

9. COMPENDIU

Deoarece nu este posibil ca disciplinele prevăzute pentru pregătirea și formarea unui specialist, indiferent de domeniul de specializare, să se predea în mod secvențial, astfel încât toate noțiunile și cunoștințele inițiale –necesare predării unei materii– să fie deplin însușite de la cursurile premergătoare (din „amonte”), s-a prevăzut –în încheiere– acest compendiu, ce cuprinde acele elemente fundamentale „din afara” Bazelor electrotehnicii, care sunt absolut necesare pentru însușirea temeinică a teoriei câmpului electromagnetic, în viziunea sistemică și de modelare – simulare pe care ne-am propus-o.

Astfel, cursul așa-numit „Matematici speciale” (care cuprinde capitolele de algebră vectorială, transformări de funcții, ecuațiile fizice-matematice etc.) se predă în paralel sau chiar după „Bazele electrotehnicii”. De aceea, s-a ajuns la necesitatea includerii aici a unui compendiu matematic, cu noțiuni strict necesare studierii teoriei macroscopice a câmpului electromagnetic și analizei circuitelor electrice.

Mai mult, noțiuni de modelare și simulare asistată de calculator (simulare numerică–discretă) nu se predau –decât ocazional– la unele discipline de automatică și informatică, dar acestea mult timp după încheierea cursului „Bazele electrotehnicii”.

În sfârșit, unele instrumente informatice performante (așa cum sunt MATLAB-ul și ANSYS EMAG), deși se găsesc implementate (sub licență) în rețele de calculatoare din U.P.G., de unde sunt simplu de accesat, nu fac obiectul nici unui curs, deși sunt mult solicitate de către doctoranzi și de participanți la cursurile postuniversitare sau la direcțiile de aprofundare.

Considerăm că, reunind în acest capitol, acele noțiuni fundamentale de teoria matematică a câmpului, modelare-simulare și utilizare de produse informatice, dăm posibilitatea cititorului să urmărească mai ușor și eficient expunerea Bazelor electrotehnicii și să realizeze numeroase aplicații practice utile consolidării pregătirii sale.

9.1. Compendiu matematic

Sunt prezentate aici acele noțiuni fundamentale referitoare la teoria matematică a câmpului, fie pentru reamintirea lor, fie pentru expunerea concisă (dar riguroasă) a unor cunoștințe noi.

9.1.1. Noțiuni de algebră vectorială

Produsul scalar a doi vectori. Fie doi vectori oarecare, $\vec{A}$ și $\vec{B}$; produsul lor scalar, care se notează cu $\vec{A} \cdot \vec{B}$, este –prin definiție– produsul modulelor celor doi vectori înmulțit cu cosinusul unghiului α dintre direcțiile acestor doi vectori, adică:

$$\vec{A} \cdot \vec{B} = AB \cos \alpha, \quad 0 \leq \alpha \leq \pi \quad (9.1)$$

rezultatul fiind o mărime scalară (de unde și numele acestui produs).

Produsul scalar poate fi nul chiar dacă $A \neq 0$ și $B \neq 0$, și anume atunci când $\vec{A} \perp \vec{B}$ deoarece în acest caz unghiul dintre direcțiile vectorilor fiind $\alpha = \pi/2$, $\cos \alpha = \pi/2 = 0$. Produsul scalar reprezintă produsul dintre A și măsura proiecției lui $\vec{B}$ pe direcția $\vec{A}$, sau reciproc (fig.9.1).

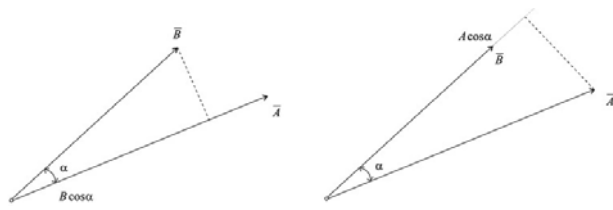


Fig. 9.1

Produsul scalar este comutativ, adică: $\vec{A} \cdot \vec{B} = \vec{B} \cdot \vec{A}$, fiind și distributiv (în adunare) adică:

$$\vec{A} \cdot (\vec{B} + \vec{C}) = \vec{A} \cdot \vec{B} + \vec{A} \cdot \vec{C}.$$

În coordonate carteziene, în care $\vec{A}$ are componentele (scalare) A_x, A_y, A_z și $\vec{B}$ componentele B_x, B_y și B_z , produsul

scalar rezultă:

$$\vec{A} \cdot \vec{B} = (A_x \vec{i} + A_y \vec{j} + A_z \vec{k}) (B_x \vec{i} + B_y \vec{j} + B_z \vec{k}) = A_x B_x + A_y B_y + A_z B_z,$$

deoarece versorii axelor ($\vec{i}, \vec{j}, \vec{k}$), între direcțiile cărora unghiurile sunt $\pi/2$ (și deci cosinusul zero), au produsele scalare:

$$\vec{i} \cdot \vec{i} = \vec{i} \cdot \vec{i} \cos 0 = i^2 = 1, \quad \vec{j} \cdot \vec{j} = j^2 = 1, \quad \vec{k} \cdot \vec{k} = k^2 = 1, \quad \vec{i} \cdot \vec{j} = 1 \cdot 1 \cos \pi/2 = 0, \quad \vec{j} \cdot \vec{k} = 0 \quad \text{și} \quad \vec{k} \cdot \vec{i} = 0.$$

Produsul vectorial a doi vectori. Acest produs, ce se notează cu $\vec{A} \times \vec{B}$, are ca rezultat o mărime vectorială care –prin definiție– are modulul egal cu produsul modulelor celor doi vectori înmulțit cu sinusul unghiului α dintre direcțiile vectorilor și direcția determinată de versorul $\vec{n}$ al perpendicularei pe planul format de cei doi vectori, orientat după regula sistemului drept (așa-zisa „regulă a burghiului drept”), adică:

$$(9.2) \quad \vec{A} \times \vec{B} = (AB \sin \alpha) \vec{n} \Leftarrow \sin \alpha > 0.$$

Regula burghiului drept se formulează astfel: se rotește primul vector al produsului vectorial spre cel de-al doilea, pe „drumul” cel mai scurt ($\sin \alpha > 0$); cu acest sens de rotație se determină sensul în care avansează un burghiu drept, care este chiar sensul versorului $\vec{n}$.

Produsul vectorial poate fi nul chiar dacă A și B sunt diferiți de zero, și anume atunci când $\vec{A}$ și $\vec{B}$ au aceeași direcție, deoarece în acest caz $\alpha = 0$ și deci $\sin \alpha = \sin 0 = 0$. Modulul produsului vectorial, adică $AB \sin \alpha$, reprezintă aria determinată de cei doi vectori – factori ai produsului (fig.9.2.).

Produsul vectorial nu este comutativ, deoarece :

$$\vec{A} \times \vec{B} = -\vec{B} \times \vec{A},$$

așa cum rezultă din definiția (9.2). Produsul vectorial este distributiv în adunare, adică :

$$\vec{A} \times (\vec{B} + \vec{C}) = \vec{A} \times \vec{B} + \vec{A} \times \vec{C}.$$

În coordonate carteziene, produsul vectorial se exprimă, prin distributivitate, astfel:

$$\begin{aligned} \vec{A} \times \vec{B} &= (A_x \vec{i} + A_y \vec{j} + A_z \vec{k}) \times (B_x \vec{i} + B_y \vec{j} + B_z \vec{k}) = \\ &= (A_y B_z - A_x B_y) \vec{i} + (A_z B_x - A_x B_z) \vec{j} + (A_x B_y - A_y B_x) \vec{k}, \end{aligned}$$

deoarece conform definiției (9.2) produsele $\vec{i} \times \vec{i}$, $\vec{j} \times \vec{j}$ și $\vec{k} \times \vec{k}$ sunt nule, iar :

$\vec{i} \times \vec{j} = \vec{k}$, $\vec{j} \times \vec{k} = \vec{i}$, $\vec{k} \times \vec{i} = \vec{j}$ și (datorită necomutativității) $\vec{j} \times \vec{i} = -\vec{k}$, $\vec{k} \times \vec{j} = -\vec{i}$ și $\vec{i} \times \vec{k} = -\vec{j}$. Acest rezultat se poate obține și prin determinantul:

$$\begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ A_x & A_y & A_z \\ B_x & B_y & B_z \end{vmatrix} \equiv \vec{A} \times \vec{B},$$

care se dezvoltă după prima linie (cu minorii formați din componentele celor doi vectori: pe linia

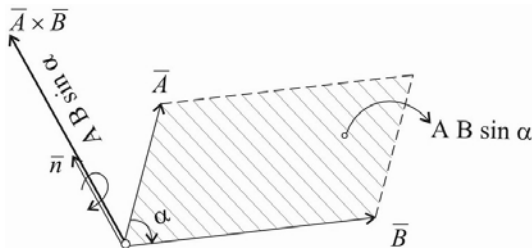


Fig. 9.2

a doua componente vectorului de la stânga produsului vectorial și pe cea de a doua componente vectorului de la dreapta).

Împărțirea cu un vector (?). O astfel de operație nu este posibilă (nu are sens !) deoarece atât produsul scalar cât și cel vectorial nu admit operația inversă (0^{-1}). De exemplu (vezi fig. 9.3), în cazul produsului scalar, determinarea unui vector $\vec{X}$, care înmulțit cu un vector $\vec{A}$ să aibă ca rezultat un scalar S , are o infinitate de soluții, căci $S = AX \cos \alpha = AX_1 \cos \alpha_1 = AX_2 \cos \alpha_2 = \dots$

Produsul mixt a trei vectori. Cu orice trei vectori oarecari, $\vec{A}, \vec{B}$ și $\vec{C}$, se poate defini așa-numitul produs mixt $\vec{ABC}$, care este o mărime scalară egală în valoare absolută cu volumul paralelipipedului construit pe cei trei vectori: Permutând ciclic factorii produsului mixt, rezultatul rămâne neschimbat, adică:

$$\vec{ABC} = \vec{BCA} = \vec{CAB},$$

dar se schimbă doi factori între ei se schimbă semnul rezultatului :

$$\vec{ABC} = -\vec{BAC} \text{ etc.}$$

Produsul mixt a trei vectori coplanari (care formează deci un triedru degenerat) este nul.

În coordonate carteziene, valoarea produsului mixt a trei vectori se poate calcula folosind determinantul :

$$\vec{ABC} = \begin{vmatrix} A_x & A_y & A_z \\ B_x & B_y & B_z \\ C_x & C_y & C_z \end{vmatrix} = A_x(B_y C_z - B_z C_y) + A_y(B_z C_x - B_x C_z) + A_z(B_x C_y - B_y C_x).$$

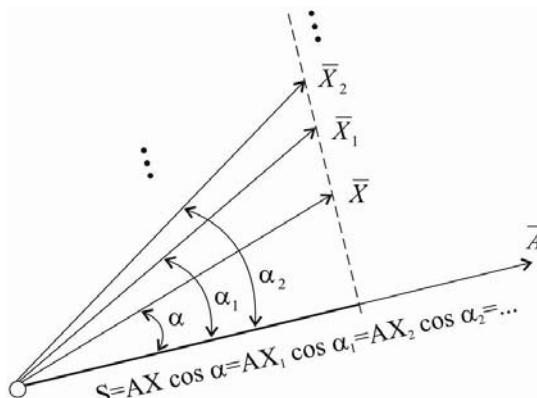


Fig. 9.3

Dublul produs vectorial. Este acel produs între trei vectori ($\vec{A}, \vec{B}$ și $\vec{C}$), scris sub forma $\vec{A} \times (\vec{B} \times \vec{C})$ prin care se obține ca rezultat un vector cuprins în planul determinat de cei doi vectori din interiorul parantezei, conform următoarelor formule de dezvoltare:

$$\vec{A} \times (\vec{B} \times \vec{C}) = (\vec{A} \cdot \vec{C}) \vec{B} - (\vec{A} \cdot \vec{B}) \vec{C}, \quad (9.4)$$

din care rezultă că vectorul dublu produs vectorial este dat prin proiecțiile sale pe cei doi vectori dintre paranteze.

Derivata unui vector. Vectorii pot fi o funcție de unul sau mai mulți parametri scalari, devenind astfel *vectori variabili*. De exemplu, dacă un vector (notat la modul generic cu $\vec{V}$) ia o infinitate de valori în funcție de un parametru scalar t , atunci $\vec{V}$ este o *funcție vectorială* de variabilă t , ceea ce se scrie sub forma:

$$\vec{V} = \vec{V}(t).$$

Prin aceleași proceduri din teoria funcțiilor scalare, se introduc și în studiul funcțiilor vectoriale noțiunile de: limită, continuitate, derivată, diferențială, derivată parțială, integrală etc. Astfel, derivata unei funcții vectoriale de un singur parametru scalar t este prin definiție:

$$\vec{V}'(t) = \frac{d\vec{V}}{dt} = \lim_{\Delta t \rightarrow 0} \frac{\vec{V}(t + \Delta t) - \vec{V}(t)}{\Delta t}, \quad (9.5)$$

iar diferențiala unei funcții vectoriale de un singur parametru scalar t este :

$$d\vec{V} = \vec{V}' dt.$$

Dacă vectorul este dat prin proiecțiile sale pe un triedru trirectangular (de exemplu: componentele vectorului reprezentat în coordonate carteziene, cu versorii $\bar{i}$ pe axa x , $\bar{j}$ pe axa y și $\bar{k}$ pe axa z), adică $\bar{V} = V_x \bar{i} + V_y \bar{j} + V_z \bar{k}$, derivata lui poate fi pusă sub forma:

$$(9.6) \quad \frac{d\bar{V}}{dt} = \frac{dV_x}{dt} \bar{i} + \frac{dV_y}{dt} \bar{j} + \frac{dV_z}{dt} \bar{k}.$$

Considerând doi vectori oarecari, $\bar{u}$ și $\bar{v}$, atunci *regulile de derivare* sunt:

$$(9.7) \quad \begin{aligned} \frac{d}{dt}(\bar{u} + \bar{v}) &= \frac{d\bar{u}}{dt} + \frac{d\bar{v}}{dt}, \\ \frac{d}{dt}(\lambda \bar{u}) &= \frac{d\lambda}{dt} \bar{u} + \lambda \frac{d\bar{u}}{dt}, \\ \frac{d}{dt}(\bar{u} \cdot \bar{v}) &= \frac{d\bar{u}}{dt} \cdot \bar{v} + \bar{u} \cdot \frac{d\bar{v}}{dt}, \\ \frac{d}{dt}(\bar{u} \times \bar{v}) &= \frac{d\bar{u}}{dt} \times \bar{v} + \bar{u} \times \frac{d\bar{v}}{dt} = \frac{d\bar{u}}{dt} \times \bar{v} - \frac{d\bar{v}}{dt} \times \bar{u}, \\ \frac{d}{dt}\{u[\varphi(t)]\} &= \frac{d\bar{u}}{d\varphi} \cdot \frac{d\varphi}{dt}, \end{aligned}$$

unde λ este un parametru scalar $\bar{u}[\varphi(t)]$ este o funcție vectorială de o funcție scalară φ de un parametru scalar t .

Derivatele parțiale ale vectorului $\bar{V} = V_x \bar{i} + V_y \bar{j} + V_z \bar{k}$ sunt:

$$(9.8) \quad \begin{aligned} \frac{\partial \bar{V}}{\partial x} &= \frac{\partial V_x}{\partial x} \bar{i} + \frac{\partial V_y}{\partial x} \bar{j} + \frac{\partial V_z}{\partial x} \bar{k}, \\ \frac{\partial \bar{V}}{\partial y} &= \frac{\partial V_x}{\partial y} \bar{i} + \frac{\partial V_y}{\partial y} \bar{j} + \frac{\partial V_z}{\partial y} \bar{k}, \\ \frac{\partial \bar{V}}{\partial z} &= \frac{\partial V_x}{\partial z} \bar{i} + \frac{\partial V_y}{\partial z} \bar{j} + \frac{\partial V_z}{\partial z} \bar{k}. \end{aligned}$$

Diferențiala vectorului $\bar{V}(x,y,z)$ este:

$$(9.9) \quad d\bar{V} = \frac{\partial \bar{V}}{\partial x} dx + \frac{\partial \bar{V}}{\partial y} dy + \frac{\partial \bar{V}}{\partial z} dz.$$

Observații cu privire la mărimile vectoriale. Așa cum s-a arătat pe larg în subcapitolul 1.2, speciile de mărimi fizice care apar în electromagnetism sunt reprezentabile în modele prin mărimi matematice ca : scalari, vectori, tensori.

i) În ceea ce privește mărimile vectoriale, există așa-numiții: *vectori legați* (de exemplu, vitezele unor corpuri), *vectori alunecători* (de exemplu, forțele) și *vectori liberi* (așa cum sunt translațiile). De aceea, când se efectuează operații cu astfel de mărimi, trebuie să se țină seama de proprietățile mărimilor fizice pe care le reprezintă. Astfel, dacă este vorba de o adunare –adică de aplicarea regulii paralelogramului– nu are sens să se adune vitezele unor puncte diferite sau forțele care acționează asupra unor particule (corpuri) diferite. De asemenea, compunerea a două rotații finite nu se poate face după regula paralelogramului, deși rotațiile finite pot fi descrise cu aceleași elemente (adică: direcție, sens, modul) ca și vectorii.

ii) La schimbarea sistemului de referință trebuie să se țină seama de faptul că mai există și alte tipuri de vectori așa cum sunt: *vectorii polari* (ca de exemplu, vectorii de poziție) și *vectorii axiali* (ca de exemplu, produsul vectorial). Primii sunt invarianți față de schimbarea orientării

sistemului de referință, pe când la trecerea de la un triedru drept la unul stâng, componentele vectorilor axiali (cum este produsul vectorial) își schimbă semnul.

iii) Mărimile scalare (prin care se reprezintă în modele mărimile fizice ca timp, masă, sarcină electrică, intensitate a curentului electric de conducție, temperatură etc.) sunt invariante față de orice schimbare a triedrului de referință (scalarii propriu-ziși), pe când produsul mixt a trei vectori (deși este un scalar) își schimbă semnul la trecerea de la sistemul drept la cel stâng (de aceea se numește, mai precis, *pseudoscalar*).

iu) Grupul aditiv al translațiilor din plan (vectori liberi) este izomorf cu grupul aditiv al numerelor complexe (dar numai pentru adunare). De aceea, înmulțirea a două numere complexe este definită cu totul altfel decât înmulțirea vectorilor (v. paragraful 9.1.3.). Din această cauză, pentru reprezentarea în planul complex ($10j$) a dreptelor (de exemplu cele ce unesc originea axelor 1 și j cu afixul –punctul ce reprezintă numărul complex) nu se utilizează denumirea de vectori, deși regula paralelogramului este comună (fiind aditivă).

9.1.2. Noțiuni de teoria câmpului

Câmp scalar. Dacă într-un domeniu Ω fiecărui punct P îi corespunde o mărime scalară φ , funcția scalară de punct („locală”) definită printr-un anume procedeu se numește câmp scalar. Din punctul de vedere al modelării (v. paragraful 9.2.1.), adică al matematicii, prin câmp scalar se înțelege și mulțimea tuturor valorilor funcției scalare de punct:

$$\varphi(P), \quad \forall P \Rightarrow P \in \Omega.$$

Exemple de câmpuri scalare sunt: câmpul temperaturilor dintr-un cuptor, câmpul presiunilor atmosferice, câmpul potențialelor electrostatice (v. subcap. 2.1.), câmpul densităților de sarcină electrică (v. subcap. 1.2.) etc.

Suprafețe echipotențiale. Locul geometric al punctelor $P \in \Omega$ în care câmpul scalar are o valoare constantă c este o suprafață echipotențială Σ_c :

$$\sum_c = \{P \mid \varphi(P) = c, P \in \Omega\} \Rightarrow \sum_c \subset \Omega. \quad (9.10)$$

Câteva exemple: suprafețele de nivel în câmpul gravitațional sunt suprafețe echipotențiale; volumul și suprafața conductorilor în regim electrostatic sunt echipotențiale (v. subcap. 2.3).

Dacă domeniul Ω este definit în plan, locul geometric al punctelor definite de modelul (9.10) este o curbă echipotențială Γ_c , adică: $\Gamma_c = \{P \mid \varphi(P) = c, P \in \Omega\} \Rightarrow \Gamma_c \subset \Omega$.

Prin trasarea curbelor echipotențiale pentru valorile: $c_0, c_0+k, c_0+2k, c_0+3k, \dots$ se obține o reprezentare geometrică intuitivă (dar și cantitativă) a câmpului scalar (v. subcap. 2.7). Împreună cu așa numitele linii de câmp (ce vor fi definite ceva mai încolo), curbele echipotențiale formează ceea ce se numește **spectrul câmpului**.

Două echipotențiale diferite nu se pot intersecta deoarece $c_1 \neq c_2 \neq \dots \neq c_n$.

Gradientul câmpului scalar. Fie un punct $P_0 \in \Omega$. În general în jurul punctului P_0 câmpul scalar φ va fi diferit de $\varphi(P_0)$ și de aceea este important de știut cum variază $\varphi(P)$ în puncte P imediat vecine lui P_0 , sub forma direcției și sensului în care variația lui φ , $\varphi(P) - \varphi(P_0)$, este cea mai mare; deci maximul acestei variații, considerată din punctul P_0 . Pentru aceasta se înconjoară punctul P_0 cu o suprafață închisă Σ , ce cuprinde volumul v (fig. 9.4), și se calculează limita (9.11) care prin definiție reprezintă gradientul câmpului scalar φ din punctul P_0 , ceea

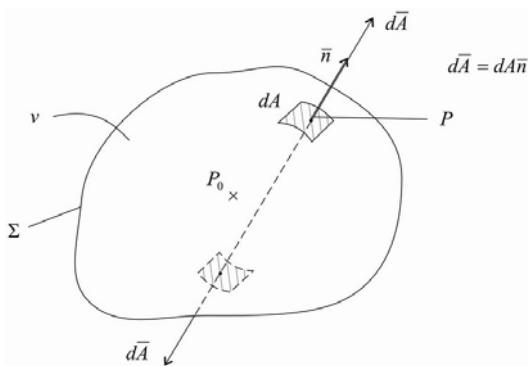


Fig. 9.4

ce se notează cu $(\text{grad } \varphi)_{P_0}$ sau $\text{grad } \varphi (P_0)$ sau –la modul generic $\text{grad } \varphi$:

$$(9.11) \quad (\text{grad } \varphi)_{P_0} = \lim_{V \rightarrow 0; P_0 \in V} \frac{\oint_{\Sigma} \varphi d\vec{A}}{V},$$

în care: φ sunt valorile pe care le ia câmpul scalar pe fiecare element de arie dA , $d\vec{A}$ sunt vectori elementari numiți *vectori de arie orientată*, iar integrala de suprafață închisă $\oint_{\Sigma} \varphi d\vec{A}$ poartă numele de *integrala de înveliș*, rezultatul ei fiind o mărime vectorială.

Elementul de arie orientat (v. fig. 9.4) este un vector elementar, având modulul egal cu suprafața elementară dA , orientarea versorului normalei $\vec{n}$ la suprafața Σ în locul elementului dA ($\vec{n}$ este versorul normalei locale la suprafața Σ și sensul ales arbitrar) de obicei se alege sensul spre exteriorul suprafeței Σ):

$$d\vec{A} = dA \vec{n}.$$

Se vede, din definiția (9.11) și figura 9.4, că dacă φ are aceeași valoare în toate punctele $P \in \Sigma$, adică $\varphi(P) = \text{const.}$ în $\forall P \in \Sigma$, atunci $(\text{grad } \varphi) = 0$, deoarece integrala de înveliș devine:

$$\oint_{\Sigma} \varphi d\vec{A} = \varphi \oint_{\Sigma} d\vec{A} = 0,$$

căci integrala pe o suprafață închisă a elementelor de arie orientată –care este o însumare de elemente vectoriale după regula paralelogramului– este nulă, fiecărui element de arie $d\vec{A}$ corespunzându-i un element de arie diametral, orientat în sens opus (v. fig. 9.4). Dacă φ crește mai mult într-o anumită direcție, atunci acea direcție va influența în cea mai mare măsură rezultatul integralei de înveliș de la numărătorul definiției (9.11).

Așa cum rezultă din definiția (9.11), gradientul câmpului scalar este un vector perpendicular pe suprafața echipotențială, fiind orientat în sensul crescător al câmpului scalar.

Se observă că definiția gradientului (9.11) este analoagă cu definiția derivatei unei funcții, fapt care va fi dovedit la punctul ce urmează.

Pentru simplificare, gradientul se poate nota și cu simbolul „nabla” (∇):

$$(9.12) \quad \text{grad } \varphi = \nabla \varphi,$$

un operator liniar cu caracter diferențial și vectorial.

În coordonate carteziene, gradientul se scrie sub forma:

$$\text{grad } \varphi = \frac{\partial \varphi}{\partial x} \vec{i} + \frac{\partial \varphi}{\partial y} \vec{j} + \frac{\partial \varphi}{\partial z} \vec{k},$$

iar în scriere simbolică:

$$\text{grad } \varphi = \left(\frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \right) \varphi = \nabla \varphi,$$

cu ajutorul căreia se pot stabili reguli formele de calcul, similare cu cele ale algebrei vectoriale, operatorul „nabla” având expresia $\frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} = \nabla$.

Derivata câmpului scalar după o direcție. Considerând un punct oarecare $P_0 \in \Omega$ situat pe o suprafață echipotențială Σ , pentru care $\varphi(P) = \varphi(P_0) \Rightarrow P, P_0 \in \Sigma$, și pornind din P_0 deplasându-ne în $\forall P \in \Sigma$, $\varphi(P)$ rămâne constant.

Dacă deplasarea se face în afara suprafeței Σ , câmpul scalar va avea o variație $\varphi(P) - \varphi(P_0) \Rightarrow P \notin \Sigma$ și $P_0 \in \Sigma$, care depinde de punctul inițial P_0 și de punctul final P , $P_0 \in \Omega$. Prin aceasta deplasare, punctul P descrie un arc P_0P al curbei Γ care admite o tangentă determinată în punctul P_0 . Fie $\vec{s}$ versorul acestei tangente, sensul său arătând sensul pozitiv de parcurgere pe curba Γ (fig. 9.5). Notând cu l_{P_0P} abscisa curbilinie a punctului P față de P_0 , rezultă că valoarea

absolută $|l_{P_oP}|$ este lungimea arcului P_oP , iar semnul arată dacă deplasarea pe curba Γ de la P_o la P se face în sensul pozitiv sau în sensul opus. punctul P_o fiind presupus fix, iar P oricare pe curba Γ , diferența $\varphi(P) - \varphi(P_o)$ este o funcție numai de l_{P_oP} .

Raportul $[\varphi(P) - \varphi(P_o)] / l_{P_oP}$ reprezintă variația medie a funcției $\varphi(P)$ pe unitatea de deplasare, în deplasarea de la P_o la P pe curba Γ . Atunci, prin definiție, limita acestui raport pentru $l_{P_oP} \rightarrow 0$ (în cazul în care există) se numește derivata funcției scalare $\varphi(P)$ după direcția $\bar{s}$ –sau derivata câmpului scalar φ sau supa o direcție $\bar{s}$ – și se notează $\frac{\partial \varphi}{\partial \bar{s}}$:

$$\frac{\partial \varphi}{\partial \bar{s}} = \lim_{l_{P_oP} \rightarrow 0} \frac{\varphi(P) - \varphi(P_o)}{l_{P_oP}}, \quad P \in \Gamma. \quad (9.13)$$

Dacă $\frac{\partial \varphi}{\partial x}$, $\frac{\partial \varphi}{\partial y}$ și $\frac{\partial \varphi}{\partial z}$ există într-o vecinătate a

punctului P_o și sunt continue în P_o , limita (9.13) există și este aceeași pentru toate curbele ($\Gamma, \Gamma', \Gamma'', \dots$) tangente la $\bar{s}$ în P_o . Într-adevăr, dacă x_o, y_o, z_o sunt coordonatele carteziene ale punctului P_o și x, y, z coordonatele lui P , în ipoteza enunțată, diferența $\varphi(P) - \varphi(P_o) = \varphi(x, y, z) - \varphi(x_o, y_o, z_o)$ se mai scrie:

$$\varphi(P) - \varphi(P_o) = \frac{\partial \varphi}{\partial x}(x - x_o) + \frac{\partial \varphi}{\partial y}(y - y_o) + \frac{\partial \varphi}{\partial z}(z - z_o) + \varepsilon_1(x - x_o) + \varepsilon_2(y - y_o) + \varepsilon_3(z - z_o),$$

în care $\varepsilon_i = \varepsilon_i(x, y, z)$ tind către zero când $P \rightarrow P_o$, iar derivatele parțiale $\frac{\partial \varphi}{\partial x}$, $\frac{\partial \varphi}{\partial y}$ și $\frac{\partial \varphi}{\partial z}$ sunt luate în punctul $P_o = (x_o, y_o, z_o)$. Împărțind în ambele părți ale egalității precedente cu l_{P_oP} și observând că, în ipoteza în care Γ admite o tangentă determinată în punctul P_o , limitele:

$$\lim_{l_{P_oP} \rightarrow 0} \frac{x - x_o}{l} = \alpha, \quad \lim_{l_{P_oP} \rightarrow 0} \frac{y - y_o}{l} = \beta \quad \text{și} \quad \lim_{l_{P_oP} \rightarrow 0} \frac{z - z_o}{l} = \gamma$$

există și sunt chiar cosinusurile directoare (α, β, γ) ale tangentei; deci:

$$\bar{s} = \alpha \bar{i} + \beta \bar{j} + \gamma \bar{k}.$$

Atunci, limita (9.13) există și are expresia:

$$\frac{d\varphi}{ds} = \alpha \frac{\partial \varphi}{\partial x} + \beta \frac{\partial \varphi}{\partial y} + \gamma \frac{\partial \varphi}{\partial z}, \quad (9.14)$$

derivatele funcției $\varphi(x, y, z)$ fiind considerate în P_o . Cum această limită este aceeași pentru toate curbele Γ care au versorul tangentei $\bar{s}$ în P_o , în particular deplasarea din P_o într-un punct oarecare din vecinătatea P se poate face chiar pe suportul $\bar{s}$.

Utilizându-se operatorul $\nabla = \frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k}$, expresia (9.14) se mai poate scrie și în forma:

$$\frac{d\varphi}{ds} = (\alpha \bar{i} + \beta \bar{j} + \gamma \bar{k}) \cdot \left(\frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k} \right) \varphi,$$

adică:

$$\frac{\partial \varphi}{\partial \bar{s}} = \bar{s} \nabla \varphi. \quad (9.15)$$

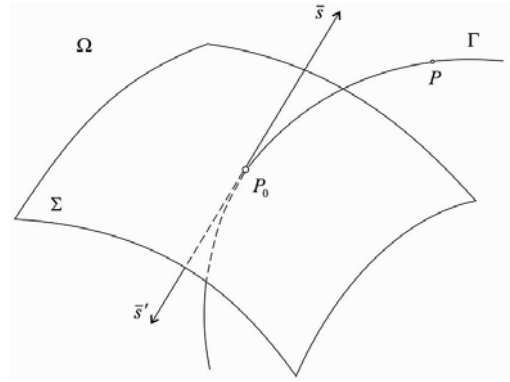


Fig. 9.5

Expresiile (9.14) și (9.15), care sunt identice reprezintă *expresia carteziană* a derivatei funcției scalare $\varphi(P)$ după direcția $\bar{s}$.

Deoarece, conform expresiei (9.12), $\nabla\varphi = \text{grad}\varphi$, derivata câmpului scalar φ după o direcție $\bar{s}$, se poate scrie și în forma:

$$(9.16) \quad \frac{\partial\varphi}{\partial s} = \bar{s}(\text{grad}\varphi),$$

care se consideră în punctul P_0 , adică $\left. \frac{d\varphi}{ds} \right|_{P_0} = \bar{s}(\text{grad}\varphi)_{P_0}$.

În expresiile acestei derivate a câmpului scalar φ după $\bar{s}$, (9.13), (9.14), (9.15) sau (9.16), „direcția $\bar{s}$ ” trebuie considerată ca o direcție înzestrată cu sensul indicat de $\bar{s}$; schimbând sensul lui $\bar{s}$, adică înlocuind pe $\bar{s}$ cu $\bar{s}'$ (vezi fig. 9.5): $\bar{s}' = -\bar{s}$, pentru aceeași deplasare din P_0 în P , lungimile de arc devin: $l'_{P_0P} = -l_{P_0P}$ și deci:

$$\frac{\varphi(P) - \varphi(P_0)}{l'_{P_0P}} = -\frac{\varphi(P) - \varphi(P_0)}{l_{P_0P}},$$

iar când P tinde către P_0 se obține:

$$\frac{d\varphi}{ds'} = -\frac{d\varphi}{ds}.$$

Prin urmare, cele două derivate –după direcțiile $\bar{s}$ și $\bar{s}'$ – sunt egale în valoare absolută și de semne contrarii, deși, în înțelesul geometriei elementare, $\bar{s}$ și $\bar{s}'$ au aceeași direcție.

Derivata câmpului scalar φ în punctul P_0 poate fi calculată după orice direcție $\bar{s}$ (din P_0); un caz posibil este acela al normalei la planul Σ în punctul P_0 ($\Rightarrow P_0 \in \Sigma$), caz în care se va considera vectorul unitar (versorul) normalei $\bar{n}$ la Σ în P_0 (fig. 9.6). Fie P și N punctele în care $\bar{s}$ și $\bar{n}$ intersectează o suprafață echiscalară vecină Σ_v și l_{P_0P}, l_{P_0N} deplasările din $P_0 \in \Sigma$ în P și N ($P, N \in \Sigma_v$) așa ca în figura 9.6.

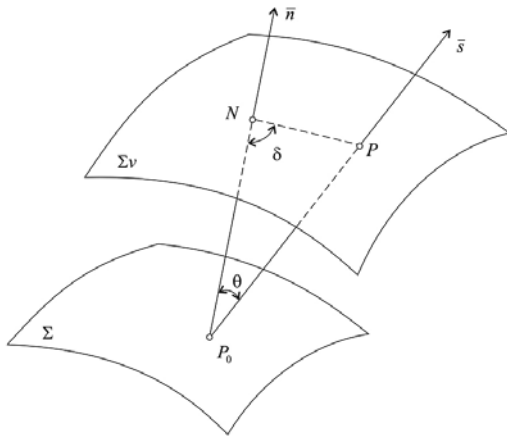


Fig. 9.6

Deoarece P și N sunt pe o aceeași suprafață echiscalară, variația funcției φ este aceeași pentru ambele deplasări:

$$\varphi(P) - \varphi(P_0) = \varphi(N) - \varphi(P_0),$$

ceea ce înseamnă că se poate scrie:

$$(9.17) \quad \frac{\varphi(P) - \varphi(P_0)}{l_{P_0P}} = \frac{\varphi(N) - \varphi(P_0)}{l_{P_0N}} \cdot \frac{l_{P_0N}}{l_{P_0P}}.$$

Aplicându-se relația sinusurilor în triunghiul P_0PN din fig. 9.6 și notând cu δ unghiul din N , se obține:

$$\frac{l_{P_0N}}{l_{P_0P}} = \frac{\sin(\theta + \varphi)}{\sin \delta} = \cos \theta + \sin \theta \cdot \text{ctg} \delta.$$

Deoarece în definiția derivatei lui φ după o direcție, limita este considerată independentă de modul cum l_{P_0P} tinde către zero, se poate presupune că P și N tind către P_0 păstrându-se mereu pe aceeași suprafață de nivel Σ_v , caz în care secanta PN (v. fig. 9.6) tinde către o tangentă în P_0 la suprafața de nivel $P_0 \in \Sigma$ și ca urmare δ tinde către $\pi/2$ și $\text{ctg} \delta \rightarrow 0$. Cum în acest proces θ rămâne constant:

$$\lim_{l_{P_0P} \rightarrow 0} \frac{l_{P_0N}}{l_{P_0P}} = \cos \theta$$

și trecând la limită în ambii membrii ai relației rezultă:

$$\frac{d\varphi}{ds} = \frac{d\varphi}{ds} \cos \theta \quad \text{sau} \quad \vec{s} \cdot (\text{grad} \varphi)_{P_0} = \vec{n} (\text{grad} \varphi)_{P_0} \cdot \cos \theta.$$

De aici rezultă ca derivate lui φ după o direcție normală la suprafața echiscalară, în sensul crescător al câmpului scalar φ este chiar $|\text{grad} \varphi|$, în timp ce după o direcție tangenta la suprafață Σ derivata este nulă.

Prin urmare, gradientul unei funcții scalare φ , într-un punct oarecare $P_0 \in \Omega$, este un vector care arată după ce direcție și în ce sens funcția scalară $\varphi(P)$ este maximă, aceasta fiind întotdeauna normală la suprafața $P_0 \in \Sigma$.

Câmpul vectorial. Dacă într-un domeniu Ω , fiecărui punct P îi corespunde o mărime vectorială $\vec{E}$ (o notație generică), funcția vectorială de punct, $\vec{E}(P)$, definită astfel se numește câmp vectorial. Prin câmp vectorial se mai înțelege și mulțimea tuturor vectorilor $\vec{E}$ din Ω , determinați de $\vec{E}(P)$:

$$\vec{E} = \{ \vec{E}(P) | \forall P \in \Omega \}.$$

Iată numai câteva exemple de câmpuri vectoriale: câmpul vitezelor unui corp în mișcare, câmpul vitezelor unui fluid în mișcare, câmpul gradientului unui câmp scalar, câmpul densității de suprafață a curenților de conducție, câmpul inducției magnetice etc.

Fluxul unui vector. Este o mărime scalară (să o notăm cu Ψ) care –pentru un vector notat cu $\vec{E}$ – se definește, printr-o suprafață Σ , de:

$$\Psi = \int_{\Sigma}^D \vec{E} \cdot d\vec{A}, \quad (9.18)$$

în care $\vec{E}$ este vectorul câmp, iar $d\vec{A}$ este elementul de arie orientată (v. fig. 9.4) într-un punct P de pe suprafață Σ . Se vede ca fluxul Ψ poate fi pozitiv sau negativ, după semnul produsului scalar. Se mai poate scrie și:

$$\Psi = \int_{\Sigma} \vec{E} \cdot \vec{n} dA = \int_{\Sigma} E dA \cos(\vec{E}, \vec{n}).$$

Termenul elementar $\vec{E} \cdot d\vec{A}$, asupra căruia operează integrala de suprafață din definiția (9.18), se numește flux elementar, printr-un element de suprafață dA oarecare:

$$d\Psi = \vec{E} \cdot d\vec{A}.$$

Se numește *tub de flux* un tub cilindric, la a cărui suprafață laterală vectorul câmp este tangent, în lungul căruia fluxul este constant.

Linii de câmp. Într-o primă definiție calitativă, liniile de câmp sunt curbele la care vectorul câmp este tangent la curbă în fiecare punct (în mod similar se definesc *suprafețele de câmp*, la care vectorul câmp este tangent în fiecare punct al suprafeței).

Mai utilă, deoarece intervine și aspectul cantitativ, este definiția prin care linia de câmp este considerată ca fiind axa *tubului de flux unitar*. Dacă se trasează axele tuturor tuburilor de flux unitar dintr-un domeniu Ω , se obține o reprezentare intuitiv-cantitativă a câmpului vectorial, câmpul fiind mai intens acolo unde liniile de câmp sunt mai dese, numărul liniilor de câmp dintr-o secțiune a desenului raportat la aria secțiunii reprezentând valoarea absolută medie a câmpului vectorial.

Divergența câmpului vectorial. Într-un câmp vectorial $\vec{E}$ din domeniul de existență Ω , se consideră un punct oarecare $P_0 \in \Omega$ și, în jurul lui, o suprafață închisă $\Sigma \subset \Omega$ ce are un volum V_{Σ} . În analiza câmpului vectorial este interesant de știut dacă fluxul vectorului $\vec{E}$ prin suprafața Σ ce înconjoară “îndeaproape” punctul P_0 este conservativ, adică dacă, în jurul punctului P_0 , fluxul prin suprafața $P_0 \in \Sigma$ este nul ($\oint_{\Sigma} \vec{E} \cdot d\vec{A} = 0$) sau –cu alte cuvinte– fluxul lui $\vec{E}$ care intră prin Σ este egal cu cel care “iese” din această suprafață închisă. Atributul de conservativ al fluxului unui vector se referă numai la fluxul calculat prin suprafețe închise; dacă fluxul printr-o astfel de

suprafață este diferit de zero, înseamnă că în interiorul suprafeței închise Σ există surse de câmp (pozitive sau negative, după cum $\oint_{\Sigma} \vec{E} \cdot d\vec{A} > 0$ sau, respectiv, $\oint_{\Sigma} \vec{E} \cdot d\vec{A} < 0$). Această situație este cu atât mai precis definită cu cât suprafața Σ este mai mică, adică volumul ei v_{Σ} tinde, la limita, către zero, punctul P_0 rămânând mereu –în timpul acestei treceri $v_{\Sigma} \rightarrow 0$ – în interiorul lui Σ ($P_0 \in v_{\Sigma}$). Făcându-se această limită, fluxul vectorului $\vec{E}$ descrie local, în $\forall P \in \Omega$, această stare de a fi conservativ sau nu. În acest fel s-a ajuns la noțiunea de divergență.

Limita raportului dintre fluxul pe suprafața închisă Σ și volumul mărginit de această suprafață (v_{Σ}), când volumul tinde către zero (conținând tot timpul punctul P_0), se numește divergența câmpului vectorial în acel punct, ceea ce se scrie astfel:

$$(9.19) \quad (\text{div} \vec{E})_{P_0} = \lim_{v_{\Sigma} \rightarrow 0, P_0 \in v_{\Sigma}} \frac{\oint_{\Sigma} \vec{E} \cdot d\vec{A}}{v_{\Sigma}} = \frac{d}{dv} \cdot \oint_{\Sigma} \vec{E} \cdot d\vec{A} \Big|_{P_0} = \nabla \cdot \vec{E}.$$

Divergența este, deci, o mărime scalară de punct (un câmp scalar) care exprimă *densitatea de volum a fluxului*, în acest fel fiind un invariant al câmpului de vectori.

În coordonate carteziene divergența, ca derivată de volum a fluxului, se scrie sub forma generică ($\forall P \in \Omega$):

$$\text{div} \vec{E} = \nabla \cdot \vec{E} = \left(\frac{\partial}{\partial x} \vec{i} + \frac{\partial}{\partial y} \vec{j} + \frac{\partial}{\partial z} \vec{k} \right) \cdot (E_x \vec{i} + E_y \vec{j} + E_z \vec{k}) = \frac{\partial E_x}{\partial x} + \frac{\partial E_y}{\partial y} + \frac{\partial E_z}{\partial z},$$

în care E_x, E_y și E_z sunt proiecțiile vectorului $\vec{E}$, din $\forall P \in \Omega$, pe triedrul $\vec{i}, \vec{j}, \vec{k}$.

Scrierea simbolică: $\text{div} \vec{E} = \nabla \cdot \vec{E}$ (adică divergența rezultă din produsul scalar dintre operatorul derivativ – vectorial ∇ și vectorul câmp $\vec{E}$) permite utilizarea regulilor de calcul existente în algebra vectorială.

Formula lui Gauss-Ostrogradski. Deoarece $\text{div} \vec{E}$ reprezintă densitatea de volum a fluxului Ψ , conform definiției (9.19), rezultă:

$$\text{div} \vec{E} = \frac{d}{dv} \oint_{\Sigma} \vec{E} \cdot d\vec{A} \rightarrow \text{div} \vec{E} \cdot dv = \oint_{\Sigma} \vec{E} \cdot d\vec{A},$$

astfel că:

$$(9.20) \quad \Psi_{\Sigma} = \oint_{\Sigma} \vec{E} \cdot d\vec{A} = \int_{v_{\Sigma}} \text{div} \vec{E} \cdot dv, \quad \forall \Sigma \subset \Omega,$$

care reprezintă formula lui Gauss – Ostrogradski (unde v_{Σ} este volumul mărginit de suprafața închisă Σ). Rezultă de aici că fluxul unui vector printr-o *suprafață închisă* se poate determina fie prin integrala de suprafață: $\oint_{\Sigma} \vec{E} \cdot d\vec{A}$ – dacă se cunosc vectorii $\vec{E}(P)$ în $\forall P \in \Sigma$, fie prin integrala

de volum a divergenței câmpului vectorial: $\int_{v_{\Sigma}} \text{div} \vec{E} \cdot dv$ – dacă se cunosc $(\text{div} \vec{E})_P$ în $\forall P \in v_{\Sigma}$. În teoria matematică a câmpului se spune despre formula (9.20) că “transformă o integrala de suprafață (dublă) într-una de volum (triplă) și invers”.

Formula (9.20) arată că un câmp de divergență zero are fluxul total printr-o suprafață închisă tot zero (fluxul care “intră” în suprafața Σ este egal cu cel ce “iese” din Σ) și –în conformitate cu definiția liniilor de câmp (prezentată anterior)– reiese că orice câmp de divergență nulă are liniile de câmp închise (curbe închise, paralele între ele). Dacă $\exists P \in \Omega \Rightarrow (\text{div} \vec{E})_P \neq 0$, atunci câmpul vectorial $\vec{E}$ din domeniul Ω are liniile de câmp concurente în punctul P (unde exista, deci, o sursă de câmp); pentru $(\text{div} \vec{E})_P < 0$, în P este un “puț” de câmp și liniile de câmp converg (se “întâlnesc”) în P .

Derivata unui vector după o direcție dată. Fie un câmp vectorial $\vec{B}(P), \forall P \in \Omega$, și $\vec{s}$ versorul tangentei la o curbă $\Gamma, \Gamma \in \Omega$, într-un punct al ei $P_0 \in \Gamma_0$. Derivata lui $\vec{B}$ după direcția $\vec{s}$, descrisă de cosinusurile ei directoare $(\alpha, \beta, \gamma): \vec{s} = \alpha\vec{i} + \beta\vec{j} + \gamma\vec{k}$, se definește prin:

$$\left. \frac{d\vec{B}}{ds} \right|_{P_0} \stackrel{D}{=} \lim_{l_{P_0P} \rightarrow 0} \frac{\vec{B}(P) - \vec{B}(P_0)}{l_{P_0P}}, \quad P, P_0 \in \Gamma \quad (9.21)$$

în care l_{P_0P} este „deplasarea” pe Γ din P_0 într-un punct imediat vecin P , în sensul pozitiv al versorului $\vec{s}$.

În coordonatele carteziene, $\vec{B}(P) = B(x, y, z)$ și $\vec{B}(P_0) = \vec{B}(x_0, y_0, z_0)$ astfel că, în condiții de existență și continuitate a derivatelor parțiale ale lui $\vec{B}$ în vecinătatea lui (x_0, y_0, z_0) se poate scrie:

$$\vec{B}(P) - \vec{B}(P_0) = \vec{B}(x, y, z) - \vec{B}(x_0, y_0, z_0) = \frac{\partial \vec{B}}{\partial x}(x - x_0) + \frac{\partial \vec{B}}{\partial y}(y - y_0) + \frac{\partial \vec{B}}{\partial z}(z - z_0)$$

și trecând la limită:

$$\lim_{l_{P_0P} \rightarrow 0} \frac{\vec{B}(x, y, z) - \vec{B}(x_0, y_0, z_0)}{l_{P_0P}} = \frac{\partial \vec{B}}{\partial x} \lim_{l_{P_0P} \rightarrow 0} \frac{x - x_0}{l_{P_0P}} + \frac{\partial \vec{B}}{\partial y} \lim_{l_{P_0P} \rightarrow 0} \frac{y - y_0}{l_{P_0P}} + \frac{\partial \vec{B}}{\partial z} \lim_{l_{P_0P} \rightarrow 0} \frac{z - z_0}{l_{P_0P}},$$

în care limitele din membrul drept sunt cosinusurile directoare (α, β, γ) ale vectorului unitar $\vec{s}$ tangent la Γ în P_0 :

$$\lim_{l_{P_0P} \rightarrow 0} \frac{x - x_0}{l_{P_0P}} = \alpha, \quad \lim_{l_{P_0P} \rightarrow 0} \frac{y - y_0}{l_{P_0P}} = \beta, \quad \lim_{l_{P_0P} \rightarrow 0} \frac{z - z_0}{l_{P_0P}} = \gamma$$

Cu acești cosinusi directori (deoarece $\vec{s} = \alpha\vec{i} + \beta\vec{j} + \gamma\vec{k}$), definiția (9.21) a derivatei unui vector după o direcție dată se scrie:

$$\left. \frac{d\vec{B}}{ds} \right|_{P_0} = \frac{\partial \vec{B}(P_0)}{\partial x} \alpha + \frac{\partial \vec{B}(P_0)}{\partial y} \beta + \frac{\partial \vec{B}(P_0)}{\partial z} \gamma \quad (9.22)$$

sau:

$$\left. \frac{d\vec{B}}{ds} \right|_{P_0} = [(\alpha\vec{i} + \beta\vec{j} + \gamma\vec{k}) \cdot (\frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial y}\vec{j} + \frac{\partial}{\partial z}\vec{k})] \vec{B}(P_0),$$

care în scriere simbolică și pentru oricare punct al lui Γ devine:

$$\frac{d\vec{B}}{ds} = [\vec{s} \cdot \nabla] \vec{B}. \quad (9.23)$$

Relația (9.22), ca și (9.23), arată ca derivata unui vector după o direcție dată, tangentă la linia de câmp în punctul considerat, are drept componente derivatele componentelor vectorului după acea direcție.

Circulația unui vector. Fie $\vec{H}$ un câmp vectorial într-un domeniu Ω și Γ o curbă în Ω ($\Gamma \subset \Omega$), pe care considerăm două puncte A și B (fig. 9.7).

Se numește integrală curbilinie a vectorului $\vec{H}$, din punctul A până în punctul B , integrala:

$$\int_{\Gamma: A \rightarrow B} \vec{H} \cdot d\vec{l}, \quad (9.24)$$

în care: indicele semnului integrală ($\Gamma: A \rightarrow B$)

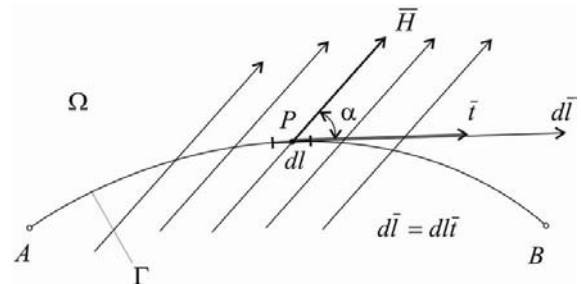


Fig. 9.7

precizează curba $\Gamma \subset \Omega$ din câmp de-a lungul căreia se face integrala și între ce puncte (cu sensul pozitiv de la primul punct indicat, A , la cel de al doilea, B); $\vec{H}$ este vectorul câmp considerat în punctele $P \in \Gamma$, între extremele A și B iar $d\vec{l}$ este așa-numitul *element de curbă orientat*. Sub semnul integralei, între vectorii $\vec{H}$ și $d\vec{l}$ se face produsul scalar $(\vec{H} \cdot d\vec{l} \cos \alpha, \text{ cu } \alpha = \angle(\vec{H}, \vec{t}))$.

Elementul de curbă orientat, $d\vec{l}$ (v. fig. 9.7) este un vector elementar, ce se determină pentru orice punct $P \in \Gamma$, tangent la curbă în punctul P , cu modulul elementar dl (un element de curbă Γ în jurul punctului P , atât de mic încât de-a lungul lui vectorul $\vec{H} = \text{const.}$), orientat după versorul $\vec{t}$ (al tangentei la curba Γ) și cu sensul acestuia (pozitiv pentru sensul $A \rightarrow B$). Deci: $d\vec{l} = dl \vec{t}$.

În coordonate carteziene, integrala curbilinie (9.24) se scrie sub forma:

$$\int_{\Gamma: A \rightarrow B} \vec{H} \cdot d\vec{l} = \int_{\Gamma: A \rightarrow B} (H_x \vec{i} + H_y \vec{j} + H_z \vec{k}) \cdot (dx \vec{i} + dy \vec{j} + dz \vec{k}) = \int_{\Gamma: A \rightarrow B} (H_x dx + H_y dy + H_z dz).$$

Exemple: lucrul mecanic (L_m) se calculează prin integrala curbilinie a unei forțe ($L_m = \int \vec{F} \cdot d\vec{l}$, tensiunea electrică în lungul firului (u_f) este definită prin integrala curbilinie

$$u_f = \int_{\Gamma: A \rightarrow B}^D \vec{E}_e \cdot d\vec{l} \text{ a intensității câmpului electric coulombian } (\vec{E}_e) \text{ dintr-un conductor etc.}$$

Pe baza celor prezentate anterior, *circulația unui vector* se definește foarte simplu ca fiind integrala curbilinie a vectorului de-a lungul oricărei *curbe închise* din domeniul de existență al câmpului vectorial:

$$(9.25) \quad \oint_{\Gamma} \vec{H} \cdot d\vec{l}, \quad \forall \Gamma \subset \Omega.$$

Exemple de circulație a unui vector: tensiunea electromotoare t.e.m. (e) definită prin circulația intensității câmpului electric $\vec{E}$, adică $e = \oint_{\Gamma}^D \vec{E} \cdot d\vec{l}$; tensiunea magnetomotoare t.m.m. (u_m)

sau solenația (θ) care se definește prin circulația intensității câmpului magnetic $\vec{H}$, adică $u_m = \oint_{\Gamma}^D \vec{H} \cdot d\vec{l} = \theta$.

Rotorul vectorului câmp. Există câmpuri în care circulația unor vectori este nulă, ca –de exemplu– lucrul mecanic al forței din câmpul gravitațional, tensiunea în câmp electrostatic etc. Dar sunt și situații în care circulația vectorului câmp este diferită de zero, ca –de exemplu– circulația vitezei unui corp pe un contur în jurul axului de rotație, tensiunea electromotoare de inducție (adică circulația intensității câmpului electric solenoidal), solenația (adică circulația vectorului intensității câmpului magnetic) etc.

Pentru a caracteriza local câmpul vectorial din acest punct de vedere (în $\forall P \in \Omega$, fără să mai fie necesară definirea conturului $\Gamma \subset \Omega$ pe care se face circulația), se introduce noțiunea de *rotor al vectorului* care –în fapt– printr-o trecere la limită reduce conturul Γ la un contur infinitesimal în jurul unui punct $P \in \Omega$. În acest scop, se consideră o suprafață deschisă A_{Γ} care se sprijină pe conturul închis $\Gamma \subset \Omega$ (domeniu în care este definit vectorul, fie acesta $\vec{H}$) ce conține punctul $P_0 \in A_{\Gamma}$ și în care rotorul, notat cu $\text{rot } \vec{H}$, se definește prin expresia:

$$(9.26) \quad \text{rot } \vec{H} \Big|_{P_0} = \vec{u} \lim_{\substack{A \rightarrow 0 \\ P_0 \rightarrow A_{\Gamma}}} \frac{\oint_{\Gamma} \vec{H} \cdot d\vec{l}}{A_{\Gamma}},$$

unde direcția versorului $\bar{u}$ reprezintă orientarea elementului de arie $A_\Gamma \rightarrow 0$ (în jurul punctului P_0) pentru care valoarea absolută $|\text{rot}\bar{H}|$ este maximă.

Cuvântul „rotor” a fost utilizat în scopul de a sugera un fapt fizic (din mecanica fluidelor), acela al „liniilor de vârtej”. Dacă într-un câmp vectorial $\exists P \in \Omega$ pentru care rotorul vectorului este diferit de zero, atunci câmpul se numește „ rotațional ” (acesta este cazul câmpului magnetic, ale cărui mărimi de stare –intensitatea câmpului magnetic $\bar{H}$, inducția magnetică $\bar{B}$ – sunt câmpuri rotaționale: $\text{rot}\bar{H} = 0$ și $\text{rot}\bar{B} = 0$).

Se poate arăta că expresia de definire a rotorului (9.26) este echivalentă cu:

$$\left. \text{rot}\bar{H} \right|_{P_0} = \lim_{\substack{v_\Sigma \rightarrow 0 \\ P_0 \in v_\Sigma}} \frac{\oint_{\Sigma} d\bar{A} \times \bar{H}}{v_\Sigma},$$

în care Σ este o suprafață închisă care înconjoară punctul P_0 , v_Σ este volumul mărginit de această suprafață, iar $d\bar{A}$ este elementul de arie orientat al lui Σ .

În coordonate carteziene rotorul are expresia:

$$\text{rot}\bar{H} = \left(\frac{\partial H_z}{\partial y} - \frac{\partial H_y}{\partial z} \right) \bar{i} + \left(\frac{\partial H_x}{\partial z} - \frac{\partial H_z}{\partial x} \right) \bar{j} + \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \bar{k},$$

ceea ce permite utilizarea tabelului:

$$\text{rot}\bar{H} = \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ H_x & H_y & H_z \end{vmatrix}, \quad (9.27)$$

care se dezvoltă ca un determinant, după prima linie.

În scriere simbolică:

$$\text{rot}\bar{H} = \nabla \times \bar{H} = \left(\frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k} \right) \times (H_x \bar{i} + H_y \bar{j} + H_z \bar{k}), \quad (9.28)$$

ceea ce permite aplicarea formală a regulilor de calcul din algebra vectorială.

Teorema lui Stokes. Pe baza definiției (9.26) se poate enunța următoarea teoremă: circulația câmpului vectorial $\bar{H}$ este egal cu fluxul rotorului acestui vector prin orice suprafață (oarecare) A_Γ mărginită de curba închisă Γ , adică:

$$\oint_{\Gamma} \bar{H} \cdot d\bar{l} = \int_{A_\Gamma} \text{rot}\bar{H} \cdot d\bar{A} \quad (9.28)$$

Ca semnificație fizică, teoremă lui Stokes (9.28) arată că în cazul câmpului rotațional ($\text{rot}\bar{H} \neq 0$), circulația vectorului câmp este diferită de zero și invers.

Din punctul de vedere al teoriei matematice, se spune că teorema lui Stokes (9.28) transformă o integrală simplă într-una dublă (de suprafață) sau invers.

Operatorul diferențial-vectorial. Acest operator se notează cu ∇ („nabla”) și este un operator liniar, ce se supune, formal, regulilor algebrei vectoriale și calculului diferențial. De exemplu, dacă ∇ se aplică produsului $\varphi \cdot \psi$ dintre două mărimi scalare, se va scrie:

$$\nabla(\varphi \cdot \psi) = \varphi \nabla \psi + \psi \nabla \varphi, \quad (9.30)$$

iar dacă se aplică, prin produs scalar, produsului dintre un scalar φ și un vector $\bar{E}$ va rezulta:

$$\nabla \cdot (\varphi \bar{E}) = \bar{E} \cdot \nabla \varphi + \varphi \nabla \cdot \bar{E}, \quad (9.30')$$

ceea ce înseamnă:

$$(9.30'') \quad \operatorname{div}(\varphi \bar{E}) = \bar{E} \operatorname{grad} \varphi + \varphi \operatorname{div} \bar{E},$$

care, în ambii membri ai egalității, conține mărimi scalare.

Alte cazuri, des întâlnite în teoria câmpului electromagnetic, sunt:

$$(9.31) \quad \begin{cases} \nabla \times (\varphi \bar{E}) = \varphi (\nabla \times \bar{E}) - \bar{E} \times \nabla \varphi \text{ sau} \\ \operatorname{rot}(\varphi \bar{E}) = \varphi \operatorname{rot} \bar{E} - \bar{E} \times \operatorname{grad} \varphi \end{cases}$$

$$(9.32) \quad \begin{cases} \nabla \cdot (\bar{u} \times \bar{v}) = \bar{v} \cdot (\nabla \times \bar{u}) - \bar{u} \cdot (\nabla \times \bar{v}) \text{ sau} \\ \operatorname{div}(\bar{u} \times \bar{v}) = \bar{v} \operatorname{rot} \bar{u} - \bar{u} \operatorname{rot} \bar{v} \end{cases}$$

$$(9.33) \quad \nabla \times (\bar{u} \times \bar{v}) = \bar{u}(\nabla \cdot \bar{v}) - \bar{v}(\nabla \cdot \bar{u}) + (\bar{v} \nabla) \bar{u} - (\bar{u} \nabla) \bar{v};$$

$$(9.34) \quad \nabla(\bar{u} \cdot \bar{v}) = \bar{v} \times (\nabla \times \bar{u}) + \bar{u} \times (\nabla \times \bar{v}) + (\bar{v} \nabla) \bar{u} + (\bar{u} \nabla) \bar{v}$$

în care $\bar{u}$ și $\bar{v}$ sunt niște funcții vectoriale, iar parantezele fixează prioritățile în executarea operațiilor în cadrul produsului a trei vectori ($\Delta, \bar{u}$ și $\bar{v}$).

Prin urmare:

- operatorul ∇ aplicat unui scalar determină vectorul numit gradient $\nabla \varphi = \operatorname{grad} \varphi = \bar{E}$

- operatorul ∇ aplicat prin produs scalar unui vector determină scalarul numit divergență ($\nabla \cdot \bar{E} = \operatorname{div} \bar{E} = \psi$)

- operatorul ∇ aplicat prin produs vectorial unui vector determină vectorul numit rotor ($\nabla \times \bar{H} = \operatorname{rot} \bar{H} = \bar{J}$)

În propozițiile acestea, φ și ψ sunt câmpuri scalare (funcții scalare de punct dintr-un anumit domeniu de existență a câmpului), iar $\bar{E}, \bar{H}$ și $\bar{J}$ sunt câmpuri vectoriale (funcții vectoriale de punct).

Operatori diferențiali de ordinul II. În analiza câmpurilor apar adesea expresii în care operatorul ∇ se repetă, ca de exemplu:

$$j) \quad \operatorname{div} \operatorname{grad} \varphi = \nabla \cdot (\nabla \varphi) = \nabla^2 \varphi = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} + \frac{\partial^2 \varphi}{\partial z^2}, \quad (9.35)$$

în care operatorul $\Delta \stackrel{D}{=} \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ se numește „laplacean” și este un operator diferențial de ordinul doi, liniar, ce a fost dedus prin regulile formale ale operatorului ∇ :

$$(9.36) \quad \nabla \cdot \nabla = \left(\frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k} \right) \cdot \left(\frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k} \right) = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \stackrel{D}{=} \Delta;$$

$$jj) \quad \operatorname{rot} \operatorname{grad} \varphi = 0 \text{ sau } \nabla \times (\nabla \varphi) = 0 \quad (9.37)$$

căci:

$$\nabla \times (\nabla \varphi) = \left(\frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k} \right) \times \left[\left(\frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k} \right) \varphi \right] = 0,$$

deoarece: $\bar{i} \times \bar{i} = \bar{j} \times \bar{j} = \bar{k} \times \bar{k} = 0$, $\bar{i} \times \bar{j} = -\bar{k}$ și $\bar{j} \times \bar{i} = \bar{k}$, $\bar{i} \times \bar{k} = -\bar{j}$ și $\bar{k} \times \bar{i} = \bar{j}$, $\bar{j} \times \bar{k} = \bar{i}$ și $\bar{k} \times \bar{j} = -\bar{i}$;

$$jjj) \quad \operatorname{div} \operatorname{rot} \bar{E} = \nabla \cdot (\nabla \times \bar{E}) = 0, \quad (9.38)$$

pentru că:

$$\begin{aligned}\nabla \cdot (\nabla \times \bar{E}) &= \left(\frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k} \right) \cdot \begin{vmatrix} \bar{i} & \bar{j} & \bar{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ E_x & E_y & E_z \end{vmatrix} = \left(\frac{\partial}{\partial x} \bar{i} + \frac{\partial}{\partial y} \bar{j} + \frac{\partial}{\partial z} \bar{k} \right) \cdot \\ &\cdot \left[\left(\frac{\partial E_z}{\partial y} - \frac{\partial E_y}{\partial z} \right) \bar{i} + \left(\frac{\partial E_x}{\partial z} - \frac{\partial E_z}{\partial x} \right) \bar{j} + \left(\frac{\partial E_y}{\partial x} - \frac{\partial E_x}{\partial y} \right) \bar{k} \right] = \\ &= \left(\frac{\partial^2 E_z}{\partial x \partial y} - \frac{\partial^2 E_y}{\partial x \partial z} \right) 1 + \left(\frac{\partial^2 E_x}{\partial y \partial z} - \frac{\partial^2 E_z}{\partial y \partial x} \right) 1 + \left(\frac{\partial^2 E_y}{\partial z \partial x} - \frac{\partial^2 E_x}{\partial z \partial y} \right) 1 = 0\end{aligned}$$

deoarece: $\bar{i}^2 = \bar{j}^2 = \bar{k}^2 = 1$; $\bar{i} \cdot \bar{j} = \bar{i} \cdot \bar{k} = \bar{j} \cdot \bar{i} = \bar{j} \cdot \bar{k} = \bar{k} \cdot \bar{i} = \bar{k} \cdot \bar{j} = 1 \cdot 1 \cos \frac{\pi}{2} = 0$ și

$\frac{\partial^2}{\partial x \partial y} = \frac{\partial^2}{\partial y \partial x}$, $\frac{\partial^2}{\partial x \partial z} = \frac{\partial^2}{\partial z \partial x}$ și $\frac{\partial^2}{\partial y \partial z} = \frac{\partial^2}{\partial z \partial y}$ (datorită comutativității operatorilor ∂ , care sunt liniari);

$$\text{jv) } \text{rot rot } \bar{A} = \text{grad}(\text{div } \bar{A}) - \Delta \bar{A}, \quad (9.39)$$

ceea ce rezultă din:

$$\text{rot rot } \bar{A} = \nabla \times (\nabla \times \bar{A}) = \nabla(\nabla \cdot \bar{A}) - \nabla^2 \bar{A} = \nabla(\nabla \cdot \bar{A}) - \Delta \bar{A},$$

unde $\nabla \times (\nabla \times \bar{A})$ este un dublu produs vectorial, care –conform formulei de dezvoltare (9.4)–se scrie:

$$\nabla \times (\nabla \times \bar{A}) = (\nabla \cdot \bar{A}) \nabla - (\nabla \cdot \nabla) \bar{A} = \nabla(\nabla \cdot \bar{A}) - \nabla^2 \bar{A}.$$

Derivata substanțială în raport cu timpul. Să presupunem că un câmp scalar de punct $\varphi(P)$, variază și în timp (t), caz deseori întâlnit în natură. Atunci, funcția scalară se scrie: $\varphi = \varphi(P, t)$, iar dacă –într-un sistem de referință cartezian– fiecărui punct P îi atașăm un vector de poziție $\bar{r}$ se va putea scrie:

$$\varphi = \varphi(P, t) = \varphi = \varphi(\bar{r}, t) \text{ unde } \bar{r} = x(t)\bar{i} + y(t)\bar{j} + z(t)\bar{k},$$

ceea ce înseamnă că însuși punctul P își modifică locul în timp.

Conform definiției, derivata lui φ în raport cu timpul va fi derivata totală:

$$\frac{d_s \varphi}{dt} = \frac{\partial \varphi}{\partial t} + \frac{\partial \varphi}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial \varphi}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial \varphi}{\partial z} \cdot \frac{dz}{dt}, \quad (9.40)$$

în care derivatele dx/dt , dy/dt și dz/dt sunt componentele (proiecțiile) unei viteze, $\bar{w}$ după triedrul $\bar{i}, \bar{j}, \bar{k}$:

$$\bar{w} = \frac{dx}{dt} \bar{i} + \frac{dy}{dt} \bar{j} + \frac{dz}{dt} \bar{k}$$

viteză cu care se deplasează punctul P . Cum punctul P este „legat” (asociat) unui suport material (un corp solid, un fluid, o particulă etc.), derivata a căpătat numele de „substanțială” (de la materialul sau substanța în care se ia punctul P) și i s-a atribuit indicele s (de la substanțială).

Explicitând viteza $\bar{w}$ relația (9.40) devine:

$$\frac{d_s \varphi}{dt} = \frac{\partial \varphi}{\partial t} + \left(\frac{\partial \varphi}{\partial x} \bar{i} + \frac{\partial \varphi}{\partial y} \bar{j} + \frac{\partial \varphi}{\partial z} \bar{k} \right) \cdot \left(\frac{dx}{dt} \bar{i} + \frac{dy}{dt} \bar{j} + \frac{dz}{dt} \bar{k} \right) = \frac{\partial \varphi}{\partial t} + \nabla \varphi \cdot \bar{w}$$

adică, derivata substanțială în raport cu timpul a câmpului scalar φ este:

$$\frac{d_s \varphi}{dt} = \frac{\partial \varphi}{\partial t} + \bar{w} \nabla \varphi + \bar{w} \text{grad} \varphi. \quad (9.41)$$

În mod similar se determină și derivata substanțială în raport cu timpul a unui câmp vectorial $\bar{E} = \bar{E}(P, t) = \bar{E}(x, y, z, t)$

$$(9.42) \quad \frac{d_s \bar{E}}{dt} = \frac{\partial \bar{E}}{\partial t} + (\bar{w} \nabla) \bar{E}.$$

Derivata substanțială (materială) a fluxului în raport cu timpul. Fie $\bar{B} \in \Omega$ un câmp vectorial în care $\bar{B}$ este o funcție de punct și variază în timp (t): $\bar{B} = \bar{B}(p, t) = \bar{B}(x, y, z, t)$, în care x, y, z sunt coordonatele unui punct $P \in \Omega$ într-un sistem de referință cartezian, și fie $\Gamma \in \Omega$ o curbă închisă (un contur "material", realizat –de pildă– sub forma unei spire a unei bobine electrice), care se deplasează în câmp cu o viteză $\bar{w} = w_x \bar{i} + w_y \bar{j} + w_z \bar{k}$, componentele acestei viteze, în coordonatele carteziene, fiind: $w_x = dx/dt$, $w_y = dy/dt$ și $w_z = dz/dt$. Fluxul vectorului $\bar{B}$ prin orice suprafață Σ_Γ , mărginită pe conturul Γ , $\varphi = \int_{\Sigma_\Gamma} \bar{B} \cdot d\bar{A}$ va fi variabil în timp deoarece atât $\bar{B}$ cât și Γ (deci și Σ_Γ) se modifică în timp: $\varphi = \varphi(t)$.

În multe situații este necesar să se calculeze derivata în raport cu timpul a acestui flux, adică:

$$\frac{d_s \varphi}{dt} = \frac{d_s}{dt} \int_{\Sigma_\Gamma} \bar{B} \cdot d\bar{A},$$

care se numește derivata materială (substanțială) a fluxului în raport cu timpul, ceea ce a impus indicele s .

Va rezulta:

$$(9.43) \quad \begin{aligned} \frac{d_s \varphi}{dt} &= \frac{d_s}{dt} \int_{\Sigma_\Gamma(t)} \bar{B}(x, y, z, t) \cdot d\bar{A} = \int_{\Sigma_\Gamma} \frac{\partial \bar{B}}{\partial t} \cdot d\bar{A} + \int_{\Sigma_\Gamma} \left(\frac{\partial \bar{B}}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial \bar{B}}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial \bar{B}}{\partial z} \cdot \frac{dz}{dt} \right) d\bar{A} = \\ &= \int_{\Sigma_\Gamma} \frac{\partial \bar{B}}{\partial t} \cdot d\bar{A} + \int_{\Sigma_\Gamma} \left[(w_x \bar{i} + w_y \bar{j} + w_z \bar{k}) \cdot \left(\frac{\partial \bar{B}}{\partial x} \bar{i} + \frac{\partial \bar{B}}{\partial y} \bar{j} + \frac{\partial \bar{B}}{\partial z} \bar{k} \right) \right] d\bar{A} = \\ &= \int_{\Sigma_\Gamma} \frac{\partial \bar{B}}{\partial t} \cdot d\bar{A} + \int_{\Sigma_\Gamma} (\bar{w} \cdot \nabla) \bar{B} \cdot d\bar{A} = \int_{\Sigma_\Gamma} \frac{\partial \bar{B}}{\partial t} \cdot d\bar{A} + \int_{\Sigma_\Gamma} \bar{w} (\nabla \cdot \bar{B}) \cdot d\bar{A} + \int_{\Sigma_\Gamma} \nabla \times (\bar{B} \times \bar{w}) \cdot d\bar{A} \end{aligned}$$

ceea ce se mai poate scrie și în forma:

$$(9.44) \quad \frac{d_s \varphi}{dt} = \int_{\Sigma_\Gamma} \frac{d_f \bar{B}}{dt} \cdot d\bar{A} = \int_{\Sigma_\Gamma} \frac{\partial \bar{B}}{\partial t} \cdot d\bar{A} + \int_{\Sigma_\Gamma} \bar{w} \operatorname{div} \bar{B} \cdot d\bar{A} + \int_{\Sigma_\Gamma} \operatorname{rot}(\bar{B} \times \bar{w}) \cdot d\bar{A},$$

în care $d_f \bar{B}/dt$ se numește derivata de flux în raport cu timpul (de unde și indicele f al derivatei d). Cum elementul de arie $d\bar{A}$ este unul oarecare, din a doua egalitate a expresiei (9.44) rezultă:

$$(9.45) \quad \frac{d_f \bar{B}}{dt} = \frac{\partial \bar{B}}{\partial t} + \bar{w} \operatorname{div} \bar{B} + \operatorname{rot}(\bar{B} \times \bar{w}).$$

Derivata de integrală de volum în raport cu timpul. Apar cazuri practice în care este necesar să se calculeze expresia:

$$(9.46) \quad \frac{d}{dt} \int_v V dv = \int_v \frac{dv V}{dt} dv,$$

în care termenul $d_v V/dt$ se numește derivata de volum în raport cu timpul a câmpului scalar variabil în timp, V . Prin urmare, V este o funcție scalară de punct $P \in v$ și timp, $V = V(P, t)$, sistemul de puncte $P \subset v$ fiind în mișcare prin raport cu sistemul de referință, cu viteza sistemului de puncte dată de $\bar{w} = w_x \bar{i} + w_y \bar{j} + w_z \bar{k} = \bar{i} dx/dt + \bar{j} dy/dt + \bar{k} dz/dt$. Atunci:

$$\begin{aligned}
\frac{d_v V}{dt} &= \frac{d_v}{dt} V(x, y, z, t) = \frac{\partial V}{\partial t} + \frac{\partial V}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial V}{\partial y} \cdot \frac{dy}{dt} + \frac{\partial V}{\partial z} \cdot \frac{dz}{dt} = \\
&= \frac{\partial V}{\partial t} + \left(\frac{\partial V}{\partial x} \bar{i} + \frac{\partial V}{\partial y} \bar{j} + \frac{\partial V}{\partial z} \bar{k} \right) \cdot \left(\frac{dx}{dt} \bar{i} + \frac{dy}{dt} \bar{j} + \frac{dz}{dt} \bar{k} \right) = \\
&= \frac{\partial V}{\partial t} + \nabla(V \bar{w}) = \frac{\partial V}{\partial t} + \operatorname{div}(V \bar{w}),
\end{aligned} \tag{9.47}$$

astfel că derivata de integrală de volum în raport cu timpul (9.46) devine:

$$\frac{d}{dt} \int_v V dv = \int_v \frac{\partial V}{\partial t} dv + \int_v \operatorname{div}(V \bar{w}) dv. \tag{9.48}$$

Aplicându-se formula (9.20), a lui Gauss-Ostrogradski, potrivit căreia integrala de volum a divergenței unui vector (deci a densității de volum a fluxului vectorului) este egală cu integrala prin suprafață închisă a fluxului elementar al vectorului, dacă volumul v este mărginit de o suprafață închisă $\Sigma(\Sigma_v)$, atunci ultimul termen al expresiei (9.48) se poate scrie:

$$\int_{v_v} \operatorname{div}(V \bar{w}) dv = \oint_{\Sigma_v} V \bar{w} \cdot d\bar{A}$$

și atunci derivata de integrală de volum (9.48) mai are și forma:

$$\frac{d}{dt} \int_v V dv = \int_v \frac{\partial V}{\partial t} dv + \int_{\Sigma_v} V \bar{w} \cdot d\bar{A}.$$

De exemplu, în mecanica fluidelor funcția scalară poate fi densitatea de volum a unui fluid ($v \equiv \rho$), caz în care $\int_{\Sigma} \rho \bar{w} \cdot d\bar{A}$ este debitul masei prin Σ (în kg/s), iar dacă se elimină ρ , $\int_{\Sigma} \bar{w} \cdot d\bar{A}$ reprezintă debitul volumic al fluidului prin Σ (în m³/s).

9.1.3. Reprezentarea în planul complex a mărimilor armonice (sinusoidale)

În studiul circuitelor electrice (așa cum s-a văzut în capitolele 7 și 8), mărimile electrice de circuit (t.e.m., curenții electrici din laturile circuitelor, tensiunile la borne, puterile instantanee etc.) se reprezintă prin mărimi matematice (modele) care sunt funcții sinusoidale de forma:

$$x = \sqrt{2} X \sin(\omega t + \varphi),$$

în care: x este valoarea instantanee a mărimii electrice de circuit considerate (t.e.m./e, i, u, p etc), deci o funcție reală de variabilă reală în timp $x(t) = x(t + kT)$, $t \in \mathbf{R}_+$, $k \in \mathbf{N}$ și T - perioada de repetiție; X - valoarea eficace (efectivă) a mărimii considerate ($X = \{E, I, U\}$); $\omega = 2\pi f$ este pulsația (frecvența unghiulară); f - frecvența de repetiție a lui x ($f = 1/T$); $(\omega t + \varphi)$ - argumentul funcției sinusoidale și φ - faza inițială (se presupune $\varphi = \text{const.}$ cu t). O funcție sinusoidală este determinată dacă se cunosc: amplitudinea, adică $x(\pi/2) = \sqrt{2} X \sin \pi/2 = \sqrt{2} X$, și argumentul $\omega t + \varphi$, iar la pulsație constantă funcția este univoc determinată prin valoarea efectivă X și faza inițială φ (ceea ce, simbolic se scrie $x|_{\varphi}$).

Avându-se în vedere acest fapt precum și acela că operațiile cu funcții de timp trigonometrice sunt greoaie și laborioase (ceea ce s-a văzut în capitolul 8, în analiza circuitelor electrice de curent alternativ, în care intervin frecvent operații de adunare, înmulțire și împărțire), s-a generalizat utilizarea unor reprezentări în planul complex a mărimilor sinusoidale, ceea ce simplifică mult calculele. S-a plecat de la observația că în planul complex $10j$ (fig. 9.8a), oricărui număr complex (a, jb) îi corespunde un punct $M(a, b)$ - numit afix, iar dreapta OM (ce unește originea planului O , cu punctul M) se poate reprezenta prin:

$$OM = a + jb = |OM| \cos \alpha + j |OM| \sin \alpha = |OM| e^{j\alpha} = \operatorname{Re} OM + j \operatorname{Im} OM,$$

astfel că orice funcție sinusoidală este partea imaginară a dreptei origine – afix din planul complex: $|\underline{OM}| \sin \alpha = \text{Im } \underline{OM}$.

În ultimele două expresii s-au folosit notațiile: 1 – unitatea reală; j – unitatea imaginară (definit prin $j^2 = -1$) și e – baza logaritmulor naturali.

Observație. În acest manual s-a făcut o abatere de la notațiile clasice și anume: unitatea imaginară (notată clasic cu i) s-a înlocuit cu j, iar e – baza logaritmulor naturali, cu e ; aceasta pentru că în cele mai uzuale aplicații – analiza circuitelor electrice – se utilizează, consacrat, notațiile i și e pentru valorile instantanee ale curenților și tensiunilor electromotoare, evitându-se astfel posibilele confuzii.

Se mai remarcă faptul că unitatea imaginară j este un operator de rotație cu unghiul $\pi/2$ (fig. 9.8.a), deoarece: $j \cdot 1 = 1 \angle \pi/2$, $j \cdot j = j^2 = -1 = 1 \angle \pi$, $j \cdot j^2 = j(-1) = -j = 1 \angle 3\pi/2$, $j^4 = j \cdot j^3 = j^2(-1) = (-1)(-1) = 1$.

În acest fel, se poate face următoarea corespondență biunivocă: dacă $x = \sqrt{2}X \sin(\omega t + \varphi)$, atunci $x \hat{=} X(\omega)$ cu $X(\omega)$ reprezentat în planul complex (fig. 9.8.b) de forma $X(\omega) = X \cos(\omega t + \varphi) + jX \sin(\omega t + \varphi) = X e^{j(\omega t + \varphi)} = X \angle (\omega t + \varphi)$, iar dacă $X(\omega) = X \angle (\omega t + \varphi)$, atunci

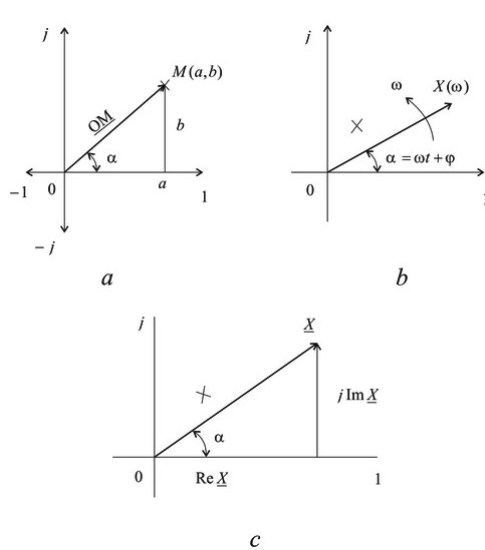


Fig. 9.8

$X(\omega) \hat{=} x$ cu x reprezentat în planul timp (tox) de forma $x = \text{Im } X(\omega) = \sqrt{2}X \sin(\omega t + \varphi)$. Deci, funcția de timp sinusoidală x i s-a asociat un așa numit vector $X(\omega)$ în planul complex, un vector învârtitor care se rotește în sens trigonometric cu viteza ω (vezi fig. 9.8.b) și are modulul X egal cu valoarea efectivă a funcției sinusoidale (deoarece în regim armonic, în orice situație $X_{\max} = \sqrt{2}X$, subînțelegându-se, astfel, că în planul timpului valoarea maximală a mărimii sinusoidale va fi modulul reprezentării în planul complex multiplicat cu $\sqrt{2}$).

Prin această corespondență biunivocă, operației de adunare a funcțiilor trigonometrice îi corespunde operația de adunare a numerelor complexe. De exemplu (v. subcap. 8.5), dacă într-un nod al unei rețele de curent alternativ sinusoidal sunt conectate trei laturi ce au curenții i_1 , i_2 și i_3 , se va putea scrie:

$$\begin{cases} i_1 + i_2 = i_3 \\ \sqrt{2}I_1 \sin(\omega t + \varphi_1) + \sqrt{2}I_2 \sin(\omega t + \varphi_2) = \sqrt{2}I_3 \sin(\omega t + \varphi_3) \end{cases}$$

$$\begin{cases} i_1 = \text{Im}[I_1 \cos(\omega t + \varphi_1) + jI_1 \sin(\omega t + \varphi_1)] = \text{Im } I_1 e^{j(\omega t + \varphi_1)} = \text{Im } I_1(\omega) \\ i_2 = \text{Im}[I_2 \cos(\omega t + \varphi_2) + jI_2 \sin(\omega t + \varphi_2)] = \text{Im } I_2 e^{j(\omega t + \varphi_2)} = \text{Im } I_2(\omega) \\ i_3 = \text{Im}[I_3 \cos(\omega t + \varphi_3) + jI_3 \sin(\omega t + \varphi_3)] = \text{Im } I_3 e^{j(\omega t + \varphi_3)} = \text{Im } I_3(\omega) \\ \text{Im}[I_1 e^{j(\omega t + \varphi_1)}] + \text{Im}[I_2 e^{j(\omega t + \varphi_2)}] = \text{Im}[I_3 e^{j(\omega t + \varphi_3)}] \Rightarrow \\ \Rightarrow I_1 e^{j(\omega t + \varphi_1)} + I_2 e^{j(\omega t + \varphi_2)} = I_3 e^{j(\omega t + \varphi_3)} \text{ sau } I_1(\omega) + I_2(\omega) = I_3(\omega), \end{cases}$$

datorită corespondenței biunivoce:

$$i_1 \hat{=} I_1 e^{j(\omega t + \varphi_1)}; i_2 \hat{=} I_2 e^{j(\omega t + \varphi_2)} \text{ și } i_3 \hat{=} I_3 e^{j(\omega t + \varphi_3)}.$$

Această dublă corespondență a elementelor și a operațiilor se numește *izomorfism* și ea dă posibilitatea să se opereze cu numere complexe în calculul *circuitelor electrice liniare* în regim armonic permanent, ceea ce acoperă un număr foarte mare de aplicații practice.

Trebuie reținut faptul că izomorfismul are loc numai pentru operația de adunare. Numerele complexe astfel obținute pot fi reprezentate și ca vectori în planul complex (vezi fig. 9.8,b), deoarece se adună după regula paralelogramului (așa cum rezultă din modul în care sunt definite operațiile cu numere complexe), însă în ceea ce privește produsele, ele nu pot fi considerate ca vectori.

Fazori. Vectorul $X(\omega)$ din figura 9.8,b, care se atașează unei funcții de timp sinusoidale x , ce rezultă din corespondența biunivocă $x=X(\omega)$ prin:

$$x = \sqrt{2} X \sin(\omega t + \varphi) = \text{Im } X(\omega)$$

$$X(\omega) = X e^{j(\omega t + \varphi)},$$

se mai poate scrie și în forma:

$$X(\omega) = X e^{j(\omega t + \varphi)} = X e^{j\varphi} \cdot e^{j\omega t} = \underline{X} e^{j\omega t},$$

în care (fig. 9.8,c):

$$\underline{X} = X e^{j\varphi},$$

se numește fazorul lui x și reprezintă acea parte a lui $X(\omega)$ ce conține numai faza inițială φ , care nu este o funcție de timp. Ca urmare fazorul este un număr complex reprezentabil în planul complex printr-un vector fix, așa ca în fig. 9.8,c, ce are modulul $|\underline{X}|$ egal cu valoarea efectivă a mărimii $x(t)$ reprezentată și unghiul față de axa reală dată de faza inițială a lui x .

Dacă pentru sistemul analizat (de exemplu un circuit electric), pulsația ω este constantă în timp și nu se fac operații de integrare sau/și derivare, atunci mărimile $x \in \{e, i, u\}$ se pot înlocui numai cu fazorul $\underline{X}$ care, în funcție de operațiile în care este implicat se pot scrie formulele:

$$\underline{X} = a + jb \text{ cu } a = \text{Re}|\underline{X}| \text{ și } b = \text{Im}|\underline{X}| \text{ sau } |\underline{X}| = X = \sqrt{a^2 + b^2},$$

$$\underline{X} = X \cos \varphi + jX \sin \varphi,$$

$$\underline{X} = X e^{j\varphi} \text{ sau } X = |\underline{X}| e^{j\varphi}, \text{ cu } \varphi = \arctg \frac{b}{a}.$$

Operații cu numere complexe. Se definesc următoarele operații:

- *adunarea:*

$$(a + jb) + (c + jd) = (a + c) + j(b + d);$$

- *înmulțirea:*

$$(a + jb) \cdot (c + jd) = (ac - bd) + j(ad + bc)$$

sau:

$$X_1 e^{j\varphi_1} \cdot X_2 e^{j\varphi_2} = X_1 X_2 e^{j(\varphi_1 + \varphi_2)};$$

- *împărțirea:*

$$\frac{a + jb}{c + jd} = \frac{(a + jb)(c - jd)}{c^2 + d^2} = \frac{ac + bd}{c^2 + d^2} + j \frac{bc - ad}{c^2 + d^2},$$

$$\frac{X_1 e^{j\varphi_1}}{X_2 e^{j\varphi_2}} = \frac{X_1}{X_2} e^{j(\varphi_1 - \varphi_2)};$$

- *extragerea rădăcinii:*

$$\sqrt[n]{\underline{X}} = \sqrt[n]{X} \left[\cos\left(\frac{\varphi}{n} + k \frac{2\pi}{n}\right) + j \sin\left(\frac{\varphi}{n} + k \frac{2\pi}{n}\right) \right], \quad k=0, 1, 2, \dots, n-1,$$

cele n rădăcinii fiind așezate în vârfurile unui poligon regulat cu n laturi.

Proprietățile operațiilor cu numere complexe și anume:

$$X^m \cdot X^n = X^{m+n}; \quad \frac{X^m}{X^n} = X^{m-n} \quad \text{și} \quad (X^m)^n = X^{m \cdot n}$$

sunt valabile pentru m și n numere raționale.

Operatorul de rotație cu 120° în planul complex. Pentru studiul sistemelor trifazate (a se revedea subcapitolul 8.6) prezintă interes rădăcinile: $a = \sqrt[3]{1}$, adică:

$$\begin{cases} a_1 = 1 \\ a_2 = -\frac{1}{2} - j\frac{\sqrt{3}}{2}, \\ a_3 = -\frac{1}{2} + j\frac{\sqrt{3}}{2} \end{cases}$$

căroră li se dă interpretarea unor operatori de rotație cu 120° , cele trei rădăcini determinând vârfurile unui triunghi echilateral. De exemplu:

$$\begin{aligned} a_1 \underline{X} &= \underline{X} = X \in^{j0} \\ a_2 \underline{X} &= \underline{X} \in^{-j120^\circ}, \end{aligned}$$

este un fazor egal în modul cu $\underline{X}$, dar rotit în sens invers sensului trigonometric cu 120° (sau în sens trigonometric cu -120°);

$$a_3 \underline{X} = \underline{X} \in^{-j240^\circ}$$

este un vector egal în modul cu $\underline{X}$, dar rotit în sens invers sensului trigonometric cu 240° (sau rotit în sens trigonometric cu -240°).

Derivarea și integrarea. Dacă $x = \sqrt{2} X \sin(\omega t + \varphi)$ este originalul și $X(\omega) = X \in^{j(\omega t + \varphi)}$ este imaginea lui x , $x \hat{=} X(\omega)$, corespondența în cazul derivării este următoarea:

$$\frac{d}{dt} \sqrt{2} X \sin(\omega t + \varphi) = \sqrt{2} X \omega \cos(\omega t + \varphi) = \sqrt{2} X \omega \sin(\omega t + \varphi + \frac{\pi}{2}),$$

dar:

$$X \sin(\omega t + \varphi) = \text{Im} [X \in^{j(\omega t + \varphi)}] = \text{Im} X(\omega),$$

iar:

$$\omega X \sin(\omega t + \varphi + \frac{\pi}{2}) = \text{Im} [\omega X \in^{j(\omega t + \varphi)}] = \text{Im} [\omega X \in^{j(\omega t + \varphi)} \in^{\frac{j\pi}{2}}],$$

însă $X \in^{\frac{j\pi}{2}} = jX$; prin urmare derivatei originalului îi corespunde derivata imaginii. Se observă că,

$X(\omega)$ prin derivare, își multiplică modulul cu ω și se rotește cu $\frac{\pi}{2}$ în sens trigonometric.

Deci, derivarea unei reprezentări în planul complex, $X(\omega)$, se efectuează conform regulilor din analiza matematică:

$$\begin{aligned} \frac{d}{dt} X(\omega) &= \frac{d}{dt} X \in^{j(\omega t + \varphi)} = \frac{d}{dt} X \in^{j\varphi} \cdot \in^{j\omega t} = \frac{d}{dt} \underline{X} \in^{j\omega t} = \\ &= j \omega \underline{X} \in^{j\omega t} = j \omega X \in^{j(\omega t + \varphi)} = j \omega X(\omega). \end{aligned}$$

Prin integrare rezultă operația inversă, adică se împarte modulul lui $X(\omega)$ cu ω și se rotește invers sensului trigonometric cu $\frac{\pi}{2}$ (neglijând constanta de integrare). Astfel, integralei originalului $x(t)$ îi corespunde integrala imaginii $X(\omega)$. Într-adevăr:

$$\int_0^t x(t) dt = \int_0^t \sqrt{2} X \sin(\omega t + \varphi) = \frac{\sqrt{2} X}{\omega} \left| -\cos(\omega t + \varphi) \right|_0^t =$$

$$= \sqrt{2} \frac{X}{\omega} \sin(\omega t + \varphi - \frac{\pi}{2})$$

și

$$\int_0^t X(t) dt = \int_0^t X e^{j(\omega t + \varphi)} dt = \int_0^t X e^{j\varphi} \cdot e^{j\omega t} dt = \int_0^t \underline{X} e^{j\omega t} dt =$$

$$= \frac{\underline{X}}{\omega} \left| e^{j\omega t} \right|_0^t = \frac{j\underline{X}}{j^2 \omega} e^{j\omega t} = -j \frac{\underline{X}}{\omega} e^{j\omega t} = -j \frac{1}{\omega} X(\omega).$$

9.1.4. Noțiuni de calcul operațional bazat pe transformata Laplace

Se numește *original* o funcție reală sau complexă de variabilă reală t care satisface următoarele condiții:

- $f(t) = 0$, pentru $t < 0$;
- $f(t)$ și $f'(t)$ sunt continue pe porțiuni, adică admit un număr finit de discontinuități de prima speță;
- $f(t)$ nu crește mai repede decât o exponențială, adică există două numere $M > 0$ și $\alpha \leq 0$ așa încât să existe condiția:

$$(9.49) \quad f(t) < M e^{\alpha t},$$

în care e reprezintă baza logaritmilor naturali (numărul “e”).

Tensiunile electrice la borne și intensitatea curentului electric de conducție din laturile unui circuit, precum și multe alte mărimi fizice din teoria câmpului electromagnetic, satisfac întotdeauna aceste condiții.

Se numește *imagea* funcției $f(t)$, transformata Laplace notată cu operatorul $\mathcal{L}$ și definită prin:

$$F(s) = \int_0^\infty f(t) e^{-st} dt = \mathcal{L}[f(t)], \quad (9.50)$$

în care s este variabila complexă:

$$s = \alpha + j\omega, \quad (9.51)$$

numită *pulsăția (frecvența) complexă*.

Planul definit de α și $j\omega$ se mai numește și *planul complex al frecvențelor*, în care α are semnificația unui factor de amortizare, iar ω semnificația frecvenței unghiulare (pulsăției).

Se demonstrează că între original $f(t)$ și imagine $F(s)$ se stabilește o corespondență biunivocă, adică există întotdeauna și transformata Laplace inversă $\mathcal{L}^{-1}$, adică:

$$f(t) = \mathcal{L}^{-1}[F(s)].$$

Aplicând transformata Laplace unei sume de mai multe funcții original, se vede –din definiția (9.50)– că rezultă suma imaginilor. Pe baza *proprietăților de liniaritate* ale operatorului $\mathcal{L}$ se poate scrie:

$$\mathcal{L}[a f_1(t) + b f_2(t)] = a \mathcal{L}[f_1(t)] + b \mathcal{L}[f_2(t)] = a F_1(s) + b F_2(s),$$

unde a și b sunt constante reale sau complexe.

Derivării originalului îi corespunde înmulțirea cu s a imaginii, cu condiția ca $f(0)=0$, deoarece:

$$\mathcal{L}\left[\frac{d}{dt} f(t)\right] = s F(s) - f(0). \quad (9.52)$$

În general, pentru derivata de ordinul n se poate scrie:

$$\mathcal{L}\left[\frac{d^n}{dt^n}f(t)\right] = s^n F(s) - s^{n-1}f(0) - s^{n-2}f'(0) - \dots - f^{(n-1)}(0),$$

unde $f' = df/dt$ și $f^{(n-1)} = d^{n-1}f/dt^{n-1}$.

Integrării originalului îi corespunde împărțirea cu s a imaginii:

$$(9.53) \quad \mathcal{L}\left[\int_0^t f(t) dt\right] = \frac{1}{s} \mathcal{L}[f(t)] = \frac{1}{s} F(s).$$

Derivarea imaginii este echivalentă cu înmulțirea originalului cu $-t$:

$$\frac{d}{ds} F(s) = \mathcal{L}[-t f(t)].$$

Integrarea imaginii este echivalentă cu împărțirea originalului la t :

$$\int_s^\infty F(s) ds = \mathcal{L}\left[\frac{1}{t} f(t)\right].$$

Problema aflării originalului când este cunoscută imaginea, așa-zisa problemă inversă $\mathcal{L}^{-1}$, se rezolvă –în general– cu ajutorul transformării inverse cu formula Mellin – Fourier :

$$f(t) = \frac{1}{2\pi j} \int_{\alpha-j\infty}^{\alpha+j\infty} F(s) e^{st} ds = \mathcal{L}^{-1}[F(s)],$$

în care a este un număr real, astfel că $a > \alpha$, aici α fiind numărul real din relațiile (9.49) și (9.51).

În cazuri particulare, ca de exemplu atunci când imaginea este o fracție rațională (situație des întâlnită în analiza circuitelor electrice în regim nestaționar), adică atunci când :

$$(9.54) \quad F(s) = \frac{P(s)}{Q(s)},$$

la care gradul numitorului este mai mare decât gradul numărătorului, iar numitorul are numai rădăcini simple $s_0, s_1, s_2, \dots, s_n$ se utilizează *teorema de dezvoltare (Heaviside)*:

$$(9.55) \quad f(t) = \mathcal{L}^{-1}[F(s)] = \sum_{k=0}^n \frac{P(s_k)}{Q'(s_k)} e^{s_k t},$$

în care s_k sunt rădăcinile ecuației $Q(s) = 0$ și $Q'(s_k) = \frac{d}{ds} Q(s) |_{s=s_k}$.

Dacă în fracția (9.54) polinomul de la numitor admite o rădăcină $s_0 = 0$, atunci polinomul $Q(s)$ se scrie în forma $Q(s) = s R(s)$ și teorema de dezvoltare (9.55) capătă forma :

$$(9.56) \quad f(t) = \mathcal{L}^{-1}\left[\frac{P(s)}{s R(s)}\right] = \frac{P(0)}{R(0)} + \sum_{k=1}^n \frac{P(s_k)}{Q'(s_k)} e^{s_k t},$$

unde $R' = dR/ds$ și s_k ($k = 1, 2, \dots, n$) sunt rădăcinile ecuației $R(s) = 0$.

Așa cum s-a văzut (în subcapitolul 8.4), însemnătatea aplicării transformatei Laplace apare atunci când trebuie rezolvate ecuații de tip integro-diferențial (cum este modelul unui dipol R, L, C serie în regim electrocinetic nestaționar oarecare: $Ri + L \frac{di}{dt} + \frac{1}{C} \int i dt = u$) care –prin transformarea Laplace devin ecuații algebrice liniare în s , foarte simplu de rezolvat (astfel, modelul din exemplul ales, devine $(R + sL + \frac{1}{sC}) I(s) = U(s)$, deci o expresie algebrică în s).

Pentru aplicațiile practice curente, trecerea de la original la imaginea Laplace, și invers, se poate face prin subrutine de calcul care conțin fișiere de tip dicționar sau (pentru aplicațiile foarte simple) utilizând tabelul 9.1 (limitat la 13 funcții mai frecvent întâlnite).

Correspondența biunivocă $f(t) \leftrightarrow F(s)$

Originalul $f(t)$	Imaginea $F(s)$	Observații
$1(t)$	$\frac{1}{s}$	Funcția treaptă a lui Heaviside
$\delta(t)$	1	Funcția impuls (Dirac)
T	$\frac{1}{s^2}$	Funcția rampă
A	$\frac{a}{s}$	$A = \text{const.}$, real sau imaginar
$\sin \omega t$	$\frac{\omega}{s^2 + \omega^2}$	Funcția sinusoidală
$\text{sh } \omega t$	$\frac{\omega}{s^2 - \omega^2}$	Funcția hiperbolic
$\cos \omega t$	$\frac{\omega}{s^2 + \omega^2}$	
$\text{ch } \omega t$	$\frac{\omega}{s^2 - \omega^2}$	
$\frac{e^{-\alpha t} - e^{-\beta t}}{\beta - \alpha}$	$\frac{1}{(s + \alpha)(s + \beta)}$	
$t e^{-\alpha t}$	$\frac{1}{(s + \alpha)^2}$	
$e^{-\alpha t} \sin \omega t$	$\frac{\omega}{(s + \alpha)^2 + \omega^2}$	
$e^{-\alpha t} \cos \omega t$	$\frac{s + \alpha}{(s + \alpha)^2 + \omega^2}$	

9.1.5. Noțiuni de bază din teoria distribuțiilor

Spații fundamentale în distribuții . Se consideră niște funcții reale φ de o variabilă reală t , care au următoarele proprietăți :

- $\varphi(t)$ are suport compact ,
- $\varphi(t)$ este indefinit derivabilă.

Se notează spațiul acestor funcții cu $\mathcal{D}$, astfel încât orice funcție cu proprietățile de mai sus aparține lui $\mathcal{D}$: $\varphi \in \mathcal{D}$. Un exemplu de astfel de funcție este :

$$\varphi(t; a, b) = \begin{cases} \exp \frac{ab}{(t-a)(t-b)} & \Rightarrow t \in [a, b] \\ 0 & \Rightarrow t \notin [a, b]. \end{cases}$$

Se observă că suportul acestei funcții este intervalul mărginit și închis $[a, b]$, prin urmare această funcție are suport compact . Faptul că este indefinit derivabilă se verifică fără dificultate ; ordinul de derivare poate fi oricât de mare, dar finit. În plus se mai observă că orice derivată a lui φ are de asemenea suport compact, deci și $\varphi^k \in \mathcal{D}$, unde $k \in \mathbb{N}$.

Spațiul $\mathcal{D}$ al funcțiilor indefinit derivabile și cu suport compact se organizează ca spațiu vectorial, normat și complet, deci este un spațiu Banach¹. Există multe alte spații fundamentale în distribuții, dar pentru aplicațiile din teoria circuitelor electrice este suficient spațiul $\mathcal{D}$.

Distribuții. Sunt considerate în continuare numai distribuțiile definite pe spațiul fundamental $\mathcal{D}$; acestea sunt funcționale² liniare și continue definite pe $\mathcal{D}$ și cu valori numere reale sau complexe. De exemplu, aplicația care fiecărei funcții $\varphi \in \mathcal{D}$ face să îi corespundă numărul $\varphi(0)$ este o funcțională liniară și continuă pe $\mathcal{D}$. Această funcțională se numește *distribuția lui Dirac* și se notează cu $\delta(t)$:

$$(9.57) \quad \varphi(t) \xrightarrow{\delta} \varphi(0).$$

Un alt exemplu de distribuții este clasa distribuțiilor generate de funcții local- integrabile, adică de funcții care sunt absolut integrabile pe orice interval mărginit (a, b) de pe axa reală. Funcția treaptă Heaviside este o funcție local-integrabilă, deoarece este absolut integrabilă pe orice interval (a, b) . Astfel, *distribuția Heaviside* este dată de funcționala:

$$(9.58) \quad \int_R h(t) \varphi(t) dt = \int_0^\infty \varphi(t) dt.$$

Această funcțională are ca valoare întotdeauna un număr finit, deoarece φ are suport compact și –prin urmare– integrala are întotdeauna limite finite, oricare ar fi $\varphi \in \mathcal{D}$. Așadar, orice funcție local-integrabilă $f(t)$ generează o distribuție prin funcționala: $\int_R h(t) \varphi(t) dt$, valoarea acestei

integrale este întotdeauna un număr finit. Aceste distribuții se numesc *distribuții de tip funcție*.

Spre deosebire de distribuția Heaviside, distribuția Dirac, $\delta(t)$, nu este o distribuție de tip funcție. Într-adevăr, să presupunem că ar exista o funcție local integrabilă $x(t)$ care ar genera distribuția lui Dirac; atunci ar fi necesar ca oricare ar fi $\varphi \in \mathcal{D}$ să existe:

$$\int_R x(t) \varphi(t) dt = \varphi(0).$$

În particular fie funcția:

$$\varphi(t) = \begin{cases} \exp(-\frac{1/n^2}{1/n^2 - t^2}) & \Rightarrow |t| < \frac{1}{n}, \forall n \in N \\ 0 & \Rightarrow |t| \geq \frac{1}{n}, \end{cases}$$

pentru care rezultă:

$$\varphi(0) = \frac{1}{e},$$

(unde e este baza logaritmilor naturali) precum și:

$$\int_{-\frac{1}{n}}^{+\frac{1}{n}} x(t) \exp(-\frac{1/n^2}{1/n^2 - t^2}) dt = \frac{1}{e},$$

Rezultă, după cum se vede, o contradicție: membrul stâng tinde către zero când n crește indefinit, iar membrul drept este mereu o constantă. Așadar, distribuția Dirac $\delta(t)$ nu este de tip funcție.

Ca notație pentru distribuții se utilizează scrierea:

¹ Spațiul vectorial E normat și complet (prin topologia normei $\|x\|$) este denumit *spațiu Banach*. Un spațiu normat E este și complet dacă orice șir x fundamental este un șir convergent, adică elementele-limită $x = \lim_{n \rightarrow \infty} x_n$ sunt continuate în spațiul E , pentru orice șir $x_n \in E$ ($n = 1, 2, \dots$).

² În analiza funcțională (și deci și în teoria distribuțiilor) se numește *funcțională* orice funcție al cărei argument este o funcție.

$$\langle f, \varphi \rangle \text{ sau } f(\varphi),$$

ceea ce conduce la:

$$\langle \delta(t), \varphi(t) \rangle = \varphi(0),$$

$$\langle h(t), \varphi(t) \rangle = \int_0^{\infty} \varphi.$$

Când argumentul integralei este cunoscut și nu există ambiguități, integrala se scrie în modul cel mai simplu:

$$\int \varphi(t) dt = \int \varphi.$$

Funcționalele care definesc distribuțiile sunt aplicații; se știe că mulțimea aplicațiilor definite pe $\mathcal{D}$ cu valori în $\mathbb{R}$ este mulțimea “duală”. Notăm această mulțime cu $\mathcal{D}'$ (spațiul dual al lui $\mathcal{D}$); orice distribuție f este un element al lui $\mathcal{D}'$. În continuare, distribuțiile f și g vor fi notate prin $f, g \in \mathcal{D}'$.

Egalitatea a două distribuții. Două distribuții sunt egale pe un interval T de pe axa reală dacă pentru $\forall \varphi \in \mathcal{D}$ pe T :

$$\langle f, \varphi \rangle = \langle g, \varphi \rangle$$

sau, echivalent:

$$\langle f - g, \varphi \rangle = 0.$$

Pentru două distribuții f și g de tip funcție, egalitatea distribuțiilor generate de ele nu înseamnă și egalitatea funcțiilor respective decât în sensul “aproape peste tot”, adică ele pot diferi pe o mulțime neglijabilă. De exemplu, distribuția $h(t)$ a lui Heaviside este dată de:

$$\langle h, \varphi \rangle = \int_0^{\infty} \varphi,$$

iar funcția:

$$\tilde{h}(t) = \begin{cases} 0 & \Rightarrow t < 0 \cap t \in \mathbb{N} \\ 1 & \Rightarrow t > 0 \cap t \notin \mathbb{N} \end{cases}$$

(numită și funcția treaptă unitate), care ia valoarea zero pentru orice t număr natural, generează distribuția:

$$\langle \tilde{h}, \varphi \rangle = \int_0^{\infty} \varphi = \langle h, \varphi \rangle.$$

În acest exemplu, cele două distribuții, $\langle h, \varphi \rangle$ și $\langle \tilde{h}, \varphi \rangle$, sunt egale, dar funcțiile ce le-au generat, h și $\tilde{h}$, nu sunt egale decât în sensul “aproape peste tot”.

Suma și produsul cu o constantă, în distribuții, rezultă imediat, deoarece funcționala este liniară:

$$\langle f, \varphi \rangle + \langle g, \varphi \rangle = \langle f + g, \varphi \rangle$$

și

$$\langle a f, \varphi \rangle = \langle f, a \varphi \rangle = a \langle f, \varphi \rangle \Rightarrow a \in \mathbb{R} = \text{const.}$$

Translația distribuțiilor. Suportul unei distribuții f este complementara mulțimii deschise pe care distribuția f se anulează; așadar, suportul unei distribuții f este o mulțime închisă. De exemplu, distribuția Dirac $\delta(t)$ are ca suport punctul $t = 0$, deoarece δ se anulează pe toată axa reală, cu excepția punctului $t = 0$, iar distribuția Heaviside $h(t)$ are ca suport semiaxa pozitivă $\mathbb{R}_+$, deoarece h se anulează pentru $t < 0$.

Translația unei distribuții cu un interval τ pe axa reală se scrie convențional astfel:

$$f(t) \rightarrow f(t + \tau).$$

Prin definiție, translația distribuției $f(t)$ cu τ este dată de:

$$\langle f(t + \tau), \varphi \rangle \stackrel{D}{=} \langle f(t), \varphi(t - \tau) \rangle.$$

Pentru distribuția de tip funcție, aplicația acestei formule duce imediat la:

$$\langle f(t + \tau), \varphi \rangle = \int_R f(t + \tau) \varphi(t) dt.$$

Notându-se $t + \tau = \sigma$, de unde $t = \sigma - \tau$, rezultă:

$$\int_R f(t + \tau) \varphi(t) dt = \int_R f(\sigma) \varphi(\sigma - \tau) d\sigma,$$

ceea ce verifică formula dată pentru translație.

În particular, translația distribuției Dirac este:

$$\langle \delta(t + \tau), \varphi(t) \rangle = \langle \delta(t), \varphi(t - \tau) \rangle = \varphi(-\tau);$$

de asemenea:

$$\langle \delta(t - \tau), \varphi(t) \rangle = \varphi(\tau),$$

ceea ce arată că suportul distribuției $\delta(t - \tau)$ este punctul τ .

Derivata distribuțiilor. Prin definiție, derivata unei distribuții f se calculează cu formula:

$$\langle Df, \varphi \rangle \stackrel{D}{=} \langle f, -D\varphi \rangle,$$

adică se derivă $\varphi(t)$ și i se schimbă semnul în funcțională. De exemplu, derivata distribuției Heaviside este:

$$\langle Dh, \varphi \rangle = \langle h, -D\varphi \rangle = - \int_0^{\infty} D\varphi = \varphi(0),$$

dar –pe de altă parte–:

$$\varphi(0) = \langle \delta, \varphi \rangle,$$

astfel că:

$$\langle Dh, \varphi \rangle = \langle \delta, \varphi \rangle,$$

adică derivata distribuției Heaviside este distribuția Dirac. Dacă saltul în origine ($t = 0$) este valoarea E , atunci rezultă:

$$\langle DEh, \varphi \rangle = \langle EDh, \varphi \rangle = E \langle Dh, \varphi \rangle = E\delta,$$

Prin urmare derivata, în sensul distribuțiilor, conține și valoarea saltului.

Cu titlu de exemplu, se va calcula –în continuare– derivata (în sensul distribuțiilor) a unei funcții derivabile $f(t)$, cu excepția unui punct t_0 în care funcția f are un salt egal cu σ . Va rezulta:

$$\begin{aligned} \langle f, \varphi \rangle &= \int_{-\infty}^{+\infty} f \varphi, \\ \langle Df, \varphi \rangle &= \langle f, -D\varphi \rangle = - \int_{-\infty}^{t_0-0} f D\varphi - \\ &- \int_{t_0+0}^{+\infty} f D\varphi = -f\varphi \Big|_{-\infty}^{t_0-0} - f\varphi \Big|_{t_0+0}^{+\infty} + \int_{-\infty}^{+\infty} Df \varphi. \end{aligned}$$

După cum se vede, s-a integrat prin părți și s-a ținut seama de faptul că φ este continuă. Deoarece φ are suport compact, la infinit se anulează, iar $\varphi(t_0 + 0) = \varphi(t_0 - 0)$ și atunci :

$$-f\varphi \Big|_{-\infty}^{t_0-0} - f\varphi \Big|_{t_0+0}^{+\infty} = [f(t_0 + 0) - f(t_0 - 0)] \varphi(t_0) = \sigma \varphi(t_0),$$

unde $\varphi(t_0)$ reprezintă însă distribuția Dirac translatată în punctul t_0 , ceea ce se scrie astfel:

$$\langle \delta(t - t_0), \varphi(t) \rangle = \varphi(t_0),$$

asa că se mai poate scrie:

$$\langle Df, \varphi \rangle = \langle \sigma \delta(t - t_0), \varphi(t) \rangle + \langle Df, \varphi \rangle,$$

sau :

$$\langle Df, \varphi \rangle = \langle Df + \sigma \delta_{t_0}, \varphi \rangle,$$

adică, prescurtat (subînțelegând funcționalele):

$$Df = f' + \sigma \delta_{t_0}, \quad (9.59)$$

care se citește astfel: derivata distribuției de tip funcție f este egală cu distribuția generată de derivata “clasică” (f') a funcției f , cu excepția punctului de salt unde apare distribuția Dirac cu suport în acel punct, înmulțită cu valoarea saltului.

Pentru aplicațiile practice (v. §8.8.1), ca o concluzie la cele de până acum, rezultă că funcțiile φ au rolul de “funcții test”. Astfel:

- înmulțirea unei distribuții cu o constantă este echivalentă cu înmulțirea lui φ cu această constantă;

- derivata unei distribuții este echivalentă cu derivarea lui φ și schimbarea semnului;

- translația unei distribuții cu τ este echivalentă cu translația lui φ cu $-\tau$.

Multiplcatori pe $\mathcal{D}$. Se consideră funcțiile ψ indefinit derivabile și cu suport oarecare. În acest caz, produsul $\psi \varphi$, unde $\varphi \in D$, are suport compact și este indefinit derivabil. Așadar se poate defini produsul dintre o distribuție $f \in D'$ și o funcție $\psi \in C^\infty$ după formula:

$$\langle \psi f, \varphi \rangle = \langle f, \psi \varphi \rangle, \quad \forall \varphi \in D.$$

În particular, produsul dintre distribuția Heaviside și o funcție $\psi \in C^\infty$ este definit fiind:

$$\psi h \in D',$$

adică o distribuție cu suport pozitiv. Rezultă că multiplicarea unei distribuții cu o funcție indefinit derivabilă este echivalentă cu multiplicarea lui φ cu această funcție, astfel încât și în acest caz φ are rolul de funcție test.

Distribuții cu suport pozitiv – spațiu $\mathcal{D}_+$. Distribuțiile care au suport mărginit la stânga (în particular cele care au suport pozitiv) formează o clasă importantă de distribuții cu proprietăți remarcabile. Pentru sistemele fizice, cauzalitatea conduce adesea la reprezentări prin distribuții cu suport pozitiv (cazul transformatorilor Laplace a distribuțiilor, din analiza circuitelor electrice). Aceste distribuții sunt elemente ale spațiului notat cu $\mathcal{D}_+$.

Produsul de convoluție. Pentru două distribuții f și g se definește așa-numitul produs de convoluție (notat cu $*$):

$$\langle f * g, \varphi \rangle \stackrel{D}{=} \langle f(t), \langle g(\tau), \varphi(t+\tau) \rangle \rangle,$$

care este o funcțională compusă. Astfel, se calculează mai întâi funcționala:

$$\langle g(\tau), \varphi(t+\tau) \rangle =: \psi(t),$$

după care se determină funcționala:

$$\langle f(t), \psi(t) \rangle, \quad \psi(t) \in \mathcal{D}$$

adică produsul de convoluție, cu condiția însă ca $\psi(t)$ să aparțină lui $\mathcal{D}$, ceea ce se întâmplă numai în anumite condiții; deci produsul de convoluție nu există întotdeauna.

Condițiile de existență a produsului de convoluție sunt:

1) f **sau** g să aibă suport compact:

2) f **și** g să aibă suportul mărginit de aceeași parte.

Prin urmare în $\mathcal{D}_+$ produsul de convoluție există întotdeauna și –în plus– în $\mathcal{D}_+$ el este comutativ și asociativ.

Derivata produsului de convoluție se obține derivând numai unul din factorii produsului, indiferent care; deci:

$$D\langle f * g \rangle = (Df) * g = f * (Dg).$$

Algebra $(\mathcal{D}_+, +, *)$. Această algebră are proprietățile:

1. $(\mathcal{D}_+, +, *)$ este un grup abelian,
2. $(\mathcal{D}_+, *)$ este un monoid cu element unitate.

Pentru orice $x, y, z \in \mathcal{D}_+$ au loc egalitățile:

$$x*(y+z) = x*y + x*z,$$

$$(x+y)*z = x*z + y*z,$$

iar inelul obținut astfel este un domeniu de integritate deoarece:

- i) inelul este comutativ,
- ii) $\delta \neq 0$ (unitatea diferită de zero),
- iii) inelul nu are divizori ai lui zero. Într-adevăr, în $\mathcal{D}_+$ din $f*g = 0$ se deduce sau $f=0$ sau $g=0$, ceea ce permite simplificarea: $x*f = x*g \Rightarrow f=g$ (în sensul distribuțiilor).

Deoarece orice inel întregu admite o scufundare într-un corp, denumit “corpul fracțiilor”, se consideră algebra $(\mathcal{D}_+, +, *)$ ca fiind un corp de fracții în care se introduce și “împărțirea” (în sensul inversului produsului de convoluție).

Ca exemplu, se arată că inversul lui δ' este h și deci ar trebui ca $\delta' * h = \delta$. Într-adevăr:

$$D\delta * h = \delta * Dh = \delta * \delta = \delta.$$

Aceasta este valabil pentru orice derivată de ordin m a lui δ , în produs de convoluție cu primitiva de ordin m a lui h ; așadar:

$$\delta^m * h^m = D^m \delta * h^m = \delta * D^m h^m = \delta * \delta = \delta,$$

fiind de reținut faptul că:

$$h^m = \frac{t^{m-1}}{(m-1)!} h.$$

Transformata Laplace a distribuțiilor (în $\mathcal{D}_+$). Transformata Laplace a unei funcții $f(t)$ – v. (9.50.) – și condițiile de existență ale transformatei – v.(9.49), au fost prezentate în subcapitolul 9.1.4.

Pentru introducerea transformatei Laplace a distribuțiilor este recomandabilă o definiție simplă și care nu reduce din generalitate în aplicații. Astfel, se consideră distribuțiile care reprezintă derivata de un ordin oarecare m a unor funcții care posedă transformata Laplace:

$$F(t) = D^m f(t);$$

Prin definiție, transformata Laplace a distribuției $F(t)$ este:

$$(9.60) \quad (\mathcal{L}F)(s) \stackrel{D}{=} s^m \mathcal{L}f,$$

în care s este numărul complex dat de expresia (9.51).

Correspondența dintre operațiile cu original F și imagine $\mathcal{L}F$, în distribuții, este: derivarea originalului corespunde cu înmulțirea cu s a imaginii, iar primitiva originalului se obține prin împărțirea cu s a imaginii. Astfel, pentru distribuția Heaviside h , pentru care:

$$Dh = \delta,$$

Adică distribuția Dirac δ , conform definiției (9.60) rezultă:

$$(9.61) \quad \mathcal{L}(Dh) = s \mathcal{L}h = s \frac{1}{s} = 1 = \mathcal{L}\delta$$

(a se vedea tabelul 9.1).

Nu trebuie uitat faptul că distribuția $h(t)$ este unică, pentru clasa de echivalență a funcțiilor $h(t)$ egale aproape peste tot. De exemplu funcțiile:

$$h_1(t) = \begin{cases} 0 & \Rightarrow t \leq 0 \\ 1 & \Rightarrow t > 0 \end{cases}; h_2 = \begin{cases} 0 & \Rightarrow t < 0 \\ 1 & \Rightarrow t \geq 0 \end{cases}; h_3 = \begin{cases} 0 & \Rightarrow t < 0 \\ \frac{1}{2} & \Rightarrow t = 0 \\ 1 & \Rightarrow t > 0 \end{cases} \text{ etc.,}$$

au aceeași transformată Laplace a distribuției $h(t)$, adică $\mathcal{L}h = 1/s$, de unde rezultă întotdeauna $\mathcal{L}\delta=1$. De aceea, “improvizațiile” care se fac pentru $\mathcal{L}h$, considerând transformata Laplace

$$\int_{-0}^{\infty} h(t) e^{-st} dt \text{ pentru a rezulta apoi } \mathcal{L}\delta=1, \text{ trebuie evitate – în condițiile din distribuții (9.61).}$$

Teoremele referitoare la transformata Laplace a funcțiilor (v. § 9.1.4) se regăsesc și în distribuții. Dintre acestea se va prezenta în continuare situația teoremei lui Borel (referitoare la produs) și anume:

$$\mathcal{L}(f * g) = \mathcal{L}(f) \cdot (\mathcal{L}g).$$

Pentru distribuțiile $F(t) = D^m f(t)$ și $G(t) = D^n g(t)$, demonstrația teoremei lui Borel în distribuții este imediată. Conform definiției (9.60) rezultă:

$$\mathcal{L}(F * G) = [D^m f(t)] * [D^n g(t)] = s^{m+n} \mathcal{L}(f * g) = s^{m+n} \mathcal{L}(f) \cdot \mathcal{L}(g) = (s^m \mathcal{L}f) \cdot (s^n \mathcal{L}g) = (\mathcal{L}F) \cdot \mathcal{L}(G).$$

Așadar, produsul de convoluție (întotdeauna în $\mathcal{D}'$) devine produsul algebric obișnuit.

Pentru rețelele electrice al căror model este de forma ecuației matriceale:

$$\dot{\mathbf{x}} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u},$$

trecând în distribuții se obține:

$$s \mathbf{D} \mathbf{x} = \mathbf{A} \mathbf{x} + \mathbf{B} \mathbf{u} + \mathbf{x}_0 \delta$$

și aplicând transformata Laplace rezultă:

$$s \mathcal{L} \mathbf{x} = \mathbf{A} \mathcal{L} \mathbf{x} + \mathbf{B} \mathcal{L} \mathbf{u} + \mathbf{x}_0,$$

care se poate scrie și în forma:

$$(s \mathbf{I} - \mathbf{A}) \mathcal{L} \mathbf{x} = \mathbf{B} \mathcal{L} \mathbf{u} + \mathbf{x}_0,$$

sau:

$$\mathcal{L} \mathbf{x} = (s \mathbf{I} - \mathbf{A})^{-1} [\mathbf{B} \mathcal{L} \mathbf{u} + \mathbf{x}_0],$$

unde $\mathbf{I}$ este matricea unitate.

Revenindu-se la original se obține:

$$\mathbf{x} = h \in \mathbf{A} t * (\mathbf{B} \mathbf{u} + \mathbf{x}_0 \delta),$$

de unde rezultă:

$$\mathcal{L}[(s \mathbf{I} - \mathbf{A})^{-1}] h \in \mathbf{A} t, \quad (9.62)$$

formulă utilizată pentru calculul direct al matricei $\in \mathbf{A} t$ (unde $\in$ reprezintă baza logaritmilor naturali – numărul e). În *Dumitrescu I. ș.a., 1983* (v. pag. 126, 127) se prezintă o metodă de calcul numeric al matricei $\in \mathbf{A} t$, prin discretizarea lui t (cu pași t_d), deoarece formula (9.62) nu este recomandabilă pentru calcule numerice.

9.2. Simularea câmpurilor potențiale

Dezvoltarea accentuată și continuă a tehnicilor automate de calcul (atât a sistemelor de calculatoare și a sistemelor de programe, cât și a procedeelelor informatice) au făcut ca azi – în domeniul electrotehnicii (al electromagnetismului în general) – să se poată utiliza, în toate fazele (de instruire/didactică, de cecetare-proiectare/analiză și sinteză a dispozitivelor electromagnetice, de producție/fabricare și de exploatare-întreținere), programe de simulare eficiente, simulare care – în prezent – precede construcția oricărui prototip fizic și a oricărei aplicații în tehnică a fenomenelor electromagnetice.

Proiectarea asistată de calculator CAD (Computer Aided Design) și ingineria asistată de calculator CAE (Computer Aided Engineering) sunt instrumente informatice care au evoluat odată cu dezvoltările tehnologiilor microelectronicii, tehnologiilor de tip program (“software”), dar și a perfecționării *algoritmilor de simulare și al modelării sistemelor*. Astăzi sunt disponibile produse “software” destinate noilor arhitecturi de calculatoare (ca SIMD: “Single-Instruction

Multiple-Data” și MIMD – calculatoare “Multiple-Instruction Multiple-Data”), iar algoritmi sistolici, algoritmi “pipeline”, algoritmi paraleli, algoritmi euristici etc. sunt frecvent utilizați în aplicațiile cele mai eficiente ale calculatoarelor și în domeniul simulării numerice.

Deoarece procedeele de modelare și simulare au devenit suverane în studiul și practica sistemelor fizice (și –cu precădere– al sistemelor electromagnetice), considerăm necesară o introducere (la esență) în practica modelării și simulării și al modelelor de simulare discretă (numerică) a câmpului electromagnetic.

9.2.1. Modelarea și simularea

Cam de prin anul 1960 a început să se contureze o nouă tehnică de studiu a sistemelor – **simularea**, care își are rădăcinile în procedeele de similitudine, de analiză cu ajutorul machetelor și în calculul analogic, utilizate încă de multă vreme la proiectarea construcțiilor hidrotehnice și aerospațiale sau la cercetarea fenomenelor din hidrogazodinamică subterană.

În prezent, simularea este un domeniu de investigare de sine stătător (*Dumitrescu I., 1983*) aflat la congruența matematicii, teoriei sistemelor și informaticii, cu o metodologie proprie (bazată pe identificarea proceselor, modelarea sistemelor și teoria algoritmilor) și cu mijloace specifice (oferite de echipamentele de calcul automat și de aparatura electronică de măsurat, preluare a datelor, rețele de comunicații și valorificare a rezultatelor).

Modelarea și simularea își au originea în studiul sistemelor. **Sistemul**, într-o accepțiune mai cuprinzătoare, poate fi definit ca 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. În această definiție este esențială ideea de scop, sistem fiind numai acea colecție de elemente capabile de interacțiune într-un astfel de mod încât să fie realizat un program dat. Pentru studierea sistemului se pleacă de la **starea** lui la un moment dat, se realizează o “descriere” a stării sistemului în funcție de starea tuturor componentelor și se stabilesc succesiunile de stări trecute ale sistemului, adică “istoricul” sistemului. Metoda de studiu cea mai indicată este aceea bazată pe experimentările efectuate direct pe sistem (dacă ele sunt posibile). Nu orice sistem poate suporta sau admite încercările directe; dacă experimentările de acest fel pot “vătăma” ireversibil sistemul (cazul în care nu este posibilă așa-zisa procedură “cut-and-try”, adică “taie și încearcă”), îi modifică starea reală, sunt de durată și nu dau rezultate în timp util, nu au precizia informațională necesară ș.a.m.d., atunci singura cale mai eficientă de studiu și analiză a sistemului este **simularea**. În prezent –la nivelul anului 2002–, simularea este o fază inițială care se utilizează întotdeauna pentru oricare sistem, chiar dacă acesta admite orice fel de experimentare directă, deoarece simularea interactivă, realizată cu produse informatice specializate (de mare precizie și viteză), permit analiza unui mare număr de variante și stări pe care sistemul real –deși le poate avea– nu este capabil să le îndeplinească efectiv în timpul experimentului. Altfel, ca în cazul unor activități de concepție-proiectare, sistemul nici nu există în mod fizic (!). Pentru realizarea unei simulări concludente trebuie ca în prealabil să se stabilească un model al sistemului, ceea ce se poate obține (eventual !) prin tehnica identificării proceselor din sistemul analizat (prin așa-numita modelare). **Modelarea** se definește ca o procedură a analizei de sistem prin idealizarea matematică a întregului sistem sau a unor părți din sistem. Rezultatul ei este elaborarea modelului sistemului, sub forma unor reprezentări matematice a relațiilor din sistem. Prin urmare –în accepțiunea din teoria sistemelor (accepțiune ce a fost adoptată în cadrul acestui manual)– prin **model** se înțelege numai reprezentările matematice (care pot descrie, în sens semiotic, stările reale ale unui sistem fizic).

În procesul de modelare și simulare, odată modelul stabilit, acesta reprezintă sistemul, orice succesiune de stări a modelului fiind interpretată ca o succesiune de stări a sistemului, așa cum se arată în figura 9.9.

Adesea, același sistem poate fi reprezentat prin mai multe modele diferite; de exemplu procesul electrocinetic, al conducției electrice, dintr-un conductor masiv poate fi descris printr-un model format din ecuații cu derivate parțiale (cazul ecuațiilor lui Helmholtz – a se vedea subcapitolul 7.2) sau uneori –mai bine (atunci când conductorul are forme geometrice neregulate)– printr-un model discret, format din ecuații cu diferențe finite sau ecuații matriceale (v. subcap. 5.6 și subcap. 7.4). Măsura în care modelul este “complet” și “fidel” depinde de problemele la care se caută răspuns, de stadiul cunoașterii sistemului și a legăturilor sale exterioare.

Scopul simulării constă în determinarea stării reale a sistemului analizat prin rezolvarea modelului ce-l reprezintă. Existând –în principiu– două căi distincte de rezolvare a ecuațiilor unui model (prin dispozitive analogice sau prin calcule analitice-numerice), există și două procedee de simulare diferite:

- *simularea analogică* sau *simularea pe sistem* care constă în reprezentarea relațiilor din sistemul studiat printr-un analog sau simil fizic denumit *simulator*;
- *simularea pe model* care constă în efectuarea calculelor de rezolvare analitică a ecuațiilor modelului, după ce acesta a fost adus (dacă este posibil) într-o formă, numită *model de simulare*, care să îndeplinească condiția de solvabilitate.

Simulatorul este și el tot un sistem, care “înlocuiește” în scopul studiului (analizei) sistemul dat; de aceea se obișnuiește (pentru o mai netă distincție) ca sistemul dat inițial (de studiat) să fie numit *sistem-obiect* sau *original*. Dacă simulatorul este de aceeași natură fizică cu originalul, simularea este de tipul unei similitudini, iar dacă sunt de naturi fizice diferite simularea este de tip analogic.

Un caz particular (deși, în prezent, generalizat) al simulării pe model îl constituie **simularea numerică**, în care un sistem de calcul automat – numeric universal (ca, de exemplu, calculatoarele așa-zise personale, de tip IBM –PC) este utilizat la prelucrarea datelor originalului reprezentate într-o formă simbolică (elemente finite, diferențe finite, matrice, calcul variațional etc.), caz asupra căruia vom reveni.

Un alt caz particular al simulării, îl constituie simularea hibridă, care combină avantajele simulării analogice cu acelea ale simulării numerice.

Prin urmare, simularea constă în utilizarea analogiilor fizice (cu ajutorul simulatoarelor) sau de calcul (cu ajutorul modelelor de simulare), ca mijloace de explorare a comportării unui sistem-obiect (original), pe baza faptului că între elementele simulatorului / modelului de simulare și ale originalului poate fi stabilită (există) o corespondență biunivocă).

Izomorfismul. Esența corespondenței biunivoce: $\text{original} \leftrightarrow \text{simulator}$ o constituie izomorfismul, prin care de altfel sunt fundamentate modelarea și simularea. După cum se știe, în teoria grupurilor din topologie izomorfismul se definește precum urmează:

fiind date două grupuri G și H , se numește izomorfism al lui G în H o aplicație f care asociază fiecărui element $a \in G$ un element $f(a) \in H$, astfel că pentru $\forall(a,b) \in G$ să existe:

$$(i) f(a \cdot b) = f(a) \cdot f(b);$$

$$(ii) f(G) = H, \text{ așa-numita proprietate de epimorfism};$$

(iii) $f(a) = f(b) \rightarrow a = b$, proprietatea de monomorfism, unde ab este un element din G asociat perechii (a,b) de elemente din G printr-o operație binară, asociativă, cu element unitate la dreapta și inversibilă la dreapta.

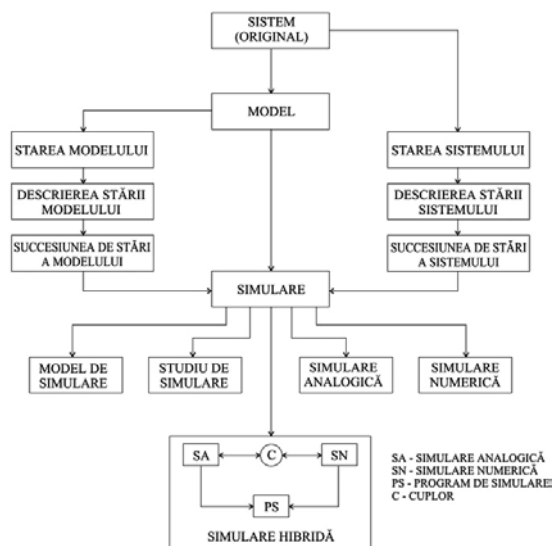


Fig. 9.9

Pe baza acestei definiții și a teoremelor de izomorfism, reiese că asociind unul din grupurile izomorfe sistemului-obiect, celălalt se asociază simulatorului; în acest fel, originalul și simulatorul alcătuiesc, pe planul modelării, două grupuri izomorfe, ceea ce constituie de fapt condiția simulării.

Prin utilizarea noțiunilor din teoria grupurilor se poate stabili un model, bazat pe izomorfism, a simulării sistemelor, precum urmează.

Fie Cs așa-numita “categorie a sistemelor” care constă:

- (j) dintr-o clasă de obiecte, notată $ObCs$ și denumită clasa sistemelor;
- (jj) pentru orice pereche ordonată de obiecte (I, E) este dată o mulțime $Cs(I, E) = \{f, g, \dots\}$ ale cărei elemente $f, g, \dots$ se numesc *morfismele* cu intrarea I și ieșirea E . Pentru un $f \in Cs(I, E)$ se scrie $f: I \rightarrow E$, care se citește “ f este un morfism de intrare (sursă) I și ieșire (adresă) E ”;
- (jjj) pentru două perechi oarecare distincte (I, E) și (I', E') se admite că $Cs(I, E) \cap Cs(I', E') = \emptyset$;

(v) pentru orice tripletă ordonată de obiecte (I, E, X) ale clasei $ObCs$ este definită o aplicație de mulțimi:

$$\theta(I, E, X): Cs(I, E) \times Cs(E, X) \rightarrow Cs(I, X),$$

numită compunerea morfismelor, care se mai poate scrie și în forma:

$$\theta(I, E, X) \circ (f, g) = g \circ f, \quad \forall f \in Cs(I, E) \text{ și } \forall g \in Cs(E, X);$$

compunerea morfismelor este asociativă, adică:

$$h \circ (g \circ f) = (h \circ g) \circ f;$$

(vj) pentru $\forall I \in ObCs$; mulțimea $Cs(I, I)$ conține cel puțin un element notat 1_I care este denumit *morfismul identic* al lui I .

Atunci, noțiunea generică de **sistem** (original) se definește printr-o categorie care constă din două obiecte (I, E) și din mulțimea morfismelor $Cs(I, E)$.

Cazuri particulare de sisteme sunt:

- *sistemul segment* care este o categorie formată din două obiecte (I, E) și trei morfisme

$$\{1_I, 1_E, f \mid f: I \rightarrow E\};$$

- *sistemul cuplu* care este format din două obiecte (I, E) și patru morfisme

$$\{1_I, 1_E, f, g \mid f, g: I \leftrightarrow E\}; \quad ?$$

- *sistemul reprezentat printr-o categorie* ale cărei obiecte sunt triplete ordonate (I, f, E) , adică categoria morfismelor lui Cs , notată MCs . Dacă $(I, f, E), (I', f', E') \in ObMCs$, atunci $MCs[(I, f, E), (I', f', E')]$ este mulțimea tuturor perechilor de morfisme $\{(\alpha; \beta) \mid \alpha: I \rightarrow I', \beta: E \rightarrow E', f' \circ \alpha = \beta \circ f\}$.

Simularea este, în esență, o activitate de investigare strict experimentală. În numeroase cazuri, studiul experimental al comportării sistemului-obiect, direct pe original, nu este posibil. În legătură cu aceasta, să presupunem că sistemul S format din tripleta de obiecte (I, f, E) nu admite observări experimentale, în timp ce un alt sistem S' format din obiectele (I', f', E') permite un studiu experimental direct.

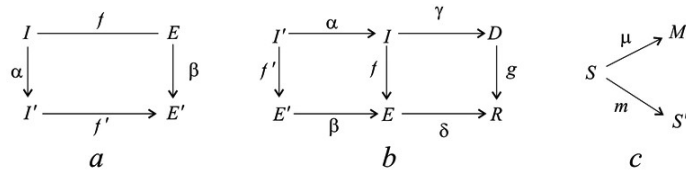


Fig. 9.10

morfismelor în categoria morfismelor lui Cs . Dacă β este un izomorfism inversibil, atunci $f = \beta^{-1} \circ f' \circ \alpha$; dacă numai α este inversibil, atunci $f' = \beta \circ f \circ \alpha^{-1}$, iar dacă α și β sunt simultan izomorfisme inversabile, atunci asocierea $f \leadsto \beta \circ f \circ \alpha^{-1}$ este o bijecție canonică de la

Avem de-a face în acest caz cu *simularea pe sistem (simulator)*, care poate fi concepută ca studiul experimentat al morfismelor f prin intermediul morfismului f' , conform diagramei comutative din figura 9.10a, adică studiul compunerii

mulțimea $Cs(I', E')$. În acest fel, diagrama comutativă din figura 9.10a nu este altceva decât o reprezentare grafică a modelului simulării pe sistem (simulator).

În ceea ce privește *simularea pe model*, să notăm cu $[S]$ clasa sistemelor izomorfe și cu S_I un “reprezentant” al acestei clase. Clasa tuturor sistemelor lui Cs izomorfe cu S se numește tipul lui S . Relația dintre două sisteme izomorfe este o relație de echivalență deoarece dacă $f: S \rightarrow S'$ este un izomorfism și $f': S' \rightarrow S''$ este tot un izomorfism, atunci și $f' \circ f: S \rightarrow S''$ este, de asemenea, un izomorfism.

Pentru sistemele segemnt izomorfe, S și S' sunt și ele categorii; deci a considera că S și S' sunt izomorfe înseamnă a admite un izomorfism functorial $F: S \rightarrow S'$, ceea ce revine la faptul că F este un functor bijectiv și deplin fidel. Aceasta conduce la concluzia că α și β sunt izomorfisme.

Pe baza acestor considerente, se poate afirma că modelul M al unui reprezentat S , al clasei sistemelor izomorfe $[S]$ este și un model al tipului S . Într-adevăr, dacă $\mu: S_1 \rightarrow M$, atunci $g: S_1 \rightarrow S_2$ și rezultă $\mu \circ g^{-1}: S_2 \rightarrow M$. În particular, dacă M este de forma $g: D \rightarrow R$, unde D reprezintă datele inițiale și R rezultatele finale, se poate construi diagrama din figura 9.10b din care rezultă:

$$\gamma \circ \alpha^{-1}: I' \rightarrow D \text{ și } \delta \circ \beta^{-1}: E' \rightarrow R.$$

Deci simularea pe model are ca obiect studiul experimental al morfismelor tipului unei clase de sisteme. Diagrama din figura 9.10b reprezintă atunci un grafic al modelului simulării în cazul mai general când S și S' sunt de tipuri diferite, iar M este model al tipului S .

În concluzie:

- dacă δ este un izomorfism inversibil, atunci simularea pe model constă în studiul morfismului $f = \delta^{-1} \circ g \circ \gamma$;
- dacă izomorfismul β este ireversibil, atunci simularea pe sistem (simulator) constă în studiul morfismului $f = \beta^{-1} \circ f' \circ \alpha$.

Din cele de mai sus rezultă posibilitatea construirii unei categorii a modelelor cu $M \in ObCm$, precum și existența unui functor $F: Cs \rightarrow Cm$. Prin alegerea unui reprezentant S_A al fiecărui tip $[S]$, se obține un schelet Cs' al categoriei Cs . Ca urmare, rezultă și un functor deplin fidel $F': Cs' \rightarrow Cm$, în virtutea faptului că pentru $\forall (S_A, S_B) \in ObCs'$ aplicația $F'(S_A, S_B)$ este o bijecție. Aspectul sub care ar putea fi studiate categoriile Cs și Cm , din punctul de vedere al simulării, este următorul: morfismul $\mu: S \rightarrow M$ se studiază pornind de la un model inițial M_I , știindu-se că $\mu_1: S \rightarrow M_I$ este retractabil. Se construiește șirul $M_I, M_2, M_3, \dots$ și se admite că simularea este un proces convergent dacă limita către care tinde acest șir este și ea un izomorfism $\mu_i: S \rightarrow M_i (i=1, 2, 3, \dots)$. Mai general, convergența simulării poate fi exprimată prin unghiul din figura 9.10c, în care morfismele m și μ devin la limită izomorfisme.

Simulatoare numerice. Calculatoarele numerice au devenit –și vor rămâne– cele mai performante simulatoare ale sistemelor fizice (obiect), oricât de complicate ar fi acestea. Pentru a fi un simulator numeric, un sistem de calcul automat (un calculator – ca parte “hardware” și un sistem de programe de bază – ca parte “software”) trebuie integrate într-o tehnică specializată activității de simulare. Așa sunt produsele informatice CAD, ale căror performanțe sunt determinate atât de factorii “software” cât și de factorii “hardware” (mai ales tipul arhitecturii sistemului gazdă).

În ceea ce privește arhitectura calculatoarelor pe care pot fi instalate produsele CAD de simulare, aceasta poate fi oricare din cele patru clase de calculatoare: de la cea mai simplă, așa-zisa SISD (“Single-Instruction Single-Data”), care sunt cunoscutele calculatoare personale de tip IBM-PC, la calculatoarele SIMD (“Single –Instruction Multiple – Data”), sau – mai noile clase MISD și MIMD (care sunt calculatoare de tip paralel), precum și la calculatoarele inter-conectate (rețele de calculatoare), calculatoare multiple și sisteme de calcul distribuite.

Produsele CAD de simulare a problemelor de câmp (printre care și câmpul electromagnetic) sunt programe de tip utilizator sau programe de firmă destinate noilor arhitecturi de calcul, care –toate– se bazează pe modele de simulare de tip numeric, așa cum sunt metodele de calcul variațional, metoda rețelilor (cu diferențe finite) și metoda elementului finit, pe care le vom prezenta în următoarele subcapitole.

9.2.2 Metode variaționale

Majoritatea fenomenelor de câmp electromagnetic au ca model original o ecuație sau un sistem de ecuații cu derivate parțiale (v. subcap. 1.4) de forma:

$$Lu=f \quad (9.63)$$

unde u este o marime de stare a câmpului electromagnetic, f este o funcție cunoscută iar L este un operator diferențial (liniar sau nelinier). Acest model se scrie pe baza datelor concrete ale sistemului fizic analizat și care se referă la: domeniul $\bar{\Omega}=\Omega \cup \Sigma$ cu $\Sigma = \text{Fr } \Omega$, natura materialului corpurilor din Ω (incluzând și eventualele suprafețe de discontinuitate $\Sigma_d \subset \Omega$), regimul în timp (static, staționar, cvasistaționar, nestaționar etc.) și cinetic (referitor la vitezele corpurilor din Ω), “sursele” și “puțurile” de câmp, condițiile pe frontiere (Σ și Σ_d)-la limită și condițiile inițiale (în timp), astfel încât –în termenii matematici– modelul (9.63) devine o problemă diferențială sau cu derivate parțiale cu condiții la limită și inițiale de tip:

- problema Dirichlet interioară relativă la Ω :

$$(9.63D \text{ i}) \quad \begin{cases} Lu = f & \text{în } \Omega \\ u_{\Sigma} = c \end{cases},$$

unde f și c sunt funcții reale, definite, $f \in C^0(\bar{\Omega})$ și $c \in C^0(\Sigma)$;

- problema Dirichlet exterioară relativă la Ω :

$$(9.63D \text{ e}) \quad \begin{cases} Lu = f & \text{în } E^n - \Omega \\ u_{\Sigma} = c \end{cases},$$

unde funcția u este continuă în $(E^n - \Omega) \cup \Sigma$, de clasă C^2 în $E^n - \Omega$, $\lim_{P \rightarrow \infty} u(P) \rightarrow 0$ uniform în orice punct P , iar $c \in C^0(\Sigma)$;

- problema lui Neumann interioară relativă la domeniul Ω :

$$(9.63N \text{ i}) \quad \begin{cases} Lu = f & \text{în } \Omega \\ D_n u_{\Sigma} = c \end{cases},$$

unde $D_n u_{\Sigma}$ reprezintă derivata parțială a funcției u după normala la Σ , $c \in C^0(\Sigma)$, funcția u fiind de clasă C^1 în Ω și de clasă C^2 în Ω :

- problema lui Neumann exterioară relativă la Ω :

$$(9.63N \text{ e}) \quad \begin{cases} Lu = f & \text{în } E^n - \Omega \\ D_n u_{\Sigma} = c \end{cases},$$

unde derivata $D_n u_{\Sigma}$ se calculează după normala exterioară la Σ , iar $c \in C^0(\Sigma)$, funcția u fiind de clasă C^1 în $(E^n - \Omega) \cup \Sigma$, de clasă C^2 în $E^n - \Omega$ și satisfăcând condiția $\lim_{P \rightarrow \infty} u(P) \rightarrow 0$ uniform;

- problema mixtă (Fourier) interioară relativă la Ω :

$$(9.63F \text{ i}) \quad \begin{cases} Lu = f & \text{în } \Omega \\ u + gD_n u_{\Sigma} = c \end{cases},$$

unde f , c și g sunt funcții reale, definite, $f \in C^0(\Omega)$ și $c, g \in C^0(\Sigma)$;

- problema mixtă (Fourier) exterioară relativă lui Ω :

$$\begin{cases} Lu = f & \text{în } E^n - \Omega \\ u + gD_n u_\Sigma = c \end{cases}, \quad (9.63 \text{ F e})$$

unde $c, g \in C^0(\Sigma)$, derivata $D_n u$ se calculează după normala exterioară la Σ , $u \in C^1[(E^n - \Omega) \cup \Sigma]$ și $u \in C^2(E^n - \Omega)$, cu satisfacerea condiției $\lim_{P \rightarrow \infty} u(P) \rightarrow 0$ uniform.

Pentru rezolvarea unei probleme de câmp, având modelul original de forma (9.63), prin tehnici de simulare trebuie să se determine modelul de simulare sau/și simulatorul corespunzător problemei original dată de (9.63). În cazul simulării numerice, modelul de simulare este un model discret care aproximează prin puncte problemele analitice cu funcții continue (9.63), prin rezolvarea lui cu un sistem de calcul automat obținându-se valori numerice $u(M)$ ale funcției de stare u în diversele puncte M ale modelului discret.

Un model numeric trebuie să satisfacă cel puțin următoarele două condiții:

- să asigure convergența soluției aproximative către soluția exactă a problemei, utilizând un număr cât mai mic de puncte (noduri) M , în care se determină valorile funcției necunoscute: $u(M) \rightarrow u(P)$, $M \in \Omega_h$ și $P \in \Omega$, unde Ω_h este un domeniu discret (o rețea de noduri) prin care se înlocuiește domeniul compact Ω ;

- să fie adaptabil lucrului pe calculatoare numerice.

În modelarea și simularea numerică a sistemelor de ecuații cu derivate parțiale de tipul (9.63), modelele numerice de aproximare a soluției se pot împărți în două mari clase:

- una este aceea în care se caută aproximarea operatorilor diferențiali de domeniu și de frontieră prin operatori mai simpli și atunci se caută soluția care să satisfacă acești operatori de aproximare. Din această clasă fac parte metodele cu diferențe finite (v. § 9.2.3), prin care operatorii diferențiali L se aproximează prin operatori diferență finită L_h (h simbolizând, generic, pasul h de discretizare a lui $\Omega \subset E^n$, pe cele n direcții de referință);

- o a doua clasă de metode păstrează forma operatorilor diferențiali și aproximează câmpul necunoscut u prin u^a și se caută algoritmul care să dea cea mai bună aproximare posibilă. În această clasă se află metodele variaționale, iar din metodele variaționale, metoda elementului finit (v. § 9.2.4) reprezintă una din metodele cu cea mai largă utilizare.

Metodele variaționale de rezolvare a ecuațiilor cu derivate parțiale, de tip (9.63), implică găsirea unei funcționale al cărei extrem este echivalent cu rezolvarea ecuației date.

Teoria modelării numerice prin calcul variațional (numit în trecut și calculul varia-țiunilor) are două lucrări de referință: R. GLOWINSKI, J. L. LIONS, R. TREMOLIERES, *Application des methodes d' optimisation, de differences et d' élément finis aux inéquations variationnelles*, Dunod, Paris, 1975 și R. GLOWINSKI, J. L. LIONS, R. TREMOLIERES, *Analyse numérique des inéquations variationnelles*, Dunod, Paris, 1976. Profesorii Lions și Glowinski, de la universitatea "Paris VI" (de matematică) și colaboratorii lor de la IRIA LABORIA (L'Institut de Recherche d'Informatique et d'Automatique) din Rocquencourt – Franța au format cea mai prestigioasă școală de simulare numerică practică a proceselor de câmp, în special de câmp electromagnetic și termic.

Formularea variațională oferă anumite avantaje față de modelul diferențial dintre care cele mai importante sunt:

- funcționala (al cărei extrem, maxim sau minim, se determină) poate avea o semnificație fizică importantă pentru însăși soluția problemei concrete date. De exemplu, în multe aplicații funcționala reprezintă energia sistemului iar rezolvarea unei probleme de câmp electric sau magnetic prin minimizarea energiei, înseamnă un volum (o masă) a dispozitivului calculat (dielectric sau miez magnetic) mai mică;

- integrandul conținând derivate de ordin mai mic decât operatorul diferențial, permite obținerea soluției problemei într-o clasă mai largă de funcții;

- pot exista formulări variaționale reciproce pentru aceeași problemă, fiecare funcțională având semnificații diferite;

- permite demonstrarea existenței problemei de câmp;
- permite tratarea unor condiții pe frontieră oricât de complicate din punctul de vedere natural.

Nu dispunem de volumul necesar pentru a putea trata echivalența modelelor variaționale cu modelele diferențiale corespunzătoare, care –de fapt– în fiecare caz concret trebuie precizată conform situației sistemului original. Totuși –în principiu– dacă sistemul dat este reprezentat de un model de forma (9.63), în care L este un operator pozitiv, adică:

$$\int_{\Omega} u Lu \, d\Omega \geq 0,$$

atunci soluția problemei diferențiale cu condiții pe frontieră omogene, de forma (9.63 D I), este unică și minimizează funcționala patratică:

$$\hat{I}(u) = \int_{\Omega} u Lu \, d\Omega - 2 \int_{\Omega} u f \, d\Omega,$$

unde $f = Lu$ din (9.63).

Cazul condițiilor pe frontieră neomogene se poate reduce la cazul cu condiții omogene printr-o transformare a funcționalei $\hat{I}(u)$, încât să includă operatori diferențiali care acționează pe frontiera domeniului spațial. De exemplu, în cazul unei probleme mixte (9.63 Fi), în E^3 cu (9.63) o ecuație Poisson, într-un sistem de referință cartezian se scrie:

$$\begin{cases} \Delta u = f(x, y, z) \text{ în } \forall (x, y, z) \in \Omega \subset E^3 \\ u_{\Sigma} = g_1(x, y, z) \\ \frac{\partial u}{\partial n} + au_{\Sigma} = g_2(x, y, z), \end{cases}$$

unde f, g_1, g_2 , și a sunt funcții cunoscute, n este direcția normalei la $\Sigma = \text{Fr}\Omega$ în punctul $(x, y, z) \in \Omega \subset E^3$ iar Δ este operatorul laplacean, funcționala echivalentă în calculul variațional fiind:

$$\hat{J}(u) = \int_{\Omega} (\nabla u)^2 \, d\Omega - 2 \int_{\Omega} f u \, d\Omega + \int_{\Sigma} (au^2 - 2g_2 u) \, d\Sigma,$$

(v. paragrafele: 2.6.3, 4.1.2 și 5.6.1).

O cale mai generală pentru găsirea modelului variațional în cazul unui sistem diferențial cu condiții pe frontieră specificate, de formele (9.63), este folosirea diferențialei Frecht. În cazul ecuațiilor neliniare descrise de operatori care nu sunt autoadjuncți, s-au căutat formulări variaționale echivalente, care însă nu posedă avantajele variaționale clasice. De exemplu, se poate întâmpla ca funcționala echivalentă ecuației date să nu fie staționară în punctele de extrem.

9.2.3. Modelele de simulare cu diferențe finite

Metoda diferențelor finite, cunoscută și sub numele de **metoda rețelelor**, este o procedură de aproximare a soluției problemelor cu condiții la limită de forma (9.63xx), arătată în subcapitolul precedent, prin modele numerice (în esență ecuații cu matrice) ușor de rezolvat prin tehnicile de calcul automat (cum ar fi un calculator de tip IBM – PC ce dispune, de exemplu, în partea sa de “software”, de produsul MATLAB – v. § 9.3.1.).

Rezolvarea problemelor cu condiții la limită prin metoda rețelelor constă, în esență, în înlocuirea (aproximarea) ecuațiilor cu derivate parțiale –de forma (9.63), adică $Lu(P) = f(P)$, $\forall P \in \Omega$ – prin ecuații cu diferențe finite – de forma $L_h u(M) = f(M)$, $\forall M \in R \subset \Omega$, unde M sunt nodurile unei rețele R cu pasul h – soluția problemei obținându-se sub forma unui tabel cu date numerice $u(M)$, care reprezintă valorile determinate de funcția u în nodurile M ale unei rețele (regulată sau neregulată, cu pas h egal sau neegal) aplicată domeniului de definiție $\bar{\Omega}$. Aceste valori numerice se determină prin rezolvarea unui sistem de ecuații algebrice liniare, care se obțin prin “discretizarea” ecuațiilor cu derivate parțiale în punctele rețelei alese. Precizia de rezolvare –

ordinul de aproximare notat cu $O(h)$ – este determinată de: forma și pasul rețelei, tipul expresiei de discretizare a derivatelor parțiale (operatorul discret L_h), modul de aproximare a condițiilor pe frontieră și tehnicile de calcul (procedee și echipamente) utilizate.

Metoda rețelelor de discretizare a ecuațiilor cu derivate parțiale. Pentru aplicarea metodei rețelelor, se aleg în domeniul Ω –de definiție a funcției u – mărginit de frontiera Σ , un sistem de puncte M , denumite *noduri*, situate discret în $\overline{\Omega} = \Omega \cup \Sigma$, care formează o mulțime R (astfel că $M \in R$), denumită *rețea de discretizare* (întotdeauna $R \subset \overline{\Omega}$). Practic nodurile M ale rețelei R se obțin ca puncte de intersecție ale unui sistem de drepte ce se translatează cu pașii h_x, h_y, h_z după direcțiile paralele cu axele unui sistem de reprezentare a punctelor $P \in \overline{\Omega} \subset E^3$. Pentru a se da o imagine sugestivă a acestui mod de alegere a nodurilor $M \in R$, se reprezintă în figura 9.11 cazul unui domeniu plan ($\Omega \subset E^2$) și a unui sistem cartezian ortogonal xOy de reprezentare a punctelor.

Alegându-se, după cele două direcții Ox și Oy , pașii h_x și respectiv h_y , care au valori “mici” (însă sunt exprimate de numere strict pozitive) și ducându-se două familii de drepte (unele paralele cu axa Ox și distanțate între ele cu h_x , iar altele paralele cu axa Oy și distanțate între ele cu pasul h_y), se obțin, la intersecția acestor drepte, o mulțime de puncte; acele dintre aceste puncte care sunt cuprinse în $\overline{\Omega} = \Omega \cup \Sigma$ și care se notează cu M , denumindu-se și noduri, formează rețeaua de discretizare R .

Deoarece fiecare nod al rețelei poate fi reprezentat prin coordonatele sale față de sistemul de referință ales (de exemplu, în cazul din figura 9.11 fiecare nod M are coordonatele $(m_x h_x, m_y h_y)$, unde m_x și m_y sunt numere întregi și pozitive care indică numărul de pași h_x și respectiv h_y , existente de la nodul M până la axele Ox și Oy , atunci rețeaua de discretizare R poate fi exprimată (în cazul tridimensional) prin mulțimea de noduri:

$$R = \{M \mid M=(m_x h_x, m_y h_y, m_z h_z); m_x, m_y, m_z \in N=\{1,2,\dots,n\}; h_x, h_y, h_z > 0\}.$$

Rețelele pot fi, după modul cum sunt aranjate nodurile în funcție de sistemul de coordonate ales, cu *pași egali* sau cu *pași neegali* (cazul precedent al lui R). La rețele cu pași egali: $h_x=h_y=h_z=h$ și ele pot fi exprimate prin mulțimea:

$$R = \{M \mid M=(m_x h, m_y h, m_z h); m_x, m_y, m_z \in N; h > 0\},$$

unde N este mulțimea numerelor naturale.

Rețelele cu pași neegali, care au “ochiurile” de formă dreptunghiulară, sunt utilizate în special în cazul domeniilor Ω a căror frontieră Σ este “foarte” neregulată (de exemplu în formă de “deget”), pentru a se putea “acoperi” mai bine porțiunile domeniului Ω din apropierea frontierei. Există, așa cum se va arăta mai încolo, și rețele cu ochiuri triunghiulare sau cu ochiuri hexagonale, precum și rețele în coordonate cilindrice, în coordonate polare, în coordonate sferice (în general, în coordonate curbilinii).

Pentru discretizarea derivatelor parțiale pe aceste rețele R se aleg (aproape în exclusivitatea cazurilor) numai noduri vecine nodului în care se face discretizarea. În legătură cu acest fapt, se introduce o nouă noțiune și anume aceea de **structură caracteristică a rețelei**, care se definește în modul următor: dacă M_i este un nod “curent”, adică un nod care “mătură” întreaga rețea R (deci $i=1,2,\dots,n$, unde n este numărul nodurilor $M \in R \subset \overline{\Omega}$), atunci se numesc *noduri vecine* mulțimea de noduri $R_q \subset R$ definite prin:

$$R_q = \{M \mid M=[(m_x+k)h_x, (m_y+k)h_y, (m_z+k)h_z]; m_x, m_y, m_z \in N; k=-1,0,1; h_x, h_y, h_z > 0\},$$

q fiind numărul de noduri vecine lui M_i care este *nodul central* (în spațiul E^1 , $q=2$; în spațiul E^2 , $q=4$ sau 8 , iar în E^3 , $q=6$ sau 18 sau 26 noduri vecine).

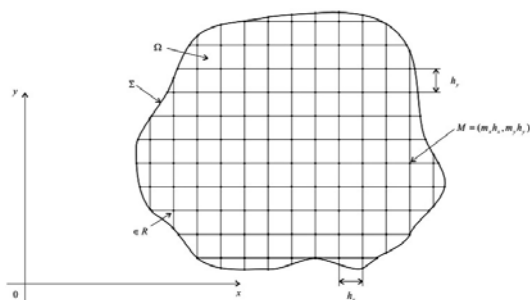


Fig. 9.11

Această mulțime $R_q \subset R$, de $q+1$ noduri vecine între ele pe R (incluzând și nodul central) este denumită structură caracteristică. Deci, o structură caracteristică într-o rețea cu ochiuri pătrate sau dreptunghiulare poate avea 3, 5, 9, 7, 19 sau 27 noduri. Într-o rețea triunghiulară R_q are 7 noduri (în E^2), iar într-o rețea hexagonală 4 noduri (în E^2).

Odată aleasă rețeaua R și structura caracteristică R_q , discretizarea ecuațiilor cu derivate parțiale se face aproximându-se operatorul L al derivatelor parțiale (adică $D_x, D_y, D_z, D_x^2, D_y^2, D_z^2, D_{xy}^2, D_{yz}^2, D_{zx}^2, D_x^4, D_y^4, D_z^4, D_{xy}^4, D_{yz}^4, D_{zx}^4, \dots$) prin *operatorii diferență finită* L_h .

Operatorii – diferență. În practica prelucrării numerice a datelor $u(M) = u_i$, pe rețele de discretizare ($M \in R, i \in N$) se utilizează următorii operatori – diferență (definiți în continuare după una din oricare axă de referință pentru M în E^3):

- diferența la dreapta Δ sau diferența înainte:

$$\Delta u_i = u_{i+1} - u_i \quad (i = 1, 2, \dots, n);$$

- diferența la stânga ∇ (diferența înapoi):

$$\nabla u_i = u_i - u_{i-1} \quad (i = 2, 3, \dots, n);$$

- translația E :

$$Eu_i = u_{i+1};$$

- diferența simetrică δ :

$$\delta u_i = u_{i+1/2} - u_{i-1/2};$$

- media μ :

$$\mu u_i = \frac{1}{2} (u_{i+1/2} + u_{i-1/2}).$$

Toți operatorii diferență definiți anterior sunt operatori liniari (asociativi, distributivi și comutativi) și admit inversul său (de exemplu: $\Delta^{-1} u_i = u_{i-1} - u_i$, și $\Delta(\Delta^{-1} u_i) = \Delta u_{i-1} - \Delta u_i = u_{i-1+1} - u_{i-1} - u_{i-1} + u_i = u_i - u_{i-1} + u_i = 1 u_i$, deci $\Delta \cdot \Delta^{-1} = 1$).

Repetarea unui operator se poate reprezenta prin puteri (exponenți reali, pozitivi sau negativi). De exemplu: $\Delta(\Delta u_i) = \Delta^2 u_i$ și are efectul: $\Delta^2 u_i = \Delta(u_{i+1} - u_i) = \Delta u_{i+1} - \Delta u_i = u_{i+2} - u_{i+1} - u_{i+1} + u_i = u_{i+2} - 2u_{i+1} + u_i$ etc. Diferențele de ordin superior, calculate succesiv, duc la așa-numitele tabele cu diferență utilizate la interpolări.

În practică –pe baza proprietăților acestor operatori-diferență– relațiile cu operatorii-diferență pot fi scrise într-o formă prescurtată, prin renunțarea la scrierea variabilelor u ; așa de exemplu: $\Delta \cdot (\Delta u_i) = \Delta^2 u_i$ se poate scrie direct $\Delta \cdot \Delta = \Delta^2$. Ca urmare, utilizându-se această scriere abreviată, se poate lucra cu acești operatori aplicându-le direct regulile algebrei elementare. În acest fel, rezultă că între operatorii-diferență există următoarele relații:

$$\begin{aligned} \Delta &= E - 1 = E\Delta, \\ \nabla &= 1 - E^{-1}, \\ \delta &= E^{1/2} - E^{-1/2}, \\ \delta^2 &= \Delta - \nabla, \\ \mu &= \frac{1}{2} (E^{1/2} + E^{-1/2}), \\ \mu^2 &= \frac{1}{4} \delta^2 \text{ etc.} \end{aligned} \quad (9.64)$$

Operatori–diferență finită. Sunt acei operatori, care utilizând operatorii–diferență (definiți la punctul precedent) aplicați funcției u în nodurile rețelei de discretizare R , aproximează operatorii derivate parțiale L prin operatori L_h cu diferențe finite între valorile $u(M)$, $M \in R_q \subset R$. În acest fel, funcția scalară de punct $u: P \in \bar{\Omega} \rightarrow u(P)$ se aproximează prin valorile discrete $U(M)$ din nodurile $M \in R \subset \bar{\Omega}$ ale rețelei de discretizare. Astfel, pentru fiecare nod $M \in R \subset \bar{\Omega}$ rezultă –prin discretizare ecuației cu derivate parțiale– o ecuație algebrică cu cel puțin $q+1$ termeni $U(M)$, $M \in R_q$ și realizându-se discretizarea ecuației $Lu = f$, cu derivate parțiale, în toate cele n noduri ale rețelei $R \subset \bar{\Omega}$ se obține un sistem de n ecuații algebrice cu n necunoscute $U(M)$. Prin rezolvarea

acestui sistem –de exemplu prin utilizarea produsului MATLAB (v. § 9.3.1), foarte ușor de aplicat, mai ales că matricea necunoscutelor $U(M)$ este o matrice rară, în formă de bandă– se obțin cele n valori $U(M)$ care aproximează funcția u din ecuația cu derivate parțiale.

Operatorii-diferențe finite L_h se obțin pentru fiecare tip de operator diferențial L , în funcție de forma rețelei de discretizare R și de spațiul lui $\bar{\Omega}$ (practic E^2 sau E^3). De exemplu pentru $u \in \Omega \subset E^1$ (pe o dreaptă, să zicem Ox) operatorul L este $Lu = \frac{du}{dx} = Du$ și rețeaua de noduri $R \subset \Omega$ este un șir de noduri $i = 0, 1, \dots, n$, adică $n+1$ noduri, în care funcția u ia valorile u_i în nodurile x_i ($i = 0, 1, \dots, n$) situate la distanțe, să zicem egale, cu pasul $h = x_{i+1} - x_i$ ($i = 0, 1, \dots, n$). Atunci:

$$Lu \triangleq \frac{du_i}{dx_i} = Du_i,$$

unde $D \equiv L$ este, aici, operatorul de derivare de ordinul 1, care este un operator liniar.

După cum se știe din analiza numerică, dacă valorile lui x_i sunt situate la intervale egale h , o valoare u_{i+1} din alt punct al șirului de noduri, poate fi exprimat în funcție de valoarea dintr-un punct imediat vecin u_i (prin dezvoltarea în serie Taylor):

$$u_{i+1} = u_i + hDu_i + \frac{h^2}{2!} D^2 u_i + \frac{h^3}{3!} D^3 u_i + \dots$$

Utilizând operatorul translație E , relația precedentă se poate scrie sub forma :

$$Eu_i = (1 + hD + \frac{h^2}{2!} D^2 + \frac{h^3}{3!} D^3 + \dots) u_i,$$

sau prescurtat:

$$E = e^{hD}$$

și folosind relațiile (9.64) se mai poate scrie:

$$\begin{aligned} e^{hD} &= (1 + \Delta), \\ e^{hD} &= (1 - \nabla)^{-1}, \\ \delta &= e^{\frac{1}{2}hD} - e^{-\frac{1}{2}hD} = 2\text{sh}\left(\frac{1}{2}hD\right), \\ \mu &= \text{ch}\left(\frac{1}{2}hD\right). \end{aligned} \quad (9.65)$$

Din primele două formule de mai sus reiese:

$$hD = \ln E = \ln(1 + \Delta) = -\ln(1 - \nabla),$$

care –dezvoltată în serie de puteri– dă:

$$D = \frac{1}{h} \left(\Delta - \frac{\Delta^2}{2} + \frac{\Delta^3}{3} - \frac{\Delta^4}{4} + \frac{\Delta^5}{5} - \dots \right), \quad (9.66d)$$

sau

$$D = \frac{1}{h} \left(\nabla + \frac{\nabla^2}{2} + \frac{\nabla^3}{3} + \frac{\nabla^4}{4} + \dots \right). \quad (9.66s)$$

Din relațiile (9.66) rezultă operatorul-diferență finită pentru aproximarea operatorului de derivare $L \equiv D$:

$$Du_i = \frac{1}{h} \left(\Delta u_i - \frac{1}{2} \Delta^2 u_i + \frac{1}{3} \Delta^3 u_i - \frac{1}{4} \Delta^4 u_i + \frac{1}{5} \Delta^5 u_i - \dots \right), \quad (9.66\Delta)$$

cu diferențe la dreapta Δ sau:

$$Du_i = \frac{1}{h} \left(\nabla u_i + \frac{1}{2} \nabla^2 u_i + \frac{1}{3} \nabla^3 u_i + \frac{1}{4} \nabla^4 u_i + \frac{1}{5} \nabla^5 u_i + \dots \right), \quad (9.66\nabla)$$

cu diferențe la stânga ∇ .

Prin ridicarea la pătrat a formulelor (9.66) se obține operatorul-diferență finită pentru aproximarea operatorului $L \equiv D^2 = d^2/dx^2$ al derivatei de ordinul doi pe axa Ox . De exemplu, utilizându-se formula (9.66 Δ), cu diferențe la dreapta Δ , rezultă:

$$(9.67) \quad D^2 u_i = \frac{1}{h^2} (\Delta^2 u_i - \Delta^3 u_i + \frac{11}{12} \Delta^4 u_i - \frac{5}{6} \Delta^5 u_i + \dots).$$

Mult mai utilizat în practică decât formulele (9.66) sunt relațiile care folosesc diferența simetrică δ . Astfel, plecând de la expresia (9.65) a lui δ rezultă:

$$hD = 2 \arg \operatorname{sh} \frac{\delta}{2},$$

din care –prin dezvoltare în serie– se obține:

$$hD = 2 \left[\frac{1}{2} \delta - \frac{1}{2} \cdot \frac{1}{3} \left(\frac{\delta}{2} \right)^3 + \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{1}{5} \left(\frac{\delta}{2} \right)^5 - \dots \right],$$

sau:

$$2hD = \delta - \frac{1^2}{2^2 \cdot 3!} \delta^3 + \frac{1^2 \cdot 3^2}{2^4 \cdot 5!} \delta^5 - \dots.$$

Înmulțindu-se această ultimă egalitate cu 1 sub forma identității:

$$\mu \left(1 + \frac{\delta^2}{4} \right)^{-1/2} = 1,$$

care rezultă din relațiile (9.64), se deduce:

$$2hD = \mu \left(1 + \frac{\delta^2}{4} \right)^{-1/2} \cdot \left(\delta - \frac{1}{24} \delta^3 + \frac{3}{640} \delta^5 - \dots \right),$$

de unde –după efectuarea operațiilor din membrul drept– rezultă formula cu diferențe simetrice pentru calculul numeric al derivatelor de ordinul 1 și anume:

$$(9.68) \quad Du_i = \frac{1}{2h} \mu \left(\delta u_i - \frac{1}{6} \delta^3 u_i + \frac{1}{30} \delta^5 u_i - \frac{1}{400} \delta^7 u_i + \dots \right).$$

Ridicându-se la puterea k ($k=2,3,4,\dots$) relația (9.68) se obțin formulele cu diferență simetrică pentru calculul numeric al derivatelor de ordin superior:

$$(9.69) \quad \begin{aligned} D^2 u_i &= \frac{1}{h^2} \left(\delta^2 u_i - \frac{1}{12} \delta^4 u_i + \frac{1}{90} \delta^6 u_i - \frac{1}{560} \delta^8 u_i + \dots \right), \\ D^3 u_i &= \frac{1}{2h^3} \mu \left(\delta^3 u_i - \frac{1}{4} \delta^5 u_i + \frac{7}{120} \delta^7 u_i - \dots \right), \\ D^4 u_i &= \frac{1}{h^4} \left(\delta^4 u_i - \frac{1}{6} \delta^6 u_i + \frac{7}{240} \delta^8 u_i - \dots \right), \end{aligned}$$

toate aceste formule ale operatorilor-diferență finită aproximând operatorii de derivare cu o anumită eroare, ordinul acestei erori –notat cu $O(h^k)$ – depinzând, evident, de numărul de termeni omiși în șirul seriei de dezvoltare și de mărimea pasului h al rețelei de discretizare R alese pe $\bar{\Omega}$.

Prin acest procedeu se poate determina ușor operatorul – diferență finită pe rețele de noduri pentru orice fel de operator L cu derivate parțiale. Astfel, în cazul tridimensional ($\Omega \subset E^3$) și o rețea de discretizare R rectangulară cu pași neegali. Notând nodul central al structurii caracteristice R_q (care este nodul curent prin care se baleiază întreaga rețea R), adică $M = (m_x \cdot h_x, m_y \cdot h_y, m_z \cdot h_z)$, într-o formă care să simplifice scrierea și anume $M = (i, j, k)$ $i, j, k \in N$ și utilizând numai primul termen al formulelor (9.66Δ) și (9.67) se obțin următorii operatori – diferență finită:

$$D_x u(M) = \frac{1}{h_x} \cdot \Delta u(M) + O(h_x^2) = \frac{1}{h_x} \cdot [U(i+1, j, k) - U(i, j, k)] + O(h_x^2),$$

unde $O(h_x^2)$ este ordinul de aproximare al derivatei parțiale de ordinul unu $\partial u / \partial x$;

$$D_y u(M) = \frac{1}{h_y} \cdot [U(i, j+1, k) - U(i, j, k)] + O(h_y^2); \quad (9.70)$$

$$D_z u(M) = \frac{1}{h_z} \cdot [U(i, j, k+1) - U(i, j, k)] + O(h_z^2);$$

$$D_{x^2}^2 u(M) = \frac{1}{h_x^2} \cdot [U(i+2, j, k) - 2U(i+1, j, k) + U(i, j, k)] + O(h_x^2);$$

$$D_{y^2}^2 u(M) = \frac{1}{h_y^2} \cdot [U(i, j+2, k) - 2U(i, j+1, k) + U(i, j, k)] + O(h_y^2); \quad (9.71)$$

$$D_{z^2}^2 u(M) = \frac{1}{h_z^2} \cdot [U(i, j, k+2) - 2U(i, j, k+1) + U(i, j, k)] + O(h_z^2).$$

Dacă se utilizează formulele cu diferențe simetrice (9.68) și (9.69), atunci se vor obține alte forme de discretizare a operatorului L (cu derivate parțiale) –scrise în continuare numai pe direcția Ox – care sunt mai simplu de aplicat tocmai datorită simetriei:

$$D_x u(M) = \frac{1}{2h_x} \cdot [U(i+1, j, k) - U(i-1, j, k)] + O(h_x^2), \quad (9.72)$$

$$D_{x^2}^2 u(M) = \frac{1}{h_x^2} \cdot [U(i+1, j, k) - 2U(i, j, k) + U(i-1, j, k)] + O(h_x^2). \quad (9.73)$$

Derivatele parțiale în raport cu y și x se scriu în mod asemănător, cu singura deosebire că se va opera asupra coeficienților j (pentru y) și k (pentru x).

„Combinând” aceste formule putem determina și operatorii-diferență finită pentru derivatele parțiale de forma $D_{x^p y^q z^n}^{p+q+n}$; astfel, spre exemplu, în bidimensional ($\Omega \subset E^2$):

$$D_{xy}^2 u(M) = \frac{1}{4h_x h_y} \cdot [U(i+1, j+1) - U(i+1, j-1) - U(i-1, j+1) + U(i-1, j-1)] + O(h_x, h_y) \quad (9.74)$$

Pentru operatorul laplacean, în plan, $L = \Delta = \partial^2 / \partial x^2 + \partial^2 / \partial y^2 = D_{x^2}^2 + D_{y^2}^2$, în cazul unei rețele pătrate cu pași egali ($h_x = h_y = h$), rezultă prin aplicarea formulei (9.72) pe o structură caracteristică cu $q=5$ noduri (unul central și celelalte 4 noduri vecine luate pe direcția axelor: $i-1, j-1$ și $j+1$):

$$\begin{aligned} \Delta u(M) &= D_{x^2}^2 u(M) + D_{y^2}^2 u(M) = \\ &= \frac{1}{h^2} \cdot [U(i-1, j) + U(i, j-1) + U(i+1, j) + U(i, j+1) - 4U(i, j)] + O(h^2). \end{aligned} \quad (9.75L)$$

Dacă, pentru simplificarea scrierii, notăm nodurile structurii cu 5 puncte prin: $M = (i, j) \hat{=} 0$ (nodul central) și $(i-1, j) \hat{=} 1$, $(i+1, j) \hat{=} 2$, $(i, j-1) \hat{=} 3$, $(i, j+1) \hat{=} 4$ (nodurile vecine imediat lui $M \hat{=} 0$ pe cele două axe), rezultă următoarea formulă simplă pentru discretizarea laplaceanului (în plan):

$$\Delta u(M) = \frac{1}{h^2} \cdot (U_1 + U_2 + U_3 + U_4 - 4U_0) + O(h^2). \quad (9.75L+)$$

Deci, pentru rezolvarea numerică a unei probleme cu condiții la limită în care intervine ecuația lui Laplace, în plan, aceasta se înlocuiește (se aproximează) prin ecuația cu diferențe finite:

$$\Delta u = 0 \rightarrow U_1 + U_2 + U_3 + U_4 - 4U_0 = 0, \quad (9.75La)$$

iar pentru ecuația lui Poisson (în plan):

$$(9.75P) \quad \Delta u = f \rightarrow U_1 + U_2 + U_3 + U_4 - 4U_0 = F_0,$$

care au ordinul de aproximare $O(h^2)$.

Structura caracteristică cu 5 puncte poate fi aleasă și prin utilizarea nodurilor vecine nodului central $M_0 = (i, j) \hat{=} 0$ situate pe diagonale (la “distanța” $\sqrt{2}h$), adică $(i-1, j-1) \hat{=} 5$, $(i+1, j-1) \hat{=} 6$, $(i-1, j+1) \hat{=} 7$ și $(i+1, j+1) \hat{=} 8$, rezultând pentru laplacean:

$$\Delta u(M) = \frac{1}{2h^2} \cdot [U(i-1, j-1) + U(i+1, j-1) + U(j+1, i-1) + U(j+1, i+1) - 4U(i, j)] + O(h^2)$$

sau:

$$(9.75LX) \quad \Delta u_0 = \frac{1}{2h^2} \cdot (U_5 + U_6 + U_7 + U_8 - 4U_0) + O(h^2),$$

care formulă nu diferă de (9.75L+). Singura diferență constă în aceea că punctele sunt mai distanțate, pasul fiind egal aici cu diagonală ochiului, adică cu $\sqrt{2}h$, însă în discretizare precizia este aceeași pentru că, față de (9.75L+) cu distanța de un pas h între noduri, în formula (9.75LX) se împarte cu $2h^2$ (în loc de h^2).

O precizie mai bună se obține dacă se combină cele două formule de discretizare (9.75L+) cu (9.75LX), luându-se (în plan) o structură cu $q = 9$ noduri vecine (toate punctele posibil vecine)

nodului central $M = (i, j) \hat{=} 0$, adică $(i-1, j) \hat{=} 1$, $(i+1, j) \hat{=} 2$, $(i, j-1) \hat{=} 3$, $(i, j+1) \hat{=} 4$, $(i-1, j-1) \hat{=} 5$, $(i+1, j-1) \hat{=} 6$, $(i-1, j+1) \hat{=} 7$ și $(i+1, j+1) \hat{=} 8$, caz în care se obține pentru laplacean un model discret ce realizează cea mai bună aproximare $O(h^4)$ în plan și anume:

$$(9.75L^{\square}) \quad \Delta u(M) = \Delta u_0 = \frac{1}{6h^2} \cdot (4U_1 + 4U_2 + 4U_3 + 4U_4 + U_5 + U_6 + U_7 + U_8 - 20U_0) + O(h^4)$$

Operatori-șablon. În practică, pentru o reprezentare mai expeditivă și în același timp intuitivă, se obișnuiește ca formulele cu operatori-diferență finită (9.72), (9.73), (9.74) și (9.75) să se reprezinte grafic, prin așa-numiții operatori-șablon, care se construiesc în modul următor: fiecărui nod al rețelei caracteristice R_q i se asociază o „căsuță (reprezentată grafic printr-un pătrat sau dreptunghi) în interiorul căreia se înscrie coeficientul corespunzător al valorii discrete U din formula de discretizare. Șablonul este simetric față de punctul central $M = (i, j, k) = M_0 \hat{=} 0$ și de aceea se scrie numai o singură dată, înțelegându-se că lui i se aplică întreg operatorul șablon. Căsuța nodului central se marchează cu un punct, iar întreg șablonul reprezintă direct operatorul-diferență finită L_h .

Redăm în continuare operatorii-șablon pentru operatorii-diferență finită arătați la punctul precedent, folosind aceeași numerotație, pentru identificarea lor, urmată de litera ș.

$$(9.72 \text{ x}) \quad \text{ș} \quad D_x u(i, j, k) = \frac{1}{2h_x} \begin{array}{|c|} \hline L_h \\ \hline -1 \\ \hline 0. \\ \hline 1 \\ \hline L_h \\ \hline \end{array} U_0 + O(h_x^2) \quad \begin{array}{c} O \\ \downarrow \\ x \end{array}$$

$$(9.72 \text{ y}) \quad \text{ș} \quad D_y u(i, j, k) = \frac{1}{2h_y} \begin{array}{|c|c|c|} \hline -1 & 0. & 1 \\ \hline \end{array} U_0 + O(h_y^2) \quad \begin{array}{c} O \\ \longrightarrow \\ y \end{array}$$

$$D_z u(i, j, k) = \frac{1}{2h_z} \begin{array}{|c|} \hline 1 \\ \hline 0. \\ \hline -1 \\ \hline \end{array} U_0 + O(h_z^2) \quad \begin{array}{c} z \\ \uparrow \\ O \end{array} \quad (9.72 \text{ z}) \S$$

$$D_x^2 u(i, j, k) = \frac{1}{2h_x^2} \begin{array}{|c|} \hline 1 \\ \hline -2. \\ \hline 1 \\ \hline \end{array} U_0 + O(h_x^2) \quad (9.73 \text{ x}) \S$$

$$D_y^2 u(i, j, k) = \frac{1}{2h_y^2} \begin{array}{|c|c|c|} \hline 1 & -2 & 1 \\ \hline \end{array} U_0 + O(h_y^2) \quad (9.73 \text{ y}) \S$$

$$D_z^2 u(i, j, k) = \frac{1}{2h_z^2} \begin{array}{|c|} \hline 1 \\ \hline -2. \\ \hline 1 \\ \hline \end{array} U_0 + O(h_z^2) \quad (9.73 \text{ z}) \S$$

$$D_{xy}^2 u(i, j, k) = \frac{1}{4h_x h_y} \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline 0 & 0. & 0 \\ \hline 1 & 0 & -1 \\ \hline \end{array} U_0 + O(h_x, h_y) \quad (9.74) \S$$

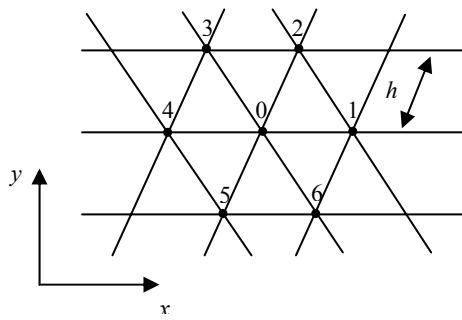
$$\Delta u(M) = \frac{1}{h^2} \begin{array}{|c|} \hline 1 \\ \hline 1 & -4. & 1 \\ \hline 1 \\ \hline \end{array} U_0 + O(h^2) \quad (9.75 \text{ L+}) \S$$

$$\Delta u(M) = \frac{1}{2h^2} \begin{array}{|c|c|c|} \hline 1 & 0 & 1 \\ \hline 0 & -4. & 0 \\ \hline 1 & 0 & 1 \\ \hline \end{array} U_0 + O(h^2) \quad (9.75 \text{ L}\square) \S$$

$$\Delta u(M) = \frac{1}{6h^2} \begin{array}{|c|c|c|} \hline 1 & 4 & 1 \\ \hline 4 & -20. & 4 \\ \hline 1 & 4 & 1 \\ \hline \end{array} U_0 + O(h^4) \quad (9.75 \text{ LX}) \S$$

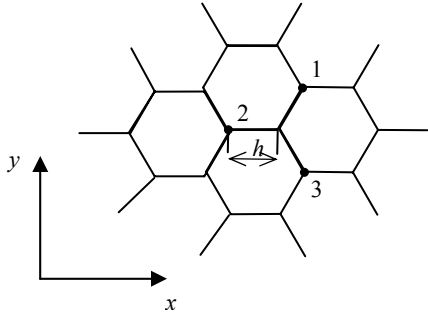
La aceștia mai adăugăm alți câțiva operatori utilizați adesea în studiul pe modele numerice a câmpului electromagnetic:

- discretizarea laplaceanului pe o rețea plană cu ochiuri triunghiulare și o structură $R_q = 7$ noduri vecine:



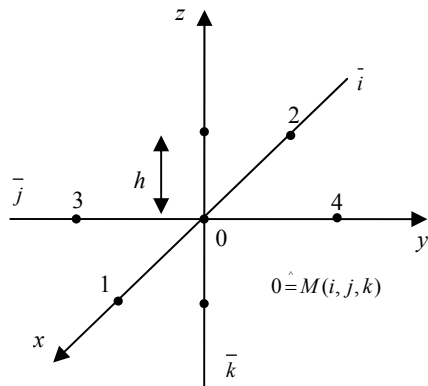
$$\Delta u_0 = \frac{1}{h^2} \cdot \frac{2}{3} \left(\sum_{s=1}^6 U_s - 6U_0 \right) + O(h^2);$$

- discretizarea laplaceanului pe o rețea plană cu ochiuri hexagonale și o structură $R_q = 4$ noduri vecine:



$$\Delta u_0 = \frac{1}{h^2} \cdot \frac{4}{3} (U_1 + U_2 + U_3 - 3U_0) + O(h^2);$$

- discretizarea laplaceanului în spațiul tridimensional pe o rețea R cubică (cu pas egal $h_x = h_y = h_z = h$) și o structură cu $R_q = 7$ noduri vecine (pe axe):



$$\Delta u(M) = \frac{1}{h^2} \begin{array}{c} \boxed{1} \\ \boxed{1} \quad \boxed{-6} \quad \boxed{1} \\ \boxed{1} \end{array} U(i, j, k) + O(h^2);$$

Aproximarea condițiilor la limită. Probleme cu derivate parțiale de tipul (9.63), pentru a putea fi rezolvate prin simulare numerică, în afara modelului discret al ecuațiilor eliptice pe care le conțin, trebuie să li se realizeze și aproximarea ecuațiilor la limită, date pe $\Sigma = \text{Fr } \Omega$, în conformitate cu problema concretă avută în vedere. Aproximarea condițiilor la limită cere cel puțin aceeași atenție și uneori un efort sporit în comparație cu aproximarea interioară a ecuațiilor cu derivate parțiale. Aceasta deoarece în multe cazuri practice $\Sigma = \text{Fr } \Omega$ are o topologie complicată și o aproximare “sumară” ar face inutilă acuratețea cu care s-au determinat ecuațiile cu diferențe finite pe nodurile $R \subset \Omega$ (interioare).

Aproximarea condițiilor la limită impune parcurgerea următoarelor etape: aproximarea conturului (frontierei) și aproximarea condițiilor la limită.

Aproximarea frontierei constă în determinarea mulțimii de noduri Σ_h care “reprezintă cel mai bine” frontiera Σ original dată, știind că $\Sigma_h = \{M \mid M \in R \subset \Sigma', M \notin \bar{R}\}$ (v. fig. 9.12), unde Σ' este o frontieră poligonală (poliedrică) construită numai pe laturile rețelei de discretizare R , care

aproximează cel mai bine pe $\Sigma = \text{Fr } \Omega$. Dacă Σ și R au forme regulate, atunci de cele mai multe ori $\Sigma \equiv \Sigma'$; în caz contrar în această fază, forma neregulată a frontierei Σ determină alegerea rețelei R sau modificarea ei adecvată (de exemplu, “îndesirea” ei) în apropierea lui Σ .

Aproximarea condițiilor la limită, date de cea de-a doua ecuație a sistemului de probleme original (9.63), se face în funcție de tipul problemei, așa cum se arată în continuare.

Aproximarea problemei Dirichlet original (“exactă”) (9.63D) se face prin următorul model cu existențe finite:

$$\begin{cases} L_h U(M) = F(M) & \text{în } \forall M \in R \subset \Omega, \\ U(M) = C(M) & \text{în } \forall M \in \Sigma_h, \end{cases} \quad (9.76)$$

$$(9.76D)$$

unde M sunt noduri ale rețelei de discretizare R , U este aproximația în $\bar{R}$ a funcției originale u definită pe $\bar{\Omega}$, F și C sunt aproximații numerice în nodurile lui R și –respectiv– Σ_h ale funcțiilor f și c date și L_h este operatorul de discretizare (diferență finită) al ecuațiilor eliptice, determinat prin formulele (9.75) pe diverse structuri caracteristice $R_q \subset R$.

În cazul unei rețele de discretizare cu pași egali, însă foarte “deasă”, Σ se aproximează mai întâi printr-o frontieră poligonală (poliedrică) Σ' , formată din laturile rețelei R cele mai apropiate de Σ . În acest caz, la scrierea ecuațiilor (9.76D) pentru fiecare nod $M \in \Sigma_h \subset \Sigma'$, apropiat de Σ , alegem un punct $P \in \Sigma$ astfel încât $|MP| < h$ să fie minim, așa ca în figura 9.12 (unde este dat un exemplu în plan) și atunci considerăm:

$$U(M_i) = c(P_i), \quad M_i \in \Sigma_h, \quad P_i \in \Sigma, \quad |M_i P_i| < h \text{ și } |M_i P_i| = \min, \quad i \in N,$$

funcția c fiind dată prin problema original (9.63D).

Dacă nu este posibilă aproximarea lui Σ prin $\Sigma' \supset R$, atunci pentru nodurile $M \in R$ din apropierea frontierei originale Σ , care au o parte din nodurile vecine în afara lui Ω (nodurile M_7, M_4, M_8 din figura 9.13), se scriu relații de interpolare pe baza nodurilor vecine situate în R și a punctelor P în care frontiera Σ intersectează laturile rețelei de discretizare R , la distanțe reprezentând fracțiuni subunitare λ din pasul h al rețelei. Astfel, în cazul $\Omega \subset E^2$, ecuația (9.76) scrisă pentru nodul central M_0 (fig. 9.13), dacă aproximarea laplaceanului se face pe $q=5$ noduri după modelul (9.75L+) sub forma:

$$\frac{\lambda_4 + 1}{4} U_0 - \frac{1}{2} U_1 - \frac{1}{2} U_2 - \frac{1}{1 + \lambda_4} U_3 = \frac{1}{\lambda_4 (1 + \lambda_4)} c_4,$$

unde $U_i = U(M_i)$, $M_i \in R$; $c_4 = c(P_4)$, $P_4 \in \Sigma$, funcția c fiind dată de condiția la limită din problema original (9.63Di) sau (9.63De) și $\lambda_4 = |M_0 P_4| / h$ (fig. 9.13).

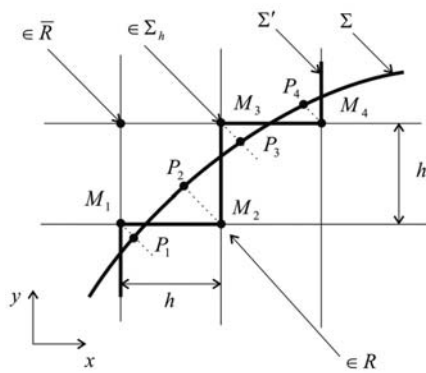


Fig. 9.12

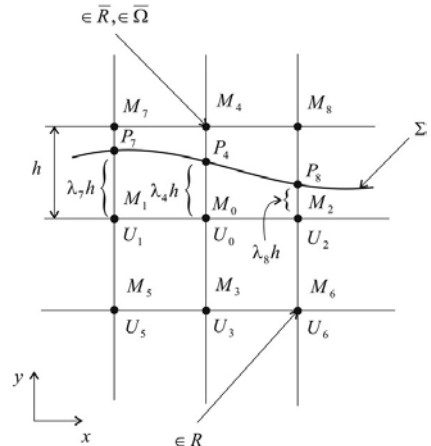


Fig. 9.13

Se poate scrie și un model numeric pe structura $R_q=9$ noduri (adică nodul central M_0 și $M_1, M_2, \dots, M_8$ din figura 9.13), însă deoarece nodurile M_4, M_7 și M_8 sunt în afara rețelei R , ele se înlocuiesc cu punctele de pe frontiera Σ , P_4, P_7 și P_8 în care –prin interpolare pe nodurile interioare $M \in R$ – se determină valorile $c(P_4), c(P_7)$ și $c(P_8)$.

Aproximarea problemei Neumann original (“exactă”) (9.63N) se face prin următorul model cu diferențe finite:

$$\begin{cases} L_h U(M) = F(M) & \text{în } \forall M \in R \subset \Omega \\ L_{hm} U(M) = C(M) & \text{în } \forall M \in \Sigma_h \end{cases} \quad (9.77)$$

unde M sunt nodurile unei rețele de discretizare R , U este aproximația în $\bar{R}$ a funcției originale u definită pe $\bar{\Omega}$, F și C sunt aproximațiile numerice în nodurile lui R și, respectiv, Σ_h ale funcțiilor f și c date, L_h este operatorul de discretizare (diferență finită) al ecuațiilor eliptice, determinat prin formulele (9.75) pe diverse structuri caracteristice $R_q \subset R$ și L_{hm} este operatorul-diferență finită ce aproximează –prin formulele (9.72)– derivata de ordinul 1 luată după direcția normalei $\bar{n}$ la Σ' .

Atunci când $\Sigma = \text{Fr } \Omega$ se aproximează prin Σ' , adică o suprafață poliedrică construită numai pe laturile rețelei de discretizare R (așa ca în figura 9.13), condiția (9.77N) se deține așa ca în cazul care urmează. Astfel, dacă $\Omega \subset E^2$ (în plan) și Σ, Σ' și Σ_h au forma din figura 9.14 (caz frecvent în practică) atunci pentru nodul M_0 –considerând cazul structurii $R_q = 5$ noduri (M_0 și cele patru puncte pe axe)– rezultă:

$$D_n u(M_i) = D_y u(M_i) \approx c(P_i), \quad M_i \in \Sigma_h, \quad P_i \in \Sigma \quad (i=0,1,2),$$

unde $c(P_i)$ reprezintă valoarea funcției c , dată prin (9.63Ni) sau (9.63Ne) ce aproximează derivata parțială D_y prin formula (9.72), după care se obține:

$$D_y u(M_0) \cong \frac{1}{2h} (U_4 - U_3) \cong c(P_0) = c_0,$$

de unde rezultă:

$$U_4 = 2h c_0 + U_3$$

și înlocuindu-l pe U_4 în modelul de aproximare pe cinci noduri (9.75L+) se deduce în final:

$$2U_0 - \frac{1}{2}U_1 - \frac{1}{2}U_2 - U_3 = hc,$$

care reprezintă (pentru cazul din figura 9.14 și $q=5$ noduri) aproximarea cu diferențe finite a condițiilor la limită (9.63Ni) sau (9.63Ne).

În cazul structurii R_9 ($M_1, M_2, \dots, M_8$ vecine nodului M_0 central), rezultă în mod similar:

$$D_y u(M_0) \cong (U_4 - U_3) / 2h \cong c(P_0) = c_0,$$

$$D_y u(M_1) \cong (U_7 - U_5) / 2h \cong c(P_1) = c_1,$$

$$D_y u(M_2) \cong (U_8 - U_6) / 2h \cong c(P_2) = c_2,$$

prin care se obțin valorile funcției discrete U în punctele din “afara” lui R :

$$U_4 = 2hc_0 + U_3, \quad U_7 = 2hc_1 + U_5 \quad \text{și} \quad U_8 = 2hc_2 + U_6,$$

care înlocuite în modelul de discretizare pe 9 puncte (9.75L□) duc la ecuația numerică:

$$5U_0 - U_1 - U_2 - 2U_3 - \frac{1}{2}U_5 - \frac{1}{2}U_6 = h(2c_0 + \frac{1}{2}c_1 + \frac{1}{2}c_2),$$

de discretizare a condițiilor la limită (9.63Ni) sau (9.63Ne). În alt caz, ce mai poate interveni adesea și ilustrat în figura 9.15, este acela în care nodul $M_4 \in E^2 - \bar{\Omega}$ (deci este în afara rețelei de discretizare R). În acest caz, nodul “de afară” M_4 se înlocuiește cu punctul $C_4 \in \Sigma$, în care frontiera Σ intersectează latura $\overline{M_0 M_4}$. Notându-se cu $\bar{n}_0, \bar{n}_1$ și $\bar{n}_2$ versorii distanțelor $\overline{M_0 P_4}$, $\overline{M_1 P_4}$ și, $\overline{M_2 P_4}$ care au expresiile:

$$\bar{n}_0 = \bar{i}\alpha_{ox} + \bar{j}\alpha_{oy} + \bar{k}\alpha_{oz}; \quad \bar{n}_1 = \bar{i}\alpha_{1x} + \bar{j}\alpha_{1y} + \bar{k}\alpha_{1z} \quad \text{și} \quad \bar{n}_2 = \bar{i}\alpha_{2x} + \bar{j}\alpha_{2y} + \bar{k}\alpha_{2z}$$

în care $\alpha_{ix}, \alpha_{iy}, \alpha_{iz}$ ($i=0,1,2$) sunt componentele verasorilor după cele trei axe ale sistemului de referință și știind –conform relației (9.15)– că derivata unei funcții scalare după direcția normalei la Σ este:

$$D_n u = \bar{\alpha}_x D_x u + \bar{\alpha}_y D_y u + \bar{\alpha}_z D_z u,$$

se poate scrie, prin discretizarea operatorilor D_x , D_y și D_z cu formulele (9.70), dar aici în plan (cu $z=0$):

$$\frac{1}{\lambda_0} [u(P_4) - u(M_0)] = D_{n_0} u(P_4) = \alpha_{0x} D_x u(P_4) + \alpha_{0y} D_y u(P_4),$$

$$\frac{1}{\lambda_1} [u(P_4) - u(M_1)] = D_{n_1} u(P_4) = \alpha_{1x} D_x u(P_4) + \alpha_{1y} D_y u(P_4),$$

$$\frac{1}{\lambda_2} [u(P_4) - u(M_2)] = D_{n_2} u(P_4) = \alpha_{2x} D_x u(P_4) + \alpha_{2y} D_y u(P_4),$$

în care λ_0, λ_1 și λ_2 sunt distanțele de la punctul $P_4 \in \Sigma$ la nodurile $M_0, M_1, M_2 \in R$ (fig. 9.15).

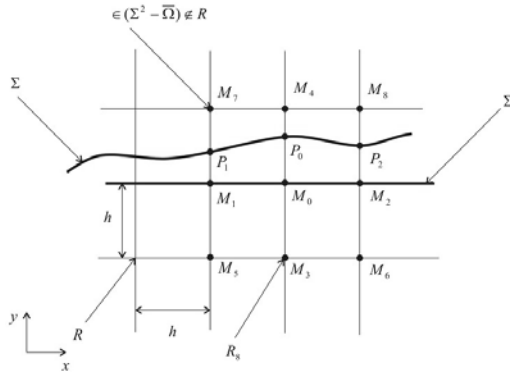


Fig. 9.14

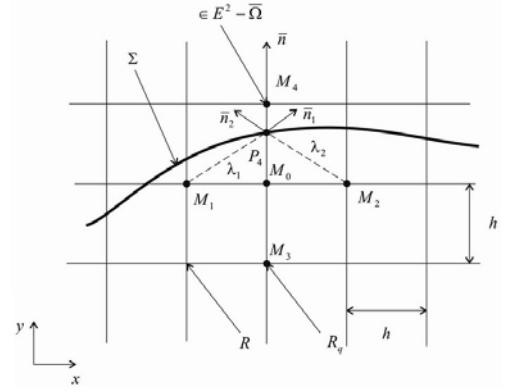


Fig. 9.15

Deoarece, conform condițiilor la limită (9.63N), $P_4 \in \Sigma$: $D_n u(P_4) = c(P_4) = c_4$, atunci, trecând la relația $u(M_i) \approx U_i$, și eliminând pe $u(P_4)$ din ecuațiile precedente rezultă:

$$U_1 - U_0 = (\lambda_0 \alpha_{0x} - \lambda_1 \alpha_{1x}) D_x u(P_4) + (\lambda_0 \alpha_{0y} - \lambda_1 \alpha_{1y}) D_y u(P_4),$$

$$U_2 - U_0 = (\lambda_0 \alpha_{0x} - \lambda_2 \alpha_{2x}) D_x u(P_4) + (\lambda_0 \alpha_{0y} - \lambda_2 \alpha_{2y}) D_y u(P_4),$$

$$c = \alpha_1 D_x u(P_4) + \alpha_2 D_y u(P_4).$$

Eliminându-se derivatele parțiale $D_x u(P_4)$ și $D_y u(P_4)$ în aceste condiții se obține în definitiv:

$$\begin{aligned} & [\lambda_1 (\alpha_1 \alpha_{1y} - \alpha_2 \alpha_{1x}) - \lambda_2 (\alpha_1 \alpha_{2y} - \alpha_2 \alpha_{2x})] U_0 + [\lambda_0 (\alpha_2 \alpha_{0x} - \alpha_1 \alpha_{0y}) - \lambda_2 (\alpha_2 \alpha_{2x} - \alpha_1 \alpha_{2y})] U_1 + \\ & + [\lambda_0 (\alpha_1 \alpha_{0y} - \alpha_2 \alpha_{0x}) - \lambda_1 (\alpha_2 \alpha_{1x} - \alpha_1 \alpha_{1y})] U_2 = \lambda_0 \lambda_1 (\alpha_{1x} \alpha_{0y} - \alpha_{0x} \alpha_{1y}) + \\ & + \lambda_0 \lambda_2 (\alpha_{0x} \alpha_{2y} - \alpha_{2x} \alpha_{0y}) + \lambda_1 \lambda_2 (\alpha_{2x} \alpha_{1y} - \alpha_{1x} \alpha_{2y}) c_4. \end{aligned}$$

Această soluție aproximează condiția la limită original (9.63Ni) sau (9.63Ne) cu o eroare de ordinul $O(h^2)$, dată de modul de aproximare a derivatelor.

9.2.4. Modele de simulare cu elemente finite

Ideea de bază a simulării numerice prin modele cu elemente finite constă în transformarea problemei variaționale inițiale (v. § 9.2.2) într-o problemă de extrem pentru funcții de mai multe variabile, rezolvabilă prin metodele analizei clasice. Metoda elementului finit este o consecință a metodei Rayleigh-Ritz, care se bazează pe aproximarea funcției necunoscute $u(x, y, z, t)$ printr-o combinație liniară a unor anumite funcții liniar independente f_i –numite funcții de coordonate:

$$u = \sum_{i=1}^n c_i f_i ,$$

unde c_i reprezintă coeficienți de pondere necunoscuți, ce urmează a fi determinați ca parametri variaționali.

Metoda elementului finit constă în următoarele etape:

1. *partiționarea* (discretizarea) *domeniului* spațial $\Omega \subset E^3$ în subdomenii disjuncte, de dimensiuni finite (și “mici”) numite elemente finite;

2. aproximarea funcției necunoscute u , definite pe Ω , la nivelul fiecărui element “e” prin combinații de forma:

$$u^e \approx \sum_i N_i(x) u_i(t),$$

în care $N_i(x)$ sunt, de obicei, polinoame de interpolare (de tip Lagrange, Hermite etc.), numite *funcții de formă sau de interpolare*, iar $u_i(t)$ reprezintă valorile discrete ale câmpului într-un anumit număr de puncte n^e numite *noduri (puncte nodale)*. Acest număr n^e reprezintă gradul de libertate al elementului finit;

3. înlocuirea funcționalei $\hat{I}(u)$ asociată câmpului prin suma “contribuțiilor” $I^e(u^e)$ ale fiecărui element finit “e” al partiției, prin:

$$(9.78) \quad \hat{I}(u) = \sum_{e=1}^m I^e(u^e);$$

4. “staționarizarea” funcționalei $\hat{I}$, adică:

$$(9.79) \quad \frac{\partial \hat{I}}{\partial u_i} = \sum_e \frac{\partial I^e}{\partial u_i} = 0,$$

unde însumarea se efectuează numai asupra acelor elemente care au vârful i comun, în număr de $m_i < m$;

5. rezolvarea sistemului de ecuații (9.79).

În scopul asigurării convergenței soluției aproximative spre soluția reală (“exactă”), funcțiile de interpolare trebuie să îndeplinească următoarele condiții:

- la *interfețele elementului*, variabila u și derivatele ei până la un ordin mai mic decât ordinul maxim al derivatelor din funcționala $I(u)$ trebuie să fie continue (*condiția de compatibilitate*);

- în *interiorul oricărui element*, funcția u și derivatele sale până la cel mai mare ordin de derivare ce apare în $I(u)$ trebuie să fie contunue (condiția de completitudine).

Din punctul de vedere al procesării (al prelucrării cu sisteme de calcul automat), metoda elementului finit comportă trei etape esențiale:

1° partiționarea domeniului analizat în elemente finite, interconectate în nodurile rețelei de discretizare;

2° aproximarea funcției u la nivelul fiecărui element și obținerea *ecuațiilor elementare*;

3° asamblarea ecuațiilor elementare obținute din condiția (9.79), de staționaritate a funcționalei (9.78), și rezolvarea modelului global rezultat.

Peste tot în această carte, prin **element finit** se înțelege ansamblul format din:

- elemente de discretizare propriu-zis (v. fig. 9.16),

- funcția de interpolare locală,

- mulțimea punctelor și a variabilelor nodale caracteristice fiecărui element (v. tabelul. 9.2).

Discretizarea domeniului. Una din etapele esențiale (și prima) în simularea numerică a proceselor electromagnetice prin utilizarea metodei elementului finit constă în alegerea *forme elementului* în care va fi partiționat domeniul spațial Ω de definiție a funcției necunoscute u .

În cadrul problemelor unidimensionale, adică a problemelor în care apare o singură variabilă spațială x , elementele sunt segmente de linie. Numărul de noduri asigurate unui anumit element

depinde de tipul variabilelor nodale, de tipul funcției de interpolare și de gradul de continuitate impus.

În problemele de câmp bidimensionale, cele mai simple elemente utilizate sunt triunghiul și dreptunghiul (fig. 9.16). Aceste elemente pot avea și laturi curbilinii, fapt ce le permite o acoperire mai bună a suprafețelor curbe sau cu contururi curbe. Particularitatea elementelor finite o constituie prezența nodurilor interioare sau a celor suplimentare față de vârfuri (v. tablul 9.2).

Discretizarea cu elemente triunghiulare prezintă o mare suplețe în aproximarea contururilor curbe. În funcție de numărul de noduri ce caracterizează un triunghi prin numărul gradelor de libertate, există elemente

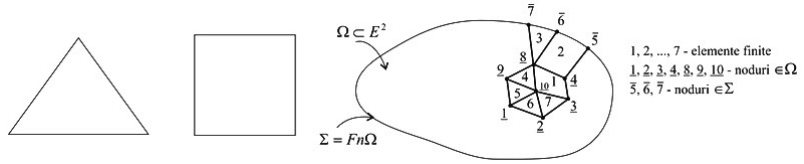


Fig. 9.16

triunghiulare liniare (element bidimensional, de ordinul 1, cu 3 noduri), cudrice (cu 6 noduri), cubice (cu 10 noduri) etc. (v. tabelul 9.2). Nodurile interioare nu parti-cipă la conexiunea elemente-lor, ci doar la determinarea modelului elementar de aproximare.

Pentru a se distinge (a nu se confunda) nodurile de elemente, numerotarea lor se face astfel: elementele cu numere întregi (scrise fără nici un “adaus” grafic), iar nodurile cu numere subliniate (cu bară jos): $\underline{1}, \underline{2}, \dots, \underline{L}$, iar cele de pe frontieră cu bară sus: $\bar{1}, \bar{2}, \dots, \bar{m}$ (v. tabelul 9.3). De preferință, nodurile se aleg în vârfurile elementelor, în puncte de simetrie, pe laturi, în puncte de simetrie, pe laturi, în centrele de simetrie ale elementelor etc. (vezi tabelul 9.2), numerotarea lor făcându-se în același sens (de obicei sensul trigonometric).

În general, într-o porțiune triunghiulară cu l elemente, un element oarecare i de ordin k are n^e noduri date de:

$$n^e = \frac{1}{2}(k+1)(k+2).$$

Din motive de simetrie cele n^e noduri sunt situate astfel: 3 în vârfurile triunghiului, $(k-1)$ noduri pe fiecare latură și $(k-1)(k-2)/2$ noduri în interiorul triunghiului (vezi tabelul 9.2).

În problemele de câmp tridimensional se utilizează elemente de tip tetraedru, prismă, tor etc. Și în acest caz, în funcție de modelul elementar de aproximare, aceste elemente pot fi liniare (de ordinul 1), cuadrice, cubice etc. (în funcție de numărul de noduri ce caracterizează fiecare element). Un tip special de element tridimensional îl constituie cel axisimetric, care poate fi tratat ca un element de suprafață datorită simetriei sale axiale (de tip tor). Astfel problemele tridimensionale pot fi modelate în spațiu bidimensional folosind modelele numerice dezvoltate pentru aplicații pe $\Omega \subset E^2$, acesta cu atât mai mult cu cât majoritatea dispozitivelor electromagnetice prezintă simetrii axiale, atât din punct de vedere geometric cât și fizic (de exemplu, după cum se știe, -v cap. 1- liniile de câmp magnetic, care este un câmp de divergență zero, sunt linii închise).

Funcții de interpolare

În metoda elementului finit funcțiile utilizate pentru reprezentarea comportării variabilei de câmp, μ , în interiorul unui element se numesc *funcții de interpolare* sau *funcții de formă* sau -încă- *funcții de aproximare*. Aceste funcții depind de structura modală a elementului și de forma lui. Deși se pot concepe multe tipuri de funcții de interpolare, cel mai frecvent se folosesc funcțiile polinomiale datorită ușurinței relative privind executarea operațiilor cu polinoame (derivare, integrare etc.). În cadrul acestei lucrări s-a optat pentru elementele finite triunghiulare de ordinul întâi din următoarele motive:

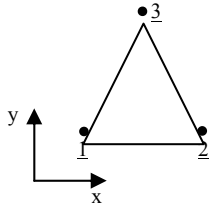
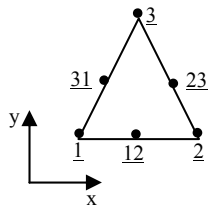
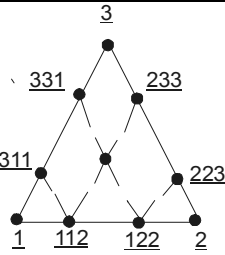
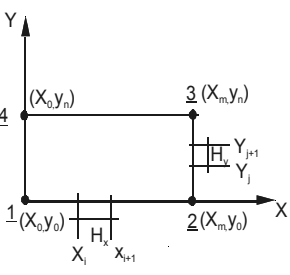
- numărul minim de noduri caracteristice elementului;

- structura simplă a funcțiilor de interpolare;
- obținerea unei matrice rare, de tip bandă, prin indexarea (notarea) convenabilă globală a nodurilor.

Pentru o mai ușoară apreciere, în tabelul 9.2 sunt prezentate câteva funcții uzuale de interpolare.

Tabel 9.2

Funcții de interpolare (aproximare) pentru diverse forme ale elementului finit și diferite structuri nodale, în planul (E^2).

Elementul finit	Nr. de noduri	Polinomul de aproximare
	3	$u^e = a_1 + a_2 x + a_3 y$ $u(x, y) = u_i = a_1 + a_2 x_i + a_3 y_i, i \in \{1, 2, 3\}$
	6	$u(x, y) = a_{00} + a_{10}x + a_{11}y + a_{20}x^2 + a_{22}y^2$
	10	$u(x, y) = a_{00} + a_{10}x + a_{10}y + a_{20}x^2 + a_{21}xy + a_{22}y^2 + a_{30}x^3 + a_{31}xy^2 + a_{32}xy^2 + a_{33}y^3$
	4	$\mu_0(x, y) = \sum_{i=0}^m \sum_{j=0}^n U_{0,ij}(x, y) U_{ij}$ $U_{0,ij} = \alpha_{ij}(x, y) U_{ij} + \beta_{i+1}(x, y) U_{i+1,j} + \gamma_{i,j+1}(x, y) U_{i+1,j+1}$ unde: $\varepsilon_{ij}(x, y) = \frac{1}{h_x h_y} (x_{i+1} - x)(y_{j+1} - y),$ $\beta_{ij}(x, y) = \frac{1}{h_x h_y} (x - x_i)(y_{j+1} - y),$ $\gamma_{ij}(x, y) = \frac{1}{h_x h_y} (x_{i+1} - x)(y - y_j),$ $\alpha_{i+1,j+1}(x, y) = \frac{1}{h_x h_y} (x - x_i)(y - y_j)$

După cum rezultă din tabelul 9.2, problema se reduce (într-o primă instanță), la determinarea coeficienților funcției (polinomului) de interpolare.

Pentru a ilustra acest fapt, se va reveni la tabelul triunghiular de ordin 1 – liniar (linia unu din tabelul 9.2), cu nodurile 1, 2, 3, la nivelul căruia funcției μ se aproximează prin:

$$u^e = a_1 + a_2 x + a_3 y,$$

Care – pentru determinarea coeficienților aproximării trebuie să respecte condițiile :

$$u_i = a_1 + a_2 x_i + a_3 y_i, \quad i \in \{1, 2, 3\},$$

Din care rezultă:

$$a_1 = \frac{(x_2 y_3 - x_3 y_2) \mu_1 + (x_3 y_1 - x_1 y_3) \mu_2 + (x_1 y_2 - x_2 y_1) \mu_3}{2D^e},$$

$$a_2 = \frac{(y_1 - y_2) \mu_1 + (y_3 - y_1) \mu_2 + (y_1 - y_2) \mu_3}{2D^e}$$

$$a_3 = \frac{(x_3 - x_2) \mu_1 + (x_1 - x_3) \mu_2 + (x_2 - x_1) \mu_3}{2D^e}$$

Introducând aceste valori în polinomul (9.80) se va obține:

$$u^e = \sum_{i=1}^3 N_i \mu_i, \quad (9.81)$$

Cu funcțiile de interpolare:

$$N_i = \frac{(a_i + b_i x + c_i y)}{2D^e},$$

În care:

valorile coeficienților a , b , c sunt :

$$a_1 = x_2 y_3 - x_3 y_2; \quad b_1 = y_2 - y_3; \quad c_1 = x_3 - x_1$$

și –prin permutări circulare asupra indicilor 1, 2 și 3– se obțin ceilalți coeficienți a_2 , b_2 , c_2 , a_3 , b_3 , c_3 ; simbolul D^e reprezintă aria triunghiului 1 2 3 și are valoarea dată prin dezvoltarea determinantului:

$$D^e = \frac{1}{2} \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix},$$

Valoarea care este pozitivă dacă numerotarea nodurilor se face în sens trigonometric (așa ca în tabelul 9.2).

Relația (9.81) se poate scrie sub formă matriceală prin:

$$u^e = [N]^e \{u\}^e,$$

în care $[N]^e$ este matricea (vectozul) linie a funcției de interpolare, iar $\{u\}^e$ – matricea (vectorul) coloană al valorilor nodale ale lui μ . Notarea cu paranteze ($[]$ pentru linie și $\{ \}$ pentru coloană) este „clasică” în aplicațiile produse informatice CAD.

Apartenența nodurilor la elemente, se stabilește sub formă de tabel. Astfel pentru o partiție triunghiulară cu l elemente simple (triunghiuri de ordinul 1, cu trei noduri în vârfuri) și n noduri, unul din tabelele cu n linii și două coloane, ceea ce se notează prin $(n, 2)$, conține coordonatele în sistemul $yo x$ ale nodurilor (pe fiecare linie sunt indicate coordonate x și y ale nodurilor, în ordinea numerotării lor, de obicei nodul central, nodurile de pe contur și apoi nodurile de pe laturile triunghiurilor – element finit; celălalt tabel are l limite și n^e (care reprezintă numărul de noduri al unui singur element finit, în capul triunghiului de ordin 1 fiind $n^e=3$) coloane, ceea ce se notează prin (l, n^e) , stabilește apartenența nodurilor la cele l elemente: pe fiecare linie a tabelului, în ordinea numerotării lor, sunt indicate numerele nodurilor din vârfurile și laturile triunghiului, scrise în sens trigonometric pozitiv.

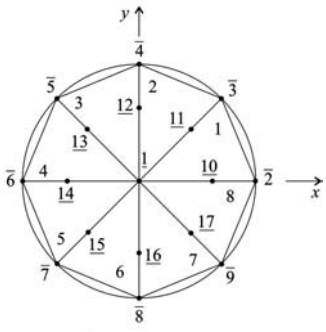
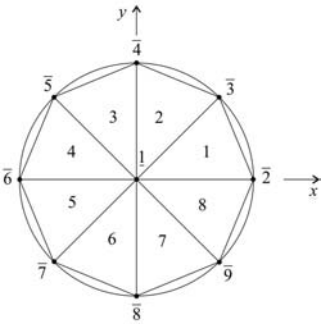
De exemplu, pentru un domeniu plan în formă de cerc cu raza egală cu 1 (o unitate), aproximat prin poligonul regulat cu 8 vârfuri, partiționat în $l=8$ triunghiuri de ordinul 1 (vezi tabelul 9.3), identificarea elementelor, nodurilor (în exemplu dat, $n=9$) și apartenența acestora la

elementele face prin introducerea în programul decalcul a următoarelor date: 18 numere reale ale primului tabel $(n,2)=(9,2)$ și 24 numere întregi ale celui de al doilea tabel $(l,n^e)=(8,3)$.

În cazul partiției cu element finit – triunghi de ordin superior, numărul datelor crește simțitor, poziționarea elementelor fiind din ce în ce mai dificilă; de aceea produsele informatice pentru analiza cu elemente finite conțin rutine (subprograme) specializate pentru operația de partiționare domeniului analizat și apartenența nodurilor la elemente. Pentru exemplificare, pe linia a doua din tabelul 9.3, se prezintă cazul în care domeniul circular cu raza unitate a fost partiționat tot în $l = 8$ triunghiuri, însă de data aceasta de ordinul 2 (cu $n^e = 5$ noduri pentru un triunghi elementar); în acest fel tabelul cu coordonatele nodurilor $(n, 2) = (17, 2)$ are 34 date (nu-

Tabelul 9.3

Apartenența nodului la elementele finite prin care se face partiționarea domeniului analizat

Partiționarea domeniului în triunghiuri elementare	Tabelul cu coordonatele nodurilor ($n, 2$)	Tabelul de aparență a nodurilor la un element finit (l, n^e)
 $l=8, n^e=5,$ $n=17$	(9, 2) $\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 1 \\ -1/\sqrt{2} & -1/\sqrt{2} \\ -1 & 0 \\ -1/\sqrt{2} & -1/\sqrt{2} \\ 0 & -1 \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$	(8, 3) $\begin{bmatrix} 1 & 2 & 3 \\ 3 & 4 & 1 \\ 1 & 4 & 5 \\ 5 & 6 & 1 \\ 1 & 6 & 7 \\ 7 & 8 & 1 \\ 1 & 8 & 9 \\ 9 & 2 & 1 \end{bmatrix}$
 $l=8, n^e=3,$ $n=9$	(17, 2) $\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} \\ 0 & 1 \\ -1/\sqrt{2} & 1/\sqrt{2} \\ -1 & 0 \\ -1/\sqrt{2} & -1/\sqrt{2} \\ -1 & 0 \\ 1/\sqrt{2} & -1/\sqrt{2} \\ 1/2 & 0 \\ 1/2\sqrt{2} & 1/2\sqrt{2} \\ 0 & 1/2 \\ -1/2\sqrt{2} & 1/2\sqrt{2} \\ 1/2 & 0 \\ -1/2\sqrt{2} & -1/2\sqrt{2} \\ 0 & -1/2 \\ 1/2\sqrt{2} & -1/2\sqrt{2} \end{bmatrix}$	(8, 5) $\begin{bmatrix} 1 & 10 & 2 & 3 & 11 \\ 3 & 4 & 12 & 1 & 11 \\ 1 & 12 & 4 & 5 & 13 \\ 5 & 6 & 14 & 1 & 13 \\ 1 & 14 & 6 & 7 & 15 \\ 7 & 8 & 16 & 1 & 15 \\ 1 & 16 & 8 & 9 & 17 \\ 9 & 2 & 10 & 1 & 17 \end{bmatrix}$

mere reale), iar tabelul de apartenență a nodurilor la un singur triunghi $(l, n^e)'(8,5)$ are 40 de date (numere întregi care reprezintă indexul/notația modului).

În aplicația 5. (din § 5.6.1), pentru calculul câmpului magnetic în tridimensional din întrefierul unui alternator, partiționarea s-a făcut în tetraedre.

Sisteme de coordonate. În modelarea numerică cu metoda elementului finit se folosesc două sisteme de referință :

- un sistem global asociat domeniului Ω de analiză (arătat în tabelul 9.3);
- un sistem local asociat fiecărui element finit.

Sistemele locale la care coordonatele unui punct sunt legate de geometria elementului și variază între 0 și 1 , numite și coordonate naturale sunt cele mai avantajoase din punctul de vedere al calculului integralelor funcțiilor de interpolare.

Coordonatele locale naturale sunt „un fel” de coordonate normalizate. Dacă se alege ca origine pentru coordonatele naturale centrul elementului , atunci domeniul de variație al acestor coordonate este $[-1,1]$ (vezi tabelul 9.3, coloana 2); dacă se aleg mai multe puncte ca origine (de pildă nodurile elementului), atunci domeniul de variație este $[0,1]$. În primul caz se obțin coordonate naturale , în cel de-al doilea caz se obțin așa-numitele coordonate L –naturale.

În cazul elementelor finite triunghiulare sau tetraedrale (v. aplicația 5.) se preferă coordonatele L –naturale, care sunt prezentate în figura 9.17 (pentru elemente triunghiulare). În acest caz ele sunt denumite și coordonate de arie datorită semnificației lor geometrice.

Pentru obținerea acestor coordonate se scrie :

$$x = \sum_{i=1}^3 L_i x_i, \quad y = \sum_{i=1}^3 L_i y_i, \quad \sum_{i=1}^3 L_i = 1$$

și notând cu A aria triunghiului rezultă:

$$L_1 = \frac{A_1}{A}, \quad L_2 = \frac{A_2}{A}, \quad L_3 = \frac{A_3}{A},$$

semnificațiile pentru A_i ($i=1,2,3$) fiind cele arătate în figura 9.17.

Există formule simple de integrare pentru coordonatele de arie pe un element triunghiular și anume :

$$\int_{A^e} L_1^\alpha L_2^\beta L_3^\delta dA = \frac{\alpha! \beta! \delta!}{(\alpha + \beta + \delta + 2)!} 2A.$$

Pentru elementele finite tridimensionale , un sistem de coordonate naturale este legat de volume . Cele două sisteme de coordonate , global (x, y, z) și local (L_1, L_2, L_3, L_4) sunt legate prin relațiile :

$$x = \sum_{i=1}^4 L_i x_i, \quad y = \sum_{i=1}^4 L_i y_i, \quad z = \sum_{i=1}^4 L_i z_i \quad \text{și} \quad \sum_{i=1}^4 L_i = 1.$$

Din rezolvarea acestor ecuații rezultă:

$$L_i = \frac{1}{6V} (a_i + b_i x + c_i y + d_i z), \quad i = 1, 2, 3, 4,$$

Unde V reprezintă volumul tetraedrului în E^3 , definit de punctele (x_i, y_i, z_i), $i=1,2,3,4$, iar coeficienții a_i, b_i, c_i, d_i sunt determinați în funcție de coordonatele geometrice ale vârfurilor .

Pentru calculul integralelor de volum se utilizează formula :

$$\int_{\Omega^e} L_1^\alpha L_2^\beta L_3^\delta L_4^\tau dV = \frac{\alpha! \beta! \delta! \tau!}{(\alpha + \beta + \delta + \tau + 3)!} 6V.$$

Atunci când variabilele de câmp se aproximează cu aceleași funcții de interpolare cu cele prin care se realizează transformarea geometrică , elementele finite se numesc *izoparametrice*. Elementele la care geometria se definește prin funcții de interpolare de grad inferior celor prin

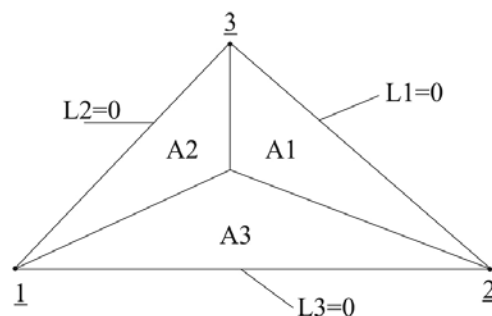


Fig. 9.17

care se aproximează variabilele de câmp, se numesc *subparametrice*; în caz contrar se numesc *supraparametrice*.

Folosirea elementelor finite implică existența a două sisteme de referință: unul local, în care se exprimă funcțiile de interpolare N_i și altul global în care se definește elementul finit (așa cum s-a arătat în tabelul 9.3) și variabilele de câmp.

Obținerea ecuațiilor elementale prin procedeul Galerkin. Pentru obținerea ecuațiilor elementale există două metode : metoda variațională și metoda reziduurilor ponderate (numită și metoda Galerkin).

Prima metodă pornește de la modelul variațional (v. § 9.2.2.) al problemei de câmp și se pune în condiția de staționaritate a funcționalei echivalente modelului diferențial. Metoda reziduurilor ponderate este mult mai generală întrucât pornește de la modelul diferențial al metodei de câmp. Acest procedeu nu are de-a face cu metoda elementului finit, dar oferă o cale simplă de obținere ecuații în elemente finite.

Metoda Galerkin implică două etape:

1. se alege o aproximare pentru variabila de câmp și se substituie această aproximare în ecuația diferențială dată , rezultând o eroare (reziduu) , după care se urmărește minimizarea acestei erori pe întreg domeniul Ω de analiză a câmpului u ;

2. rezolvarea ecuațiilor care rezultă după terminarea etapei precedente.

Astfel , pentru o ecuație de tipul (9.63) , adică $Lu = f$, se aproximează –în prima etapă– u prin u^e , metoda lui Galerkin cerând ca :

$$\int_{\Omega^e} [Lu^e - f^e] N_i d\Omega = 0 \quad , \quad i = \underline{1}, \underline{2}, \underline{3}, \dots, \underline{n} .$$

Integrând prin părți aceste integrale , se pot introduce în mod convenabil condițiile pe frontiera elementului , însă în procesul de asamblare rămân numai condițiile pe frontiera Σ a domeniului spațial și nu cele de pe interfețele elementelor , în cazul în care variabila de câmp este continuă pe aceste interfețe . În cazul unor discontinuități la interfețele elementelor , acestea pot fi introduse fără dificultate în modelul numeric global .

Asamblarea elementelor finite. Asamblarea este procesul de reunire a elementelor finite și de sinteză a domeniului Ω de analiză considerat . Din punctul de vedere geometric, asamblarea înseamnă refacerea domeniului inițial, iar din punctul de vedere funcțional, obținerea modelului numeric global al corpului studiat. Între discretizarea domeniului spațial și asamblarea elementelor finite are loc etapa de obținere a modelului numeric elemental. Asamblarea elementelor finite se poate face în două moduri: după noduri și după elemente.

În prima variantă se iau nodurile globale ale sistemului, unul câte unul, și se assemblează elementele finite în jurul fiecărui nod. În cel de-al doilea caz, asamblarea după elemente se face luându-se element cu element, în ordinea crescândă a numerotării acestora și scriindu-se ecuațiile elementale pentru fiecare element.

Asamblarea după noduri se face atunci când derivarea modelului numeric elemental se face prin metoda variațională, iar asamblarea după elemente se preferă atunci când modelul numeric se obține prin procedeul Galerkin.

În **asamblarea după noduri** se precizează o *matrice de conexiuni* prin care se stabilesc, pentru fiecare nod, care sunt elementele care îl conțin. Pentru fiecare nod i se va nota cu l_i numărul de elemente vecine care îl conțin. În acest fel funcționala care descrie problema de câmp devine – prin discretizare:

$$I(u) = \sum_{e=1}^l I^e(u^e)$$

în care l este numărul total de elemente în care s-a discretizat domeniul Ω de analiză . Condiția de minimum pentru $I(u)$ în raport cu variabila u_i este :

$$(9.82) \quad \frac{\partial I}{\partial u_i} = \sum_{e=1}^{l_i} \frac{\partial I^e}{\partial u_i}$$

întrucât contribuții în evaluarea derivatei parțiale $\partial I / \partial u_i$ aduc numai elemente ce conțin nodul i .

Scriind ecuația (9.82) pentru fiecare nod se obține modelul numeric global. Identificarea elementelor l_i pentru fiecare nod i se face cu ajutorul matricei de conexiuni (de forma celei din tabelul 9.4). Componentele acestei matrice sunt elementele care conțin nodul de pe linia specificată.

Tabelul 9.4

Matricea de conexiuni după noduri				
Nodul	Elementele vecine			
1 ⋮ ⋮	e_1	e_2	...	e_{l1}
i ⋮ ⋮	e_1	e_2	...	e_{li}

Sistemul de ecuații rezultat în urma asamblării se rezolvă după implementarea condițiilor pe frontieră.

Asamblarea după elemente cuprinde, în esență, două etape: expandarea și asamblarea propriu-zisă. Prima etapă (expandarea) constă în raportarea modelului matriceal elemental la sistemul global de noduri, folosindu-se matricea de conexiuni după elemente (care este de forma celei din tabelul 9.5). Această matrice cuprinde pentru fiecare element numerele nodurilor care îl delimitează. Coeficienții matriceali nu se modifică, ci se transformă numai poziția lor prin trecerea de la un sistem local de numerotare a nodurilor la unul global.

Tabelul 9.5

Matricea de conexiuni după elemente				
Elemente	Noduri			
	1	2	...	n
e_1 ⋮ ⋮	n_{11}	n_{12}	...	$n_{1\ e1}$
e_i ⋮ ⋮	n_{i1}	n_{i2}	...	$n_{i\ ei}$

Faza de asamblare propriu-zisă constă în suprapunerea modelelor elementale expandate, astfel încât coeficienții matriceali din două elemente vecine să se însumeze în nodurile comune. În mod similar se adună vectorii termenilor liberi corespunzători modelelor numerice elementare.

Includerea condițiilor pe frontieră. Condițiile pe frontieră considerate în modelul diferențial (original) trebuie incluse în modelul numeric obținut prin discretizare spațială. Implementarea acestor condiții se face în funcție de tipul condiției considerate. Introducerea în modelul numeric se poate face fie după obținerea modelului numeric global, fie pe durata obținerii modelului numeric global, adică la nivel de element.

În cazul problemelor de câmp electromagnetic (9.63), condițiile de tip Neumann (9.63N) omogene sunt satisfăcute în mod „automat” și deci nu necesită un efort de calcul suplimentar. Introducerea condițiilor de tip Dirichlet (9.63D) se poate face numai după obținerea modelului numeric global și singura restricție în această operație este aceea că modificările în modelul numeric să fie reduse la minimum. În acest ultim caz, se pot utiliza mai multe metode, unele păstrând dimensiunea matricei globale a modelului numeric, altele reducând dimensiunea sistemului de ecuații rezultat prin eliminarea unui număr de ecuații egal cu numărul de puncte în care funcția este precizată.

9.3. Compendiu de informatică

În activitatea de simulare numerică a problemelor de câmp electromagnetic prin utilizarea sistemelor de calcul automat, de tipul CAD/CAM (produse foarte răspândite în prezent), dintre instrumentele informatice cele mai utile în problemele de câmp electric și magnetic și de analiză-sinteză a rețelelor electrice și circuitelor magnetice (pe care le-am și folosit în aplicațiile prezentate în capitolele 2-8), două sunt de maximă importanță practică:

- produsul de tip limbaj de programare MATLAB, pentru calculul cu matrice, evaluări de funcții și prelucrări de obiecte grafice;
- pachetul de programe ANSYS pentru utilizarea metodei elementului finit.

În continuare, vor fi prezentate aceste două produse, la nivel de utilizator, în scopul de a permite cititorului însușirea rapidă a utilizării acestor instrumente informatice.

9.3.1. Introducere în MATLAB

MATLAB este un produs-program de tip utilizator elaborat de firma americană *The MathWorks, Inc.*, destinat realizării de calcule numerice (științifice, economice, ingineresti etc.) de mare performanță, cu facilități deosebite de vizualizare grafică a rezultatelor.

MATLAB integrează analiza numerică, statistica matematică, calculul matriceal și prelucrarea de obiecte grafice într-un mediu ușor accesibil, în care soluțiile problemelor se obțin mai ușor decât în cazul utilizării programării tradiționale.

Prezentarea generală a produsului-program MATLAB. Denumirea MATLAB provine din prescurtarea sintagmei *MATrix LABoratory* (laborator matriceal), deoarece elementul de bază este o matrice care nu trebuie dimensionată și poate fi, în particular, un vector sau un scalar.

Principalele facilități oferite de acest produs-program, în versiunea sa cea mai recentă, sunt următoarele:

- realizarea de calcule numerice cu precizie ridicată;
- operații cu matrice și vectori precum și cu funcții speciale aplicabile acestora;
- prelucrări statistice de date și realizarea de prognoze;
- funcții matematice speciale, precum și funcții pentru integrarea și derivarea numerică;
- funcții de optimizare și pentru aplicarea unor metode numerice;
- realizarea unei game largi de reprezentări grafice în culori, bi- și tridimensionale;
- sinteza de imagini de obiecte și prelucrări de obiecte grafice (rotiri, efecte de umbre și lumini, animație etc.);
- funcții speciale de tip *debugger* pentru depanarea programelor (fișierelor de instrucțiuni MATLAB);
- operarea cu fișiere de date numerice sau grafice și cu fișiere de instrucțiuni (adevărate programe de calcul MATLAB);
- posibilitatea interfetării cu limbajele de programare C sau FORTRAN, în vederea preluării unor subprograme.

Produsul MATLAB a cunoscut mai multe versiuni, câte una pentru fiecare sistem de operare mai răspândit. Astfel, au existat sau există versiunile PC-MATLAB (pentru sistemul de operare MS-DOS), MacMATLAB (pentru calculatoarele personale compatibile *Macintosh*), PRO-MATLAB (pentru sistemul de operare UNIX), precum și versiunile MATLAB 4.0 și superioare special concepute pentru WINDOWS. Totuși, principalele instrucțiuni și funcții MATLAB sunt identice pentru toate versiunile existente.

În încheiere, menționăm că principala caracteristică a produsului MATLAB este ușoara lui extensibilitate, ce permite oricărui utilizator să creeze noi funcții, care să se adauge celor elaborate de realizatorii produsului. Acest fapt a permis evoluția și dezvoltarea continuă a produsului, prin înglobarea de noi aplicații, elaborate în diverse medii științifice și universitare.

Instrucțiuni fundamentale și de exploatare

Generalități. În cele ce urmează, vor fi prezentate, pe categorii, principalele instrucțiuni și funcții MATLAB, necesare înțelegerii aplicațiilor, cu demonstrarea lor practică cu ajutorul versiunii MATLAB 5.0, pentru WINDOWS 95 sau superior.

Odată intrați în ambientul MATLAB, sunt posibile două moduri diferite de lucru:

1) interactiv, prin lansarea în execuție a câte unei instrucțiuni sau apelarea a câte unei funcții, prin scrierea acesteia de la tastatură în dreptul promptului dublu al MATLAB-ului, `>>`, urmată de afișarea imediată pe ecran a rezultatului. Acest mod de lucru se recomandă în cazul unor prelucrări mai puțin ample de date sau care se efectuează o singură dată;

2) utilizarea MATLAB-ului ca limbaj de programare, prin crearea unor fișiere de instrucțiuni (*.m) cu ajutorul unui editor de texte (versiunile pentru WINDOWS –începând cu 5.0– sunt prevăzute cu un editor propriu, combinat cu un debugger „M-file editor/debugger”). Aceste fișiere, care sunt, de fapt, un fel de programe MATLAB, sunt apoi lansate în execuție prin introducerea de la tastatură a numelui lor (fără extensia m) în dreptul promptului `>>`.

Pentru terminarea unei sesiuni de lucru MATLAB, se pot utiliza, pe lângă procedura uzuală de încheiere a unei aplicații sub WINDOWS și instrucțiunile **quit** sau **exit**.

Instrucțiuni de informare. Acestea sunt următoarele:

Instrucțiunea **help** care pune la dispoziție informații *on-line* despre toate instrucțiunile și funcțiile MATLAB disponibile la un moment dat, inclusiv cele nou create de către utilizator. Astfel, folosind-o ca instrucțiune simplă (neurmată de argumente), are ca rezultat afișarea pe ecran a unei liste a tuturor directorilor ce conțin fișiere legate de produsul MATLAB, fiecărui director corespunzându-i un anumit domeniu de interes. În plus, este descris succint acest domeniu pentru fiecare director.

Dacă se utilizează aceeași instrucțiune sub forma:

help < nume director >

atunci este afișată lista tuturor funcțiilor și instrucțiunilor MATLAB din directorul respectiv, fiecare fiind urmată de o caracterizare foarte succintă.

În sfârșit, dacă se utilizează instrucțiunea sub forma:

help < nume fișier >

atunci este prezentată o descriere detaliată a funcției sau instrucțiunii MATLAB realizate de către fișierul respectiv.

Instrucțiunea **helpwin** conduce la deschiderea unei ferestre de tip „menu” care conține lista tuturor instrucțiunilor MATLAB, grupate în domenii. Accasul la informația despre o anumită instrucțiune este posibil prin selectarea acesteia cu „mouse”-ul.

Instrucțiunile **who/whos** ne informează asupra variabilelor existente în memorie la un moment dat. Precizăm că, odată create și introduse în memorie, variabilele nu pot fi eliminate decât cu ajutorul instrucțiunii **clear**.

Instrucțiunea **who** conduce la afișarea pe ecran a listei complete a tuturor variabilelor rezidente în memorie la un moment dat, iar instrucțiunea **whos** indică, în plus, și dimensiunea fiecărei variabile, oferind astfel indicații asupra variabilelor ce ar putea fi eliminate pentru a se crea un spațiu disponibil în memorie pentru introducerea de noi variabile.

Precizăm că, în cazul în care nu mai este spațiu în memorie pentru definirea de noi variabile, MATLAB-ul afișează următorul mesaj de eroare:

>> Out of memory

În cazul apariției unui astfel de mesaj, se impune eliminarea unora din variabilele rezidente în memorie, cu ajutorul instrucțiunii **clear**, ce va fi prezentă în paragraful următor.

Instrucțiuni pentru importare/exportare date. Acestea au structura :

Instrucțiunea **save** se folosește pentru salvarea într-un fișier de date (de tip *.mat) a unora din variabilele existente la un moment dat în spațiul de lucru (memorie). Această instrucțiune creează în același timp fișiere de date numerice MATLAB.

Dacă se utilizează ca instrucțiune simplă (fără argumente), toate variabilele existente la un moment dat în memorie sunt salvate și încărcate într-un fișier de date numit **matlab.mat**.

Dacă însă această instrucțiune este utilizată sub forma:

save < nume fișier >

atunci toate variabilele din memorie sunt încărcate într-un fișier cu numele ales, la care se adaugă extensia **mat** ce nu trebuie inclusă în numele fișierului.

În sfârșit dacă instrucțiunea este utilizată sub forma:

save [< nume fișier >] < listă variabile >

atunci sunt salvate doar acele variabilele care sunt cuprinse în lista explicită menționată.

Fișierele de date numerice astfel create pot fi utilizate pentru exportarea de date din MATLAB în alte produse-program sau pot fi tipărite.

Instrucțiunea **load** se folosește pentru încărcarea în memorie (spațiul de lucru) a unor variabile aflate într-un fișier de date numerice (cu extensia **mat**). Acest fișier poate fi creat și cu ajutorul altor produse-program, fiind astfel posibilă importarea de date în ambientul MATLAB. Această instrucțiune are o sintaxă practic identică cu instrucțiunea **save** și anume:

load [< nume fișier >] [< listă variabile >]

Astfel, dacă nu este menționat numele fișierului (care nu trebuie să conțină și extensia), variabilele sunt preluate din fișierul **matlab.mat**. De asemenea, dacă nu este precizată lista de variabile, sunt încărcate toate variabilele din fișierul precizat;

Instrucțiunea **clear** se utilizează pentru eliminarea (ștergerea) unor variabile rezidente în memorie și prezintă următoarea sintaxă:

clear [<listă de variabile>]

Efectul instrucțiunii constă în eliminarea definitivă din spațiul de lucru a variabilelor cuprinse în listă. Dacă lipsește o listă explicită de variabile, sunt eliminate toate variabilele existente în memorie la un moment dat. Precizăm că se recomandă ca, la începutul unui program MATLAB (fișier de tip *.m), să fie prezentă instrucțiunea **clear**, pentru a se crea spațiul necesar în memorie pentru variabilele din program;

Instrucțiunile **dir**, **delete**, **type** au ca rezultat respectiv listarea numelui tuturor fișierelor din directorul curent, ștergerea unui fișier (al cărui nume plus extensie trebuie menționat după **delete**) și tipărirea la imprimantă a unui fișier (al cărui nume trebuie, de asemenea, menționat). Mai precizăm că, ne aflăm în ambient MATLAB, este posibilă lansarea în execuție a unui program extern sau a unei instrucțiuni MS-DOS, dacă se tastează, înainte de numele acesteia, simbolul „!”.

Structuri fundamentale de program. Produsul MATLAB este prevăzut cu toate instrucțiunile necesare realizării oricărei structuri fundamentale de program (secvență, selecție, iterație), fiind posibilă codificarea, ca în orice limbaj de programare clasic, a oricărui algoritm de calcul.

Astfel, există instrucțiunea **for**, destinată unui ciclu (bucă) de instrucțiuni de un număr predeterminat de ori, instrucțiunea **while**, pentru repetarea unui ciclu de instrucțiuni un număr indefinit de ori, sub controlul unei condiții logice, instrucțiunea **if...elseif...else** pentru realizarea unei selecții pe baza unui număr oricât de mare de condiții logice, și instrucțiunea **break** care realizează ieșirea automată dintr-un ciclu de instrucțiuni:

Instrucțiunea **for** permite, după cum s-a precizat, repetarea unui ciclu de instrucțiuni de un număr finit, predeterminat, de ori. Este echivalentul instrucțiunii similare din **PASCAL** sau al instrucțiunii **DO** din **FORTRAN** și are următoarea sintaxă:

for v = < expresie > ,
< secvență de instrucțiuni > ;
end,

În această formulare, **expresie** este, de fapt, o matrice, care, de cele mai multe ori, se reduce la un vector. Coloanele acestei matrice (sau elementele acestui vector) sunt atribuite pe rând variabilei **v** (care poate deci fi un vector coloană sau un scalar), secvența de instrucțiuni din interiorul buclei fiind apoi reluată pentru fiecare valoare atribuită lui **v**. Pentru a obține, în

particular, repetarea secvenței de **n** ori, **expresie** trebuie să fie un vector ce cuprinde numerele naturale de la 1 la n.

Instrucțiunea **if..elseif..else** permite realizarea de selecții conform următoarei sintaxe:

```
if < expresie logică 1 > ,  
    < secvență de instrucțiuni 1 > ;  
[ elseif < expresie logică 2 > , ]  
    [ < secvență de instrucțiuni 2 > ; ]  
[ else < expresie logică 3 > , ]  
    [ < secvență de instrucțiuni 3 > ; ]  
end,
```

În formularea de mai sus, rândurile cuprinse între paranteze drepte pot lipsi, iar efectul acestei instrucțiuni constă în executarea acelei secvențe de instrucțiuni (1,2 sau 3) care corespunde expresiei logice care este adevărată. În plus, se pot include mai mult de trei condiții (alternative), prin repetarea cuvântului-cheie **elseif** de un număr oarecare de ori.

Constante MATLAB. Definirea variabilelor. Instrucțiuni de atribuire

Constante MATLAB. În ambientul MATLAB se pot utiliza următoarele tipuri de constante: reale, complexe și de tip text (șir de caractere). Primele două tipuri pot alcătui vectori și matrice. Pentru scrierea constantelor complexe este definită, în MATLAB, unitatea imaginară **i**.

În continuare, vom face câteva precizări referitoare la constantele reale. Acestea pot fi scrise folosind notația zecimală convențională, în care punctul zecimal poate lipsi, deoarece nu se face nici o distincție între constante reale și întregi. Se poate de asemenea utiliza scrierea exponențială în puterile lui zece. Prezentăm câteva exemple de scriere corectă:

3	-87	.01	9,2568	1.325e-10	6.235e25	1.235E+8
----------	------------	------------	---------------	------------------	-----------------	-----------------

În ceea ce privește precizia, MATLAB lucrează cu numere având 16 cifre semnificative la partea zecimală. Numerele reale pot fi cuprinse între 10^{-308} și 10^{308} . În plus, în MATLAB există și variabile permanente **Inf** (infinit) și **NaN** (ce corespunde nedeterminării 0/0 sau **Inf/Inf**). Prezența acestor variabile permite continuarea calculelor chiar în situația în care apare o împărțire cu zero sau o nedeterminare, programul avertizând însă asupra apariției unor astfel de situații;

Instrucțiunea **format** este destinată fixării formatului de scriere (afișare pe ecran) a constantelor și variabilelor reale. Formatul de scriere poate fi schimbat în orice moment în timpul unei sesiuni de lucru MATLAB. Sintaxa acestei instrucțiuni este următoarea:

format < specificator de format >

Specificatorii de format sunt următorii:

short pentru scriere în virgula mobilă, cu cinci cifre semnificative. Acesta este formatul de scriere adoptat automat, fără a fi necesar utilizarea instrucțiunii **format**

short e pentru scriere exponențială cu cinci cifre semnificative, adoptată automat în cazul în care numărul este prea mare pentru a fi scris în format short

long pentru scriere în virgula mobilă, cu 16 cifre semnificative,

long e pentru scriere exponențială cu 16 cifre semnificative,

hex, pentru scriere în sistemul hexazecimal.

Definirea variabilelor și instrucțiuni de atribuire. În cele ce urmează, vom prezenta modul de definire a variabilelor reale (scalari, vectori, matrice). Acestea pot fi definite practic în două moduri:

- ca listă explicită, precizându-se direct constanta atribuită fiecărei variabile;
- prin intermediul unor instrucțiuni de atribuire, în care unei variabile **i** se atribuie o expresie, care este compusă din constante (scalari, matrice), nume de alte variabile, funcții, operatori și alte caractere speciale.

Precizăm că nu există declarații de tip sau dimensiune în MATLAB. Spațiul necesar în memorie pentru fiecare variabilă, indiferent de tipul acesteia, este alocat automat până la nivelul

disponibil pentru calculatorul utilizat. Acest fapt simplifică mult programele scrise în limbajul MATLAB și constituie una din principalele facilități ale produsului.

Instrucțiunea de atribuire în MATLAB are următoarea sintaxă:

< nume variabilă > = < expresie > [;]

Dacă lipsește simbolul final “ ; ” rezultatul evaluării expresiei (valoarea efectivă a variabilei) este afișată pe ecran, iar în caz contrar acest rezultat este doar reținut în memorie.

În ceea ce privește scrierea explicită a unei matrice sau vector, aceasta se realizează prin înșirarea elementelor sale între paranteze drepte, cu spații între ele. Matricele se scriu linie cu linie, două linii succesive fiind delimitate de simbolul “ ; ”. Prezentăm mai jos un exemplu de generare a unei matrice cu două rânduri și trei coloane:

A = [23.1 .36 58 ; 210 3.69 .236] ;

De asemenea, o matrice sau un vector pot fi definiți element cu element, fiecăruia atribuindu-i-se o expresie sau constantă, ca în următorul exemplu:

A(i,j) = 2*i +6/j ;

Un caz foarte des întâlnit este cel al generării unui vector ale cărui elemente sunt primele n numere naturale ordonate crescător. Astfel, un vector care are drept elemente numerele de la 1 la 5 se obține cu ajutorul următoarei instrucțiuni de atribuire:

x = 1:5;

Instrucțiunea de mai sus este un caz particular, însă des întâlnit, al unei instrucțiuni a cărei sintaxă este următoarea:

< nume vector > = < element inițial > : [< pas >] : < element final >;

Această instrucțiune generează un vector pentru care se dau primul și ultimul element (sub forma unor numere reale), precum și pasul dintre două elemente succesive. Dacă pasul lipsește, el este considerat ca fiind egal cu unitatea.

Funcții speciale de intrare/ieșire. Pentru început precizăm că aceste funcții, destinate importării și exportării de date au fost preluate, cu mici modificări, din limbajul C. În continuare, vom prezenta doar două dintre aceste funcții, care vor fi utilizate în cadrul aplicațiilor prezentate în capitolele 2 ÷ 8:

Funcția input este folosită pentru introducerea datelor în mod interactiv, în cadrul unui program MATLAB. Sintaxa acestei funcții este următoarea:

< nume variabilă > = input ('< șir de caractere >')

În momentul execuției, șirul de caractere menționat este afișat pe ecran, după care se așteaptă introducerea unei constante numerice sau de tip șir de caractere, care va fi apoi atribuită variabilei al cărei nume a fost precizat.

Funcția fprintf este utilizată pentru convertirea unor date numerice în șiruri de caractere și afișarea lor pe ecran sau introducerea lor într-un fișier date. În cazul în care se urmărește afișarea pe ecran a valorii unei variabile, această funcție are următoarea sintaxă:

fprintf ('< format > ', <nume variabilă >)

Formatul de scriere este precizat de un șir de caractere de control ce conține specificații de conversie (care controlează modul de scriere și sunt precedați de simbolul %) și text care se copiază direct pe ecran. În text poate fi inclus și specificatorul pentru rând nou: \n . Dintre specificații de conversie, menționăm %e pentru scriere cu exponenți și %f pentru scriere cu virgula fixă.

Operații și operatori MATLAB

Operații cu matrice și scalari. Operatorii matematici fundamentali utilizați în limbajul MATLAB sunt următorii:

, realizează transpunerea unei matrice sau vector (transformarea unui vector linie în vector coloană sau invers);

+ realizează suma a doi scalari, vectori sau matrice;

– realizează diferența între doi scalari, vectori sau matrice;

* realizează produsul a doi scalari, produsul între un scalar și un vector sau matrice sau produsul matriceal a două matrice sau vectori (cu respectarea regulilor cunoscute privind dimensiunile matricelor care se înmulțesc);

*. realizează produsul, element cu element, a două matrice sau vectori de dimensiuni identice;

./\ realizează împărțirea, la stânga sau la dreapta, a doi scalari, vectori sau matrice. Pentru aceștia din urmă, această împărțire se face în sens matriceal, astfel:

dacă $X * A = B$, vectorul X se poate calcula cu relația: $X = B/A$,

dacă $A * X = B$, vectorul X se poate calcula astfel: $X = A/B$, cu condiția ca matricea A să fie nesingulară (să aibă o inversă);

./ realizează împărțirea, element cu element, a două matrice sau vectori de dimensiuni identice;

^ realizează ridicarea la putere a unui scalar sau a unei matrice pătrate (prin înmulțirea cu ea însăși în sens matriceal);

.^ realizează ridicarea la putere, element cu element, a unui vector sau matrice de dimensiuni oarecare.

Cu ajutorul acestor operatori, se pot realiza atât operații de calcul matriceal cât și operații de calcul pentru fiecare element al unei matrice. Operatorii / și \ fac posibilă rezolvarea rapidă a oricărui sistem de ecuații liniare, care poate fi ușor transpus sub formă matriceală.

Operatori logici și de relație. În MATLAB, se folosesc următorii operatori logici: & operator ȘI, | operator SAU, ~ operator NU (negație). Cu ajutorul acestor operatori, se pot construi expresii logice, a căror valoare poate fi 0 (fals) sau 1 (adevărat).

În ceea ce privește operatorii de relație, aceștia sunt cei clasici: >, >=, <, <=, == (de egalitate), ~= (diferit).

De asemenea, în MATLAB există toate funcțiile matematice uzuale (exponențiale, trigonometrice, logaritmice etc.) sau speciale. Aceste funcții au și variante aplicabile, element cu element, unor matrice.

Funcții speciale MATLAB. Reprezentarea polinoamelor.

Produsul MATLAB este prevăzut cu o serie întreagă de funcții speciale. Astfel, amintim funcțiile pentru prelucrări statistice, între care menționăm: **max** (pentru determinarea valorii maxime), **min** (determinarea valorii minime), **mean** (calculul valorii medii), **std** (calculul abaterii).

În continuare, vor fi prezentate întâi unele funcții aplicabile matricelor și vectorilor, iar în continuare o serie de funcții aplicabile polinoamelor, reprezentate prin vectori.

Funcții speciale aplicate matricelor. Cele mai importante sunt:

Funcția **leng** care este destinată determinării lungimii unui vector. Ca urmare, argumentul acestei funcții trebuie să fie un vector. Prin aplicarea acestei funcții, rezultă un număr natural care reprezintă numărul de elemente al vectorului (linie sau coloană) considerat. Acest număr poate fi eventual atribuit unei variabile reale.

Funcția **size** se utilizează pentru determinarea dimensiunilor unei matrice și deci trebuie să aibă argument o matrice. Prin aplicarea acestei funcții, se indică două numere naturale, care reprezintă dimensiunile matricei considerate (numărul de linii și respectiv de coloane). Cele două numere pot fi atribuite unor variabile astfel:

$$[m,n] = \text{size}(X)$$

unde **m** este numărul de linii al matricei **X**, iar **n** numărul de coloane al aceleiași matrice.

Funcții **inv** și **eig** realizează respectiv inversarea unei matrice pentru care o astfel de operație este posibilă și calculul valorilor proprii ale unei matrice. Funcția **eig** permite și calculul vectorilor proprii, dacă se utilizează sub forma:

$$[X, D] = \text{eig}(A)$$

unde **X** este matricea vectorilor proprii (așezați pe coloane) ai matricei **A**, iar **D** este matricea diagonală a valorilor proprii.

Reprezentarea polinoamelor prin vectori. Polinoamele sunt reprezentate în MATLAB prin vectori linie care conțin coeficienții în ordinea descrescătoare a puterilor. De exemplu, polinomul $p = 3x^3 + 2x^2 - x + 6$ este reprezentat prin vectorul $\mathbf{p} = [3 \ 2 \ -1 \ 6]$. În continuare, vor fi descrise o serie de funcții aplicabile polinoamelor reprezentate sub această formă vectorială:

Funcția **polyder** simulează operația de derivare a unui polinom. Astfel, fiind dat un polinom reprezentat prin vectorul coeficienților săi, această funcție generează vectorul coeficienților polinomului obținut prin derivarea celui dat;

Funcția **roots** calculează rădăcinile (zerourile) unui polinom reprezentat sub forma vectorului coeficienților. Rezultatul este reprezentat sub forma unui vector coloană, în care rădăcinile sunt ordonate în ordine strict descrescătoare. Dacă se dorește realizarea operației inverse (calculul vectorului coeficienților polinomului fiind dat vectorul rădăcinilor sale), se aplică funcția **poly** ultimului vector. Dacă însă se aplică funcția **poly** unei matrice pătrate, se obține vectorul coeficienților ecuației caracteristice a matricei respective;

Funcția **polyval** realizează evaluarea unui polinom în unul sau mai multe puncte (unui punct îi corespunde practic un număr real). Astfel, dacă **p** este vectorul coeficienților polinomului, iar **s** punctul în care se dorește evaluarea, se utilizează instrucțiunea:

$$\mathbf{val} = \mathbf{polyval}(\mathbf{p}, \mathbf{s})$$

unde **val** este valoarea polinomului în punctul **s**. Dacă **s** este un vector sau o matrice, **val** este la rândul său un vector sau o matrice cuprinzând valoarea polinomului **p** pentru fiecare din punctele cuprinse în vectorul sau matricea **s**.

Facilități de grafică MATLAB

Realizarea de grafice bidimensionale. Graficele se obțin utilizând următoarele instrucțiuni:

Instrucțiunea **plot** realizează grafice bidimensionale. Astfel, dacă **Z** este un vector, instrucțiunea **plot (Z)** construiește o reprezentare grafică în plan a elementelor vectorului considerat în funcție de indexul lor (numărul de ordine). Acest grafic va fi vizibil pe ecran într-o nouă fereastră (cea „grafică”), independentă de fereastra sesiunii de lucru MATLAB. De asemenea, scara graficului este fixată automat, astfel ca toate datele să fie vizibile. Există însă și posibilitatea alegerii scării de către utilizator, folosind instrucțiunea **axis**.

Dacă **X** și **Y** sunt doi vectori având, în mod obligatoriu, aceeași lungime, atunci instrucțiunea **plot (X, Y)** realizează reprezentarea grafică liniară bidimensională a dependenței elementelor vectorului **Y** de elementele vectorului **X**. De cele mai multe ori, **X** reprezintă vectorul punctelor în care se dorește evaluarea unei funcții, iar **Y** conține valorile funcției considerate pentru fiecare punct al vectorului **X**. Dacă vectorii **X** sau **Y** (sau doar unul dintre aceștia) sunt înlocuiți cu matrice de aceleași dimensiuni, instrucțiunea de mai sus realizează un grafic cu mai multe linii, fiecare reprezentând dependența dintre o coloană a lui **Y** și coloana corespunzătoare a lui **X**.

În sfârșit, dacă se consideră mai multe perechi de vectori, **X1, Y1, ..., Xn, Yn**, instrucțiunea **plot(X1, Y1, X2, Y2, ..., Xn, Yn)** realizează un grafic cu linii multiple, fiecare linie reprezentând dependența dintre vectorii uneia din cele **n** perechi, vectori care trebuie să aibă aceeași lungime. În graficele cu linii multiple, în lipsa unor indicații suplimentare, fiecare linie este reprezentată cu altă culoare. Există posibilitatea, pentru fiecare linie (curbă) a graficului să se aleagă o anumită culoare și un anumit tip de linie;

Culori și tipuri de linii. În versiunea MATLAB 5.0, este posibilă utilizarea următoarelor culori: galben (cu simbolul **y**), magenta (simbol **m**), cyan (**c**), roșu (**r**), verde (**g**), albastru (**b**), alb (**w**), negru (**k**). În ceea ce privește tipurile de linii disponibile, acestea sunt următoarele: linie continuă (simbol -), linie întreruptă (--), linie-punct (-.), și linie punctată (:). De asemenea, este

posibilă realizarea unor reprezentări grafice prin puncte (în loc de linii continue), puncte care pot fi marcate cu unul din următoarele simboluri: ., +, *, o, x.

În vederea stabilirii, pentru o anumită curbă, a tipului de linie și a culorii dorite, se introduc, în cadrul instrucțiunii **plot**, între ghilimele simple, simbolurile pentru tipul de linie și pentru culoare, ca în exemplul de mai jos:

plot (X1, Y1, '-r', X1, Y2, '--b', X2, Y2, '+c')

Instrucțiuni pentru inscripționarea graficului (**xlabel**, **ylabel**, **title**, **grid**, **text**). Odată realizate, graficele MTLAB pot fi prevăzute atât cu o grilă (caroiaj), alcătuită din linii punctate și obținută cu ajutorul instrucțiunii **grid**, aplicată după instrucțiunea **plot**, cât și cu linii de text. Astfel, pentru scrierea de explicații pe cele două axe ale graficului se folosesc instrucțiunile **xlabel** (pentru axa absciselor) și **ylabel** (pentru axa ordonatelor), iar pentru scrierea unui titlu al graficului (având însă un singur rând) instrucțiunea **title**. Aceste trei instrucțiuni vor fi obligatoriu urmate de textul (variabila de tip șir de caractere) ce se dorește a fi aplicat pe grafic, introdus între ghilimele simple și apoi între paranteze rotunde.

Dacă este necesară scrierea unui rând de text în interiorul graficului, pornind din punctul de coordonate x,y (exprimate fie prin valoarea lor numerică, fie prin elemente ale vectorilor ce au fost reprezentați grafic), se utilizează instrucțiunea **text** sub forma:

text (< coordonata x >, < coordonata y > , '< șir de caractere >')

Alte tipuri de grafice bidimensionale. În afara graficelor obișnuite (liniare), se pot realiza și grafice având scări logaritmice. Astfel, instrucțiunea **semilogx** produce o reprezentare grafică cu scara logaritmică pe axa absciselor, **semilogy** realizează un grafic cu scara logaritmică pe axa ordonatelor, iar **loglog** un grafic având scări logaritmice pe ambele axe. De asemenea, există instrucțiuni pentru realizarea de grafice în coordonate polare (**polar**) , a unor grafice de tip cu bară (**bar**) sau a unor histograme (**hist**). În sfârșit, menționăm și instrucțiunea **fill**, care realizează „umplerea” (colorarea la interior) a unei arii poligonale oarecare.

Realizarea de grafice tridimensionale. Dintre instrucțiunile destinate realizării unor grafice tridimensionale (în spațiu), vom prezenta succint doar pe cele mai utilizate: **plot3** și **mesh**.

Instrucțiunea **plot3** este echivalentul tridimensional al instrucțiunii **plot**. Astfel, fiind dați trei vectori de aceeași lungime, **x**, **y**, **z**, instrucțiunea **plot3 (x, y, z)** realizează o reprezentare tridimensională a dependenței dintre cei trei vectori, sub forma unei curbe în spațiu. Graficele tridimensionale obținute cu această instrucțiune pot fi inscripționate și utilizează aceleași tipuri de linii și culori ca și cele obținute prin aplicarea instrucțiunii **plot**.

Instrucțiunea **mesh** realizează o reprezentare tridimensională (în spațiu, 3D) a unei suprafețe, sub forma unei rețele (grile) de curbe. Practic, fiind dată o matrice **Z**, ale cărei elemente reprezintă cotele suprafeței ce se dorește a fi reprezentată, instrucțiunea **mesh(Z)** realizează o perspectivă tridimensională a elementelor matricei **Z**.

9.3.2. Produsul informatic ANSYS EMAG

Pe piața „software”-ului de aplicație în domeniul dispozitivelor electromagnetice există o mare diversitate de programe de simulare, atât în domeniul frecvențelor joase cât și al celor înalte (în radiofrecvență). În ultimii ani, după anul 1996, au fost introduse zeci de pachete „software” performante pentru simularea numerică a dispozitivelor electromagnetice. Înainte de apariția acestor pachete de programe–produs, simularea câmpurilor electromagnetice și termice era apanajul unui număr redus de ingineri care aveau cunoștințe atât de analiză numerică și simulare, cât și de programare a calculatoarelor. Astăzi, la începutul mileniului trei, un inginer electrician –fără să cunoască detalii privind simularea numerică și programarea calculatoarelor– poate, în câteva ore, să utilizeze pachete ultrasofisticate. Desigur că rezultatele bune se pot obține când utilizatorul unui produs CAD are cunoștințe de bază privind metodele de simulare și performanțele relative ale acestora, fiindu-i suficiente numai noțiunile prezentate în acest compendium (!); în plus, astăzi există un număr mare de pagini WEB pe Internet (unde poate fi

găsit și acest manual), care dau descrieri sumare (mai mult sau mai puțin cu caracter promoțional) despre tehnicile cele mai utilizate în diversele produse informatice de firmă disponibile pe piața „software”.

La alegerea produsului informatic de aplicație, inginerul proiectant, în afara trăsăturilor generale (cum ar fi resursele „hardware” necesare, costul și domeniul de aplicație) trebuie să țină seama și de trăsături speciale care au un impact profund într-o alegere adecvată. Aceste trăsături speciale se referă la: *modulele principale ale programului și modul în care interacționează, ușurința cu care se construiește modelul de intrare, algoritmi utilizați și libertatea utilizatorului de a-i manipula*, precum și *flexibilitatea prezentării rezultatelor*.

Nu este posibil ca –în volumul redus de care poate dispune un compendiu– să se prezinte diversitatea (mare) a problemelor care apar în practica ingineriei electrice, însă o clasificare după anumite criterii (atât de necesară) poate fi realizată și aici. Astfel, dacă se ia numai criteriul domeniului frecvențelor, instrumentele informatice existente pentru simularea numerică prin metoda elementului finit a problemelor de câmp electromagnetic se pot identifica trei clase mari de aplicații:

- pentru dispozitivele electromagnetice de joasă frecvență (industrială);
- pentru dispozitivele electronice de bandă largă;
- pentru dispozitivele electromagnetice de înaltă frecvență.

Prima clasă include dispozitive, aparate, mașini etc. care lucrează în câmp electromagnetic alternativ de joasă frecvență așa cum sunt transformatoarele electrice, motoarele asincrone, electromagneții etc. Pentru această clasă există o mare diversitate de pachete de programe – produs bazate pe metoda elementului finit în spațiul 2D (plan) și 3D (simulare tridimensională). Aceste pachete sunt utilizate pentru: calculul câmpului electric, magnetic și termic, calculul forțelor (cuplurilor de forțe), calculul impedanțelor, calculul pierderilor în fier și transformarea lor în căldură etc. Firmele producătoare pentru astfel de programe sunt: Ansoft (Pittsburgh, SUA), Magsoft (New York, SUA), Infolytica (Montreal, Canada), Ansys, Inc. (SUA) ș.a.

Clasa pentru dispozitivele electronice de bandă largă se referă –în special– la componentele calculatoarelor electronice. Instrumentele informatice pentru această clasă permit predicția integrității semnalului electric, a discuției încrucișate (așa-zisul „cross – talk”) etc. Firmele cele mai cunoscute, producătoare de pachete de programe pentru simularea numerică în acest domeniu sunt: Sonnet Software, Bay Technology și Hewlett Packard (California, SUA).

A treia clasă de aplicații se referă la dispozitivele electromagnetice de radio frecvență și microunde așa cum sunt: antenele, sistemele radar și ghidurile de undă. Câteva produse CAD pentru această clasă sunt disponibile la firmele: Ansoft, Vector Fields, Hewlett Packard, Electromagnetic Applications (Colorado, SUA).

Dezvoltarea modulară a instrumentelor informatice. Programele de simulare a câmpurilor electromagnetice și termice prin metoda elementului finit trebuie să realizeze o echilibrare între general și particular, în sensul că un produs informatic este necesar să asigure o simulare numerică pe modele generale (pentru a putea fi utilizat într-o clasă largă de probleme), care –însă– prin anumite specificații particulare să permită utilizarea produsului la aplicații concrete – individualizate. Acest deziderat se obține prin crearea de programe formate din numeroase module (de caz general) interconectabile între ele după o „schemă” impusă de o anumită aplicație particulară, concret formulată.

Astfel, dezvoltarea modulară a produselor „software” pentru simularea numerică a sistemelor cu parametri distribuiți, folosind metoda elementului finit, coincide într-un fel cu structura generală a algoritmului de rezolvare a unei probleme de câmp prin această metodă. După cum s-a arătat (în § 9.2.4), metoda elementului finit implică trei faze distincte:

1. realizarea modelului geometric al obiectului fizic studiat și generarea rețelei de discretizare, ceea ce în „plan computațional” (al produsului CAD) se cheamă *preprocesare*;

2. rezolvarea (găsirea soluției) modelului numeric obținut în faza precedentă printr-o discretizare spațio-temporală, care se reflectă în produsul informatic prin faza de *procesare* (calcul automat) și

3. transformarea rezultatelor (datelor) numerice obținute în faza precedentă în informații, adică în noțiuni și mărimi cu semnificație inginerescă concretă, ceea ce în produsul informatic se realizează prin așa – numita *postprocesare*.

Modulele „software” legate de procesare și postprocesare sunt specifice aplicației concrete sau unei clase de probleme similare; modulele „software” legate de preprocesare au un caracter mult mai general. Firmele (cele mai multe) care realizează produse CAD, ce merg pe linia dezvoltării modulare oferă utilizatorului o gamă largă de module care pot fi livrate separat, în funcție de aplicație (magnetism, electrostatică, electrocinetică, rezistența materialelor, căldură etc.). Acesta este și cazul produsului ANSYS, la care ne vom referi în final.

Preprocesarea este faza în care utilizatorul produselor CAD destinate câmpurilor electromagnetice și/sau termice specifică toate datele necesare obținerii modelului în mod interactiv. Astfel, utilizatorul selectează sistemul de coordonate și tipurile de elemente, definește geometria sistemului fizic și sursele de câmp, structura dispozitivului (corpului) electromagnetic și proprietățile de material (de exemplu, prin curba de magnetizare a materialului folosit), creează modele și rețele de discretizare, modifică rețeaua și definește restricțiile problemei, precum și ecuațiile de cuplare. Rețeaua de discretizare spațială poate conține elemente de același tip (triunghi, dreptunghi, tetraedru, prismă, piramidă etc.) sau de tipuri diferite. Construirea rețelei de discretizare a domeniului spațial este o problemă critică, în sensul că rezultatele simulării ca și eficiența calculului depind de *numărul și tipul elementelor*. De aceea, preprocesoarele moderne realizează rețeaua în mod automat sau semiautomat / interactiv (utilizatorul apreciază fie frontiera domeniului spațial și interferențele între elementele structurale ale dispozitivului electromagnetic, fie utilizatorul generează manual o rețea inițială grosieră după care are loc o rafinare automată). Multe produse-program folosesc *generatoare de rețele adaptive*; aceste produse ajustează automat mărimea elementelor finite și numărul lor, în funcție de gradientul câmpului sau a violării anumitor restricții fizice (de exemplu, discontinuitatea unei variabile de câmp). Generarea adaptivă este un procedeu iterativ, numărul de iterații fiind impus de un criteriu de eroare dat apriori.

Tot în faza de preprocesare sunt estimate dimensiunile fișierelor de date și necesarul de memorie pentru rularea programului în vederea obținerii soluției. Datele furnizate de utilizator în faza de preprocesare devin parte integrantă a bazei de date a produsului CAD. Baza de date este organizată în tablouri ce conțin coordonatele nodurilor, topologia rețelei, proprietățile de material și caracteristicile la nivel de element.

Produsele-program actuale operează cu anumite concepte care tind spre standardizare. Astfel, în modelarea corpurilor se utilizează noțiuni ca: noduri (puncte, vârfuri), linii, arii și volume. Nodurile sunt utilizate pentru localizarea elementelor în spațiu, iar elementele definesc conectivitatea modelului (v. § 9.2.4). Modelul de element finit este definit prin specificarea nodurilor rețelei și a atributelor elementelor (mărime, formă, conectivitate). Alte entități ca: zonă, subrețea, rotire, focalizare ș.m.a. permit o localizare, o reprezentare grafică pe „display” și o prelucrare eficientă folosindu-se dispozitive de intrare adecvate (ca „mouse”, „trackball” etc.). Anumite comenzi ale preprocesorului (care este un modul inițial de program) permit utilizatorului să aranjeze graficul (desenul) pe „display” în forma dorită (mărime / scară, axe, rotire, vedere etc.), să-l copieze pe un anumit suport, să scaleze sau să extindă, prin simetrie, un model numeric inițial.

Procesorul este un al doilea modul al produsului-program care realizează *soluționarea* modelului (problemei) numeric. Procesarea implică generarea modelului numeric elementar și global (v. § 9.2.4) pornind de la baza de date creată de preprocesor și soluționarea acestui model. Există o mare diversitate de algoritmi numerici ca: metoda elementelor de frontieră (BEM, de la „Boundary Element Method”), metoda momentului (MM – „Moment Method”) etc. Totuși,

indiferent de metoda folosită, utilizatorul trebuie să specifice pentru problema analizată: regimul de lucru (static, staționar, nestaționar, dinamic ș.a), tipul sistemului (liniar sau nelinier), sistemul de coordonate utilizat, metoda de soluționare (directă sau iterativă și în acest caz criteriul de calitate etc.).

Unele procesoare CAD folosesc în faza de soluționare module de rezolvare a modelelor numerice rezultate în faza de asamblare, module numite „solver”-e, care –în general– se bazează pe calculul matriceal.

Postprocesarea asigură interpretarea fizică – inginerească a soluției numerice dată de procesor, ceea ce înseamnă –în general– reprezentarea grafică a soluției (pe corp), cu indicații numerice și cromatice.

Postprocesoarele actuale afișează rezultatul simulării în spațiul dorit (2D sau 3D), cu trasarea liniilor de câmp (de flux unitar), a liniilor (suprafețelor) echiscale etc. Simulatoarele performante (din clasa a treia de frecvențe) pot prezenta rezultatele în formă animată (de exemplu propagarea undelor electromagnetice produse de o antenă). Rezultatele pot fi și stocate (memorate) în diverse fișiere pentru o prelucrare ulterioară „off-line”.

Pachetul de programe ANSYS. Este un produs informatic destinat anume analizei sistemelor prin metoda elementului finit, care s-a remarcat prin performanțe deosebite (precizie, versatilitate, ușurința utilizării) de simulare a proceselor modelate prin ecuațiile fizicii matematice și în special a proceselor electromagnetice. Dintre toate variantele posibile, am ales pentru o sumară descriere aici, produsul ANSYS – 386 ED („Educational”), atât din motive didactice, cât și pentru faptul că se află implementat în rețeaua de calculatoare a catedrei „Electrotehnică – Electronică” din U.P.G. Ploiești (sub licență). Pentru învățarea utilizării programului ANSYS în mod eficient, cititorul va trebui să facă apel permanent la comanda help pentru a obține „on – line” descrierea amănunțită a tuturor comenzilor programului, pe care le va putea studia direct pe ecranul monitorului calculatorului, dar și separat pe un „out – print” listat în prealabil.

Produsul ANSYS a fost lansat de firma americană Swanson Analysis Systems, Inc. (Houston, USA), încă din anul 1970, de către dr. John Swanson, numele acestui pachet de programe fiind o abreviere a cuvintelor „ANalysis SYStems”. De la lansare și până acum (ne referim la anul 2002), produsul a fost permanent îmbunătățit pentru a putea face față oricăror aplicații de simulare prin metoda elementului finit, existând acum numeroase variante specializate.

Programul ANSYS complet are toate modulele necesare utilizării generale a metodei elementului finit, pentru orice structuri liniare și neliniare. Cu el se pot efectua: analize de sistem în regim static, dinamic, tranzitoriu și de transfer (ca, de exemplu, răspunsul unui etaj electronic la semnale armonice); studiul proceselor termice, magnetice, fluidice (hidraulice și pneumatice), termoelectrice și acustice; modelarea numerică și simularea solidelor, precum și optimizarea proiectării.

Familia programelor modularizate *ANSYS – PC*, executabile pe calculatoare personale, are disponibile următoarele module: *ANSYS – PC / LINEAR* (utilizat în special pentru rezolvarea problemelor de flambaj al sistemelor liniare), *ANSYS – PC / THERMAL*, *ANSYS – PC / SOLID* (utilizat în faza de preprocesor cu un preprocesor general PREP 7 – pentru modelarea numerică a solidelor în aplicațiile cu *PC / LINEAR* și *PC / THERMAL*), *ANSYS – PC / OPT* (pentru optimizarea proiectării cu produsele *PC / LINEAR* și *PC / THERMAL*).

Dintre ultimele instrumente ANSYS sunt de evidențiat: *ANSYS / Emag 5.2* (pentru analiză și proiectare în domeniul electromagnetic), *ANSYS / Emag & FLOTRAN* (o asociere a modulelor ANSYS / Emag cu modulele FLOTRAN de procesare prin folosirea ecuațiilor Navier – Stokes în problemele cu simetrie axială, cu folosirea a 141 elemente finite) *ANSYS / ED™* (produs educațional, pentru mediile universitare, de simulare pe sisteme PC compatibile IBM, sub Windows sau UNIX), *ANSYS / Structural* (pentru analiza structurală în mecanică), *ANSYS / Mechanical* (un produs CAD / CAE în domeniul ingineriei mecanice, al rezistenței materialelor,

al plasticității etc.), *ANSYS / LS – DYN™* (dotat cu un „Explicit Dynamic Solver” pentru mai toate domeniile de activitate de la cel electric la cele biomedicale, componente electronice, plasticitate, simularea ruperii etc.).

Toate produsele ANSYS realizează simularea numerică prin metoda elementului finit cu parcurgerea celor trei faze „clasice”: preprocesarea (care se face cu un preprocesor denumit PREP 7 de tip general automat), procesarea (soluționarea) și postprocesarea (care se face cu modulele POST 1 și / sau POST 26, prin care rezultatele sunt redată grafic –în 2D sau 3D– și/sau prin tabele cu date, cu listare, redare cu „digital incremental plotter”, în culori, cu animație pe ecranul monitorului etc.).

Produsul ANSYS / Emag ED™. Este varianta „Educational” a produsului cu aplicații ingineresti – industriale „Electromagnetic (field)”, cu absolut toate facilitățile și performanțele acestuia, dar pentru dimensiuni mult reduse (număr maxim de noduri 500), pe care îl prezentăm aici datorită calităților sale educaționale (de instruire – învățare) și de care dispune (prin licență) catedra „Electrotehnică – Electronică” din U.P.G. Ploiești, în laboratorul său de „Tehnologia predării”.

Resursele instrumentului informației ANSYS / Emag ED™ sunt:

- numărul maxim de domenii (subdiviziuni structurale) în care poate fi divizat câmpul Ω este „Domain Of Field” $DOF=1000$;

- noduri 500;
- numărul maxim de elemente (triunghiulare) 250;
- numărul maxim de elemente structurale $p=50$;
- numărul maxim al elementelor principale $DOF_{max}=50$;
- numărul maxim al punctelor cheie $k=100$;
- numărul maxim al liniilor $l=100$;
- numărul maxim al suprafețelor $A=50$;
- numărul maxim al volumelor $V=100$.

Produsul cu performanțele dimensionale arătate anterior necesită o platformă „hardware” de tip PC 486 Intel sau Pentium cu resursele: 16 MB RAM („Real Memory”), 125 MB pe „Disk Space Full Install”, 42 MB de „Disk Space Less Doc. Files”, 120 MB de „Disk Space for Swap”, instrumentele de grafică ale sistemului Windows 95 sau Windows NT cu un monitor / „display” cu o rezoluție minimă de 1024×768 picșeli și un sistem de operare Windows 95, Windows NT3.5 sau superior.

Capabilitățile („Capabilities”) produsului ANSYS / Emag™ sunt cele legate de aplicarea performantă, automată, a analizei structurilor electromagnetice prin metoda elementului finit care –în esență– constau în:

- caracteristicile produsului: transfer de date prin sistemul IGES, modelarea (numerică) a corpului solid (discretizarea solidelor), optimizarea proiectării (de exemplu, proiectarea unui ventil electromagnetic prin minimizarea energiei $\int_{\Omega} \frac{1}{2} BH \cdot d\Omega$), reprezentări grafice în 2D și în 3D, animația imaginii, colorarea zonelor și a conturilor, modelarea proceselor neliniare, algoritmi de rezolvare a sistemului de ecuații modale prin metoda Newton – Raphson (pentru

ecuațiile liniare) și algoritmi de calcul pe porțiuni liniarizate de curbă (a ecuațiilor modale neliniare), calcule magnetice de grafică (cu redarea cantitativă a spectrelor de câmp în 2D sau 3D), generator automat adaptiv pentru modelarea cu elemente finite, calcule de transfer termic, calcule dinamice și multe altele;

- interfețe pentru utilizarea produsului: interferența de grafică IGU (de la „Intuitive Graphical User”) de tip intuitiv – interactiv, documentația „on – line” prin help cu „hypertext links”, instalator / aranjor de meniuri, blocuri de dialog, hărți / tabele funcționale pentru comenzi;

- diverse tipuri de analize ale sistemelor electromagnetice, statice, staționare și dinamice, liniare și neliniare, efectele termice ale câmpului electromagnetic etc.;

- programe utilitare ca: „solver”-e de mare capacitate și viteză, algoritmi de calcul numeric (atât pentru sisteme liniare cât și pentru cele neliniare), programe de preprocesare (de exemplu preprocesorul PREP 7), programe de postprocesare (cum sunt POST 1 și POST 26), programe de lucru cu fișierele etc.;

- programe de bibliotecă: reprezentări în 2D și 3D (de bare, conducte, circuite magnetice, solide, suprafețe de contact, cavități, învelișuri, întrefier ș.a.), tratarea suprafețelor asimetrice și a solidelor fără simetrie axială, tratarea (în 2D sau 3D) a corpurilor solide superelastice, tratarea corpurilor solide neuniforme (neomogene și / sau anizotrope), prelucrarea matricelor de tip general, proiectarea servomotoarelor liniare, proiectarea/analiza circuitelor electrice cu conductoare masive (în 3D), analiza / proiectarea circuitelor magnetice (în 2D și 3D), analiza câmpurilor electromagnetice cu frontieră la infinit, studiul corpurilor dielectrice / condensatoare în câmp electrostatic (în 2D și 3D) etc.

Programele ANSYS. Pentru rezolvarea unei probleme prin metoda elementului finit, programele ANSYS realizate de utilizatori folosesc următoarele elemente: descrierea analizei, comenzile (de utilizare, de preprocesare generală – PREP 7, de procesare, de postprocesare generală POST 1, de postprocesare „Time – History” – POST 26) și de bibliotecă.

Comenzile utilitare ANSYS sunt foarte numeroase și se referă la: controlul grafic (procedee de tip /MENU /SHOW /VIS; imagini, aspect, scală, numere, simboluri, contururi etc.); selecția logică (pentru manipularea nodurilor și elementelor în care s-a descompus domeniul, ca de exemplu: NALL ce adaugă noduri pentru definirea suprafețelor selectate etc.); parametri; controlul fișierelor și altele.

Comenzile ANSYS de preprocesare generală (pentru PREP 7) sunt cele pentru: ajustare / „Set Up” (descriere) a analizei (prin specificarea: tipului analizei, opțiunile de analiză, definirea elementelor tip – de exemplu ETLIST este comanda pentru definirea tipului de elemente, proprietăți geometrice și de material); de model; pentru datele de încărcat în fișierul pentru prelucrare; de asamblare / „Wrap Up” ș.a.

Comenzi pentru procesare ANSYS (v. fig. 9.18) care sunt: /CHECK prin care se activează controlul rulării (execuției calculului), /INPUT, 27 ce comandă preluarea datelor pentru prelucrare din fișierul „File 27” și FINISH – comandă pentru ieșirea normală (prin fișierul „File 12”) a rezultatelor necesare pentru realizarea ultimei faze.

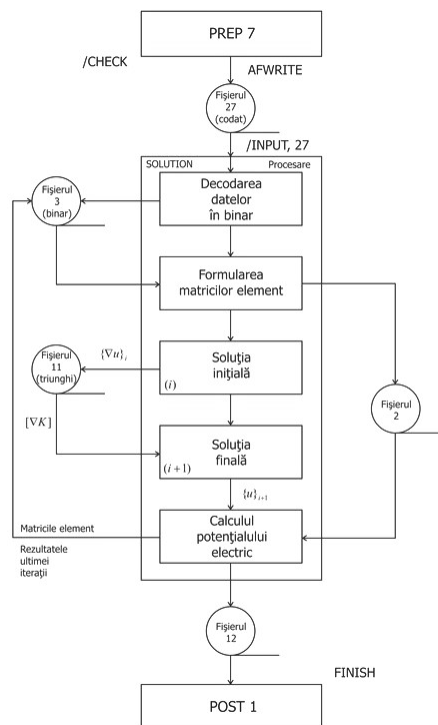


Fig. 9.18

În figura 9.18 este prezentată schema logică a procesării datelor în faza de soluționare (rezolvare a modelului numeric).

S-a considerat, în figura 9.18, cazul analizei statice a comportării mecanice a stratului dielectric dintr-un condensator plan, pentru determinarea locală a deformațiilor, eforturilor sau forțelor de reacție în regim electrostatic (un condensator plan, de exemplu un dielectric din foi de mică sau strat ceramic, supus forțelor electrostatice care acționează prin armături). În acest caz, modelul numeric de rezolvat este: $[K] \cdot \{u\} = \{F\}$, unde $[K]$ este matricea coeficienților de rigiditate ai dielectricului, $\{u\}$ este vectorul deformațiilor nodale (necunoscut) și $\{F\}$ este vectorul forțelor electrostatice. Se consideră că dielectricul este liniar și elastic, fără deformații inițiale, efectele inițiale și de histerezis sunt neglijabile, iar forțele se aplică staționar (fără mișcare) și reprezintă condiții la limită (de pe armăturile condensatorului).

În faza premergătoare (de preprocesare prin PREP7), pe baza datelor privind geometria, dimensiunile, proprietățile de material și condițiile la limită se generează (determină) modelul numeric al condensatorului $[K] \cdot \{u\} = \{F\}$. Pașii de bază ai preprocesării sunt:

- „Set Up” (de „așezare” a modelului numeric), prin care se definesc: tipul de analiză, opțiunile analizei, *tipurile de elemente* și proprietățile geometrice și de material ale corpului (placa dielectrică a condensatorului);
- „Model”, prin care se construiește modelul cu elemente finite (noduri și elemente), prin generare automată;

- „Load – Data”, prin care se specifică condițiile la limită (restricții și sarcini) și se încarcă opțiunile de optimizare;

- „Wrap – Up” (asamblarea), prin care se scriu toate informațiile necesare într-o formă corespunzătoare fazei de soluționare (procesare).

În cele de mai sus, prin *tipul de element* se înțelege determinarea gradului de libertate pentru fiecare nod și a formei caracteristice.

Soluționarea (faza de procesare) se face conform schemei de calcul din figura 9.18 prin care: la comanda /CHECK începe rularea prelucrării și prin comanda AFWRITE toate datele (codate) ale modelului numeric sunt înscrise într-un fișier special (fișierul 27). Apoi (prin comanda /INPUT, 27) datele din fișierul 27 încep să fie prelucrate (se decodifică în binar toate datele de calcul din fișierul 27, care sunt introduse în fișierul 3, după care se construiesc matricele elementale care se stochează în fișierul 2 și se soluționează numeric în mod iterativ). După ultima iterație –care dă rezultatele finale de aproximare numerică a lui $\{u\}$, care sunt salvate într-un fișier (fișierul 12)– se execută comanda FINISH de reîntoarcere la următoarea suprafață din panou și apoi începerea fazei de postprocesare (prin apelarea postprocesorului POST 1).

Prin urmare, în faza de procesare se lucrează cu cinci fișiere: „File 2” – cu datele de model (geometria) și matricele elementale; „File 3” – cu datele modelului numeric în binar; „File M” – cu matricele coeficienților de material $[K]$ triunghiularizate $[\nabla K]$; „File 12” – cu datele finale (rezultatele soluționării) necesare fazei următoare de postprocesare și „File 27” – cu datele (în cod) ale modelului numeric realizat în faza precedentă de preprocesare.

Comenzile de postprocesare ANSYS apelează posprocesoarele POST 1 (general) sau / și POST 26 (cu evoluția în timp) prin care se reprezintă pe „display” rezultatele simulării.

Biblioteca ANSYS conține subrutine cu privire la trăsăturile generale ale elementelor finite și descrierea elementelor.