținute prin rezolvarea modelului numeric construit cu ajutorul pachetului ALGOR și valorile determinate experimental.

Având la dispoziție doar curbele experimentale de variație a temperaturii în punctele de măsurare, conform figurii 4.7, compararea rezultatelor se va face doar pentru valorile corespunzătoare acestor puncte.

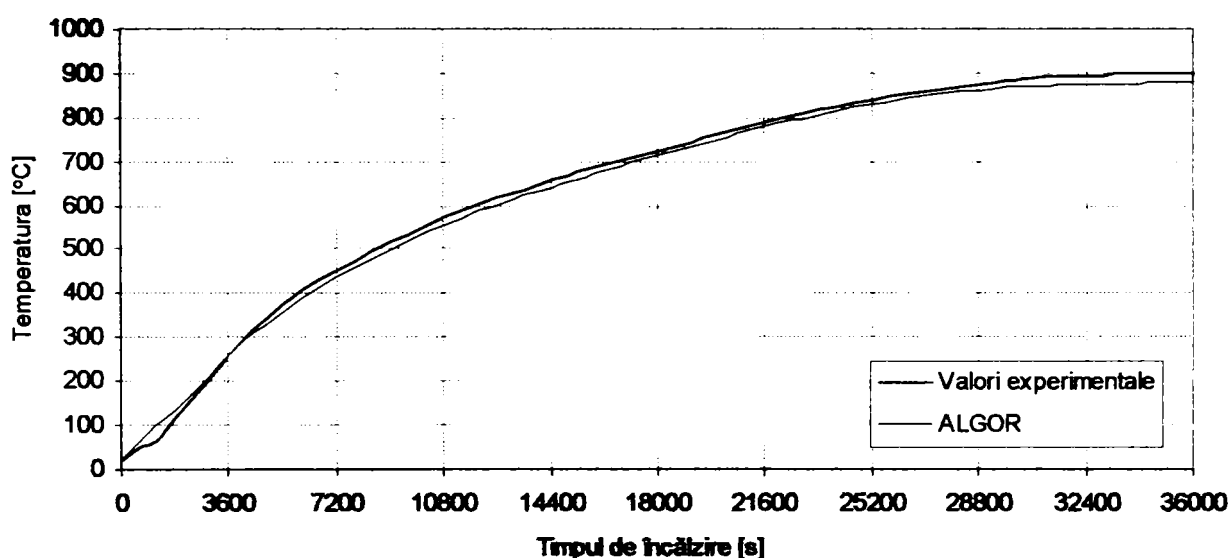


Fig.7.40. Rezultate comparative corespunzătoare punctului 1 de măsurare

În figura 7.40 s-au reprezentat comparativ variațiile în timp ale temperaturii obținute prin măsurători experimentale și calculate numeric cu ALGOR, pentru punctul caracteristic nr.1. Semnificația graficelor din figurile 7.41, 7.42, 7.43 și 7.44 este aceeași, dar pentru punctele caracteristice 2, 3, 4, și respectiv 5.

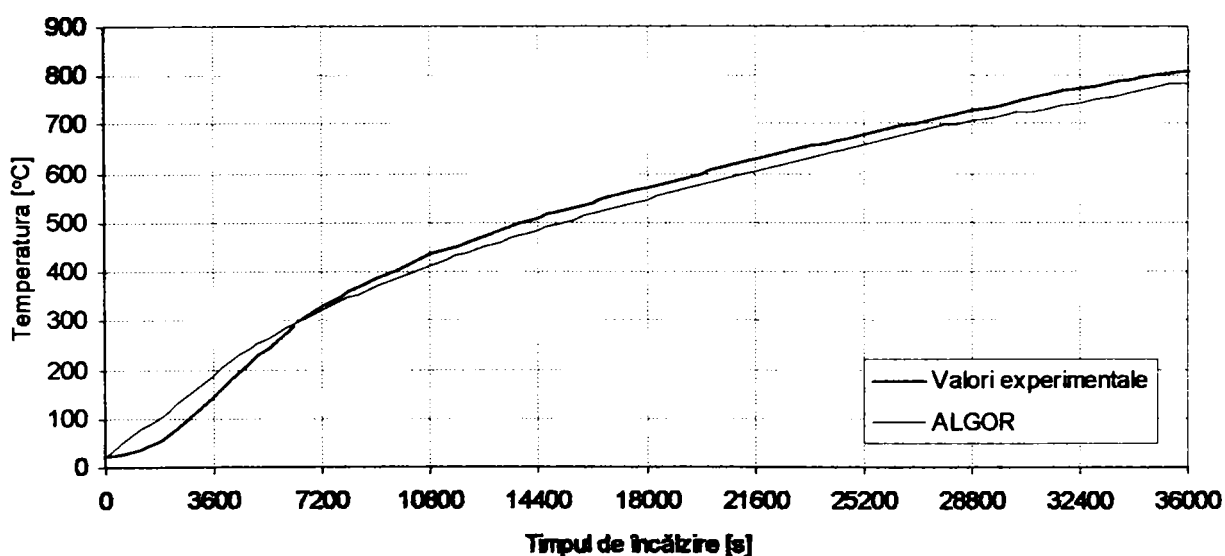


Fig.7.41. Rezultate comparative corespunzătoare punctului 2 de măsurare

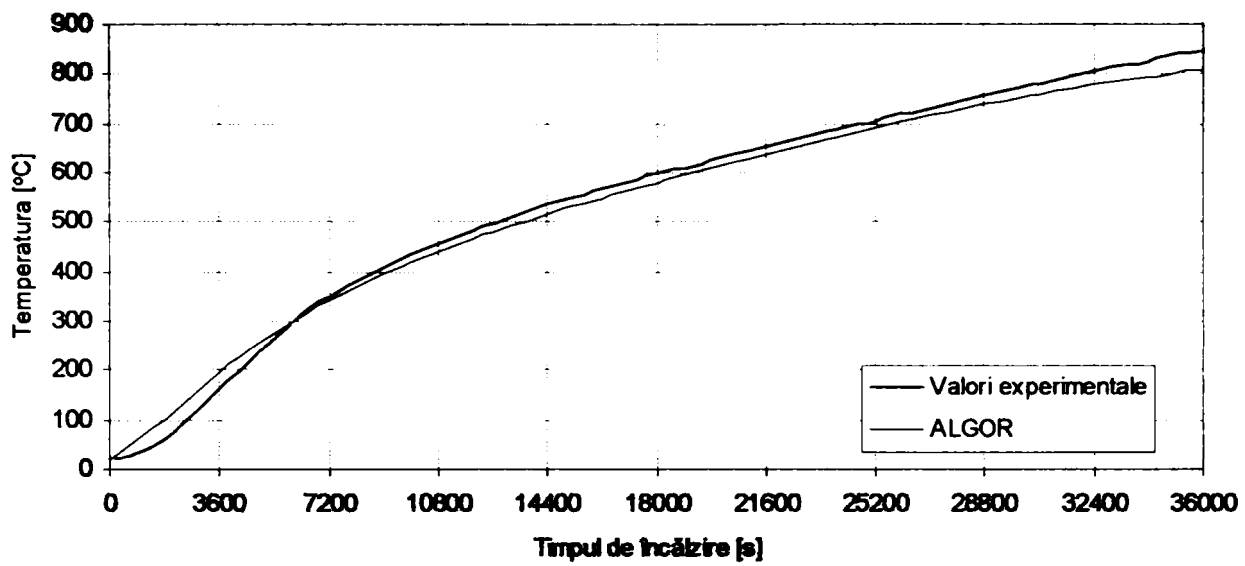


Fig.7.42. Rezultate comparative corespunzătoare punctului 3 de măsurare

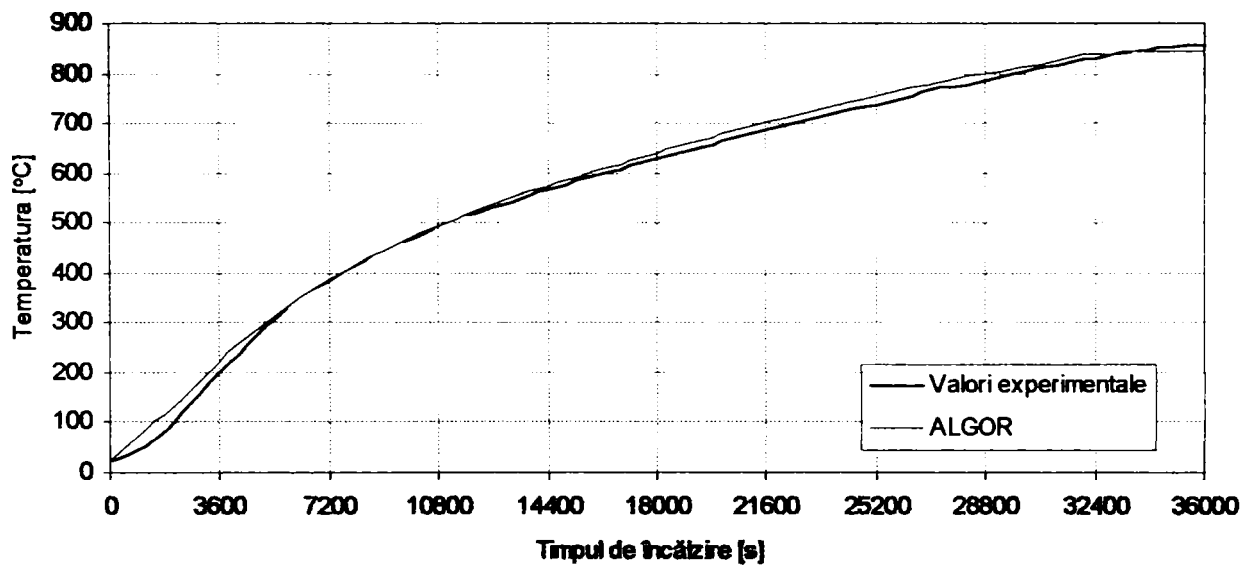


Fig.7.43. Rezultate comparative corespunzătoare punctului 4 de măsurare

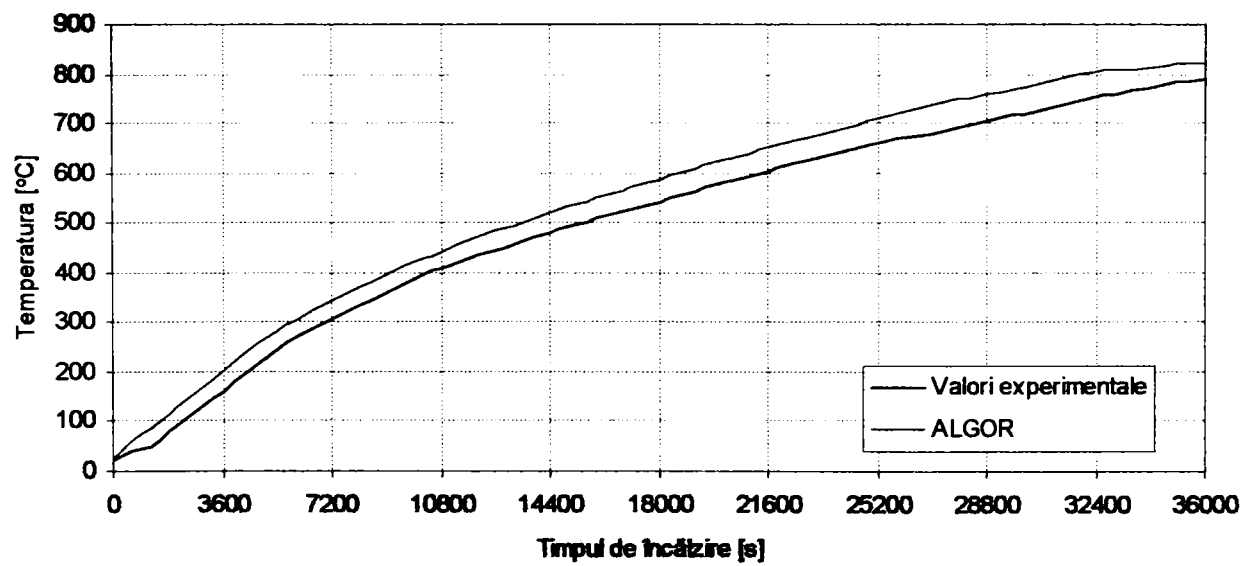


Fig.7.44. Rezultate comparative corespunzătoare punctului 5 de măsurare

În figura 7.45 s-a reprezentat, pentru întreg intervalul de încălzire, diferența efectivă între valorile măsurate experimental, considerate etalon și cele obținute numeric cu ALGOR, pentru toate cele cinci puncte (respectiv noduri caracteristice). În plus, figura 7.46 prezintă eroarea relativă a valorilor calculate numeric cu ALGOR față de cele măsurate experimental, pentru aceleași cinci puncte. Modul de calcul al acestor valori este următorul:

⇒ Pentru diferența efectivă: $\Delta t = t_{exp} - t_{ALGOR} [^{\circ}C]$

⇒ Pentru eroarea relativă: $\epsilon_r = \frac{t_{exp} - t_{ALGOR}}{t_{exp}} \cdot 100 [\%]$

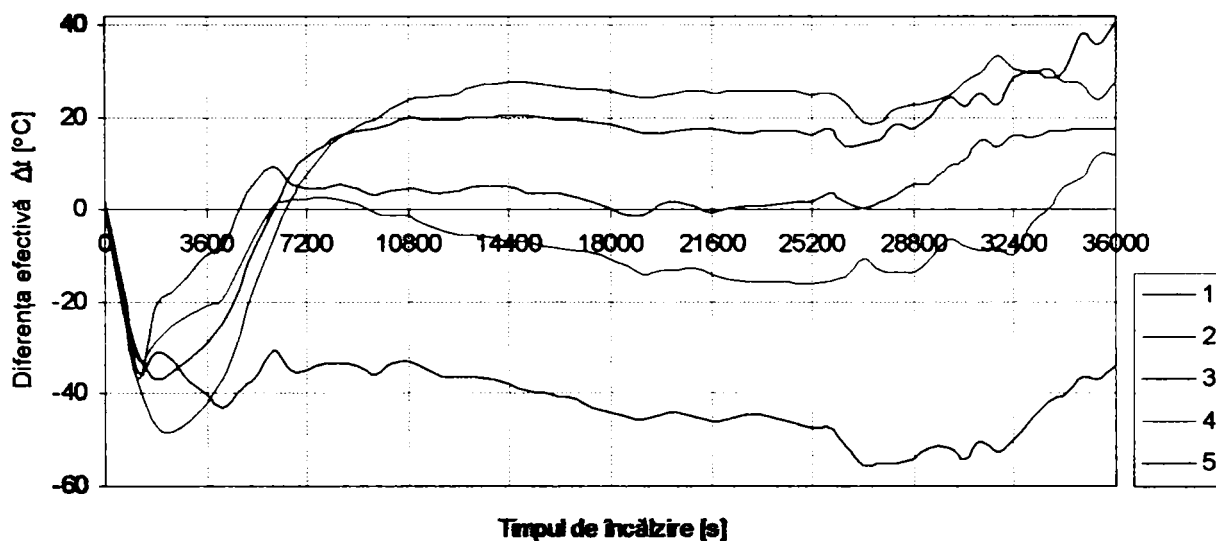


Fig.7.45. Diferența efectivă între valorile măsurate experimental și cele obținute cu ALGOR

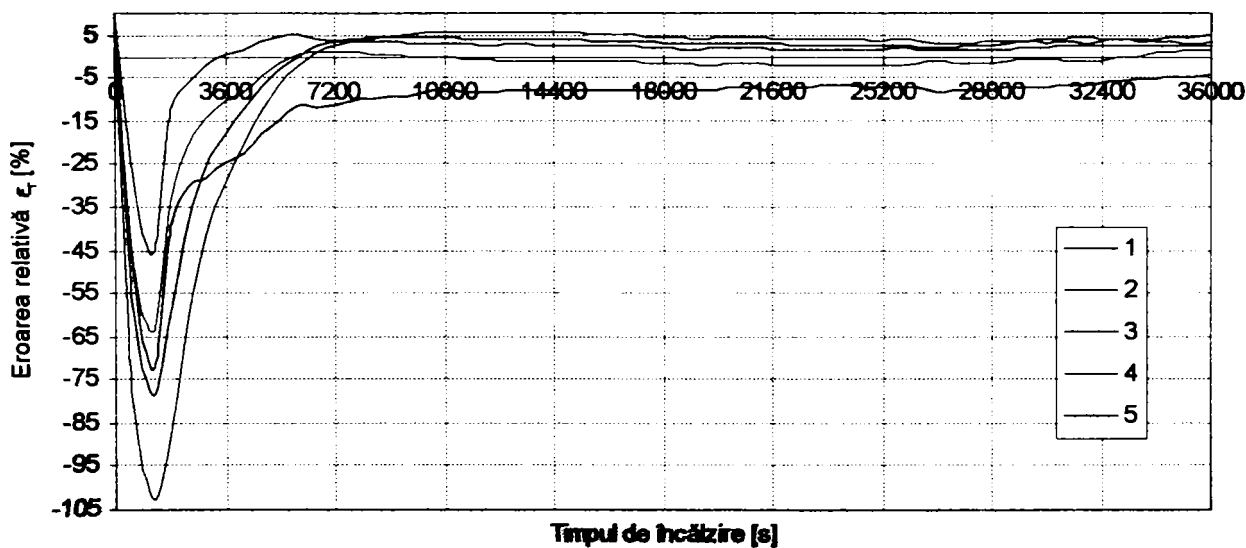


Fig.7.46. Erorile relative ale valorile obținute cu ALGOR față de cele măsurate experimental

□ Privind în ansamblu dinamica variației temperaturii măsurate experimental în cele cinci puncte, putem spune că valorile calculate cu programul ALGOR în toate cele cinci noduri caracteristice, corespunzătoare punctelor de măsurare, urmăresc fidel această variație.

□ Diferențele, în valoare absolută, între valorile experimentale și cele furnizate de ALGOR sunt mai mari pentru nodurile caracteristice corespunzătoare punctelor 2 și 5. Cauza esențială a acestor abateri semnificative (în special pentru punctul 5) o reprezintă poziția geometrică a punctelor în discuție (vezi figurile 4.7 și 4.9). După cum s-a arătat în capitolul 4, infiltrarea aerului fals prin fanta obținută intenționat datorită închiderii parțiale a ușii instalației face ca partea inferioară a blumului, împreună cu suprafața sa verticală din imediata vecinătate a ușii, să fie mai reci decât în cazul în care încălzirea ar fi avut loc cu ușa complet închisă. Pătrunderea aerului rece în interiorul incintei de lucru a instalației experimentale, cu consecințe reflectate în diluarea și deci răcirea zonală a gazelor arse, nu a putut fi luată în considerare în realizarea modelului numeric în ALGOR. Modelul numeric se apropie mai mult de realitate în zonele în care temperatura mediului gazos nu diferă semnificativ de cea măsurată experimental cu termocuplul notat cu nr.0 în figura 4.9. Un exemplu în acest sens îl reprezintă punctele 1 și 4.

□ În altă ordine de idei, exceptând punctul 5, în toate celelalte cazuri, la începutul perioadei de încălzire (după aproximativ 1...2 h) valorile calculate cu ALGOR sunt mai mari decât cele măsurate experimental, urmând ca după această perioadă situația să se inverseze. Explicația constă în faptul că, în realitate, temperatura suprafeței interioare a zidăriei, considerată suprafață radiantă termic, crește continuu pe toată durata încălzirii, atingând o valoare maximă la sfârșitul acesteia. Modelul numeric însă, ia în calcul o valoare medie a temperaturii zidăriei, valoare care stă la baza calculului schimbului de căldură prin radiație între suprafața interioară a zidăriei și semifabricat. Mai mult, radiația de la zidărie este cu atât mai pregnantă cu cât temperatura este mai ridicată. Considerarea, într-o proporție mai mare decât în realitate, a radiației zidăriei în prima perioadă a încălzirii (prin atribuirea unei temperaturi mai mari decât cea reală pentru suprafața interioară a zidăriei) constituie motivul obținerii unor valori mai mari cu programul ALGOR decât cele măsurate, în prima perioadă a încălzirii.

□ Concentrându-ne atenția asupra figurii 7.46 putem spune că, în prima oră de încălzire, erorile relative ale modelului numeric sunt destul de mari, atingând chiar procente de 100 %, urmând ca în următoarele etape ale încălzirii acestea să se micșoreze semnificativ, încadrându-se într-o plajă acceptabilă de 5 % (excepție făcând nodul caracteristic corespunzător punctului 5 de măsurare).

□ Ca și o concluzie generală relativ la valorile de temperatură obținute cu modelul numeric implementat în ALGOR, putem spune că acesta furnizează date exacte, comparativ cu cele măsurate experimental și considerate de referință (exceptând perioada de început a încălzirii), simularea numerică dovedindu-se și în acest caz o alternativă viabilă în studiul fenomenului de transfer termic. Faptul că modelul numeric se apropie mult de realitate este dovedit, odată în plus și de imaginile din figura 7.47.

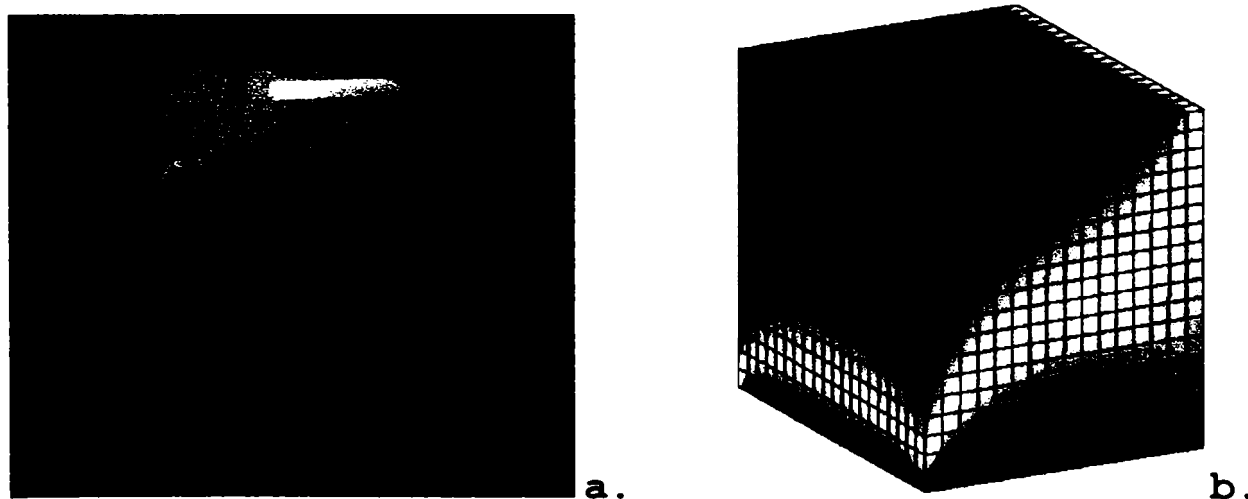


Fig.7.47. Comparație vizuală între câmpul termic stabilit în blum în cazul fenomenului real de încălzire (a) și câmpul termic obținut cu modelul numeric în ALGOR (b)

□ Trebuie precizat faptul că, în figura 7.47, comparația s-a făcut pentru momentul de sfârșit al încălzirii, moment care, pentru cazul real, coincide cu cel în care a avut loc întreruperea alimentării cu combustibil a instalației experimentale și deschiderea completă a ușii (figura 7.47.a), iar pentru modelul numeric reprezintă pasul 60 de timp. Sunt evidente, în ambele cazuri, efectele încălzirii asimetrice și neuniforme datorită poziției arzătorului relativ la spațiul incintei de lucru, precum și cele datorate contactului direct al semifabricatului cu vatra.

□ În ceea ce privește calculul tensiunilor termice apărute în interiorul materialului, putem concluziona că, pe tot intervalul de încălzire, acestea au valori mult sub limitele admisibile. Astfel, nu se pune problema apariției fisurilor sau deteriorării semifabricatului. Acest lucru se datorează, pe de o parte materialului din care este realizat blumul (oțel carbon), iar pe de altă parte vitezei moderate de încălzire (gradienți termici pe secțiune – în limite normale). După cum se poate observa, programul a calculat valori rezultante ale tensiunilor termice, după teoria a V-a de rezistență, deci tensiuni rezultante von Mises, în $[N/m^2]$.

□ De asemenea, sunt puse la dispoziția utilizatorului date referitoare la mărimea deformațiilor în fiecare nod al rețelei de discretizare și la fiecare pas de timp. Am considerat ca fiind semnificative deformațiile apărute la mijlocul perioadei de încălzire, motiv pentru care distribuția acestora a fost arătată doar pentru acel moment (figura 7.36). Pentru o vedere de ansamblu și mai intuitivă, în figura 7.36 s-a prezentat structura deformată a semifabricatului, la o scară mult mărită. Se poate observa că în zona de contact a blumului cu vatra, unde temperaturile sunt mai scăzute, dilatările sunt mult mai mici decât în întreaga masă a materialului. Această imagine (figura 7.36) scoate în evidență, odată în plus, efectul negativ, din punct de vedere al uniformității încălzirii, al contactului direct cu vatra a semifabricatului.

□ În final, pot preciza faptul că, folosind programul ALGOR, am descoperit că acesta nu are doar capacitatea de analiză de care este nevoie în studiul modelării câmpurilor termice la încălzirea oțelurilor dar, în plus, posedă una din cele mai intuitive și ușor de utilizat interfețe grafice, în comparație cu celelalte pachete FEA (Finite Element Analysis), împreună cu o serie întreagă de facilități deosebit de utile în procesul de postprocesare. Rezultatele obținute de mine cu ALGOR prezintă o eroare, în raport cu măsurătorile experimentale, ce se încadrează într-un procent de 5...7%. Pot spune că aceste rezultate sunt cât se poate de satisfăcătoare excepție făcând cele obținute la începutul perioadei de încălzire. Eroarea mare a acestora nu se datorează însă neajunsurilor legate de program, ci ipotezelor simplificatoare utilizate în construirea modelului.

Capitolul 8

Concluzii finale, contribuții personale, perspective

- 8.1. Concluzii finale
- 8.2. Contribuții personale
- 8.3. Perspective

Capitolul 8

Concluzii finale, contribuții personale, perspective

8.1. Concluzii finale

Scopul acestui studiu constă în obținerea unui model de simulare numerică, general valabil și aplicabil tuturor cazurilor particulare de încălzire a oțelurilor în cuptoare cu flacără, model cu ajutorul căruia să poată fi studiată, prin intermediul computerului, posibilitatea optimizării transferului termic în cazurile respective, cu un cost minim (necesar acoperirii cheltuielilor cu echipamentul de calcul și software) și într-un timp foarte scurt.

Pentru realizarea țelului propus au fost atinse următoarele obiective:

⇒ prezentarea unui studiu legat de conceptele moderne referitoare la modelarea și simularea fenomenelor și proceselor în general, din care reiese ideea de bază conform căreia simularea este un domeniu de investigare aflat la confluența matematicii, teoriei sistemelor și informaticii, cu o metodologie proprie (bazată pe identificarea proceselor, modelarea sistemelor și teoria algoritmilor) și cu mijloace specifice (oferite de echipamentul de calcul automat, de aparatura electronică de măsură, de prelevare a datelor și de verificare a rezultatelor). Ea pune la dispoziția cercetătorului o metodă prin care acesta să-și poată verifica puterea de înțelegere a fenomenelor din jurul său și îl ajută să obțină rezultate mai bune într-un timp mult mai scurt. De asemenea reprezintă un instrument eficient de comunicare, arătând modul de desfășurare al unui proces și stimulează gândirea creativă în legătură cu posibilitățile de îmbunătățire și optimizare a acestuia.

⇒ structurarea bazei teoretice a transferului termic în cuptoarele industriale. În acest sens a fost evidențiat caracterul extrem de complex al proceselor de schimb de căldură care au loc în interiorul unui cuptor industrial de încălzire cu flacără, cu prezentarea fenomenelor care apar la exteriorul încărcăturii metalice (schimbul termic de la mediul de încălzire spre încărcătură) și respectiv interiorul acesteia (transferul de energie termică de la suprafață spre interiorul încărcăturii);

⇒ prezentarea metodelor numerice utilizate în tratarea problemelor de transfer termic, realizându-se o comparație între metoda diferențelor finite și cea cu elemente finite. Au fost evidențiate astfel avantajele și dezavantajele fiecărei metode.

⇒ realizarea unei instalații experimentale pilot care să reproducă, la scară redusă, fenomenele specifice încălzirii unui semifabricat din oțel și care să permită efectuarea unor măsurători experimentale, dificil sau chiar imposibil de realizat la scară industrială, pe modele reale. Metoda modernă folosită pentru achiziția semnalelor transmise de termocupluri, utilizând modulul ADAM-4018 și respectiv convertorul de semnal ADAM-4520, face posibilă obținerea unor date exacte, erorile de măsurare reducându-se considerabil și datorită software-ului de prelucrare a datelor care permite realizarea unor corecții automate funcție de condițiile concrete de măsurare. De asemenea frecvența de achiziție a semnalului la fiecare 5 s, pe întreaga durată de încălzire de 10 h, oferă posibilitatea urmăririi fidele a modificărilor de temperatură care apar.

⇒ elaborarea a trei modele de simulare a transferului termic, pe baza fenomenelor reale din instalația experimentală pilot. Având la dispoziție date concrete privind evoluția câmpului termic în interiorul semifabricatului supus încălzirii, date obținute prin măsurători experimentale directe, s-a dorit realizarea unei comparații între acestea și cele obținute virtual, cu modelele concepute în lucrare.

În continuare sunt prezentate comparativ rezultatele măsurătorilor experimentale și cele obținute cu modelele analitic, numeric cu diferențe finite și numeric cu element finit, pentru cele cinci puncte de măsurare caracteristice.

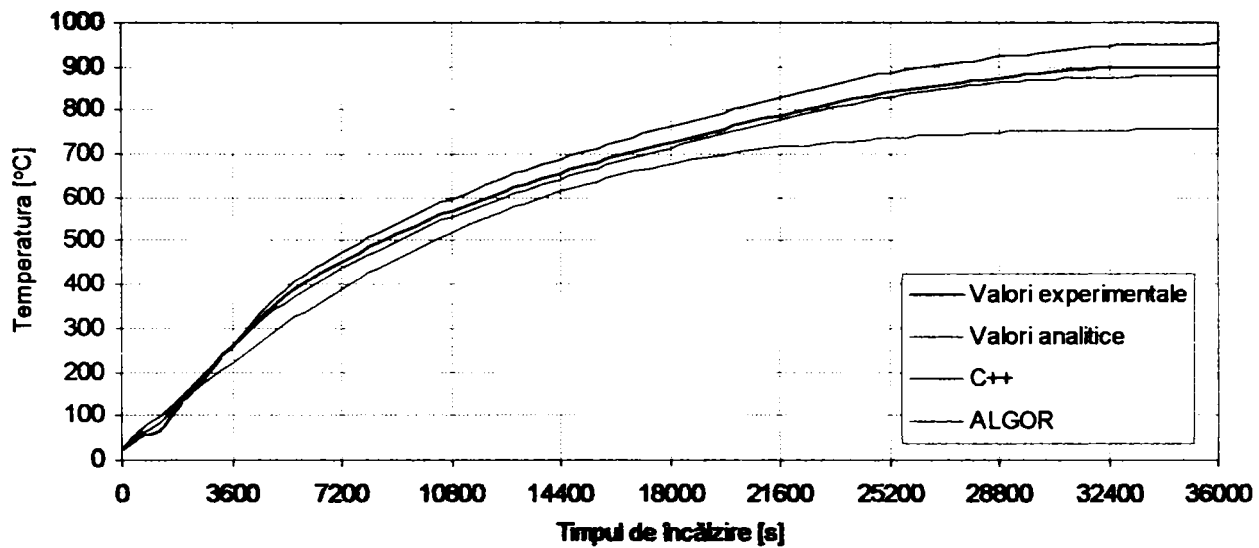


Fig.8.1. Evoluție comparativă a valorilor de temperatură pentru punctul 1

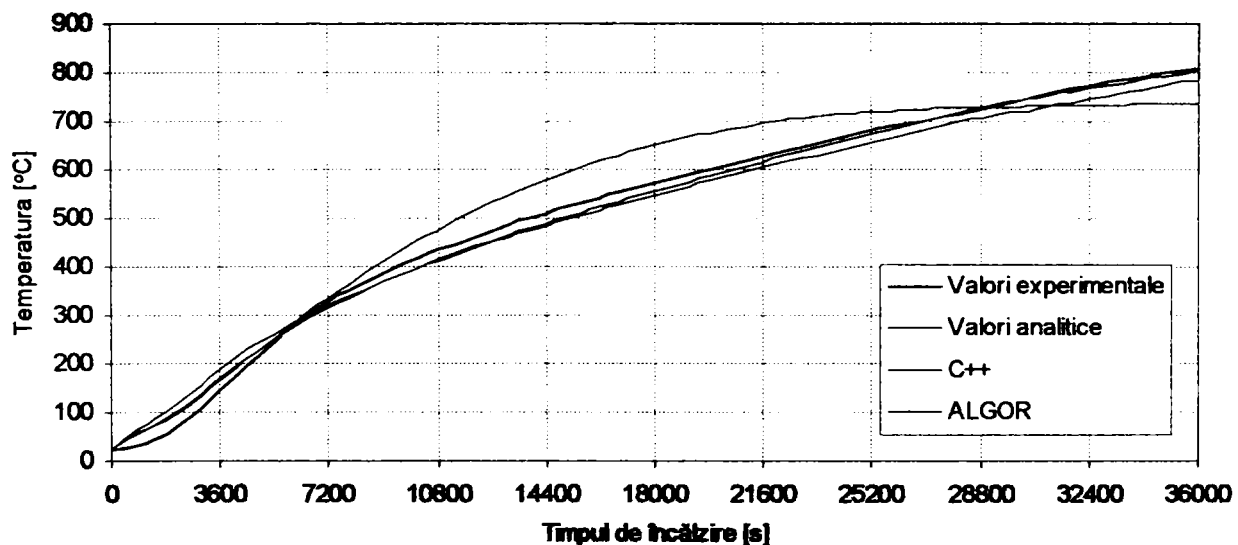


Fig.8.2. Evoluție comparativă a valorilor de temperatură pentru punctul 2

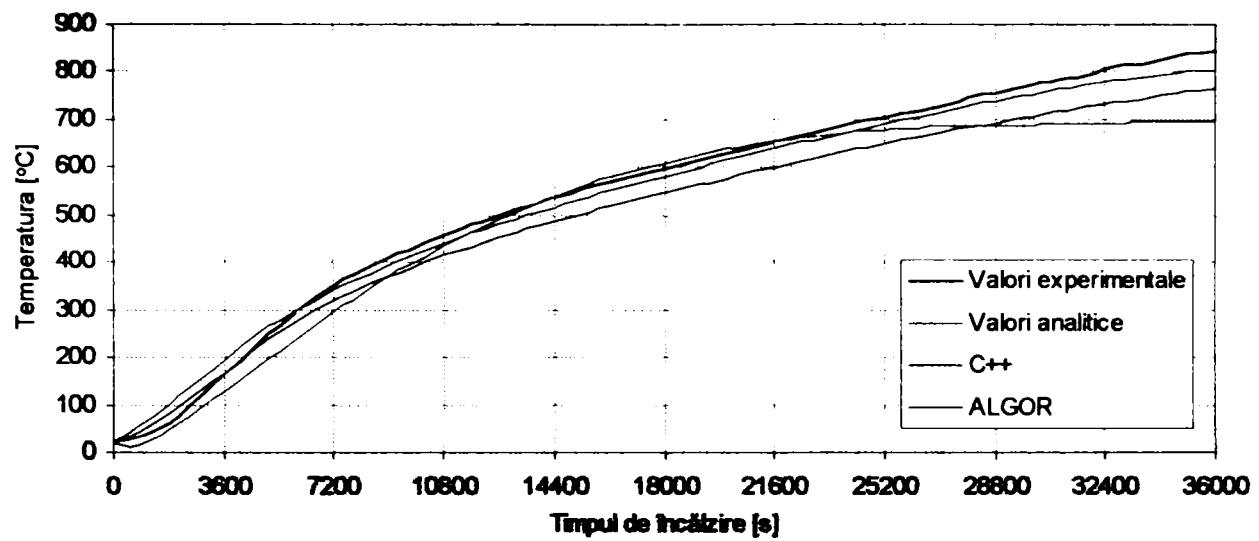


Fig.8.3. Evoluție comparativă a valorilor de temperatură pentru punctul 3

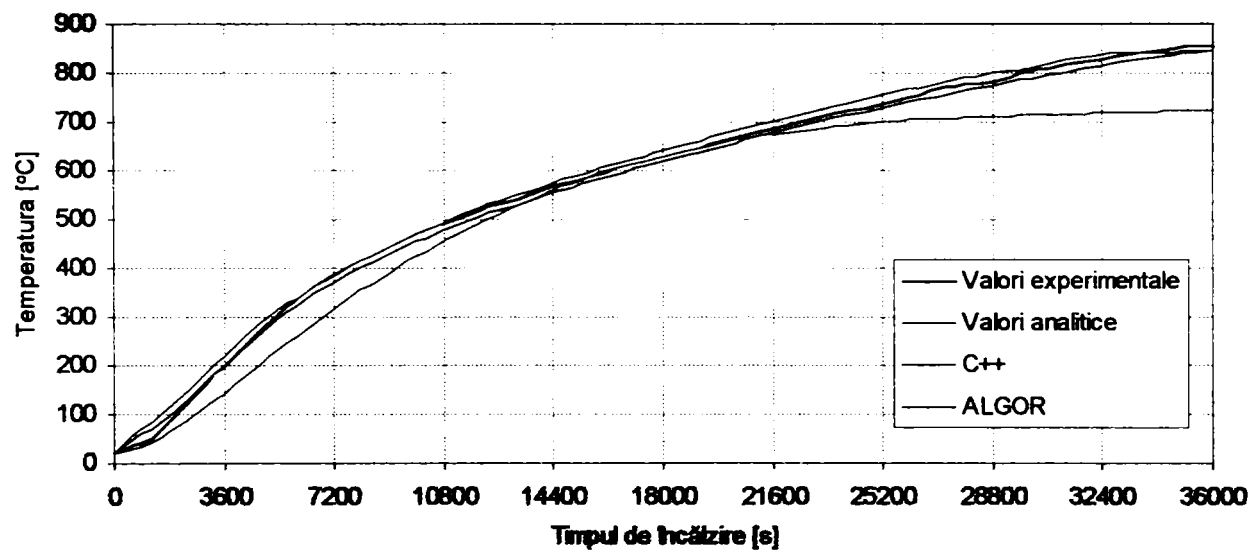


Fig.8.4. Evoluție comparativă a valorilor de temperatură pentru punctul 4

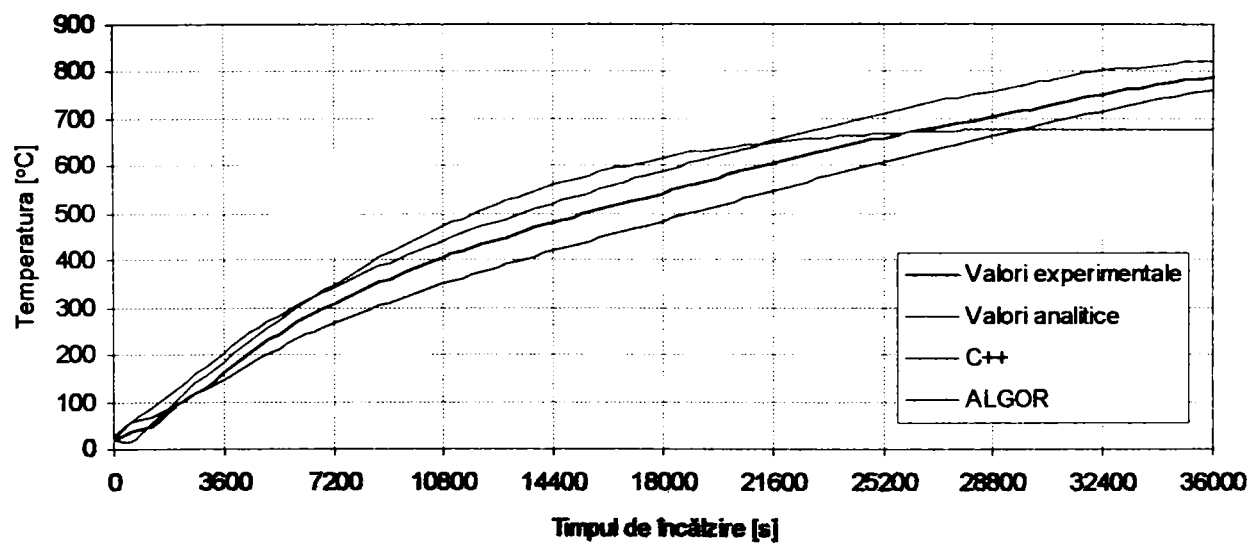


Fig.8.5. Evoluție comparativă a valorilor de temperatură pentru punctul 5

□ Aruncând o privire de ansamblu asupra rezultatelor obținute cu cele trei modele de simulare se poate observa o diferență semnificativă între valori, lucru explicat prin modul conceptual diferit care a stat la baza elaborării modelelor.

□ Pentru rezolvarea analitică a ecuației conducției termice în interiorul blumului s-a construit un model de simulare simplificat, având la bază o mulțime de ipoteze care au permis obținerea soluției analitice prin metoda seriilor Fourier. Ipoteza încălzirii simetrice a blumului, precum și considerarea constantelor termo-fizice de material ca fiind invariabile cu temperatura, au condus la erori considerabile. De aceste erori se face responsabilă și ipoteza conform căreia fenomenul de radiație termică este complet neglijat, încălzirea având loc exclusiv convectiv, cu un coeficient de convecție mare, astfel încât suprafața exterioară a metalului să poată urmări fidel variația de temperatură a atmosferei incintei de încălzire. Se poate spune că aceste ipoteze au fost impuse de aparatul matematic folosit. Pe de altă parte, erorile mari înregistrate în cazul valorilor corespunzătoare sfârșitului perioadei de încălzire sunt datorate convergenței seriilor Fourier pentru valori mari ale variabilei timp.

□ Modelul numeric cu diferențe finite este conceput și implementat de către autor în mediul de programare C++. În acest caz, ecuația diferențială cu derivate parțiale este transformată într-o ecuație cu diferențe finite rezolvabilă de către calculatorul electronic. Metoda permite eliminarea unor ipoteze simplificatoare și în consecință obținerea unor rezultate mai puțin eronate decât cele furnizate de modelul analitic.

□ Modelul numeric cu elemente finite se dovedește a fi cel mai în măsură să simuleze fenomenul real de încălzire. Implementat cu ajutorul unui program profesional – ALGOR, modelul furnizează date apropiate de cele obținute prin măsurători experimentale, ipotezele simplificatoare fiind reduse la minimum. Interfața grafică completă, posibilitățile multiple de particularizare a fenomenelor specifice, precum și numărul mare de facilități deosebit de utile în faza de postprocesare, îl recomandă în studiul virtual al acestui tip de problemă.

□ În concluzie se poate spune că alegerea metodei de simulare a fenomenului de transfer de căldură în incintele instalațiilor de încălzire trebuie făcută cu grijă, în urma unui studiu atent al particularităților care apar. În cazul în care introducerea unor ipoteze simplificatoare nu conduce la diferențe esențiale între fenomenul real și model, simularea poate fi realizată și cu ajutorul modelului analitic. Dacă însă fenomenul real nu permite simplificări, iar rezultatele simulării se doresc a fi mai mult decât orientative, pe baza lor dezvoltându-se procese de optimizare, este de dorit a se utiliza simularea numerică, fie folosind metoda cu diferențe finite, fie cea cu element finit, datorită preciziei rezultatelor furnizate. În plus, dacă se dispune de o licență pentru un program specializat, împreună cu suportul hardware adecvat, este preferabil utilizarea acestuia datorită multiplelor facilități oferite.

8.2. Contribuții personale

- ⇒ 1. Realizarea unei sinteze, pe baza unui studiu vast, cuprinzând un număr important de repere bibliografice și informații culese din rețeaua Internet, a problemicii modelării și simulării fenomenelor și proceselor în general și a celor legate de transferul termic, prin metode numerice, în particular.
- ⇒ 2. Expunerea clară și concisă a fundamentelor teoretice privitoare la fenomenele complexe de transfer termic la încălzirea semifabricatelor din oțel în cuptoare cu flacăra.
- ⇒ 3. Extragerea din literatura de specialitate a problemelor teoretice esențiale legate de metodele numerice utilizate în lucrare.

- ⊖ 4. Conceperea și realizarea practică a unei instalații experimentale pilot care să reproducă, la scară redusă, fenomenele specifice încălzirii unui semifabricat din oțel și care să fie capabilă să furnizeze toate datele necesare dezvoltării și validării modelelor numerice implementate ulterior.
- ⊖ 5. Realizarea modelului geometric tridimensional (simularea geometrică) la nivel de detaliu al instalației experimentale utilizând pachetele software AUTOCAD 2002, Mechanical Desktop și Autodesk Inventor.
- ⊖ 6. Conceperea, pe baza informațiilor teoretice din literatura de specialitate, a unui model analitic al fenomenului studiat și implementarea acestuia în software-ul matematic MATLAB.
- ⊖ 7. Conceperea și dezvoltarea, în mediul de programare C++, a unui model de simulare numerică cu metoda diferențelor finite a transferului termic stabilit în interiorul semifabricatului din oțel supus încălzirii, respectând condițiile de desfășurare a fenomenelor din instalația experimentală.
- ⊖ 8. Realizarea unui model de simulare numerică, folosind metoda elementului finit, a fenomenelor complexe de transfer termic din interiorul spațiului de lucru al instalației experimentale, cu ajutorul pachetului software profesional ALGOR.
- ⊖ 9. Compararea rezultatelor obținute prin cele trei metode (rezolvarea analitică și rezolvările numerice prin diferențe finite și respectiv element finit) și validarea acestora de către valorile furnizate de măsurătorile experimentale.

8.3. Perspective

⊖ Lucrarea de față deschide un drum nou în studiul încălzirii oțelurilor în vederea deformării plastice. Realizarea unui model de simulare numerică, validat prin date experimentale, reprezintă punctul de plecare în procesul de optimizare a fenomenului considerat. În cazul de față, modelul numeric corespunzător fenomenului de transfer termic din incinta de lucru a instalației experimentale pilot (prezentată în lucrare) poate fi extrapolat și adaptat cazurilor concrete din cuptoarele industriale folosite la încălzirea semifabricatelor ce urmează a fi deformate la cald prin diverse procedee.

⊖ Modelul implementat cu ajutorul calculatorului oferă posibilitatea cercetătorului să experimenteze virtual diferite situații care, în condiții reale de exploatare, ar fi imposibil de realizat. Astfel, realizată într-un timp foarte scurt și implicând costuri minime, simularea fenomenului de încălzire duce la obținerea unor date care pot fi folosite în diverse scopuri.

⊖ În cazul cuptoarelor aflate în uz, programul de simulare poate fi pus în legătură cu un altul care permite modificarea și respectiv corectarea în timp real a parametrilor tehnologici și funcționali ai cuptorului, funcție de datele furnizate de modelul de simulare, în scopul optimizării încălzirii semifabricatelor.

⊖ În faza de proiectare a cuptoarelor, folosind modelul de simulare numerică adecvat, se pot experimenta diverse situații concrete în ideea alegerii soluției constructive optime.

⊖ Este evident faptul că modelele de simulare concepute în lucrare sunt perfectibile. Eliminarea, pe cât posibil a ipotezelor simplificatoare reprezintă un studiu de perspectivă și o preocupare continuă a autorului. Spre exemplu, modelul cu elemente finite poate fi îmbunătățit prin realizarea unei simulări numerice distincte a fenomenelor gazo-dinamice complexe care au loc în incinta de încălzire, folosind un program specializat în acest domeniu (*3D Flow, Fluent, etc.*). Rezultatele astfel obținute, concretizate prin stabilirea câmpului de viteze și temperaturi în masa de gaze arse ce înconjoară semifabricatul, pot deveni date inițiale pentru modelul de simulare propriu-zis. În acest fel determinarea coeficientului global de schimb de căldură între suprafața metalului și atmosfera care-l înconjoară devine mai precisă.

Bibliografie

- [1] Muntean, S. *Metode numerice pentru determinarea câmpurilor tridimensionale în rotoarele turbinelor Francis* - Teză de doctorat, Universitatea „Politehnica” din Timișoara, 2002;
- [2] Leca, A., ș.a. *Transfer de căldură și masă – O abordare inginerescă*, Editura Tehnică București, 1998;
- [3] Popa, B., Vintilă, C. *Termotehnică și mașini termice*, Editura Didactică și Pedagogică, București, 1977;
- [4] Negru, L.D. *Transmiterea căldurii și dinamica gazelor – Partea I*, Litografie Institutul Politehnic „Traian Vuia” Timișoara, 1990;
- [5] Brunklaus, J.H. *Cuptoare industriale – Seria Termo – Frig*, Editura Tehnică, București, 1977;
- [6] Samoilă, C., ș.a. *Cuptoare și instalații de încălzire*, Editura Didactică și Pedagogică, București, 1983;
- [7] Udrescu, L. *Transfer termic și masic la aliaje metalice*, Editura Politehnica, Timișoara, 1999;
- [8] Lienhard, J.H.IV, Lienhard, J.H.V *A Heat Transfer Textbook*, Third Edition, Phlogiston Press, Cambridge, Massachusetts;
- [9] Goldfarb, E.M., ș.a. *Calculul cuptoarelor de încălzire – sub redacția Prof.N.Iu. Taiț*, Traducere din limba rusă, Editura Tehnică, București, 1960;
- [10] Ilca, I., ș.a. *Contribuții la studiul încălzirii lingourilor în cuptoare industriale*, Revista Metalurgia, nr.10, 1982;
- [11] Rațiu, S., ș.a. *Considerații cu privire la încălzirea oțelurilor de calitate*, Sesiunea de comunicări științifice cu participare internațională, Hunedoara, octombrie 2000, pag. 122...129;
- [12] Popescu, N., Vintilescu, C. *Tehnologia tratamentelor termice*, Editura Tehnică, București, 1974;
- [13] Dragotă, I., Petrehuș, V. *Metode numerice pentru ecuații diferențiale*, Editura Orizonturi Universitare, Timișoara, 2002;
- [14] Marinescu, M., ș.a. *Termodinamică tehnică*, volumul 3, Editura Matrix Rom, București, 1998;
- [15] Olariu, V., Brătianu, C. *Modelarea numerică cu elemente finite*, Editura Tehnică, București, 1986;
- [16] Gârbea, D. *Analiză cu elemente finite*, Editura Tehnică, București, 1990;
- [17] Rațiu, S. *Instalație experimentală pentru măsurarea temperaturii semifabricatelor din oțel supuse încălzirii în vederea deformării plastice*, Analele Facultății de Inginerie Hunedoara, 2002, Tomul IV, Fascicola 3, pag. 67...70;
- [18] SR EN 10025 + A1:1994 *Produse laminate la cald din oțeluri de construcție nealiatate. Condiții tehnice de livrare;*

- [19] STAS 500/2 – 80 *Oțeluri de uz general pentru construcții*. Mărci;
- [20] * * * *Manualul inginerului termotehnician*, volumul 3, Editura Tehnică, București, 1986;
- [21] STAS 8703 – 70 *Mijloace pentru măsurarea temperaturii*. Terminologie;
- [22] STAS 8421 – 87 *Mijloace de măsurare a temperaturii*. Termocupluri tehnice. Condiții tehnice generale de calitate;
- [23] Asavinei, I., Niculescu, C. *Ghid pentru utilizarea termocuplurilor în măsurători industriale*, Editura Tehnică, București, 1981;
- [24] Rațiu, S., Nica, G. *Metode directe în rezolvarea problemelor neliniare de transfer termic la încălzirea semifabricatelor în cuptoare cu propulsie*, Știință și Inginerie, Lucrările celei de A doua Conferințe Naționale «Prof. Dorin Pavel – fondatorul hidroenergeticii românești» Sebeș 2002, Editura AGIR, București 2002, pag. 189...194;
- [25] Godounov, S. *Équations de la Physique Mathématique*, Éditions MIR, Moscou, 1973;
- [26] Rațiu, S. *Utilizarea seriilor Fourier în rezolvarea ecuației conducției termice în regim nestaționar*, Analele Facultății de Inginerie Hunedoara, 2002, Tomul IV, Fascicola 3, pag. 71...78;
- [27] Maksay, Șt.I. *Calcul numeric*, Editura Politehnica, Timișoara, 2002;
- [28] Smirnov, V. *Cours de Mathématiques Supérieures*, Tome II, Éditions MIR, Moscou, 1970;
- [29] Ghinea, M., Firețeanu, V. *Matlab. Calcul numeric – Grafică – Aplicații*, Editura Teora, 1999;
- [30] Rațiu, S., ș.a. *The solving of a non-linear problem of thermal transfer by a particular analytical method*, 7-th International Symposium of Interdisciplinary Regional Research, Hungary-Romania-Yugoslavia, Hunedoara, 25-26 September 2003, pag. 689...694;
- [31] Rațiu, S., Kiss, I. *Particular Method for the Analytical Solution of a Non-Linear Temperature Transfer Problem*, Revista MAŠINSTVO -Journal of Mechanical Engineering, nr.4, 2003 – Zenica, pag. 209...216;
- [32] Tripșa, I., ș.a. *Optimizarea proceselor de turnare, solidificare și încălzire a lingourilor de oțel*, Editura Tehnică, București, 1975;
- [33] Rațiu, S., ș.a. *Modelarea numerică a fenomenului de conducție termică*, Sesiunea de comunicări științifice cu participare internațională, Hunedoara, octombrie 2000, pag. 130...135;
- [34] Prejban, I. *Tratamente termice – Îndrumar pentru lucrări de laborator și proiect*, litografia UPTVT, 1990;
- [35] Usache, M., Chircă, D. *Proprietățile metalelor*, Editura Didactică și Pedagogică, București, 1982;
- [36] Dulămiță, T., Florian, E. *Tratamente termice și termochimice*, Editura Didactică și Pedagogică, București, 1982;
- [37] Pănoiu, M., Pănoiu, C. *Bazele informaticii*, Editura Mirton, Timișoara, 2002;
- [38] Pănoiu, C., ș.a. *Utilizarea și programarea calculatoarelor*, Editura Mirton, Timișoara, 2002;
- [39] Swan, T. *Învățăm C...pas cu pas – Type & Learn C*, traducere de Dan Somnea, Editura Tehnică, București, 1996;

- [40] Jamsa, K. *C++ Manualul începătorului*, Editura Teora, București, 1999;
- [41] Negrescu, L. *Limbajele C și C++ pentru începători*, vol. I și II, Editura Microinformatica, Cluj-Napoca, 1999;
- [42] Jamsa, K., Klander, L. *Totul despre C și C++*, Editura Teora, București, 2000;
- [43] Pătruț, B. *Aplicații în C și C++*, Editura Teora, București, 2000;
- [44] Negrescu, L. *Limbajul Turbo C*, Editura Libris, Cluj-Napoca, 1992;
- [45] Muscalagiu, D., ș.a. *Culegere de probleme în limbajul C*, Editura Mirton, Timișoara, 2001;
- [46] Rațiu, S., ș.a. *Rezolvarea ecuației conducției termice prin metoda elementului finit*, Sesiunea de comunicări științifice cu participare internațională, Hunedoara, octombrie 2000, pag. 136...143;
- [47] * * * *Mică enciclopedie matematică*, Editura Tehnică, București, 1980;
- [48] Rațiu, S., ș.a. *Studiul evoluției câmpului termic în lingourile încălzite în vederea deformării plastice, cu ajutorul metodei elementului finit*, Lucrările științifice ale Simpozionului Internațional "Universitaria Ropet 2001" Petroșani, octombrie 2001, ISBN 973-85398-5-4, pag. 207...210;
- [49] Rațiu, S. *Studiu comparativ al rezultatelor obținute pe cale numerică și determinate experimental, în cazul câmpului termic stabilit într-un lingou încălzit în vederea deformării plastice*, Conferința Națională de Metalurgie și Știința Materialelor, București, septembrie 2001, ISBN 973-652-389-6, pag. 363...368;
- [50] Rațiu, S., ș.a. *Comparative analysis of the results obtained regarding the problems concerning the thermal transfer when heating the billets for future rolling*, ZBORNIK RADOVA SA NAUČNO-STRUČNOG SKUPA ISTRAŽIVANJE I RAZVOJ MAŠINSKIH ELEMENTA I SISTEMA, JAHORINA-IRMES'2002, 2/2, SRPSKO SARAJEVO-JAHORINA, 19 i 20 Septembar 2002, pag. 553...558;
- [51] Rațiu, S., ș.a. *Aspecte privind studiul transferului termic la încălzirea țagelilor în cuptoare cu propulsie*, Lucrările științifice ale Simpozionului Internațional "Universitaria Ropet 2002" Petroșani, octombrie 2002, ISBN 973-8435-22-6, pag. 157...160;
- [52] * * *
- [53] * * * *ALGOR – User's Guide*
- [54] Jurca, I. *Programarea orientată pe obiecte*, Editura Eurobit, Timișoara, 1992
- [55] Mușlea, I. *Programarea orientată pe obiecte*, Editura Ardealul, Cluj-Napoca, 1992;
- [56] Apostol, C., ș.a. *Introducere în programare*, Editura Viața Românească, București, 1993;
- [57] Mocanu, M., ș.a. *333 de probleme de programare*, Editura Teora 1993;

- [58] Litvovschi, L., Georgescu, H. *Analiza și sinteza algoritmilor*, Editura Științifică și Enciclopedică, București, 1986;
- [59] Roșca, I., ș.a. *Introducere în programarea și utilizarea calculatoarelor*, Editura Alfar, Rm. Vâlcea, 1992;
- [60] Norton, P., Goodman, J. *PC – Totul despre calculatoarele personale*, Editura Teora, București, 2000;
- [61] Kelly, J. *Utilizare Microsoft Excel 2000*, Editura Teora, București, 2000;
- [62] Blattner, P. *Microsoft Excel 2000*, Editura Teora, București, 2002;
- [63] Habraken, J. *Microsoft Office 2000, 8 în 1*, Editura Teora, București, 2000;
- [64] Ionesc, B., ș.a. *Instrumente informatice pentru birotică – Word, Power Point, Outlook, Front Page*, Editura DualTech, 2000;
- [65] Florescu, V., ș.a. *Baze de date – Fundamente teoretice și practice*, Editura Infomega, București, 2002;
- [66] Tamaș, I., ș.a. *Bazele informaticii*, Editura Infomega, București, 2000;
- [67] Ionescu, B., ș.a. *Aplicații informatice pentru birotică și comunicații*, Editura Infomega, București, 2001;
- [68] Ionescu, I., ș.a. *Birotică 2000*, Editura Infomega, București, 2001;
- [69] * * * *AutoCAD 12*, Editura Teora, București, 1995;
- [70] Stăncescu, C. *AutoLisp*, Editura Fast 2000, București, 1996;
- [71] Popescu, A., ș.a. *AutoCAD*, Editura Teora, București, 1994;
- [72] Manolea, D. *Practica în AutoCAD 2D*, Editura Microinformatica, Cluj-Napoca, 1994;
- [73] Burchard, B., ș.a. *Secrete AutoCAD 14*, Editura Teora, București, 1998;
- [74] Harington, D., ș.a. *AutoCAD 2002*, Editura Teora, București, 2002;
- [75] Miklos, Zs., Cioată, V.G. *Desenare 2D cu AutoCAD 2002*, Editura Mirton, Timișoara, 2003;
- [76] Pozdărcă, A., ș.a. *AutoCAD – Modelare spațială*, Editura Universității „Petru Maior”, Tg. Mureș, 2001;
- [77] Simion, I. *AutoCAD 2000 – Aplicații*, Editura Teora, București, 2000;
- [78] * * * *AutoCAD 2002 – User Guide*;
- [79] * * * *AutoCAD Mechanical 2002 – User Guide*;
- [80] * * * *Mechanical Desktop R6 – User Guide*;
- [81] * * * *AutoDESK Inventor 5, - User Guide*;
- [82] Ilca, I., Rațiu, S. *Numerical simulation of heat transfer at ingot heating*, Simpozionul anual al Institutului de Mecanica Solidelor SISOM 2000, Bucuresti, octombrie 2000, pag. 123...128;
- [83] Rațiu, S., Pinca, B.C. *The Analysis of the Thermal field of a Shears Blade used in hot Cutting of Blooms, by means of the Finite Elements Method*, Buletinul Științific al UPT, 2001, Tom 46(60) Fascicola 1, ISSN 1224-6077, pag. 7...12;
- [84] Pinca, B.C., Rațiu, S. *Analytical Study of the Temperature field from a Shear Knife for Blooms hot Cutting*, Buletinul Științific al UPT, 2001, Tom 46(60) Fascicola 1, ISSN 1224-6077, pag. 1...6;
- [85] Rațiu, S. *Rezolvarea unei probleme neliniare de transfer termic printr-o metodă analitică particulară*, Al IV-lea Simpozion Internațional „Tinerii și cercetarea multidisciplinară” ACM-V, 14-15 nov. 2002, Timișoara;

- [86] Rațiu, S. *Scurtă prezentare a programului COSMOS/M 2.5., utilizat la calculul câmpurilor de temperaturi*, A III-a Sesiune de Comunicări Științifice cu participare internațională, Hunedoara octombrie 2001, pag. 62...65;
- [87] Puțan, V., Rațiu, S. *Rezolvarea ecuațiilor diferențiale caracteristice proceselor termogazodinamice*, Analele Facultății de Inginerie din Hunedoara, Tom IV, Fascicola 1, Hunedoara – 2002, pag. 103...106;
- [88] Simionescu, I., ș.a. *Metode numerice în tehnică*, Editura Tehnică, București, 1995;
- [89] Vraciu, G., Popa, A. *Metode numerice cu aplicații în tehnica de calcul*, Scrisul românesc, Craiova, 1982;
- [90] Cristici, B. *Matematici speciale*, litografie IPTVT, 1971;
- [91] Cosma, D., ș.a. *Optimisation of Solidification, Cooling and Reheating of Steel Ingots*, Steel Times, October, 1972;
- [92] Minkowycz, W.J., ș.a. *Handbook of Numerical Heat Transfer*, John Wiley & Sons, 1988;
- [93] Minkowycz, W.J., ș.a. *Advances in Numerical Heat Transfer, Volume I*, Taylor & Francis, 1996;
- [94] Cebeci, T., Bradshaw, P. *Physical and Computational Aspects of Convective Heat Transfer*, Springer - Verlag, 1998;
- [95] Necati Ozisik, M. *Finite Difference Methods in Heat Transfer*, CRC Press, 1994;
- [96] Yogesh, J., Torrance, K. *Computational Heat Transfer*, Hemisphere Publishing Corporation, 1986;
- [97] Baehr, H.D., Stephan, K. *Heat and Mass Transfer*, Springer-Verlag, Berlin, 1998;
- [98] Mills, A.F. *Basic Heat and Mass Transfer*, Prentice-Hall, Inc., Upper Saddle River, NJ, 2nd edition, 1999;
- [99] Bejan, M. *În lumea unităților de măsură*, Editura AGIR, București, 2000;
- [100] Neacșu, I. *Metode și tehnici de învățare eficientă*, Editura Militară, București, 1990.

Anexa

Programul sursă C++

În program apar uneori referințe către clase a căror nume este prefixat cu "C" (de la *class*), de ex. CProgressDialog. Acestea sunt clase de interfață cu utilizatorul, cum sunt ferestrele de dialog – iar prefixarea este parte a convenției de numire folosită de Microsoft. De asemenea, există clase a căror nume începe cu "Abstract" (de ex. **AbstractWindow**) – acestea sunt clase abstracte (generice). Deoarece majoritatea codului din acest program (aprox. 65%) ține de programarea "pură", irelevantă pentru subiectul tezei, prezentarea acestuia va fi omisă.

MetalDataFile este cea mai importantă clasă a programului. Este responsabilă de gestionarea fișierelor care conțin datele simulărilor împreună cu toate datele inițiale și opțiunile folosite pentru rularea simulării. Se folosesc fișiere de date și nu memoria pentru că s-a dorit ca durata simulării și numărul de puncte de discretizare să fie arbitrare, ceea ce ar putea conduce la dimensiuni foarte mari ale datelor obținute.

Deoarece durata unei simulări poate fi de ordinul minutelor, este necesar să nu se blocheze tratarea mesajelor Windows de către ferestrele de dialog. Aceasta înseamnă că nu se poate porni o simulare ca urmare a unui simplu eveniment de tip "apasa butonul start", pentru că pe durata tratării evenimentului (minute) interfața ar fi blocată. Simulările sunt realizate prin lansarea unui fir de execuție (*thread*) paralel cu cel principal, ceea ce evită situația blocării.

Există o singură instanță a MetalDataFile în program.

```

//////////////////////////////////////
MetalDataFile.h - header file
//////////////////////////////////////
#ifndef __METAL_DATA_FILE_H
#define __METAL_DATA_FILE_H

#include "Material.h"
#include "SurfaceData.h"
#include "HeatFlowSequence.h"
#include "SequenceInterpolator.h"

typedef (*ProgressCallback)(LPVOID, HeatFlowSequence*);

#define SURF_IO 0
#define SURF_IM 4
#define SURF_JO 1
#define SURF_JM 3
#define SURF_K0 5
#define SURF_KM 2

class MetalDataFile
class MetalDataFile {
    friend class HeatFlowSequence;
public:    //antetul fisierului

```

```

LPCTSTR fileSignature;           //semnatura fisierului
int headerSize, seqSize;         //dimensiunea headerului si a secventei
int imax, jmax, kmax;           //nr. puncte discretizare
double tini;                     temperatura initiala
double dimx, dimy, dimz;         geometria
double alpha[6];                //alfa pt. cele 6 suprafete
double max_dh;                  //maximul variatiei entalpiei / iteratie
double max_dt;                  //maximul timpului / iteratie
double stop_time;               //timp de oprire [s]
double save_time_step;          //minimul intervalului intre 2 salvari pe disc

SurfaceData surface;
Material mat;                   //caracteristicile materialului
double max_time;                //durata totala
int steps, iterations;          //nr. de pasi / iteratii

private: //date de uz intern
DataFile data;                 //fisierul de date
BOOL candraw, fileIsTemp;
FloatBuffer timeBuffer;
double currentTime, savedTime;
int currentTimeStep;
SequenceInterpolator seqInt1, seqInt2;

private: //functii de uz intern
int calculateSeqSize();
void seekTime(float t);

public: //constructor & destructor
MetalDataFile();
~MetalDataFile();

public: //functii publice
void close();
void assign(LPCTSTR filename);
void open();
BOOL isOpen() {return data.isOpen();}
CString getFileName() {return data.getName();}
static void defaults(MetalDataFile *target);
BOOL writeHeader(DataFile *df = NULL);
BOOL readHeader(DataFile *df = NULL);
BOOL isTemp() {return fileIsTemp;}
BOOL canDraw() {return candraw;}
BOOL copy(LPCTSTR name);
void calculate(ProgressCallback pc, LPVOID param);
float getValue(float x, float y, float z, float t);
};

MetalDataFile *getMDF();
#endif//__METAL_DATA_FILE_H

//////////////////////////////////////
// MetalDataFile.cpp - class implementation
//////////////////////////////////////
#include "stdafx.h"
#include "MetalDataFile.h"
MetalDataFile metalDataFile;

```



```

MetalDataFile *getMDF() {return &metalDataFile;}

class MetalDataFile
MetalDataFile::MetalDataFile() {
    fileSignature= "MetalDataFile";
    currentTime = currentTimeStep = -1;
    candraw = FALSE; fileIsTemp = FALSE;
    defaults(this);
}
MetalDataFile::~MetalDataFile() {
    close();
    if (fileIsTemp) data.remove();
}
void MetalDataFile::defaults(MetalDataFile *target) {
    target->imax = target->jmax = target->kmax = 20;
    target->tini = 20;
    target->dimx = 0.24;
    target->dimy = 0.3;
    target->dimz = 0.27;
    for (int k = 0; k < 5; k++) target->alpha[k] = 1000;
    target->alpha[5] = 100000;
    target->mat.defaults();
    target->max_dh = 100;
    target->max_dt = 100;
    target->max_time = target->steps = 0;
    target->stop_time = 36000;
    target->save_time_step = 30;
}
void MetalDataFile::close() {
    if (data.canWrite()) writeHeader();
    candraw = FALSE;
    data.close();
}
void MetalDataFile::assign(LPCTSTR filename) {
    fileIsTemp = !filename;
    close();
    data.assign(filename);
}
int MetalDataFile::calculateSeqSize() {
    return 4*sizeof(double) + 3*sizeof(int) +
        imax * jmax * kmax * sizeof(float);
}
void MetalDataFile::open() {
    close();
    data.openRead(); if (!data.isOpen()) return;
    if (!readHeader()) {data.close(); return;}
    timeBuffer.empty(); timeBuffer.setSize(steps);
    int k;
    for (k = 0; k < steps; k++) {
        data.seek(headerSize + k*seqSize);
        timeBuffer[k] = data.readDouble();
    }
}
BOOL MetalDataFile::writeHeader(DataFile *df) {
    bool full = (df == NULL);
    if (full) df = &(MetalDataFile::data);
}

```

```

if (!df->isOpen()) df->openCreate();
if (!df->isOpen()) return FALSE;
df->seek(0);
df->writeString(fileSignature);
long sizepos = df->tell();
df->writeInt(0); !aici va fi dimensiunea headerului
seqSize = calculateSeqSize();
df->writeInt(seqSize);
df->writeInt(imax);
df->writeInt(jmax);
df->writeInt(kmax);
df->writeDouble(tini);
df->writeDouble(dimx);
df->writeDouble(dimy);
df->writeDouble(dimz);
df->write(alpha, 6*sizeof(double));
df->writeDouble(max_dh);
df->writeDouble(max_dt);
df->writeDouble(stop_time);
df->writeDouble(save_time_step);
df->writeDouble(full ? max_time : 0);
df->writeInt(full ? steps : 0);
mat.writeToFile(df);
headerSize = df->tell();
df->seek(sizepos);
df->writeInt(headerSize);
df->seek(headerSize);
df->flush(); return TRUE;
}
BOOL MetalDataFile::readHeader(DataFile *df) {
if (!df) df = &(MetalDataFile::data);
if (!df->isOpen()) df->openRead();
if (!df->isOpen()) return FALSE;
df->seek(0);
CString signature = df->readString();
if (signature != fileSignature) return FALSE;
headerSize = df->readInt(); seqSize = df->readInt();
imax = df->readInt();
jmax = df->readInt();
kmax = df->readInt();
tini = df->readDouble();
dimx = df->readDouble();
dimy = df->readDouble();
dimz = df->readDouble();
df->read(alpha, 6*sizeof(double));
max_dh = df->readDouble();
max_dt = df->readDouble();
stop_time = df->readDouble();
save_time_step = df->readDouble();
max_time = df->readDouble();
steps = df->readInt();
mat.readFromFile(df);

if ((headerSize != df->tell()) ||
(seqSize != calculateSeqSize()) ||
(df->size() != headerSize + seqSize*steps)) {
candraw = FALSE;

```

```

    df->close(); return FALSE;
}
candraw = (steps > 5);
return TRUE;
}
void MetalDataFile::seekTime(float t) {
    int tstep = timeBuffer.binarySearch(t);
    if (currentTimeStep != tstep) {
        data.seek(headerSize + seqSize * tstep);
        currentTimeStep = tstep;
        seqInt1.readFromFile(&data);
        seqInt2.readFromFile(&data);
    }
}
float MetalDataFile::getValue(float x, float y, float z, float t) {
    ASSERT(candraw && (x >= 0) && (x <= 1) && (y >= 0) &&
        (y <= 1) && (z >= 0) && (z <= 1));
    seekTime(t);
    float th1 = seqInt1.interpolate(x, y, z);
    float t1 = timeBuffer[currentTimeStep];
    float th2 = seqInt2.interpolate(x, y, z);
    float t2 = timeBuffer[currentTimeStep + 1];
    return ((t - t1)*th2 + (t2 - t)*th1)/(t2 - t1);
}
void MetalDataFile::calculate(ProgressCallback pc, LPVOID param) {
    iterations = 0; candraw = FALSE;
    close();
    if (fileIsTemp) data.remove();
    data.openCreate();
    writeHeader();
    HeatFlowSequence *oldhfs, *hfs;
    hfs = new HeatFlowSequence(this, NULL);
    hfs->calculate(); hfs->writeToFile();
    if (pc) pc(param, hfs);
    oldhfs = hfs;
    double savedTime = 0;
    BOOL cont = TRUE;
    steps = 1;
    DWORD callback_time, call_time;
    callback_time = call_time = GetTickCount();
    for (iterations = 1; cont; iterations++) {
        hfs = new HeatFlowSequence(this, oldhfs);
        hfs->calculate();
        if (hfs->time - savedTime >= save_time_step) {
            steps++;
            hfs->writeToFile();
            max_time = savedTime = hfs->time;
            call_time = GetTickCount();
            cont = (hfs->time < stop_time);
            if (pc && (call_time - callback_time >= 250)) {
                callback_time = call_time;
                if (!pc(param, hfs)) cont = FALSE;
            }
        }
        delete oldhfs; oldhfs = hfs;
    }
    delete hfs;
}

```

```

    open();
    if (pc) pc(param, NULL);
}

```

HeatFlowSequence realizează calculele pentru un moment de timp în funcție de momentul anterior. Deoarece rezultatul aplicării formulelor este o viteză de variație a entalpiei (dH/dt), iar modelul este cu atât mai precis cu cât această viteză este mai mică, s-a folosit un pas de timp variabil prin limitarea variației entalpiei la o iterație. După realizarea calculelor, **HeatFlowSequence** va scrie rezultatul în fișierul de date. Intern, **HeatFlowSequence** operează cu numere în virgulă flotantă dublă precizie (64 biți) pentru a minimiza propagarea erorilor – de fapt simularea este un șir foarte lung de calcule iterative. Mărirea și mai mult a preciziei nu este posibilă folosind compilatorul Microsoft, deoarece acesta nu suportă tipul **long double** (de fapt îl consideră identic cu **double**). Extern (în fișier) se reprezintă numai valorile temperaturilor ca virgulă flotantă simplă precizie (32 biți), fiindcă aici nu se mai pune problema propagării erorilor, însă se pune cea a unor fișiere excesiv de mari.

```

////////////////////////////////////
HeatFlowSequence.h - header file
////////////////////////////////////
#ifndef __HEAT_FLOW_SEQUENCE_H
#define __HEAT_FLOW_SEQUENCE_H

class HeatFlowSequence {
    friend class MetalDataFile;
    friend class CProgressDialog;
private:
    MetalDataFile *mdf;
    HeatFlowSequence *prev;
    double time; //timpul
    double tmin, tmed, tmax; //temp. minima, medie si maxima
    DoubleSpace dsf, dst, dsh;
private: //functii de uz intern
    double dhdt(int i, int j, int k);
    void init();
public:
    HeatFlowSequence(MetalDataFile *mdf, HeatFlowSequence *prev);
    void writeToFile();
    void calculate();
};

#endif// __HEAT_FLOW_SEQUENCE_H

////////////////////////////////////
HeatFlowSequence.cpp - class implementation
////////////////////////////////////
#include "stdafx.h"
#include "headers.h"
HeatFlowSequence::HeatFlowSequence(MetalDataFile *mdf, HeatFlowSequence *prev) {
    HeatFlowSequence::mdf = mdf;
    HeatFlowSequence::prev = prev;
    dsf.alloc(mdf->imax, mdf->jmax, mdf->kmax);
    dst.alloc(mdf->imax, mdf->jmax, mdf->kmax);
    dsh.alloc(mdf->imax, mdf->jmax, mdf->kmax);
    time = tmin = tmed = tmax = 0;
}

```

```

void HeatFlowSequence::writeToFile() {
    DataFile *df = &mdf->data;
    df->writeDouble(time);
    df->writeDouble(tmin);
    df->writeDouble(tmed);
    df->writeDouble(tmax);
    dst.writeToFileAsFloatSpace(df);
}
void HeatFlowSequence::init() {
    double t = mdf->tini;
    time = 0; tmin = tmed = tmax = t;
    dst.fill(t);
    dsf.fill(mdf->mat.fi_t(t));
    dsh.fill(mdf->mat.h_t(t));
}

#define DOW(SURF) (mdf->alpha[SURF] * (mdf->surface.get(SURF+1, ptime) - pdst))
double HeatFlowSequence::dhdt(int i, int j, int k) {
    int im = mdf->imax - 1, jm = mdf->jmax - 1, km = mdf->kmax - 1;
    double W, pfix, pfiy, pfiz;
    double x = mdf->dimx / mdf->imax;
    double y = mdf->dimy / mdf->jmax;
    double z = mdf->dimz / mdf->kmax;
    double pdsf = prev->dsf(i, j, k), pdst = prev->dst(i, j, k),
           ptime = prev->time;
    if (i == 0) {
        W = DOW(SURF_I0);
        pfix = 2*(prev->dsf(i+1, j, k) - pdsf + W*x / mdf->mat.lambda_0());
    } else if (i == im) {
        W = DOW(SURF_IM);
        pfix = 2*(prev->dsf(i-1, j, k) - pdsf + W*x / mdf->mat.lambda_0());
    } else pfix = prev->dsf(i+1, j, k) + prev->dsf(i-1, j, k) - 2*pdsf;
    if (j == 0) {
        W = DOW(SURF_J0);
        pfiy = 2*(prev->dsf(i, j+1, k) - pdsf + W*y / mdf->mat.lambda_0());
    } else if (j == jm) {
        W = DOW(SURF_JM);
        pfiy = 2*(prev->dsf(i, j-1, k) - pdsf + W*y / mdf->mat.lambda_0());
    } else pfiy = prev->dsf(i, j+1, k) + prev->dsf(i, j-1, k) - 2*pdsf;
    if (k == 0) {
        W = DOW(SURF_K0);
        pfiz = 2*(prev->dsf(i, j, k+1) - pdsf + W*z / mdf->mat.lambda_0());
    } else if (k == km) {
        W = DOW(SURF_KM);
        pfiz = 2*(prev->dsf(i, j, k-1) - pdsf + W*z / mdf->mat.lambda_0());
    } else pfiz = prev->dsf(i, j, k+1) + prev->dsf(i, j, k-1) - 2*pdsf;
    return (mdf->mat.lambda_0() / mdf->mat.ro()) *
           (pfix/sq(x) + pfiy/sq(y) + pfiz/sq(z));
}
void HeatFlowSequence::calculate() {
    if (!prev) {init(); return;}
    int i, j, k, im, jm, km;
    im = mdf->imax; jm = mdf->jmax; km = mdf->kmax;
    double t, dhdt, mdhdt = 0, dt;
    bool flag = true;
    Material *m = &mdf->mat;
    for (i = 0; i < im; i++) for (j = 0; j < jm; j++) for (k = 0; k < km; k++) {

```

```

    dsf(i, j, k) = dhdt = HeatFlowSequence::dhdt(i, j, k);
    dhdt = fabs(dhdt);
    if (mdhdt < dhdt) mdhdt = dhdt;
}
if (mdhdt != 0) {
    dt = mdf->max_dh / mdhdt;
    if (dt > mdf->max_dt) dt = mdf->max_dt;
} else dt = mdf->max_dt;
for (i = 0; i < im; i++) for (j = 0; j < jm; j++) for (k = 0; k < km; k++) {
    dsh(i, j, k) = t = prev->dsh(i, j, k) + dt * dsf(i, j, k);
    dst(i, j, k) = t = m->t_h(t);
    dsf(i, j, k) = m->fi_t(t);
}
time = prev->time + dt;
tmin = dst.minimum(); tmax = dst.maximum();
tmed = dst.average();
}

```

SequenceInterpolator este o clasă simplă folosită exclusiv pentru interpolarea multiliniară a temperaturilor în interiorul materialului. Interpolarea efectivă este totuși realizată de clasa container **FloatSpace**. Motivul utilizării unei astfel de clase este acela că încapsularea oferă posibilitatea unei întrețineri mult mai ușoare a programului. O eventuală modificare a structurii fișierului de date se va reflecta numai în **HeatFlowSequence** (sursa datelor) și această clasă ("consumatorul" datelor).

```

// SequenceInterpolator.h - header file
////////////////////////////////////////////////////////////////////
#ifndef __SEQUENCE_INTERPOLATOR_H
#define __SEQUENCE_INTERPOLATOR_H
class SequenceInterpolator {
    double time; // timpul
    double tmin, tmed, tmax; //temp. minima, medie si maxima
    FloatSpace st;
public:
    SequenceInterpolator() {}
    SequenceInterpolator(DataFile *df);
    void readFromFile(DataFile *df);
    double getTime() {return time;}
    double getTMin() {return tmin;}
    double getTMed() {return tmed;}
    double getTMax() {return tmax;}
    float interpolate(float x, float y, float z);
};
#endif//__SEQUENCE_INTERPOLATOR_H

////////////////////////////////////////////////////////////////////
// SequenceInterpolator.cpp - class implementation
////////////////////////////////////////////////////////////////////
#include "stdafx.h"
#include "headers.h"
SequenceInterpolator::SequenceInterpolator(DataFile *df) {
    readFromFile(df);
}
void SequenceInterpolator::readFromFile(DataFile *df) {

```

```

    time = df->readDouble();
    tmin = df->readDouble();
    tmed = df->readDouble();
    tmax = df->readDouble();
    st.readFromFile(df);
}
float SequenceInterpolator::interpolate(float x, float y, float z) {
    return st.interpolate(x, y, z);
}

```

Clasa **Material** este folosită pentru descrierea unor proprietăți fizice a materialului folosit și a dependenței lor de temperatură. Acestea sunt: densitatea ρ (considerată constantă), conductibilitatea termică $\lambda(t)$ și entalpia $H(t)$. De asemenea se calculează temperatura redusă $\Phi(t)$ și funcțiile inverse $t(H)$ și $t(\Phi)$. Această clasă are și funcții pentru *serializare* (scrierea într-un fișier a tuturor parametrilor) și *deserializare* (procesul invers).

```

// Material.h - header file
// Material.cpp - class implementation

#ifndef __MATERIAL_H
#define __MATERIAL_H
#include "buffers.h"
#define MATERIAL_STEEL 1

class Material {
    int type;
    double a, b, c, tpr; //pt. lambda
    double lpr, tpr2, tpr3, fi pr; //pt. accelerarea calculelor lambda
    FloatBuffer ht, hh; //entalpia prin puncte
    double _ro; //densitatea
    void precalc();
public:
    Material();
    Material(Material& mat);
    void clone(Material *mat);
    void defaults(int type = MATERIAL_STEEL);
    void writeToFile(DataFile* df);
    void readFromFile(DataFile* df);

    double ro() {return _ro;}
    double lambda_0() {return c;}
    double lambda_t(double t);
    double fi_t(double t);
    double t_fi(double fi);
    double h_t(double t);
    double t_h(double h);
};
#endif// __MATERIAL_H

// Material.cpp - class implementation

#include "stdafx.h"
#include "headers.h"

```

```

Material::Material() {
    defaults();
}
Material::Material(Material& mat) {
    mat.clone(this);
}
void Material::clone(Material *mat) {
    mat->_ro = _ro;
    mat->a = a; mat->b = b; mat->c = c;
    mat->tpr = tpr;
    mat->ht.fromFloatBuffer(ht);
    mat->hh.fromFloatBuffer(hh);
    mat->precalc();
}
void Material::defaults(int type) {
    Material::type = type;
    switch (type) {
    default: ASSERT(false); break;
    case MATERIAL_STEEL:
        coeficientii pentru calculul conductibilitatii termice
        // lambda = a*t*t + b*t + c pentru t > tpr
        // si lambda = lpr = a*tpr*tpr + b*tpr + c (constanta)
        pentru t <= tpr
        a = -3.3253477562765E-05;
        b = -8.6288455272252E-03;
        c = 5.09286185768560E+01;
        tpr = 768;
        //valorile pentru entalpii (hh) la diverse temperaturi (ht)
        ht.fromString(
            "100 200 300 400 500 600 "
            "700 800 900 1000 1100 1200 "
            "1300 1440 1500 1600");
        hh.fromString(
            "11.2 22.6 36.2 49.9 64.3 82.0 "
            "101.0 130.8 148.1 164.6 181.7 198.6 "
            "207.5 220.0 284.0 310.7");
        //entalpiile sunt date in kcal, dar le vrem in J
        hh.multiply(J_KCAL);
        _ro = 7850;
        break;
    }
    precalc();
}
void Material::precalc() {
    tpr2 = tpr*tpr; tpr3 = tpr2*tpr;
    lpr = a*tpr2 + b*tpr + c;
    fipr = tpr + (tpr2/c)*(tpr*a/3 + b/2);
}
void Material::writeToFile(DataFile *df) {
    df->writeInt(type);
    df->writeDouble(a);
    df->writeDouble(b);
    df->writeDouble(c);
    df->writeDouble(tpr);
    df->writeDouble(_ro);
    ht.writeToFile(df);
    hh.writeToFile(df);
}

```



```

}
void Material::readFromFile(DataFile *df) {
    type = df->readInt();
    a = df->readDouble();
    b = df->readDouble();
    c = df->readDouble();
    tpr = df->readDouble();
    _ro = df->readDouble();
    ht.readFromFile(df);
    hh.readFromFile(df);
}
double Material::lambda_t(double t) {
    if (type && (t < tpr)) return a*t*t + b*t + c;
    else return lpr;
}
double Material::fi_t(double t) {
    if (t < tpr) return t + (t*t/c)*(t*a/3 + b/2);
    return fipr + (t - tpr)*lpr/c;
}
double Material::t_fi(double fi) {
    if (fi >= fipr) //ecuatie de gr. I, altfel de gr. III
        return tpr + (c/lpr)*(fi - fipr);
    double fit, dfidt, t;
    t = tpr*fi/fipr;
    while (true) { //metoda aproximarii succesive (Newton)
        fit = t + (t*t/c)*(t*a/3 + b/2);
        if (fabs(fi - fit) < 1e-6) return t;
        dfidt = (t/c)*(a*t + b) + 1;
        t += (fi - fit)/dfidt;
    }
}
double Material::h_t(double t) {
    int n = ht.binarySearch(t);
    float t1 = ht[n], t2 = ht[n+1];
    return ((t-t1)*hh[n+1] + (t2-t)*hh[n])/(t2-t1);
}
double Material::t_h(double h) {
    int n = hh.binarySearch(h);
    float h1 = hh[n], h2 = hh[n+1];
    return ((h-h1)*ht[n+1] + (h2-h)*ht[n])/(h2-h1);
}
}

```

SurfaceData conține metode pentru determinarea condițiilor de limită. Condițiile implicite sunt implementate direct în program (*hard-coded*). De asemenea, și această clasă conține funcții pentru serializare și deserializare.

```

// SurfaceData.h
SurfaceData.h - header file
////////////////////////////////////////////////////
#ifndef __SURFACE_DATA_H
#define __SURFACE_DATA_H

#include "buffers.h"

class SurfaceData {
    FloatBuffer **bufs;
    void addLine(float *line);
}

```

```

    void init();
    void clear();
    void defaults();
public:
    SurfaceData();
    ~SurfaceData();
    void writeToFile(DataFile *df);
    void readFromFile(DataFile *df);
    float get(int surface, float time);
    float maxTime();
};

#endif//__SURFACE_DATA_H

// SurfaceData.cpp - class implementation
#include "stdafx.h"
#include "SurfaceData.h"

SurfaceData::SurfaceData() {
    bufs = NULL; defaults();
}
SurfaceData::~SurfaceData() {
    clear();
}
void SurfaceData::clear() {
    if (!bufs) return;
    for (int k = 0; k < 7; k++) DEL(bufs[k]);
    delete bufs;
}
void SurfaceData::init() {
    clear();
    bufs = new FloatBuffer*[7];
    for (int k = 0; k < 7; k++) bufs[k] = new FloatBuffer();
}
void SurfaceData::addLine(float *line) {
    for (int k = 0; k < 7; k++)
        bufs[k]->add(line[k]);
}
void SurfaceData::writeToFile(DataFile *df) {
    for (int k = 0; k < 7; k++) bufs[k]->writeToFile(df);
}
void SurfaceData::readFromFile(DataFile *df) {
    for (int k = 0; k < 7; k++) bufs[k]->readFromFile(df);
}
float SurfaceData::get(int surface, float time) {
    ASSERT((surface > 0) && (surface < 7));
    FloatBuffer *fb = bufs[surface];
    int k = bufs[0]->binarySearch(time);
    float t1 = (*bufs[0])[k], t2 = (*bufs[0])[k + 1];
    float res = ((*fb)[k + 1] * (time - t1) + (*fb)[k] * (t2 - time)) / (t2 - t1);
    ASSERT(res >= 0);
    ASSERT(res < 1000);
    return res;
}
float SurfaceData::maxTime() {

```

```

    return (*bufs[0])[bufs[0]->size() - 1];
}
void SurfaceData::defaults() {
    init();
    FloatBuffer fb;
    fb.fromString("0 20.0 20.0 20.0 20.0 20.0 20.0"); addLine(&fb[0]);
    fb.fromString("600 58.6 56.3 68.6 75.5 47.8 42.3"); addLine(&fb[0]);
    fb.fromString("1200 77.2 66.7 87.2 91.1 58.6 45.1"); addLine(&fb[0]);
    fb.fromString("1800 127.9 83.7 137.9 125.2 89.0 49.4"); addLine(&fb[0]);
    fb.fromString("2400 171.6 110.0 181.6 162.5 117.4 55.5"); addLine(&fb[0]);
    fb.fromString("3000 217.8 140.6 227.8 200.0 142.9 62.7"); addLine(&fb[0]);
    fb.fromString("3600 265.0 173.5 275.0 238.0 171.2 70.9"); addLine(&fb[0]);
    fb.fromString("4200 309.7 207.8 319.7 275.7 199.3 79.3"); addLine(&fb[0]);
    fb.fromString("4800 350.2 242.1 360.2 312.7 227.1 87.6"); addLine(&fb[0]);
    fb.fromString("5400 384.7 274.7 394.7 345.9 252.8 96.5"); addLine(&fb[0]);
    fb.fromString("6000 416.3 305.1 426.3 377.7 280.6 105.2"); addLine(&fb[0]);
    fb.fromString("6600 440.2 333.9 450.2 403.8 298.1 113.8"); addLine(&fb[0]);
    fb.fromString("7200 461.9 357.1 471.9 425.4 316.6 121.8"); addLine(&fb[0]);
    fb.fromString("7800 483.7 378.2 493.7 445.6 335.5 129.7"); addLine(&fb[0]);
    fb.fromString("8400 505.7 397.7 515.7 464.7 353.2 137.6"); addLine(&fb[0]);
    fb.fromString("9000 525.7 415.4 535.7 482.4 370.0 145.1"); addLine(&fb[0]);
    fb.fromString("9600 545.0 432.7 555.0 499.5 385.6 153.1"); addLine(&fb[0]);
    fb.fromString("10200 563.4 448.8 573.4 515.5 402.5 160.9"); addLine(&fb[0]);
    fb.fromString("10800 580.5 464.3 590.5 531.0 417.5 169.0"); addLine(&fb[0]);
    fb.fromString("11400 596.0 477.6 606.0 544.4 430.0 176.4"); addLine(&fb[0]);
    fb.fromString("12000 611.5 490.8 621.5 557.7 442.5 183.8"); addLine(&fb[0]);
    fb.fromString("12600 625.9 503.4 635.9 570.3 455.1 191.3"); addLine(&fb[0]);
    fb.fromString("13200 640.5 516.2 650.5 583.1 467.3 200.0"); addLine(&fb[0]);
    fb.fromString("13800 653.8 528.2 663.8 594.8 479.2 207.2"); addLine(&fb[0]);
    fb.fromString("14400 667.2 539.8 677.2 606.2 490.8 215.4"); addLine(&fb[0]);
    fb.fromString("15000 678.8 551.1 688.8 617.7 501.0 223.8"); addLine(&fb[0]);
    fb.fromString("15600 691.1 561.6 701.1 628.0 512.2 231.1"); addLine(&fb[0]);
    fb.fromString("16200 703.6 572.0 713.6 638.8 523.0 239.0"); addLine(&fb[0]);
    fb.fromString("16800 714.8 582.5 724.8 649.4 534.0 247.1"); addLine(&fb[0]);
    fb.fromString("17400 725.1 592.6 735.1 659.7 543.2 255.4"); addLine(&fb[0]);
    fb.fromString("18000 734.5 601.9 744.5 668.9 552.9 263.0"); addLine(&fb[0]);
    fb.fromString("18600 744.0 610.8 754.0 677.9 563.0 270.3"); addLine(&fb[0]);
    fb.fromString("19200 754.8 619.6 764.8 686.5 573.3 277.6"); addLine(&fb[0]);
    fb.fromString("19800 768.1 630.0 778.1 697.7 584.6 285.8"); addLine(&fb[0]);
    fb.fromString("20400 779.7 639.8 789.7 708.1 595.8 293.7"); addLine(&fb[0]);
    fb.fromString("21000 789.7 649.5 799.7 718.4 605.2 301.3"); addLine(&fb[0]);
    fb.fromString("21600 799.3 658.3 809.3 727.2 614.7 309.0"); addLine(&fb[0]);
    fb.fromString("22200 809.3 667.5 819.3 735.9 624.8 316.9"); addLine(&fb[0]);
    fb.fromString("22800 818.2 676.3 828.2 744.3 635.1 324.3"); addLine(&fb[0]);
    fb.fromString("23400 826.7 685.2 836.7 752.8 644.6 332.1"); addLine(&fb[0]);
    fb.fromString("24000 835.6 693.6 845.6 761.1 652.9 340.2"); addLine(&fb[0]);
    fb.fromString("24600 844.3 702.2 854.3 769.4 661.5 348.2"); addLine(&fb[0]);
    fb.fromString("25200 851.6 710.0 861.6 777.7 669.7 355.8"); addLine(&fb[0]);
    fb.fromString("25800 860.0 720.0 870.0 786.0 679.0 363.0"); addLine(&fb[0]);
    fb.fromString("26400 865.0 727.0 875.0 795.0 685.0 370.0"); addLine(&fb[0]);
    fb.fromString("27000 870.0 732.0 880.0 807.0 690.0 380.0"); addLine(&fb[0]);
    fb.fromString("27600 875.0 740.0 885.0 812.0 698.0 388.0"); addLine(&fb[0]);
    fb.fromString("28200 880.0 750.0 890.0 818.0 705.0 395.0"); addLine(&fb[0]);
    fb.fromString("28800 885.0 758.0 895.0 824.0 713.0 403.0"); addLine(&fb[0]);
    fb.fromString("29400 888.0 765.0 898.0 833.0 722.0 410.0"); addLine(&fb[0]);
    fb.fromString("30050 894.0 773.0 904.0 844.0 730.0 418.0"); addLine(&fb[0]);
    fb.fromString("30600 897.0 782.0 907.0 850.0 736.0 423.0"); addLine(&fb[0]);
}

```

```

fb.fromString("31200 903.0 790.0 913.0 856.0 748.0 428.0"); addLine(&fb[0]);
fb.fromString("31800 903.0 800.0 913.0 864.0 754.0 435.0"); addLine(&fb[0]);
fb.fromString("32400 906.0 804.0 916.0 867.0 762.0 440.0"); addLine(&fb[0]);
fb.fromString("33000 906.0 810.0 916.0 876.0 770.0 445.0"); addLine(&fb[0]);
fb.fromString("33600 908.0 818.0 918.0 881.0 777.0 453.0"); addLine(&fb[0]);
fb.fromString("34200 908.0 822.0 918.0 887.0 782.0 460.0"); addLine(&fb[0]);
fb.fromString("34800 909.0 830.0 919.0 890.0 790.0 468.0"); addLine(&fb[0]);
fb.fromString("35400 909.0 836.0 919.0 896.0 794.0 473.0"); addLine(&fb[0]);
fb.fromString("36000 909.0 840.0 919.0 896.0 797.0 483.0"); addLine(&fb[0]);
}

```

Fișierele Buffers.* conțin clase "container" pentru implementarea unor vectori și matrici (inclusiv tridimensionale).

```

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
Buffers.h - header file
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

#ifndef __BUFFERS_H
#define __BUFFERS_H
#include "DataFile.h"
#define DECLARE_MINIMUM(type, buf, len) \
    type minimum() {int k; type result = buf[0]; \
    for (k = 1; k < len; k++) if (result > buf[k]) result = buf[k]; \
    return result;}
#define DECLARE_MAXIMUM(type, buf, len) \
    type maximum() {int k; type result = buf[0]; \
    for (k = 1; k < len; k++) if (result < buf[k]) result = buf[k]; \
    return result;}
#define DECLARE_AVERAGE(type, buf, len) \
    type average() {int k; type result = 0; \
    for (k = 0; k < len; k++) result += buf[k]; return result / len;}

//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
class FloatBuffer
class FloatBuffer {
    float *buf;
    int n;
public:
    FloatBuffer();
    FloatBuffer(FloatBuffer& fb);
    FloatBuffer(float *fb, int count);
    FloatBuffer(double *fb, int count);
    ~FloatBuffer();
    DECLARE_MINIMUM(float, buf, n);
    DECLARE_MAXIMUM(float, buf, n);
    DECLARE_AVERAGE(float, buf, n);
    void empty() {setSize(0);}
    bool isEmpty() {return (n == 0);}
    void setSize(int size);
    int size() {return n;}
    void add(float x);
    void insert(float x, int pos);
    void remove(int pos);
    void sort();
    void multiply(float x);
    int binarySearch(float x);
}

```

```

CString toString();
void fromString(LPCTSTR s);
void fromFloatBuffer(FloatBuffer& fb);
void fromArray(float *fb, int count);
void fromArray(double *fb, int count);
void writeToFile(DataFile *df);
void readFromFile(DataFile *df);
float& operator[] (UINT index) {return buf[index];}
float interpolate(float x);
};

class FloatArray
class FloatArray {
    float *buf;
    int n, m;
public:
    FloatArray();
    FloatArray(FloatArray& fa);
    ~FloatArray();
    DECLARE_MINIMUM(float, buf, n*m);
    DECLARE_MAXIMUM(float, buf, n*m);
    void alloc(int n, int m);
    void clear();
    float& operator() (int i, int j);
    int nsize() {return n;}
    int msize() {return m;}
    void fill(float x);
    void copy(float *px);
    void multiply(float x);
    float sum();
    float average();
    BOOL isEmpty() {return (buf == NULL);}
    void writeToFile(DataFile *df);
    void readFromFile(DataFile *df);
    void writeToTextFile(FILE *file);
    float *getBuffer() {return buf;}
    float interpolate(float x, float y);
};
////////////////////////////////////////////////////
class FloatSpace
class FloatSpace {
    float *buf;
    int n, m, p;
public:
    FloatSpace();
    FloatSpace(FloatSpace& fa);
    ~FloatSpace();
    DECLARE_MINIMUM(float, buf, n*m*p);
    DECLARE_MAXIMUM(float, buf, n*m*p);
    void alloc(int n, int m, int p);
    void clear();
    float& operator() (int i, int j, int k);
    int nsize() {return n;}
    int msize() {return m;}
    int psize() {return p;}
    void fill(float x);
    void copy(float *px);
};

```

```

void multiply(float x);
float sum();
float average();
BOOL isEmpty() {return (buf == NULL);}
void writeToFile(DataFile *df);
void readFromFile(DataFile *df);
float *getBuffer() {return buf;}
float interpolate(float x, float y, float z);
};

```

```

class DoubleArray
class DoubleArray {
    double *buf;
    int n, m;
public:
    DoubleArray();
    DoubleArray(DoubleArray& fa);
    ~DoubleArray();
    DECLARE_MINIMUM(double, buf, n*m);
    DECLARE_MAXIMUM(double, buf, n*m);
    void alloc(int n, int m);
    void clear();
    double& operator() (int i, int j);
    int nsize() {return n;}
    int msize() {return m;}
    void fill(double x);
    void copy(double *px);
    void multiply(double x);
    double sum();
    double average();
    BOOL isEmpty() {return (buf == NULL);}
    void writeToFile(DataFile *df);
    void readFromFile(DataFile *df);
    void writeToFileAsFloatArray(DataFile *df);
    void writeToTextFile(FILE *file);
    double *getBuffer() {return buf;}
    double interpolate(double x, double y);
};

```

```

//class DoubleVector
class DoubleVector {
    double* buf; int n;
public:
    DoubleVector();
    DoubleVector(DoubleVector& dv);
    ~DoubleVector();
    DECLARE_MINIMUM(double, buf, n);
    DECLARE_MAXIMUM(double, buf, n);
    void alloc(int n);
    void clear();
    double& operator[] (int i);
    int size();
    void fill(double x);
    void multiply(double x);
    double sum();
    double average();
};

```

```

    BOOL isEmpty() {return (buf == NULL);}
    double interpolate(double x);
};

class DoubleSpace
class DoubleSpace {
    double *buf;
    int n, m, p;
public:
    DoubleSpace();
    DoubleSpace(DoubleSpace& fa);
    ~DoubleSpace();
    DECLARE_MINIMUM(double, buf, n*m*p);
    DECLARE_MAXIMUM(double, buf, n*m*p);
    void alloc(int n, int m, int p);
    void clear();
    double& operator() (int i, int j, int k);
    int nsize() {return n;}
    int msize() {return m;}
    int psize() {return p;}
    void fill(double x);
    void copy(double *px);
    void multiply(double x);
    double sum();
    double average();
    BOOL isEmpty() {return (buf == NULL);}
    void writeToFile(DataFile *df);
    void readFromFile(DataFile *df);
    void writeToFileAsFloatSpace(DataFile *df);
    double *getBuffer() {return buf;}
    double interpolate(double x, double y, double z);
};
#endif//__BUFFERS_H

// ***** //
Buffers.cpp - class implementations
// ***** //

#include "stdafx.h"
#include "DataFile.h"
#include "Buffers.h"

#define BUF2(i, j) buf[((i)*m + (j))]
#define BUF3(i, j, k) buf[((i)*m + (j))*p + (k)]

bool isFloatPart(char ch) {
    return isdigit(ch) || (ch == '.') || (tolower(ch) == 'e');
}

class FloatBuffer
FloatBuffer::FloatBuffer() {
    n = 0; buf = NULL;
}
FloatBuffer::FloatBuffer(FloatBuffer& fb) {
    n = 0; buf = NULL;
    fromFloatBuffer(fb);
}

```

```

FloatBuffer::FloatBuffer(float *fb, int count) {
    n = 0; buf = NULL;
    fromArray(fb, count);
}
FloatBuffer::FloatBuffer(double *fb, int count) {
    n = 0; buf = NULL;
    fromArray(fb, count);
}
FloatBuffer::~FloatBuffer() {
    if (buf) delete buf;
}
void FloatBuffer::setSize(int size) {
    if (size <= 0) {
        if (buf) delete buf; n = 0; buf = NULL;
        return;
    }
    if (buf == NULL) {
        buf = new float[size]; n = size;
        memset(buf, 0, sizeof(float)*size);
        return;
    }
    int nx = (size < n) ? size : n;
    float *nwbuf = new float[size];
    memset(nwbuf, 0, sizeof(float)*size);
    memcpy(nwbuf, buf, sizeof(float)*nx);
    delete buf; buf = nwbuf;
    n = size; return;
}
void FloatBuffer::add(float x) {
    setSize(n+1);
    buf[n-1] = x;
}
void FloatBuffer::insert(float x, int pos) {
    setSize(n+1);
    memmove(buf+pos+1, buf+pos, sizeof(float)*(n-pos-1));
    buf[pos] = x;
}
void FloatBuffer::remove(int pos) {
    if (n <= 1) {setSize(0); return;}
    memmove(buf+pos, buf+pos+1, sizeof(float)*(n-pos-1));
    setSize(n-1);
}
void FloatBuffer::sort() {
    if (n <= 1) return;
    bool done;
    int k; float aux;
    do {
        done = true;
        for (k = 0; k < n-1; k++) {
            if (buf[k] > buf[k+1]) {
                done = false;
                aux = buf[k];
                buf[k] = buf[k+1];
                buf[k+1] = aux;
            }
        }
    } while(!done);
}

```



```

}
void FloatBuffer::multiply(float x) {
    if (!buf) return;
    for (int k = 0; k < n; k++) buf[k] *= x;
}
int FloatBuffer::binarySearch(float x) {
    if (!buf) return 0;
    int a = 0, b = n - 1, c = (a + b) >> 1;
    if (x >= buf[b]) return b-1;
    while (c != a) {
        if (x >= buf[c]) a = c;
        else b = c;
        c = (a + b) >> 1;
    }
    return c;
}
CString FloatBuffer::toString() {
    int k; CString res, cs;
    for (k = 0; k < n; k++) {
        cs.Format("%g", buf[k]);
        res += cs;
        if (k < n-1) res+=' ';
    }
    return res;
}
void FloatBuffer::fromString(LPCTSTR s) {
    float x;
    DEL(buf); n = 0;
    if (s == NULL) return;
    int j = 0, len = strlen(s);
    do {
        for (; j < len; j++) if (isFloatPart(s[j])) break;
        if (j < len) if (sscanf(s+j, "%g", &x)) add(x);
        for (; j < len; j++) if (!isFloatPart(s[j])) break;
    } while(j < len);
}
void FloatBuffer::fromFloatBuffer(FloatBuffer& fb) {
    empty();
    if (fb.buf == NULL) return;
    buf = new float[fb.n]; n = fb.n;
    memcpy(buf, fb.buf, sizeof(float)*n);
}
void FloatBuffer::fromArray(float *fb, int count) {
    empty();
    if (count <= 0) return;
    buf = new float[count]; n = count;
    memcpy(buf, fb, sizeof(float)*count);
}
void FloatBuffer::fromArray(double *fb, int count) {
    empty();
    if (count <= 0) return;
    buf = new float[count]; n = count;
    for (int k = 0; k < count; k++) buf[k] = (float)fb[k];
}
void FloatBuffer::writeToFile(DataFile *df) {
    df->writeInt(n);
    df->write(buf, n*sizeof(float));
}

```

```

}
void FloatBuffer::readFromFile(DataFile *df) {
    empty();
    n = df->readInt();
    setSize(n);
    df->read(buf, n*sizeof(float));
}
float FloatBuffer::interpolate(float x) {
    int k = (int)(x * n);
    if (k < 0) k = 0;
    if (k > n - 2) k = n - 2;
    float kx = x * n;
    return buf[k + 1] * (kx - k) + buf[k] * (k + 1 - kx);
}
////////////////////////////////////////////////////
class FloatArray
FloatArray::FloatArray() {
    buf = NULL; n = m = 0;
}
FloatArray::FloatArray(FloatArray& fa) {
    buf = NULL; n = m = 0;
    alloc(fa.n, fa.m);
    memcpy(buf, fa.buf, sizeof(float)*n*m);
}
FloatArray::~FloatArray() {
    clear();
}
void FloatArray::clear() {
    DEL(buf); n = m = 0;
}
void FloatArray::alloc(int n, int m) {
    clear();
    CHKNEW(buf = new float[n*m]);
    FloatArray::n = n; FloatArray::m = m;
}
float& FloatArray::operator () (int i, int j) {
    ASSERT(buf && (i>=0)&&(i<n) && (j>=0)&&(j<m));
    return buf[i*m+j];
}
void FloatArray::fill(float x) {
    if (!buf) return; int nm = n*m;
    for (int k = 0; k < nm; k++) buf[k] = x;
}
void FloatArray::copy(float *px) {
    memcpy(buf, px, sizeof(float)*n*m);
}
void FloatArray::multiply(float x) {
    ASSERT(buf); int nm = n*m;
    for (int k = 0; k < nm; k++) buf[k] *= x;
}
float FloatArray::sum() {
    ASSERT(buf); int k, nm = n*m; float s = 0;
    for (k = 0; k < nm; k++) s += buf[k]; return s;
}
float FloatArray::average() {
    return sum()/(n*m);
}
}

```

```

void FloatArray::writeToFile(DataFile *df) {
    df->writeInt(n); df->writeInt(m);
    df->write(buf, n*m*sizeof(float));
}
void FloatArray::readFromFile(DataFile *df) {
    clear();
    n = df->readInt(); m = df->readInt();
    alloc(n, m);
    df->read(buf, n*m*sizeof(float));
}
void FloatArray::writeToTextFile(FILE *file) {
    int i, j;
    for (j = 0; j < m; j++) {
        for (i = 0; i < n; i++) {
            fprintf(file, "%g", buf[i*m+j]);
            if (i < n-1) fprintf(file, "\t");
        }
        if (j < m-1) fprintf(file, "\n");
    }
}
float FloatArray::interpolate(float x, float y) {
    int i = (int)(x*n), j = (int)(y*m);
    if (i < 0) i = 0; if (i > n - 2) i = n - 2;
    if (j < 0) j = 0; if (j > m - 2) j = m - 2;
    float nx = n * x, my = m * y;
    float px1 = BUF2(i+1,j) * (nx - i) + BUF2(i,j) * (i + 1 - nx);
    float px2 = BUF2(i+1,j+1) * (nx - i) + BUF2(i,j+1) * (i + 1 - nx);
    return px2 * (my - j) + px1 * (j + 1 - my);
}
// ****

class FloatSpace
FloatSpace::FloatSpace() {
    buf = NULL; n = m = p = 0;
}
FloatSpace::FloatSpace(FloatSpace& fa) {
    buf = NULL; n = m = p = 0;
    alloc(fa.n, fa.m, fa.p);
    memcpy(buf, fa.buf, sizeof(float)*n*m*p);
}
FloatSpace::~FloatSpace() {
    clear();
}
void FloatSpace::clear() {
    DEL(buf); n = m = p = 0;
}
void FloatSpace::alloc(int n, int m, int p) {
    clear();
    CHKNEW(buf = new float[n*m*p]);
    FloatSpace::n = n; FloatSpace::m = m; FloatSpace::p = p;
}
float& FloatSpace::operator () (int i, int j, int k) {
    ASSERT(buf && (i>=0)&&(i<n) && (j>=0)&&(j<m) && (k>=0)&&(k<p));
    return buf[(i*m + j)*p + k];
}
void FloatSpace::fill(float x) {
    if (!buf) return; int nmp = n*m*p;
    for (int k = 0; k < nmp; k++) buf[k] = x;
}

```

```

}
void FloatSpace::copy(float *px) {
    memcpy(buf, px, sizeof(float)*n*m*p);
}
void FloatSpace::multiply(float x) {
    ASSERT(buf); int nmp = n*m*p;
    for (int k = 0; k < nmp; k++) buf[k] *= x;
}
float FloatSpace::sum() {
    ASSERT(buf); int k, nmp = n*m*p; float s = 0;
    for (k = 0; k < nmp; k++) s += buf[k]; return s;
}
float FloatSpace::average() {
    return sum()/(n*m*p);
}
void FloatSpace::writeToFile(DataFile *df) {
    df->writeInt(n); df->writeInt(m); df->writeInt(p);
    df->write(buf, n*m*p*sizeof(float));
}
void FloatSpace::readFromFile(DataFile *df) {
    clear();
    n = df->readInt(); m = df->readInt(); p = df->readInt();
    alloc(n, m, p);
    df->read(buf, n*m*p*sizeof(float));
}
float FloatSpace::interpolate(float x, float y, float z) {
    ASSERT((n > 1) && (m > 1) && (p > 1));
    int i = (int)(x*n), j = (int)(y*m), k = (int)(z*p);
    if (i < 0) i = 0; if (i > n - 2) i = n - 2;
    if (j < 0) j = 0; if (j > m - 2) j = m - 2;
    if (k < 0) k = 0; if (k > p - 2) k = p - 2;
    float t1, t2;
    t1 = p * (z - (float)k/p); t2 = p * ((float)(k + 1)/p - z);
    float p00 = BUF3(i, j, k+1) * t1 + BUF3(i, j, k) * t2;
    float p01 = BUF3(i, j+1, k+1) * t1 + BUF3(i, j+1, k) * t2;
    float p10 = BUF3(i+1, j, k+1) * t1 + BUF3(i+1, j, k) * t2;
    float p11 = BUF3(i+1, j+1, k+1) * t1 + BUF3(i+1, j+1, k) * t2;
    t1 = m * (y - (float)j/m); t2 = m * ((float)(j + 1)/m - y);
    float px0 = p00 * t2 + p01 * t1;
    float px1 = p10 * t2 + p11 * t1;
    t1 = n * (x - (float)i/n); t2 = n * ((float)(i + 1)/n - x);
    return px0 * t2 + px1 * t1;
}
////////////////////////////////////////////////////
//class DoubleArray
DoubleArray::DoubleArray() {
    buf = NULL; n = m = 0;
}
DoubleArray::DoubleArray(DoubleArray& fa) {
    buf = NULL; n = m = 0;
    alloc(fa.n, fa.m);
    memcpy(buf, fa.buf, sizeof(double)*n*m);
}
DoubleArray::~DoubleArray() {
    clear();
}
void DoubleArray::clear() {

```

```

    DEL(buf); n = m = 0;
}
void DoubleArray::alloc(int n, int m) {
    clear();
    CHKNEW(buf = new double[n*m]);
    DoubleArray::n = n; DoubleArray::m = m;
}
double& DoubleArray::operator () (int i, int j) {
    ASSERT(buf && (i>=0)&&(i<n) && (j>=0)&&(j<m));
    return buf[i*m+j];
}
void DoubleArray::fill(double x) {
    if (!buf) return; int nm = n*m;
    for (int k = 0; k < nm; k++) buf[k] = x;
}
void DoubleArray::copy(double *px) {
    memcpy(buf, px, sizeof(double)*n*m);
}
void DoubleArray::multiply(double x) {
    ASSERT(buf); int nm = n*m;
    for (int k = 0; k < nm; k++) buf[k] *= x;
}
double DoubleArray::sum() {
    ASSERT(buf); int k, nm = n*m; double s = 0;
    for (k = 0; k < nm; k++) s += buf[k]; return s;
}
double DoubleArray::average() {
    return sum()/(n*m);
}
void DoubleArray::writeToFile(DataFile *df) {
    df->writeInt(n); df->writeInt(m);
    df->write(buf, n*m*sizeof(double));
}
void DoubleArray::readFromFile(DataFile *df) {
    clear();
    n = df->readInt(); m = df->readInt();
    alloc(n, m);
    df->read(buf, n*m*sizeof(double));
}
void DoubleArray::writeToFileAsFloatArray(DataFile *df) {
    int nm = n*m; df->writeInt(n); df->writeInt(m);
    for (int k = 0; k < nm; k++) df->writeFloat((float)buf[k]);
}
void DoubleArray::writeToTextFile(FILE *file) {
    int i, j;
    for (j = 0; j < m; j++) {
        for (i = 0; i < n; i++) {
            fprintf(file, "%lg", buf[i*m+j]);
            if (i < n-1) fprintf(file, "\t");
        }
        if (j < m-1) fprintf(file, "\n");
    }
}
double DoubleArray::interpolate(double x, double y) {
    int i = (int)(x*n), j = (int)(y*m);
    if (i < 0) i = 0; if (i > n - 2) i = n - 2;
    if (j < 0) j = 0; if (j > m - 2) j = m - 2;
}

```

```

    double nx = n * x, my = m * y;
    double px1 = BUF2(i+1,j) * (nx - i) + BUF2(i,j) * (i + 1 - nx);
    double px2 = BUF2(i+1,j+1) * (nx - i) + BUF2(i,j+1) * (i + 1 - nx);
    return px2 * (my - j) + px1 * (j + 1 - my);
}

class DoubleVector
DoubleVector::DoubleVector() {
    buf = NULL; n = 0;
}
DoubleVector::DoubleVector(DoubleVector& dv) {
    buf = NULL; n = 0;
    alloc(dv.n);
    memcpy(buf, dv.buf, sizeof(double)*n);
}
DoubleVector::~DoubleVector() {
    clear();
}
void DoubleVector::clear() {
    DEL(buf); n = 0;
}
void DoubleVector::alloc(int n) {
    clear();
    CHKNEW(buf = new double[n]);
    DoubleVector::n = n;
}
double& DoubleVector::operator[] (int i) {
    ASSERT(buf && (i>=0)&&(i<n));
    return buf[i];
}
void DoubleVector::fill(double x) {
    if (!buf) return; int k;
    for (k = 0; k < n; k++) buf[k] = x;
}
void DoubleVector::multiply(double x) {
    ASSERT(buf); int k;
    for (k = 0; k < n; k++) buf[k] *= x;
}
double DoubleVector::sum() {
    ASSERT(buf); int k; double s = 0;
    for (k = 0; k < n; k++) s += buf[k]; return s;
}
double DoubleVector::average() {
    return sum()/n;
}
double DoubleVector::interpolate(double x) {
    int k = (int)(x * n);
    if (k < 0) k = 0;
    if (k > n - 2) k = n - 2;
    double kx = x * n;
    return buf[k + 1] * (kx - k) + buf[k] * (k + 1 - kx);
}

/class DoubleSpace
DoubleSpace::DoubleSpace() {
    buf = NULL; n = m = p = 0;
}

```

```

DoubleSpace::DoubleSpace(DoubleSpace& fa) {
    buf = NULL; n = m = p = 0;
    alloc(fa.n, fa.m, fa.p);
    memcpy(buf, fa.buf, sizeof(double)*n*m*p);
}
DoubleSpace::~DoubleSpace() {
    clear();
}
void DoubleSpace::clear() {
    DEL(buf); n = m = p = 0;
}
void DoubleSpace::alloc(int n, int m, int p) {
    clear();
    CHKNEW(buf = new double[n*m*p]);
    DoubleSpace::n = n; DoubleSpace::m = m; DoubleSpace::p = p;
}
double& DoubleSpace::operator () (int i, int j, int k) {
    ASSERT(buf && (i>=0)&&(i<n) && (j>=0)&&(j<m) && (k>=0)&&(k<p));
    return buf[(i*m + j)*p + k];
}
void DoubleSpace::fill(double x) {
    if (!buf) return; int nmp = n*m*p;
    for (int k = 0; k < nmp; k++) buf[k] = x;
}
void DoubleSpace::copy(double *px) {
    memcpy(buf, px, sizeof(double)*n*m*p);
}
void DoubleSpace::multiply(double x) {
    ASSERT(buf); int nmp = n*m*p;
    for (int k = 0; k < nmp; k++) buf[k] *= x;
}
double DoubleSpace::sum() {
    ASSERT(buf); int k, nmp = n*m*p; double s = 0;
    for (k = 0; k < nmp; k++) s += buf[k]; return s;
}
double DoubleSpace::average() {
    return sum()/(n*m*p);
}
void DoubleSpace::writeToFile(DataFile *df) {
    df->writeInt(n); df->writeInt(m); df->writeInt(p);
    df->write(buf, n*m*p*sizeof(double));
}
void DoubleSpace::readFromFile(DataFile *df) {
    clear();
    n = df->readInt(); m = df->readInt(); p = df->readInt();
    alloc(n, m, p);
    df->read(buf, n*m*p*sizeof(double));
}
void DoubleSpace::writeToFileAsFloatSpace(DataFile *df) {
    int k, nmp = n*m*p;
    df->writeInt(n); df->writeInt(m); df->writeInt(p);
    for (k = 0; k < nmp; k++) df->writeFloat((float)buf[k]);
}
double DoubleSpace::interpolate(double x, double y, double z) {
    ASSERT((n > 1) && (m > 1) && (p > 1));
    int i = (int)(x*n), j = (int)(y*m), k = (int)(z*p);
    if (i < 0) i = 0; if (i > n - 2) i = n - 2;
}

```

```

if (j < 0) j = 0; if (j > m - 2) j = m - 2;
if (k < 0) k = 0; if (k > p - 2) k = p - 2;
double t1, t2;
t1 = p * (z - (double)k/p); t2 = p * ((double)(k + 1)/p - z);
double p00 = BUF3(i, j, k+1) * t1 + BUF3(i, j, k) * t2;
double p01 = BUF3(i, j+1, k+1) * t1 + BUF3(i, j+1, k) * t2;
double p10 = BUF3(i+1, j, k+1) * t1 + BUF3(i+1, j, k) * t2;
double p11 = BUF3(i+1, j+1, k+1) * t1 + BUF3(i+1, j+1, k) * t2;
t1 = m * (y - (double)j/m); t2 = m * ((double)(j + 1)/m - y);
double px0 = p00 * t2 + p01 * t1;
double px1 = p10 * t2 + p11 * t1;
t1 = n * (x - (double)i/n); t2 = n * ((double)(i + 1)/n - x);
return px0 * t2 + px1 * t1;
}

```

DataFile.h - header file

```

#ifndef __DATA_FILE_H
#define __DATA_FILE_H

class DataFile {
    FILE *file;
    CString filename;
    BOOL bWrite, bRead;
    void tempName();
public:
    DataFile();
    ~DataFile();
    void assign(LPCTSTR filename);
    CString getName() {return filename;}
    void openRead();
    void openWrite();
    void openExisting();
    void openCreate();
    void close();
    void remove();
    BOOL rename(LPCTSTR name);
    BOOL copy(LPCTSTR name);
    BOOL isOpen() {return (file != NULL);}
    FILE *getFile() {return file;}
    void flush();
    void seek(long pos);
    long tell();
    long size();
    BOOL canRead() {return bRead;}
    BOOL canWrite() {return bWrite;}

    void writeChar(char x);
    void writeByte(BYTE x);
    void writeShort(short x);
    void writeWord(WORD x);
    void writeInt(int x);
    void writeDword(DWORD x);
    void writeFloat(float x);
    void writeDouble(double x);
    void writeString(LPCTSTR x);

```



```

void write(LPVOID x, size_t length);

char readChar();
BYTE readByte();
short readShort();
WORD readWord();
int readInt();
DWORD readDword();
float readFloat();
double readDouble();
CString readString();
void read(LPVOID buf, size_t length);
};

```

DataFile.cpp - class implementation

```

#include "stdafx.h"
#include "DataFile.h"

DataFile::DataFile() {
    file = NULL; bRead = bWrite = FALSE;
}
DataFile::~DataFile() {
    close();
}
void DataFile::close() {
    if (file) {fclose(file); file = NULL;}
    bRead = bWrite = FALSE;
}
void DataFile::remove() {
    close();
    DeleteFile(filename);
    filename.Empty();
}
BOOL DataFile::rename(LPCTSTR name) {
    close();
    BOOL res = MoveFileEx(filename, name,
        MOVEFILE_COPY_ALLOWED | MOVEFILE_REPLACE_EXISTING);
    if (res) filename = name;
    return res;
}
BOOL DataFile::copy(LPCTSTR name) {
    close();
    return CopyFile(filename, name, FALSE);
}
void DataFile::assign(LPCTSTR filename) {
    close();
    DataFile::filename = filename;
}
void DataFile::openRead() {
    close(); if (filename.IsEmpty()) tempName();
    file = fopen(filename, "rb");
    if (file) bRead = TRUE;
}
void DataFile::openWrite() {
    close(); if (filename.IsEmpty()) tempName();
}

```

```
    file = fopen(filename, "wb");
    if (file) bWrite = TRUE;
}
void DataFile::openExisting() {
    close(); if (filename.IsEmpty()) tempName();
    file = fopen(filename, "r+b");
    if (file) bRead = bWrite = TRUE;
}
void DataFile::openCreate() {
    close(); if (filename.IsEmpty()) tempName();
    file = fopen(filename, "w+b");
    if (file) bRead = bWrite = TRUE;
}
void DataFile::flush() {
    ASSERT(file); fflush(file);
}
void DataFile::seek(long pos) {
    ASSERT(file); fseek(file, pos, SEEK_SET);
}
long DataFile::tell() {
    ASSERT(file); return ftell(file);
}
long DataFile::size() {
    ASSERT(file);
    return _filelength(_fileno(file));
}
void DataFile::tempName() {
    char temppath[MAX_PATH], fn[MAX_PATH];
    GetTempPath(MAX_PATH, temppath);
    GetTempFileName(temppath, "termo-", 0, fn);
    filename = fn;
}

/* writing
void DataFile::writeChar(char x) {
    ASSERT(file); fwrite(&x, sizeof(char), 1, file);
}
void DataFile::writeByte(BYTE x) {
    ASSERT(file); fwrite(&x, sizeof(BYTE), 1, file);
}
void DataFile::writeShort(short x) {
    ASSERT(file); fwrite(&x, sizeof(short), 1, file);
}
void DataFile::writeWord(WORD x) {
    ASSERT(file); fwrite(&x, sizeof(WORD), 1, file);
}
void DataFile::writeInt(int x) {
    ASSERT(file); fwrite(&x, sizeof(int), 1, file);
}
void DataFile::writeDword(DWORD x) {
    ASSERT(file); fwrite(&x, sizeof(DWORD), 1, file);
}
void DataFile::writeFloat(float x) {
    ASSERT(file); fwrite(&x, sizeof(float), 1, file);
}
void DataFile::writeDouble(double x) {
    ASSERT(file); fwrite(&x, sizeof(double), 1, file);
}
*/
```

```
}  
void DataFile::writeString(LPCTSTR x) {  
    ASSERT(file);  
    size_t len = strlen(x);  
    fwrite(x, 1+len, 1, file);  
}  
void DataFile::write(LPVOID x, size_t length) {  
    ASSERT(file); fwrite(x, length, 1, file);  
}  
  
reading  
char DataFile::readChar() {  
    ASSERT(file);  
    char x; fread(&x, sizeof(char), 1, file); return x;  
}  
BYTE DataFile::readByte() {  
    ASSERT(file);  
    BYTE x; fread(&x, sizeof(BYTE), 1, file); return x;  
}  
short DataFile::readShort() {  
    ASSERT(file);  
    short x; fread(&x, sizeof(short), 1, file); return x;  
}  
WORD DataFile::readWord() {  
    ASSERT(file);  
    WORD x; fread(&x, sizeof(WORD), 1, file); return x;  
}  
int DataFile::readInt() {  
    ASSERT(file);  
    int x; fread(&x, sizeof(int), 1, file); return x;  
}  
DWORD DataFile::readDword() {  
    ASSERT(file);  
    DWORD x; fread(&x, sizeof(DWORD), 1, file); return x;  
}  
float DataFile::readFloat() {  
    ASSERT(file);  
    float x; fread(&x, sizeof(float), 1, file); return x;  
}  
double DataFile::readDouble() {  
    ASSERT(file);  
    double x; fread(&x, sizeof(double), 1, file); return x;  
}  
CString DataFile::readString() {  
    ASSERT(file);  
    CString cs; char ch;  
    for (ch = fgetc(file); ch && (!feof(file)); ch = fgetc(file))  
        cs += ch;  
    return cs;  
}  
void DataFile::read(LPVOID buf, size_t length) {  
    ASSERT(file); fread(buf, length, 1, file);  
}
```