

UNIVERSITATEA ${\bf POLITEHNICA}$ DIN BUCUREȘTI

Școala Doctorală de <u>INGINERIE ELECTRICĂ</u>

Teză de Doctorat PhD Thesis

Denumirea tezei în română

Tehnici de Modelare și Simulare a Dispozitivelor MEMS

Denumirea tezei în engleză MEMS MODELING AND SIMULATION TECHNIQUES

Autor: As. Ing. Mihai POPESCU Conducător de doctorat: Prof. dr. ing. Daniel IOAN

BUCUREŞTI 2021

În memoria părinților mei

Cuprins

	Mu	Mulțumiri		
	Introducere			11
1	Calcul științific de înaltă performanță			17
	1.1	Gener	al	17
	1.2	Arhite	ecturile sistemelor multiprocesor $\ldots \ldots \ldots \ldots \ldots \ldots \ldots$	20
		1.2.1	Conceptele arhitecturii unui sistem de calcul	20
		1.2.2	Arhitectura tip von Neumann	22
		1.2.3	Nivele de memorie. Paralelism la nivel uniprocesor $\ . \ . \ .$	24
	1.3	Sisten	me multiprocesor 	26
		1.3.1	Sisteme multiprocesor cu memorie comună $\ .\ .\ .\ .\ .$	29
		1.3.2	Sisteme multiprocesor cu memorie distribuită, conectate prin	
			rețea	30
		1.3.3	Clustere. Avantaje și dezavantaje	31
	1.4	Model	le de programare paralelă	34
		1.4.1	Paralelismul datelor. Granularitate	35
		1.4.2	Paralelismul instrucțiunilor (task-urilor)	37
	1.5	Limitări ale calculelor paralele		42
		1.5.1	Legea lui Amdahl	42
		1.5.2	Legea lui Gustafson	43
	1.6 Instrumente software pentru programare paralelă $\ .$		mente software pentru programare paralelă $\ldots \ldots \ldots \ldots$	44
		1.6.1	Biblioteca Pthreads	45
		1.6.2	Biblioteca OpenMP	47
		1.6.3	Medii software pentru sisteme cu memorie distribuită. Mode-	
			lul MPI	49
	1.7	Server	rul HPC Atlas al LMN. Concluzii	52

2	Fizi	ca		55		
	2.1	1 Modelare multifizică				
	2.2	Mărin	ni fizice	58		
	2.3	electromagnetic	58			
		2.3.1	Mărimile caracteristice locale ale câmpului electromagnetic	58		
		2.3.2	Mărimile locale ale corpurilor în interacțiune cu câmpul			
			electromagnetic	59		
		2.3.3	Mărimile globale ale câmpului electromagnetic	60		
		2.3.4	Legile generale ale câmpului electromagnetic	61		
		2.3.5	Teoremele de conservare ale câmpului electromagnetic	64		
		2.3.6	Ecuații constitutive	68		
	2.4	Teoria	h elasticității	70		
		2.4.1	Configurații	72		
		2.4.2	Forțe și cupluri	74		
		2.4.3	Starea de tensiune	77		
		2.4.4	Starea de deformare	79		
		2.4.5	Legile de conservare ale mecanicii mediilor continue $\ . \ . \ .$	83		
		2.4.6	Ecuații constitutive	88		
	2.5	Concl	uzii referitoare la aspectele fizice	90		
3	Mat	temati	ca	91		
	3.1	Consid	derațiuni preliminare	91		
	3.2	2 Formulări slabe				
		3.2.1	Distribuții	95		
		3.2.2	Spații vectoriale Sobolev	97		
			3.2.2.1 Definiții	97		
			3.2.2.2 Domenii Lipschitz	99		
			3.2.2.3 Proprietăți de densitate	101		
		3.2.3	Operatori diferențiali pe domeniul Ω tridimensional	104		
		3.2.4	Spații Sobolev și operatori diferențiali pe frontiera $\partial \Omega$	106		
		3.2.5	Spațiul H1	110		
		3.2.6	Spațiul H(rot)	112		
		3.2.7	Spațiul H(div)	117		
		3.2.8	Complexul De Rham	118		
		3.2.9	Existența și unicitatea soluțiilor formulărilor slabe	120		
	3.3	Formu	llări matematice pentru o ecuație eliptică	122		
		3.3.1	Formulare matematică tare	123		

		3.3.2	Unicitatea soluției în formulare tare		
		3.3.3	Formulare matematică slabă		
		3.3.4	Existența și unicitatea soluției în formulare slabă		
		3.3.5	Forme particulare de ecuații eliptice		
	3.4	Ecuaț	ii ale elasticității liniare $\ldots \ldots 134$		
		3.4.1	Forma tare		
		3.4.2	Forma slabă		
		3.4.3	Concluzii referitoare la aspectele matematice		
4	Cal	cul nu	meric 139		
	4.1	Discre	tizarea cu elemente finite		
		4.1.1	Discretizarea spațială		
		4.1.2	Coordonate baricentrice. Elementul master		
		4.1.3	Transformări afine		
		4.1.4	Elementul finit conform în H^1		
		4.1.5	Elementul finit conform în $H(\mathbf{rot})$		
		4.1.6	Elementul finit conform în $H(\mathbf{div})$		
		4.1.7	Elementul finit conform în L^2		
		4.1.8	Complexul De Rham discret		
		4.1.9	Concluzii referitoare la aspectele numerice		
5	Asp	ecte c	omputationale 179		
	5.1	Dispo	zitive MEMS		
	5.2	Un me	odel simplu de microcomutator capacitiv RF		
	5.3	Proble	ema 1D pentru microșuntul electrostatic		
	5.4	Un model 2D pentru suntul capacitiv			
		5.4.1	Formulări tari		
		5.4.2	Formulări slabe		
	5.5	Model	ul 3D pentru șuntul capacitiv		
		5.5.1	Formulări tari		
		5.5.2	Formulări slabe		
	5.6	Rezoly	vări numerice prin metoda elementelor finite $\ldots \ldots \ldots \ldots \ldots 200$		
		5.6.1	Pachetul software FreeFEM++		
		5.6.2	Rezolvare numerică iterativă		
		5.6.3	Problema2D		
		5.6.4	Evaluarea rețelei de elemente finite cu elemente de ordin I $$ 207		
		5.6.5	Influența modului de determinare a forței electrostatice 219		
		566	Caracteristica $V_a(u)$		
		0.0.0			

		5.6.7 Problema 3D	224		
	5.7	Procesarea paralelă	233		
	5.8	Concluzii referitoare la aspecte computaționale	238		
6	6 Concluzii				
	6.1	Concluzii generale	241		
	6.2	Lucrări publicate	245		
	6.3	Contribuții originale	246		
	6.4	Perspective	249		
Anexa 1. Cod FreeFemm++ 2D					
	Anexa 2. Cod FreeFemm++ 3D				
	Anexa 3. Cod FreeFem++ DDM - paralel				
Bi	Bibliografie				

Mulțumiri

Tematica esențială a prezentei lucrări este legată de modelarea multifizică: un domeniu fascinant care care s-a dezvoltat foarte mult în ultimii zece ani, în special datorită creșterii deosebite a posibilităților hardware disponibile pentru prelucraea unor volume mari de date cu algoritmi complecși. A fost nevoie să parcurg lucrări de fizică din mai multe domenii, precum și să descopăr modele de programare noi - din zona algoritmilor paraleli. Etapa cea mai dificilă însă, a fost aceea de a-mi forma o bază solidă de cunoștințe în domeniul analizei funcționale, fără de care această lucrare nu ar fi fost posibil de scris. Nu aș fi reușit să mă descurc prin hățișurile acestor multor teorii matematice fără îndrumările conducătorului de doctorat, dl. prof. dr. ing. Daniel Ioan, care au avut darul esențial de a-mi canaliza eforturile pe căile inginerești de abordare ale modelării multifizice și mai puțin pe căile pur matematice. Acestea din urmă sunt extrem de atractive, dar de cele mai multe ori sunt duse căre rafinamente matematice, care cu greu își găsesc imaginea în aplicațiile tehnice reale, rămânând în apanajul cercetătorilor dedicați matematicii pure.

Multe mulțumiri d-nei prof. dr. ing. Gabriela Ciuprina, care, cu binecunoscuta-i rigurozitate, a adus corecții pe parcursul scrierii acestei teze, menite să dea mai multă claritate expunerilor matematice și fizice, astfel încât fiecare afirmație din teză să își găsească locul potrivit în structura logică a expunerii. De menționat totodată că participarea la câteva conferințe prestigioase din domeniu, cu lucrări legate de tematica tezei, au fost posibile datorită proiectului de cercetare PN-II-PT-PCCA-2011-3 (http://mems.lmn.pub.ro/), desfășurat în perioada 2012 - 2016, inițiat și condus de D-na prof. Ciuprina.

Muţumiri speciale aduc și celorlalți colegi din echipa Laboratorului de Modelare Numerică, ș.l. dr. ing. Sorin Lup și ș.l. dr. ing. Ruxandra Bărbulescu, pentru buna colaborare la scrierea lucrărilor publicate, precum și sprijinul acordat ce mi-a permis să acord mai mult timp scrierii prezentei lucrări. Aici trebuie menționat faptul că una din lucrările publicate au fost sprijinite de proiectul UPB – GEX 2017. Ctr. Nr. 05/25.09.2017, condus de dl. ș.l. dr. ing. Sorin Lup. Nu în ultim rând, aduc mulțumiri domnilor prof. dr. ing. Mihai Iordache și prof. dr. ing. Valentin Ioniță, membri în comisia de îndrumare, atât pentru încurajări și pentru sfaturile punctuale care m-au ajutat pentru a-mi clarifica pe deplin anumite aspecte teoretice abordate în teză, precum și pentru sfaturile referitoare la anumite completări necesare spre a spori consistența textului lucrării.

Totodată doresc să mulțumesc tuturor colegilor din Departamentul de Electrotehnică, în special d-lui conf. dr. ing Mihai Maricaru pentru încurajările și suportul moral acordat.

In final, aduc calde mulțumiri întregii mele familii, care m-a sprijinit moral în toate momentele dificile și m-a suportat cu drag și răbdare mai ales în perioada de definitivare a tezei.

Introducere

Ultimele două decenii ne-au făcut martorii unei creșteri exponențiale a cererii de putere de calcul, atât din partea comunității științifice, cât și din partea aplicațiilor comerciale. Pe măsură ce costurile de proiectare și testare a noilor produse de înaltă tehnologie sunt în continuă creștere, simularea acestor dispozitive complexe cu ajutorul calculatorului a devenit din ce în ce mai necesară. Faptul a dus la dezvoltarea completă a unei ramuri științifice de sine stătătoare și anume *modelarea multifizică*.

Modelarea echipamentelor moderne necesită simularea cuplajului dintre mai multe fenomene fizice care interacționează pe parcursul funcționării respectivului dispozitiv. Frecvent, sunt întâlnite cuplaje între fenomene electromagnetice, mecanice, termice, de curgere a fluidelor. Ca urmare, în cadrul procesului de modelare multifizică punem accentul pe *modul de interacțiune dintre trei domenii: fizică, matematică și calcul numeric*, adică pe *interdisciplinaritatea* dintre cele trei domenii științifice. Această abordare este cunoscută sub numele de *CSE* -*Computational Science and Engineering*.

Primul capitol al lucrării pune în evidență resursele de calcul atât hardware, cât și software strict necesare implementării algoritmilor generați în cadrul *CSE*. Sunt punctate, pe scurt, etapele ce au marcat dezvoltarea arhitecturilor hardware ale unităților centrale de procesare ale computerelor, pornind de la arhitectura Neumann și terminând cu pocesoarele multinucleu (în engleză *multicore*), dotate fiecare cu câte o memorie temporară rapidă (*cache*), capabile de a rula seturi de instrucțiuni în paralel. La ora actuală, volumul mare de prelucrări solicitate de modelarea multifizică este efectuat în paralel, pe grupări de computere (*cluster*-e) dotate cu procesoare multinucleu. Pentru a putea beneficia de puterea de calcul oferită de procesarea paralelă a fost necesară schimbarea modelului clasic de abordare a algoritmilor seriali, cu modele de programare paralelă descrise în subcap.1.4. Fenomenul a atras după sine și dezvoltarea unor instrumente software utilizate pentru implementarea practică a noilor modele. In subcap.1.6 sunt discutate cele mai populare clase de instrumente software utilizate actual pentru implementările paralele: bibliotecile Pthreads și OpenMP pentru procesarea datelor stocate în zone de memorie partajată, respectiv modelul MPI utilizat pentru procesarea datelor stocate în zone de memorie distribuite, zone conectate prin intermediul unui sistem de mesaje.

Universul multifizic poate fi caracterizat, din punct de vedere material, printr-o mulțime de domenii spațiale disjuncte sau nu, în care se manifestă simultan sau secvențial mai multe fenomene fizice. Fiecare dintre acestea reprezintă obiectul unei teorii științifice caracterizată de:

- mărimi fizice primitive și derivate;
- un set de axiome (legi) independente, consistente (necontradictorii), respectiv complete;
- teoreme demonstrate pe baza setului de axiome menționat.

Capitolul 2 se referă la aspectele fizice de mai sus ce caracterizează teoria câmpului electromagnetic, respectiv teoria elasticității în medii liniare. Aspectele fizice ale fiecărei dintre cele două teorii sunt descrise și în limbaj matematic prin ecuații diferențiale, respectiv integrale. Exprimările matematice prin ecuații cu derivate parțiale constituie premisele aparatului matematic utilizat pentru modelarea multifizică.

Capitolul 3 este dedicat laturii matematice care permite, în mod natural, abordarea fenomenelor multifizice și anume *analiza funcțională*, prin conceptul ei fundamental de *spațiu abstract*. Este discutată clasa spațiilor vectoriale Banach din care derivă spațiile Sobolev ca spații ale soluțiilor ecuațiilor cu derivate parțiale. Din această ultimă clasă este reținută categoria spațiilor Hilbert, înzestrate cu produs scalar din care derivă norma spațiului respectiv, precum și definirea proprietății de ortogonalitate. Se arată că elementele spațiilor Sobolev pot fi aproximate oricât de bine cu funcții de test definite pe \mathbb{R}^N .

Analiza funcțională transformă formulările matematice caracteristice universului fizic (denumite *formulări tari*) în formulări corespunzătoare (denumite *formulări slabe*) universului spațiilor abstracte. In acest scop este necesară demonstrarea echivalenței dintre sistemele cu ecuații cu derivate parțiale, asociate unei anumite formulări tari, cu sistemele de ecuații ale formulării slabe corespondente, scrise cu funcții generalizate din spațiile Sobolev. Practic trebuie demonstrat că transformarea păstrează proprietățile de continuitate în sens slab. Deoarece frontierele ce închid domeniul pe care se definesc formulările conțin un număr finit de porțiuni singulare (pe care funcțiile nu mai sunt derivabile, respectiv continue) spațiile Sobolev asociate acestora nu sunt aceleași cu cele asociate interiorului domeniului de simulare. Este motivul pentru care capitolul 3 conține discuții separate referitoare la:

- spațiile Sobolev întregi și operatorii diferențiali care pot echivala formulările tari definite pe interiorul domeniului asociat problemei de rezolvat;
- spațiile Sobolev fracționare și operatorii diferențiali superficiali asociați, care pot echivala formulările tari definite pe frontiera domeniului asociat problemei;
- legătura care se face între cele doua spații Sobolev cu ajutorul operatorilor de urmă, reprezentând restricții pe frontieră ale funcțiilor definite în interiorul domeniului problemei.

Transformarea se efectuează prin proiecția formulărilor tari pe un set de funcții de bază ce generează spațiul abstract respectiv. Forma slabă a problemelor multifizice, presupune existența în membrul stâng a unei funcționale biliniare - $\mathcal{A}(u, v)$ - respectiv în membrul drept a unei funcționale liniare - $\mathcal{F}(v)$. Pe baza formulării slabe se demonstrează existența și unicitatea soluțiilor (v.subcap.3.2.9). Cum ecuațiile de rezolvat sunt diferențiale, soluțiile lor aparțin unui spațiu funcțional Sobolev întreg, iar condițiile de frontieră (excitațiile) apațin unui spațiu Sobolev fracționar (v.subcap.3.2.2). Totodată este interesant de menționat că pe spațiile funcționale abstracte se poate defini noțiunea de (spațiu) infinit, putându-se găsi formulări funcționale slabe echivalente formulărilor tari definite pe spații infinite.

Pentru fiecare tip de operator diferențial generalizat (*grad, rot, div*) ce intervin în expresia formulării slabe se alege un spațiu funcțional compatibil, care să asigure continuitatea slabă a funcționalelor definite cu acești operatori pe întreg domeniul problemei. Legătura dintre aceste spații, precum și metoda de construcție a unui spațiu dintr-altul este evidențiată de secvența exactă De Rham (subcap.3.2.8).

Ultimele două secțiuni ale cap. 3 conțin tratarea completă din punct de vedere funcțional a unei ecuații eliptice în formele sale generale și particulare, precum și a unui sistem de ecuații caracteristic unei probleme de elasticitate liniară: forma tare, forma slabă, demonstrarea existenței și a unicității soluției, cazuri particulare.

Capitolul 4 este dedicat rezolvării numerice a sistemelor de ecuații cu derivate parțiale, în forma lor slabă, prin metoda elementului finit, utilizând tehnica Galerkin. Aici sunt evidențiate aspectele esențiale ce însoțesc trecerea ecuațiilor din universul funcțional abstract continuu în universul digtal (discret):

• în universul digital noțiunea de infinit se pierde, fiind înlocuită cu relații de evaluare a erorii cu care se aproximează această noțiune;

- elementul finit este de fapt un obiect ce însumează atât caracteristici de ordin pur geometric, cât și caracteristici de ordin funcțional: pe domeniul geometric se definesc două spații duale, cel al funcțiilor de formă, respectiv cel al gradelor de libertate; conform tehnicii Galerkin ambele spații admit ca bază același set de funcții de test;
- în funcție de forma diferențială ce descrie formularea slabă, spațiile elementelor finite sunt spații Hilbert, grad-conforme, rot-conforme, div-conforme, respectiv L²-conforme reprezentând spații funcționale discrete, care păstrează proprietatea esențială de continuitate pe întreg domeniul de calcul, conform proprietăților garantate de către fiecare formă diferențială în parte;
- spațiile funcționale ale elementelor finite menționate mai sus se pot genera unul dintr-altul conform secvenței De Rham discrete, echivalentă cu cea din universul funcțional continuu.

Capitolul 5 tratează sub toate aspectele discutate în capitolele precedente o problemă mutifizică de actualitate și anume modelarea unui microșunt capacitiv. Dispozitivul face parte din categoria sistemelor microelectromecanice (presurtat în limba engleză - MEMS). Sunt parcurse practic, toate etapele procesului de modelare (cf. subcap.2.1):

- 1. este elaborat un model conceptual în două variante 2D, respectiv 3D;
- 2. pentru ambele variante sunt descrise formulările matematice tari și slabe;
- este descris un model analitic aproximativ 1D ce admite o soluție analitică care permite evaluarea calitativă a funcționării dispozitivului; aici este evidențiat caracterul neliniar al cuplajului dintre două probleme liniare: una de electrostatică, iar cealaltă de elasticitate liniară;
- 4. sunt construite modelele numerice 2D și 3D prin tehnica elementului finit, cu ajutorul unui software specializat: **FreeFEM++** . Acest software permite alegerea elementelor finite conforme exact după definițiile discutate în capitolul precedent. Descrierea problemei se face direct prin scrierea expresiilor formulărilor slabe, așa cum au fost discutate în cap.3. Anexele 1 și 2 conțin listingurile programelor scrise pentru problemele 2D, respectiv 3D, listinguri din care rezultă clar abordarea funcțională a problemelor.
- 5. sunt prezentate și discutate rezultatele obținute pentru diverse simulări efectuate cu elemente grad-conforme de ordin I și II;

- 6. sunt prezentate și comparate rezultate între simulările 2D și 3D, atât din punct de vedere al acurateței simulării, dar și al timpului de calcul;
- 7. în final este folosită o metodă de descompunere a domeniilor astfel încât să devină posibilă prelucrarea numerică pe mai multe procesoare în paralel, fiecărui procesor fiindu-i atribuit un subdomeniu; sunt prezentate rezultate comparative, sub aspectul timpului de calcul, între rezolvarea serială și cea paralelă;

Ultimul capitol al prezentei teze este unul concluziv, referitor la tematicile abordate în fiecare capitol și încadrarea lor în parcursul general al lucrării. Sunt evidențiate contribuțiile originle autorului, contribuții menite a oferi un sprijin celor care doresc să studieze mai de aproape universul modelărilor multifizice. Totodată sunt precizate lucrările științifice prezentate la diverse conferințe dedicate domeniului, precum și intențiile de dezvoltare a cercetărilor expuse în prezenta teză.

In final, merită menționată și lista bibliografică, ce conține multe lucrări de referință în domeniile analizei funcționale, al definirii și utilizării elementului finit pentru rezolvarea numerică a ecuațiilor cu derivate parțiale, precum și al utilizării tehnicilor paralele pentru implementarea practică a algoritmilor utilizați în modelarea multifizică.

Capitolul 1

Calcul științific de înaltă performanță. Arhitecturi multiprocesor

1.1 General

In 1965 Gordon Moore (co-fondator al Companiei Intel) a enunțat, într-un articol publicat în The Electronics Magazine, o afirmație conform căreia – cel puțin în următorii 10 ani (până în 1975) - densitatea de tranzistoare pe cipurile semiconductoare se va dubla la aproximativ fiecare 18 luni [115]. Din date statistice publicate în diferite lucrări de specialitate [3] [77] [89] rezultă Tab.1.1 din care se poate observa o continuare a dublării densității de tranzistoare pe un cip de microprocesor odată la aproximativ 2 ani și după 1975. În prezent această constatare este cunoscută sub numele de "Legea lui Moore", caracterul de "lege" fiindu-i atribuit deoarece fenomenul este în continuă desfășurare.

Părerile specialiștilor converg spre a accepta că tendința de dublare a densității de componente pe unitatea de structură semiconductoare se va manifesta și în următorii 10-15 ani [142], ea fiind limitată de nivelele fizice de miniaturizare considerate acum ca posibile pentru tranzistoare (nivel atomic).

Până acum câțiva ani (2004 – 2005) nivelul de miniaturizare a fost îmbunătățit în scopul de a permite utilizarea unor frecvențe de lucru din ce în ce mai mari. Acest fenomen, permitea programelor cu execuție secvențială să ruleze din ce în ce mai repede. Se constată totuși că în ultimii ani creșterea frecvenței de lucru tinde să se plafoneze. Aceasta deoarece, odată cu creșterea frecvenței de lucru, a apărut și necesitatea de a consuma mai multă energie electrică pentru răcirea procesoarelor

Familie	An	Număr	Frecvența	
microprocesor	lansare	tranzistoare	de lucru	TDP
		pe cip		
8080	1974	$6 \ge 10^3$	2 MHz	indisponibil
8086	1978	$25 \ge 10^3$	10 MHz	indisponibil
80286	1982	$134 \ge 10^3$	$25 \mathrm{~MHz}$	indisponibil
Intel Pentium	1993	$3.100 \ge 10^3$	$450 \mathrm{~MHz}$	47 W
Intel Pentium 4	2001	$42.000 \ge 10^3$	$1.5~\mathrm{GHz}$	75 W
Dual Core (Intel)	2007	$240.000 \ge 10^3$	$2.66~\mathrm{GHz}$	$95 \mathrm{W}$
Quad Core (Intel i7)	2010	$900.000 \ge 10^3$	3.20 GHz	130 W

Capitolul 1. Calcul științific de înaltă performanță

Tabela 1.1: Evoluția numărului de tranzistoare pe cip, a frecvenței de lucru și a puterii necesare pentru răcirea microprocesorului(TDP – definite cf. [5]).

(v. Tab. 1.1). Faptul a dus la un fenomen sintetizat de către Patterson și Hennesy în [77] astfel: "In curând vom dispune pe un cip de mai multe tranzistoare decât vom putea alimenta." Este relevant în acest sens graficul din Fig. 1 1 publicat în [89].

Puterea disipată de un microprocesor poate fi exprimată cu ajutorul următoarei formule [89] [24]:

$$P_{disipat} = C \times V^2 \times F \tag{1.1}$$

unde:

- C reprezintă capacitatea comutată la fiecare tact de ceas;
- V reprezintă tensiunea de alimentare;
- F reprezintă frecvența de lucru.

Aceasta deoarece energia maximă acumulată în condensatoarele cipului este $W = C \times V^2$, ceea ce implică o putere transferata P = W/T, în care T = 1/F reprezintă perioada ceasului de lucru al procesorului. Creșterea densității de tranzistoare pentru a permite creșterea frecvenței duce implicit și la creșterea direct proportională a capacitații comutate. Prin urmare o dublare a densității de integrare însoțită de dublarea frecvenței de lucru va duce la o **creștere de patru ori** a puterii disipate. Tot printr-o mărire a densității de tranzistoare se obține și integrarea pe același spațiu semiconductor a mai multor structuri de microprocesoare simple numite nuclee (în engleză – *core*) care să ruleze același program în paralel. O



Figura 1.1: Limitarea frecvenței de lucru pentru microprocesoare impusă de nivelul puterii disipate de către acestea [89]

astfel de tehnică nu implică mărirea capacității comutate C din relația 1.1. Definim performanța unei astfel de sctructuri *multi-core* prin relația:

$$Perf = (N_{core} \times F) \times k \tag{1.2}$$

unde:

- Ncore reprezintă numărul de nuclee ce lucrează în paralel;
- k reprezintă un parametru real strict pozitiv ce pune în evidență eficiența trecerii de la un singur procesor la *Ncore* procesoare ce rulează în paralel. În mod ideal k = 1.

Dacă se dublează densitatea de integrare, astfel încât să se obțină două procesoare identice care lucrează în paralel (Ncore = 2), se poate dubla performanța în condițiile menținerii frecvenței de lucru constantă și k = 1. Prin urmare, comparând numeric cu relația 1.1, ajungem la concluzia că, în acest ultim caz, se obține aceeași performanță ca și în primul caz, dar cu **o pătrime din puterea disipată** în prima variantă. Concluzia este că, o soluție pentru a depăși bariera termică (creșterea excesivă a temperaturii cauzată de puterea disipată crescută), este să se proiecteze cipuri micro-procesor (CPU) cu mai multe nuclee.

In altă ordine de idei mai putem enumera și următoarele avantaje:

- structurile integrate de tip microprocesor ce se conectează în paralel sunt mai simple și prin urmare mai ușor de proiectat și de testat;
- dintr-un set de n structuri conectate în paralel se pot utiliza numai n-1 asigurându-se astfel o îmbunătățire a siguranței în funcționare a sistemului prin redundanță.

Există și limitari ale sistemelor multi-core, limitari ce vor fi enunțate aici și tratate, pe parcursul lucrării. În general aceste limitări duc la scăderea performanței de viteză a sistemului (k subunitar) și la apariția de erori în timpul rulării programelor – erori ce sunt adeseori foarte dificil de identificat:

- Comunicarea interprocesoare: oridecâteori un procesor generează un rezultat ce trebuie folosit de către alt procesor, se consumă un timp suplimentar pentru această comunicație; această situație nu apare în cazul sistemelor uniprocesor (cu rulare secvențială);
- Sincronizarea: de multe ori este nevoie ca o anumită rutină rulată de către un procesor să fie încheiată înainte ca celelalte procesoare sa poată trece la execuția pasului de program următor; acest fapt introduce timp de așteptare suplimentar;
- Incărcarea neuniformă a procesoarelor: în mod ideal numărul total de cicluri mașină de executat pentru rularea completă a unui program ar trebui împărțit în mod egal între procesoarele implicate; acest lucru nu este posibil în totalitate, astfel încât, în permanență există procesoare care sunt în așteptare (nu sunt încărcate).

Există metode atât software cât și hardware pentru a reduce din efectele cauzelor de mai sus. Acestea vor fi prezentate in continuare.

1.2 Arhitecturile sistemelor multiprocesor

1.2.1 Conceptele arhitecturii unui sistem de calcul

Arhitectura unui sistem de calcul (computer) poate fi privită ca o suprapunere a trei concepte:



Figura 1.2: Componența **ISA**. Reprezentarea utilizează ierarhia de memorie figurată în [27]. ISA operează cu diferite tipuri de date pe care le vehiculează – în diferite moduri - între modulele de memorie, unitatea de prelucrare și diverse dispozitive de Intare/Ieșire (I/E).

- i Setul de instrucțiuni (Instruction Set Architecture), la care se va face referire în continuare prin prescurtarea ISA;
- ii Organizarea calculatorului (în engleză se folosește direct termenul *organization*), adică structurarea schemei sale;
- iii Hardware componentele fizice ale calculatorului.

ISA reprezintă întreaga structură software (v. Fig. 1.2) pe baza căreia se programează funcționarea unui sistem de calcul [77] [25]. ISA este descrisă de un set instrucțiuni software (coduri mașină alcătuite din descriptori de operații și operanzi) executate de către o unitate de prelucrare. Operanzii sunt diferite tipuri de date [23] (de ex. Intregi, boolean, caractere, numere reale reprezentate în virgulă mobilă, șiruri de caractere) ce sunt manipulate între diferitele tipuri de memorii și de dispozitive de intrare/ieșire (I/E).

Eficiența ISA este puternic dependentă de celelalte două concepte [77][25]:

• Organizarea – ce constă atât în modul în care (soft) și căile prin care (hard) sunt vehiculate datele de la un modul la altul al sistemului de calcul, cât și a modului de organizare internă a unității centrale de prelucrare a datelor (de ex. microprocesorul, sau setul de microprocesoare); în acest sens, pot exista două unități centrale de prelucrare care să utilizeze aceleași ISA, dar să aibă performanțe extrem de diferite datorită organizărilor diferite [115] [22] [26].

• Hardware – în multe cazuri denumit și implementare [77], care este rezultatul direct al tehnologiei de execuție și concepție ale a tuturor componentelor fizice ale sistemului de calcul, pornind de la circuitele integrate de orice natură și terminând cu placa de circuit imprimat care asigură interconectarea, respectiv poziționarea relativă a acestora.

1.2.2 Arhitectura tip von Neumann

Primul computer considerat "de uz general" – ENIAC - [8], putea fi programat numai prin modificări de cablare ce se obțineau cu ajutorul unor comutatoare, sau modificare de conexiuni cu fișe. Activitatea de programare pentru ENIAC dura un timp de ordinul săptămânilor. A doua generație reprezentată de computerul EDVAC, conținea o arhitectură ce permitea memorarea atât a programelor cât și a datelor în aceeași memorie devenind prototipul calculatoarelor moderne ce au urmat. Programarea acestui tip de computer se făcea în numai câteva zile. Arhitectura acestui tip de calculator poartă numele matematicianului care a descris-o: John Von Neumann [154] [111]. Ea conține toate elementele unei arhitecturi de computer moderne:

- unitate de prelucrare ce conține o structură simplă de procesor: unitate logică aritmetică cu registre pentru date și o unitate de control ce conține un contor și un registru de instrucțiuni;
- unitate de memorie în care se stocau atât datele cât și programele;
- o magistrală comună pentru date și instrucțiuni: nu era posibilă citirea simultană a unei instrucțiuni și a unei date; aceasta este de fapt caracteristica principală a arhitecturilor de computer de tip Neumann utilizată și în ziua de astăzi într-o formă perfecționată (vezi fig. 1.3) în care toate dispozitivele exterioare microprocesorului (memorii, interfețe periferice) sunt accesate printr-o magsitrala de adresare și control unitară.

Problema principală a arhitecturii von Neumann este faptul că întreaga activitate de transfer dintre microprocesor și memorii nu se poate efectua simultan pentru instrucțiuni și pentru datele manipulate de către acestea. Chiar dacă la ora actuală dispunem atât de procesoare foarte rapide cât și de spații mari de memorie la un preț acceptabil, prelucrarea bancurilor mari de date este mult încetinită de



Figura 1.3: Arhitectură von Neumann în varianta modernă cu magistrală de sistem unică [21]

limitarea adusă de însăși unul din principiile organizării arhitecturii von Neumann, decris mai sus. Limitarea este cunoscută azi sub același nume ca și arhitectura pe care o caractreizează: "limitarea Neumann".

Efectele limitării Neumann sunt mai mult accentuate de faptul că viteza microprocesoarelor actuale este de câteva zeci de ori mai mare decât viteza de răspuns a memoriilor la care sunt conectate [90] [9]. Practic, în permanență microprocesorul este în așteptare de date de la memorie, înregistrând mulți timpi morți.

Există mai multe modalități de a atenua efectele de mai sus [90] [12]:

- i Utilizarea unor memorii rapide (cache) conectate direct la microprocesor, sau fac parte din acesta fiind incluse pe cipul lui, memorii în care procesorul iși va stoca datele de prelucrat la un moment dat, sau în vederea reutilizării rapide la un pas de program ulterior;
- ii Utilizarea unor memorii rapide (cache) conectate direct la microprocesor, utilizate separat atât pentru date cât și pentru instrucțiuni – arhitectură Harvard modificată;
- iii Optimizarea programelor prin utilizarea tehnicilor de lucru paralele pe sisteme uniprocesor moderne, caracterizate de [90]:

- a înlănțuirea instrucțiunilor (pipelining),
- b utilizarea ierarhiei memoriilor (spațiu mic, dar cu acces foarte rapid -> spre spațiu mare, dar cu acces mai lent),
- c utilizarea capabilității procesoarelor de a efectua o singură instrucțiune simultan pe mai mulțe date stocate într-un registru – Single Instruction Multiple Data – SIMD)
- iv Utilizarea procesării paralele prin tehnici de accesare neuniformă a memoriei (NUMA) [77] [12] [21].

Câteva din tehnicile de mai sus (cele mai utilizate) sunt prezentate în subcapitolul următor.

1.2.3 Nivele de memorie. Paralelism la nivel uniprocesor

In anii 1944 – 1952 IBM a dezvoltat o serie de calculatoare cu o arhitectură diferită de cea de tip von Neumann, arhitectură ce poartă numele de Harvard după numele celor patru calculatoare realizate în această perioadă: Harvard I - IV [9] [10] [11]. Diferența esențială între cele două tipuri de arhitecturi este că, în cazul modelului Harvard datele și instrucțiunile sunt stocate în zone separate de memorie, ele fiind accesate prin magistrale diferite (v. Fig. 1.4). Mai mult, adresele de memorie pentru date sunt mai simple decât cele pentru instrucțiuni fiind determinate prin algoritmi mai simpli și mai rapizi.

In final a rezultat o mașină de calcul mai rapidă decât cea pe baza arhitecturii von Neumann, dar cu un set de instrucțiuni mai complicat.

Ţinând cont de diferența mare de viteză dintre microprocesoarele și memoriile moderne (v.cap. 1.2.2), performanțele arhitecturii Havard suferă de aceeași problemă (denumită în engleză "memory bound") a timpilor morți de așteptare ai procesorului ca și în cazul von Neumann. O bună și des utilizată soluție pentru rezolvarea acestei probleme este utilizarea unei arhitecturi hibride von Neumann – Harvard arhitectură ce utilizează noțiunea de nivele de memorie, nivele definite conform ierarhiei din Fig.1.8. Microprocesorul are acces direct la o memorie ultrarapidă (cache) care este scumpă, dar de capacitate mică. Există două astfel de zone de memorie: una pentru instrucțiuni, iar alta pentru date. In esență arhitectura Harvard este utilizată pentru prelucrarea datelor din imediata apropiere a microprocesorului, iar arhitectura von Neumann este utilizată pentru tratarea datelor în zonele de memorare cu capacitate mare, dar mai lente și totodată mai



Figura 1.4: Arhitectură Harvard standard [9]

ieftine. Acest hibrid poartă denumirea de arhitectură Harvard modificată și este, la ora actuală, cea mai utilizată pentru computerele de uz general.

O altă metodă de optimizare a performanțelor pe procesor este utilizarea capabilității acestora de execuție "înlănțuită" a instrucțiunilor; în engleză procesul poartă denumirea de "pipelining" [90] [16]. Instrucțiunile se execută tot secvențial, timpul de execuție al acesteia nefiind afectat, cu excepția faptului că procesorul nu așteaptă încheierea unei instrucțiuni pentru o a prelua pe următoarea. Beneficiind de un sistem de buffere și de câteva circuite suplimentare de interblocare logică, procesorul poate prelua și executa instrucțiunea următoare, imediat după ce s-au încheiat de executat anumite prime etape din instrucțiunea curentă.

In exemplul din Fig. 1.5:

- execuția secvențială durează: 4 x 90min = 6 ore cu o viteză de 4/6 procese pe oră;
- execuția înlănțuită durează: 30min + 4 x 40min + 20 min = 3,5 ore, rezultând o viteză de 4/3,5 procese pe oră (aproape dublu).

Se observă că tehnica de pipe-lining îmbunătățește strict viteza de execuție a programului, dar nu și pe cea a procesului în sine (care rămâne la 90 min).

In fine, o ultimă tehnică des utilizată în microprocesoarele moderne este aceea de a executa o singură instrucțiune, simultan, cu mai multe date stocate într-un registru al procesorului. Tehnica este notată prescurtat SIMD ("Single Instruction Multiple Data") v. Fig. 1.6 [90] [18].





Figura 1.5: Exemplul Patterson [77] [90] - Pipelining: 4 persoane își spală rufele la o spălătorie:procesul constă în: spălare(30 min) + uscare (40 min) + împachetare (20 min) = 90 min -durată

Ţinând cont de cele expuse pâna acum, înainte de a aborda procesarea paralelă, se impun câteva concluzii referitoare la sistemele uniprocesor și anume [90]:

- Trebuie să fim siguri că dispunem de un procesor serial performant (optimizat);
- Trebuie să cunoaștem paralelismul disponibil la nivel de procesor (pipelining, SIMD);
- Trebuie să cunoaștem și eventual să putem controla ierarhia de memorie din sistemul de calcul de care dispunem.

1.3 Sisteme multiprocesor

Un sistem multiprocesor poate fi definit, la modul general, ca un set de procesoare conectate între ele printr-o rețea de comunicație.

Prima clasificare a arhitecturii acestor sisteme este reprezentată de taxonomia Flynn([68], [62]), a fost enunțată în 1966, și a fost efectuată în funcție de posibilitățile sistemului de calcul de a trata în paralel seturile de date, respectiv de instrucțiuni:

• SISD = Single Instruction, Single Data – tipică arhitecturii von Neumann (v. fig.1.2.2);



Figura 1.6: O singură instrucțiune poate prelucra simultan toate datele aflate în cele două registre X, respectiv Y – proces SIMD

- SIMD = Single Instruction, Multiple Data caracteristică în general sistemelor multiprocesor cu memorie comună (v. fig.1.8, fig.1.9) cum ar fi de exemplu: procesoare vectoriale pentru aplicații științifice care au dominat piața supercomputerelor în perioada 1970 - 1990 (v. platforma CRAY-1, [54]), respectiv pentru procesarea informației grafice (GPU) în variantele populare în perioada 1990 - 2000 ([125], [4]);
- MISD = Multiple Instruction, Single Data arhitectură folosită pentru procesoare specializate, necomerciale, de exemplu în tehnicile de control al zborurilor ([146]);
- MIMD = Multiple Instruction, Multiple Data caracteristică majorității procesoarelor actuale - cu mai multe nuclee (multicore), fiecare nucleu putând procesa mai multe secvențe de instrucțiuni simultan (multithread). Majoritatea computerelor fabricate după 2013 sunt de tip MIMD; la fel și procesoarele grafice de performanță ([5]) (v.fig.1.7).

Această clasificare este pur teoretică, abstractă. După cum se poate vedea mai jos, în realitate apar multe alte detalii și variante influențate și de modul în care se poate "paraleliza" un algoritm matematic oarecare. De exemplu o o variantă de SIMD, este SPMD - Single Program Multiple Data.

Unul din idealurile proiectanților de arhitecturi de calcul este de a crea supercomputere prin conectarea celor mici – deja existente. In general este agreat



Capitolul 1. Calcul științific de înaltă performanță

Figura 1.7: Sistem de calcul cu arhitectură MIMD - conform taxonomiei lui Flynn: seturile de procese P1, P2, ..., Pn se desfășoară în unități de procesare ale aceluiași sistem; fiecare set prelucrează simultan k1, k2, ..., kN secvențe de date T1, ..., TkN; în terminologie engleză un astfel de sistem este denumit "multicore/multithread".

faptul că în loc de a proiecta un singur microprocesor de "super-performanță" este mai eficient de a dezvolta un sistem de tip cluster, sau multiprocesor – compus din mai multe mașini de calcul puse sa lucreze în paralel [77].

Pentru a înțelege funcționarea acestora trebuiesc dezvoltate următoarele aspecte:

- 1. Modul în care procesoarele paralele își partajează datele
- 2. Modul în care procesoarele se sincronizează unele cu altele
- 3. Modul de determinare a numărului de procesoare necesare pentru un anumit scop

Partajarea datelor între procesoare paralele se poate face în mai multe moduri, din care cele mai utilizate sunt: sisteme multiprocesor cu memorie comună, sisteme multiprocesor conectate prin rețea, clustere, la care memoria este distribuită. În prezent se folosesc tot mai mult soluții hibride, clustere în care fiecare nod este multiprocesor cu memorie a nodului comună.

1.3.1 Sisteme multiprocesor cu memorie comună

Mai multe microprocesoare sunt conectate la o magistrală comună. Prin intermediul acesteia ele au acces la memoria sistemului v. Fig. 1.8. Fiecare microprocesor beneficiază de memoria cache proprie fiind posibilă astfel o reducere a traficului pe magistrală. Acest tip de procesoare comunica unul cu celălalt prin intermediul unor fanioane de stare (flag-uri sau semafoare, care controlează cine are acces la magistrală, utililizate în comun.



Figura 1.8: Sistem multiprocesor cu memorie și magistrală comună. Dimensiunea tipică este între 2 și 32 de procesoare

Procesoarele conectate la o magistrală comună sunt, de regulă, identice, iar timpul de acces la orice adresă a memoriei comune (numită și memorie partajată, sau în engleză "shared memory") este același pentru fiecare dintre procesoare. Astfel de sisteme sunt de tipul cu acces uniform la memorie (UMA) sau multiprocesoare simetrice (SMP) [77] [93] [20].

O problemă ce apare la acest tip de sistem este necesitatea asigurării coerenței datelor din memoriile cache [77] [6]. Există protocoale de menținere a coerenței acestor memorii prin care controlerele lor monitorizează magistrala comună în scopul de a se asigura că în cache se regăsește o copie corectă a blocului de memorie partajată de procesoarele ce lucrează în paralel.

Acțiunile procesoarelor trebuie să se sincronizeze astfel încât unul dintre ele să nu intervină asupra datelor comune înainte ca acestea să fi fost prelucrate complet de către un altul. Cel mai utilizat mod de sincronizare este "sincronizarea prin coerență". Pentru aceasta se utilizează un semafor [77] [17] asociat unuia dintre procesoare care practic blochează toată magistrala de date pentru a-și executa operațiunile de citire/scriere în zona de memorie comună. Odată ce se încheie aceste acțiuni, fanionul de blocare se resetează, semnalizând celorlalte procesoare că magistrala este liberă.

Dezavantajul major al acestui sistem paralel ramâne limitarea numărului de procesoare ce se pot conecta datorită capacității de transfer reduse a magistralei comune (max 36 la ora actuală) [19]. Peste această limită eficiența scade, deoarece magistrala devine ocupată în majoritatea timpului, iar procesoarele nu o mai pot folosi când au nevoie. Pentru a putea conecta mai multe procesoare în sistem cu memorie partajată ar trebui utilizate mai multe magistrale, iar aceasta ar însemna un sistem de tipul celui din Fig. 1.9.

1.3.2 Sisteme multiprocesor cu memorie distribuită, conectate prin rețea

Aceste sisteme sunt caracterizare de faptul că fiecare procesor are memoria proprie, spre deosebire de cazul cu magistrală comună și memorie partajată, accesata uniform (UMA). În acest caz conexiunea dintre procesoare folosește exclusiv la comunicția interprocesor, iar memoria este tot comună, dar de această dată, distribuită. Ținând cont că un procesor poate accesa memoria altui procesor, acest acces este neuniform depinzând de adresa de memorie în/din care procesorul trebie să scrie/citească (NUMA – Non Uniform Memory Access) [77] [13].

Comunicarea, respectiv sincronizarea între procesoare se face prin mesaje. Deși comunicarea mesajelor se poate efectua printr-un protocol normal de rețea (transmisie/recepție), datorită întârzierilor mari ce însoțesc pachetele de date conform acestor protocoale, se preferă utilizarea unor mesaje de tip încarcă/preia din memorie. Aceste mesaje sunt mult mai economice d.p.d.v al duratei de transfer. Memoriile procesoarelor vecine vor fi "văzute" ca **memorie partajată virtuală**, iar organizarea este de tipul cu **memorie distribuită partajată** (**D**istributed **S**hared **M**emory – **DSM**) [77] [13]. Totuși, datorită timpilor mari de transfer de la un bloc de memorie la altul, în cadrul acestei organizări, memoria va fi accesată numai atunci când este nevoie de datele stocate, și nu doar pentru a putea fi reutilizată mai târziu, pe parcursul programului.



Figura 1.9: Organizarea unui sistem multiprocesor, cu memorie comună distribuită și procesoare conectate prin rețea.

1.3.3 Clustere. Avantaje și dezavantaje.

Organizarea clustere-lor [77] [91] [7] constă dintr-un set de computere de sine stătătoare, conectate în rețea prin itermediul unor dispozitive de de interconectare (de tip hub, switch, router) și programate să lucreze în paralel pentru execuția unor aplicații solicitante cum ar fi: baze de date, servere de fișiere, servere de Web, calcule științifice de înaltă performanță. Există avantaje și dezavantaje ale clustere-lor față de sistemele multiprocesor cu memorie partajată [77] [91]:

- Cel mai mare avantaj al clustere-lor față de sistemele multiprocesor este prețul; acest avantaj este cu atât mai mare cu cât sistemul multiprocesor cu care se compară cluster-ul conține mai multe procesoare;
- Un dezavantaj al clustere-lor ar fi costurile de administrare: administrarea unui cluster cu N mașini echivalează cu de N ori costul de administrare al unui calculator singular, pe când costurile de administrare pentru un sistem multiprocesor se apropie foarte mult de costul de administrare al unui calculator monoprocesor;
- Capacitatea de transfer a magistralei unui sistem multiprocesor cu memorie



Capitolul 1. Calcul științific de înaltă performanță

Figura 1.10: Organizare de cluster cu mașini independente. Comunicația interprocesoare se efectuează prin interfețele de rețea. De aceasta dată, rețeaua nu mai este o simplă magistrală, ci un dispozitiv de comunicație (un "switch") de la care pleaca linii de comunicație către calculatoare.

partajată (SMP) este mult mai bună decât capacitatea de transfer a magistralei de I/E ale computerelor ce compun cluster-ul;

- Un cluster are memoria totală divizată în N sisteme de operare, pe când un SMP are la dispoziție întreaga memorie a sistemului; astfel, un program secvențial va rula mai bine pe un SMP decât pe un cluster;
- Dezavantajul de mai sus este însă și un avantaj pentru clustere: în cazul unui defect al unuia dintre procesoare, acesta se poate înlocui – la cluster – fără a opri întreg sistemul, pe când la SMP acest lucru nu este posibil.
- Totodată cluster-ul este mult mai flexibil, numărul de procesoare putând fi extins fără afectarea celor care rulează deja, pe când la SMP acest lucru nu este posibil.

Pentru sistemele de clustere mari se utilizează organizări hibride ce conțin atât sisteme SMP cât și calculatoare independente.

In cadrul organizărilor hibride complexe:

- Calculele se execută atât în procesoare cât și în procesoarele grafice (GPU)
- Se elimină limitarea de calcul în simplă precizie



Figura 1.11: Organizare de cluster hibridă cu mașini independente și SMP. Se combină avantajele sistemelor multiprocesor cu memorie partajată și ale sistemelor multi-computer. Astfel se pot realiza sisteme paralele de mari dimensiuni. Comunicația interprocesoare este foarte complexă și se efectuează prin interfețele de rețea.



Figura 1.12: Organizare de cluster hibridă cu mașini independente și SMP. În mașinile independente este folosit și procesorul grafic (GPU).

- Se poate echilibra foarte bine încărcarea procesoarelor
- Este necesară utilizarea unor biblioteci și compilatoare dedicate (CUDA [28], OpenCL [29]
- Comunicația interprocesoare este foarte complexă și se efectuează prin interfețele de rețea.

1.4 Modele de programare paralelă

Modalitățile de paralelizare ale unui algoritm nu se pot înscrie unor așa zise rețete ci, mai degrabă, procesul de trecere, al algoritmului de la o descriere serială la una paralelă, trebuie să respecte anumite principii, având ca scop esențial îndeplinirea unor criterii generale de eficiență ale programului. In cele ce urmează, vom reprezenta un algoritm cu ajutorul unui graf astfel:

- Noduri care corespund instrucțiunilor, sau blocurilor de instrucțiuni;
- Ramuri orientate care leagă aceste noduri și care specifică dependințele dintre instrucțiuni, indicând ordinea în care ele sunt executate.

Un algoritm dedicat unei probleme științifice poate fi descompus în mai multe căi ce pot fi figurate în paralel; în final, pentru a obține rezultatul dorit, aceste căi se unesc într-una singură. Căile reprezentabile în paralel reprezintă acea parte din algoritm ce poate fi implementată pe procesoare paralele, iar calea ce rămâne reprezintă acea porțiune de algoritm ce nu poate fi rulată decât secvențial. Prin urmare, pentru descrierea unui algoritm paralel ar trebui parcurși urmatorii pași [34] [35]:

- 1. Descompunerea problemei în etape de rezolvat (task-uri) mai mici; de regulă aici se caută stabilirea unor porțiuni de algoritm (reflectate mai apoi în porțiuni de cod) care se pot refolosi și pentru alte implementări;
- 2. Determinarea acelor etape ce pot fi rezolvate în paralel unele cu altele;
- 3. Determinarea acelor etape care trebuiesc rezolvate secvențial și ordinea în care trebuiesc rezolvate;
- 4. Descrierea etapelor de la puntele 2 și 3 de mai sus printr-un graf orientat, conform celor de mai sus;
- 5. Determinarea variantei optime de descompunere paralelă și secvențială și alocarea task-urilor pe procesoarele disponibile.

Această metodologie de paralelizare a fost formalizată de Foster în 1995 ([69]) prin următoarea secvență: "Partition -> Communication -> Agglomeration -> Mapping" (prescurtat PCAM).

Primele două etape, generează niște modele abstracte întrucât nu țin cont de resursele de calcul disponible în mod concret (procesoare, memorie). Este rolul următoarelor două etape de a adapta și optimiza modelul de algoritm paralel abstract la sistemul de calcul pe care acest algoritm se poate real implementa: Etapa "Partition" poate fi abordată prin două tehnici ce pot fi aplicate, în funcție de problemă, simultan sau doar una dintre ele:

- partițonarea datelor care se înscrie în așa numita clasă de metode de *descompunere a domenului*;
- partiționare la nivel funcțional constând în împărțirea întregului proces în unele (funcții/task-uri) mai mici, care pot rula în paralel, fiecare pe câte un procesor dintr-un ipotetic set de unități de calcul independente.

1.4.1 Paralelismul datelor. Granularitate

Paralelismul datelor este caracterizat de faptul că fiecare procesor execută aceleași operațiuni, dar pe seturi de date diferite. Exemplu: înmulțirea unei matrice A_{nxn} cu un vector b_n . Pornim de la următorul pseudocod al algoritmului (fără a evidenția instrucțiunile de I/E):

```
intreg n, i, j
tablou real A[n, n], b[n], r[n]

pentru i = 1, n
    r[i] = 0 ; initializare
    pentru j = 1, n
        r[i] = r[i] + A[i, j] * b[i] ; calcul r[i]
retur
```

Se observă că fiecare termen al vectorului rezultat \mathbf{r} se calculează independent pentru fiecare i în parte. Prin urmare se pot descrie n ramuri de program în paralel pentru fiecare i (v. Fig 1.14).

Din exemplul de mai sus se poate observa că algoritmul se poate rula pe maxim n procesoare în paralel. Același algoritm se poate rula și pe mai puține procesoare, de exemplu – presupunând că n este multiplu de 3, ramurile în paralel se pot grupa



Figura 1.13: Imagine schematică a metodei de paralelizare propusă de Foster. Este exemplificat un proces serial, care admite o ipotetică descompunere funcțională în opt procese elementare, fiecare dintre aceste procese putând prelucra câte o partiție din datele de intrare; în etapa a doua sunt identificate comunicațiile necesare între procesele elementare ce rulează în paralel: se pot observa direcțiile de transfer ale datelor după sensurile săgeților. Etapa de aglomerare grupează procesele astfel încât să se reducă efortul solicitat de comunicațiile intertask-uri: aglomerarea proceselor verde deschis albastru, respectiv portocaliu - gri elimină două comunicații, iar grupările roz-verde închis și roșu-magenta elimină două comunicații (cele verticale) și reduc la jumătate latența comunicației orizontale între cele două grupări ([155]). În final grupările rezultate pot fi asociate (mapate), în funcție de resursele de calcul disponibile, pe două, respectiv patru unități de procesare (de exemplu patru nuclee (cores).


Figura 1.14: Algoritmul de înmulțire a unei matrice cu un vector se poate descrie prin n ramuri în paralel, deci se poate implementa prin n procesoare. Fiecare procesor execută aceleași operatiuni, dar pe seturi de date diferite (linii diferite ale matricei A).

câte trei, iar tot programul se poate rula pe n/3 procesoare. Se poate discuta deci, de o finețe de împărțire a problemei pe un anumit număr de procesoare în paralel, deci se poate defini o anumită **granularitate** a algoritmului. In general, există un optim de granularitate: pe de o parte cu cât aceasta este mai fină, cu atât este de așteptat ca algoritmul să ruleze mai repede, dar, pe de altă parte vor crește întârzierile datorate intercomunicației dintre procese [92]. Rezultă deci, că fiecare problemă are o granularitate optimă.

1.4.2 Paralelismul instrucțiunilor (task-urilor)

Paralelismul instrucțiunilor este caracterizat de faptul că fiecare procesor execută un set de instrucțiuni diferite, pe același set de date [63]. Exemplu: asamblarea matricei nodale $G_{N\times N}$ și a termenului liber asociat $I_{s_{N\times 1}}$ pentru un circuit electric rezistiv, liniar cu **N** noduri și **L** laturi. Pentru simplitate, presupunem că fiecare latură de circuit conține o rezistență nenulă R_k și o sursă independentă de tensiune E_k care poate fi și nulă, unde $k \in \{1, 2, ..., L\}$.

Pseudocodul procedurii de asamblare este următorul [51]:

```
procedura asamblare_nodal (N, L, ni, nf, R, E)
```

```
intreg N ;numar de noduri
intreg L ;numar de laturi
tablou intreg ni(L) ;noduri initiale
tablou intreg nf(L) ;noduri finale
tablou real R(L) ;rezistentele laturilor
```

Capitolul 1. Calcul științific de înaltă performanță



Figura 1.15: Algoritm paralel de asamblare a matricei unui circuit liniar rezistiv corespunzător metodei nodală. Fiecare procesor rulează seturi de instrucțiuni diferite pe aceeași structură de date.

```
tablou real E(L) ;t.e.m. ale laturilor
tablou real G(N,N) ;matricea conductante nodale
tablou real is(L) ;vector injectii de curent
pentru i = 1,N
                     ; initializez matrice sistem
    is(i) = 0
   pentru j = 1, N
       G(i, j) = 0
pentru k = 1, L ;parcurge laturi
   n1 = ni(k)
   n2 = nf(k)
   G(n1,n1) = G(n1,n1) + 1/Rk; procesorul 1
    G(n2,n2) = G(n2,n2) + 1/Rk; asamblare termeni diagonali
   G(n1,n2) = G(n1,n2) + 1/Rk; procesorul 2
    G(n2,n1) = G(n2,n1) + 1/Rk; asamblare termeni nediagonali
    is(n1) = is(n1) - Ek/Rk ; procesorul 3
    is(n2) = is(n2) + Ek/Rk; asamblare termen liber
retur
```

Procesul se poate paraleliza conform diagramei din Fig. 1.15.

Etapa "Communication" se referă la stabilirea necesităților de comunicare între procesele rezultate în urma etapei de partiționare: în cele mai multe cazuri, este nevoie ca procesele să-și transmită de la unul la altul seturi de date, sau semnale necesare sincronizării. Comunicațiile sunt globale, atunci când pentru prelucrarea unui set de date este necesară participarea majorității proceselor paralele ale algoritmului, sau locale atunci când schimbul de date se efectuează între un număr restrâns de procese ([155]). Aceste comunicații nu apar în cazul proceselor seriale, deci în cazul paralel reprezintă un efort de timp de calcul suplimentar (în lb. engleză *overhead*) important. Partiționarea funcțională trebuie concepută astfel încât timpul solicitat de comunicațiile inter task-uri să fie minim și să fie uniform distribuit între acestea.

Ultimele două etape (Agglomeration și Mapping) reprezintă o problemă de optimizare a structurii abstracte de date, funcții și comunicații elaborate în primele două etape. In urma aglomerării, procesele se pot grupa astfel încât să se reducă efortul de timp de calcul suplimentar datorat comunicațiilor dintre task-urile paralele; în schimb, se mărește granularitatea implementării paralele (crește efortul de calcul pe fiecare proces). Acestei optimizări i se adaugă și obiectivul de a putea implementa numărul de procese paralele proiectat pe resursa hardware disponibilă, reprezentată, în majoritatea cazurilor, de un set de unități de procesare cu memorie distribuită.

In 2004, Mattson [109] propune o sistematizare a procesului de programare paralelă, grupând multitudinile de tehnici tipice acestui proces - împărțit în patru etape (v. fig.1.16) - în grupe de *modele (patterns)* definite la nivel abstract.

Această abordare este mai generală decât a lui Foster, încercând să acopere toate aspectele esențiale ale metodologiei de implementare paralelă a unui algoritm. Prima etapă (v. fig.1.16), asemănătoare cu cea descrisă de Foster, este urmată de o a doua care are ca finalitate stabilirea unei structuri a algoritmului paralel. Acest scop se poate atinge urmând un arbore decizional (v. fig.1.17) pornind de la unul dintre criteriile următoare, prioritatea criteriilor fiind decisă în urma partiționărilor efectuate în prima etapă:

- task-uri (descompunere funcțională) care se pot grupa în două tipuri principale:
 - independente, respectiv între care există anumite interdependențe impuse de sincronizarea execuției lor în timp, sau de modul de partajare a datelor între procese; în această situație procesele respective pot fi rulate în paralel, fiecare pe câte o unitate de procesare.
 - a căror acțiune poate fi descrisă printr-o procedură recursivă, descriptibilă de exemplu, printr-un arbore binar; în acest caz se pot adopta metode din clasa "Divide and Conquer" prin divizarea unui task în două sub-task-uri recursive, care pot rula în paralel pe unități de procesare diferite, urmând



Figura 1.16: Etapele procesului de implementare paralelă a unui algoritm, conform diagramei propusă de Mattson [109]; la fel ca și la metoda propusă de Foster, prima etapă este dedicată partiționării funcționale, respectiv ale datelor; următoarea este dedicată stabilirii celui mai potrivit mod de organizare a algoritmului paralel în funcție de modul de partiționare stabilit în prima etapă; etapa a treia este dedicată alegerii structurilor de funcții/proceduri și de date astfel încît acestea să poată realiza cel mai bine implementarea modului de organizare stabilit în etapa a doua, pe hardware-ul disponibil; în sfîrșit a patra etapă se referă la tehnicile practice specifice necesare implementării procedurilor și structurilor de date alese în etapa a treia pe sistemele de calcul disponibile.



Figura 1.17: Procesul decizional de stabilire a modelului de programare paralelă potrivit unui anumit algoritm, propus de Mattson în [109].

ca rezultatele parțiale ale fiecărui sub-task să se combine la sfârșit pentru generarea rezultatului final al task-ului din vârful arborelui.

- descompunerea datelor (decompunerea domeniilor) ce se pot grupa tot în două tipuri principale:
 - tipul asimilat unei descompuneri geometrice, fiind caracterizat de descompunerea spațiului datelor de intrare, în subspații, prin discretizare conform unei anumite grile; soluțiile parțiale se pot determina, în paralel, pentru fiecare subspațiu, pe câte o unitare de procesare independentă, fiecare soluție parțială fiind influențată doar de datele din câteva subspații vecine (de ex. metoda elementului finit, sau cea a diferențelor finite); soluția finală a problemei se obține prin suprapunerea soluțiilor obținute pentru subspațiile respective;
 - tipul de date ce se poate organiza în liste înlănțuite, administrarea lor putându-se efectua prin proceduri recursive;
- desfășurarea fluxului de date, criteriu care se folosește atunci când ordinea de procesare a datelor este determinantă pentru stabilirea ordinei de apelare a procedurilor de prelucrare a lor; și în acest caz, conform Mattson, se pot determina două tipologii:
 - lanț de proceduri tip "Pipeline" caracteristic unui flux de date unidirecțional, constant, în care rezultatele fiecărei proceduri folosesc ca date de intrare următoarei proceduri din lanțul de execuție; în acest caz, fiecare dintre procedurile pomenite, se pot implementa pe câte o unitate de procesare independentă, astfel încât, după câțiva pași inițiali, toate procesoarele implicate vor lucra în paralel, preluând date de la predecesor și furnizându-le celui următor;
 - set de task-uri ce pot rula în paralel, pe unități de procesare independente, activitatea lor fiind declanșată de apariția, regulată sau spontană, a unor evenimente rezultate din datele de intrare, sau intermediare din algoritm.

Ca o concluzie din cele prezentate mai sus, putem afirma că unul din cele mai potrivite modele de paralelizare ale unui algoritm complex, este un model hibrid de programare paralelă, reprezentativ pentru majoritatea cazurilor ce apar în practică și având următoarele caractresitici:

• Există o singură sursă pentru cod ce se transmite tuturor procesoarelor;

- Codul poate conține un set de condiții de selecție a procesorului care va executa un anumit set de instrucțiuni pe un anumit set de date;
- Toate procesoarele încep rularea codului simultan, iar apoi comunică și se sincronizează între ele după necesități;

Acest model de programare poartă denumirea de "Single Program Multiple Data" (SPMD) - v. fig. 1.18



Figura 1.18: Modelul de programare SPMD

1.5 Limitări ale calculelor paralele

1.5.1 Legea lui Amdahl

Toate programele paralele conțin atât segmente ce pot fi implementate pe procesoare paralele, dar și segmente ce nu pot fi tratate altfel decât secvențial. Acestea din urmă limitează avantajele aduse de segmentele paralele. Creșterea de viteză posibilă prin paralelizare a fost exprimată de Gene Amdahl in 1960 printr-o lege ce se poate enunța astfel [15] [63]:

Notăm cu α fracțiunea din efortul de timp al unui program pe care acesta îl dedică execuției segmentelor neparalelizabile. Prin paralelizarea segmentelor de program ce permit acest lucru se obține o creștere de viteză $S = T_s/T_p$ (in care T_s este timpul execuției seriale, iar ${\cal T}_p$ este timpul execuției paralele. Această accelerare are expresia:

$$S = \frac{p}{(\alpha \cdot p + 1 - \alpha)} \tag{1.3}$$

în care p este numărul de procesoare care lucrează în paralel. La limită, pentru un număr foarte mare de procesoare se obține:

$$S = \lim_{p \to \infty} \frac{1}{\frac{1-\alpha}{p} + \alpha} = \frac{1}{\alpha}$$
(1.4)

Deci dacă 10% din efortul de calcul este dedicat porțiunii obligatoriu seriale a programului, atunci creșterea de viteza obținută, în mod ideal $(p \to \infty)$, este de maxim 10 ori, oricâte procesoare ar fi disponibile! De fapt lucrurile sunt și mai rele datorită întârzierilor adugate de:

- Încărcarea neuniformă a procesoarelor;
- Timpii de așteptare în scopul sincronizării procesoarelor, sau a câștigării accesului la memoria comună;
- Timpii pierduți prin protocoalele de comunicație;
- Intârzierile introduse de perifericele de I/E.

1.5.2 Legea lui Gustafson

Legea lui Gustafson determină viteza de execuție câștigată prin utilizarea a p procesoare paralele:

$$S(p) = p - \alpha \cdot (p - 1) = \alpha + p \cdot (1 - \alpha) \tag{1.5}$$

Legea lui Amdahl presupune independența efortului de calcul de numărul de procesoare, pe când Gustafson presupune că efortul de calcul în paralel variază liniar cu numărul de procesoare. Legea lui Amdahl se referă la probleme de dimensiune fixă (executate în timpi diferiți) și oferă o viziune mai pesimistă asupra accelerării decât Legea lui Gustafson, care se referă la un timp de execuție fixat, dar probleme de mărime diferită.

1.6 Instrumente software pentru programare paralelă

Clasificarea instrumentelor software utilizate pentru implementatea algoritmilor paraleli este similară modului de clasificare a arhitecturilor sistemelor multiprocesor (v. cap. 1.3). Nu putem vorbi însă de același grad de similaritate între rapiditatea ce caracterizează procesul de dezvoltare hardware a sistemelor multiprocesor, și viteza de dezvoltare al mediilor software necesare implementării aplicațiilor paralele. Aceasta datorită în special faptului că, trecerea de la implementarea algoritmilor secvențiali, realizată cu ajutorul unor compilatoare performante, dotate cu puternice funcții de optimizare automată, la implementări paralele, nu se mai poate face (cel puțin la ora actuală) cu ajutorul unor "compilatoare automate" dedicate acestei conversii de la serial la paralel. Particularitățile aplicațiilor software, mai ales cele legate de modelarea fenomenelor fizice, nu permit un proces de implementare paralelă automată, similară modului de tratament al altor algoritmi, de exemplu acelora dedicați gestionării bazelor de date. Se pot desprinde două clase mari de instrumente software în domeniul aplicațiilor paralele, după modul de gestionare a datelor:

- procesare a datelor stocate în zone de memorie partajată; tipic acestui model este că datele sunt plasate într-un banc de memorie (partajată) la care au acces toate procesele (implicit și procesoarele) ce rulează în paralel în cadrul aceluiași program (v.fig.1.8). Datele sunt accesate de către procesele paralele respective, în mod asincron.
- procesare a datelor stocate în zone de memorie distribuite "conectate" prin intermediul unui sistem de mesaje.

Fiecare clasă de aplicații din cele enumerate mai sus, au avantaje și dezavantaje care pot fi evidențiate urmărind următoarele aspecte:

- modul în care, pe parcursul execuției procesului principal al programului, se inițiază și se încheie sub-procese, sau secvențe scurte (threads) care rulează în paralel pe mai multe procesoare și/sau pe mai multe nuclee ale aceluiași procesor;
- modul în care sunt rezolvate situațiile critice care pot apărea pe parcursul rulării proceselor parlele pomenit la puctul anterior și aici putem vorbi de două mari clase de astfel de situații - numite și *porțiuni critice de cod*:

- un set de date este accesat la momente nepotrivite şi în mod necontrolat de către diverse secvențe de program ce rulează în paralel: fie datele sunt neprelucrate suficient pentru a fi preluate de către procesul respeciv, fie modificarea lor de către procesul în cauză ar fi prematură întrucât mai sunt şi alte procese care ar trebui să preia acele date în forma curentă (nemodificată încă);
- două sau mai multe procese/secvențe se interblochează între ele așteptând unul de la altul instrucțiuni pentru continuarea activității;
- modul în care se utilizează instrumentele software respective: sub formă de biblioteci de funcții, sau sub formă de seturi de directive de compilare, care generează în momentul etapei de compilare codul necesar inițierii setului de procese ce vor rula în paralel.

1.6.1 Biblioteca Pthreads

Pthreads este o bibliotecă de funcții C dedicate transformării unui program serial, într-unul cu porțiuni în care execuția se împarte pe mai multe subprocese ce rulează în paralel pe procesoarele sau nucleele disponibile în hardware-ul curent. Aceste subprocese sunt de fapt secvențe scurte de instucțiuni - numite *"threads*"([40]), fără a avea structura completă a unui proces complet: zonă de date privată, contor și stivă de instrucțiuni proprii, descriptori de stare dedicați. Thread-urile reprezintă pentru un proces, un mod rapid de accesare a unui set de date din memoria comună, sau a unui flux de date de I/E. În general un anumit proces poate deveni *"multithread"* (multisecvență) dacă, la un moment dat, va genera astfel de secvențe ce vor rula în paralel, partajând resursele procesului părinte și dispunând numai de stiva proprie de instrucțiuni. Același proces va controla activitatea secvențelor paralele generate, în final fiind responsabil de închiderea acestora. Caracterul *"P"* din denumirea bibliotecii este datorat faptului că biblioteca Pthreads este disponibilă pentru sistemele de operare conforme cu standardul POSIX (UNIX, Linux, Mac OS X)([30]).

Inițierea unui "thread" se efectuează în mod dinamic, în funcție de necesitățile curente ale programului: de exemplu, în cazul rezolvării unui sistem de ecuații liniar poate fi suficient un sigur thread dacă dimensiunea sistemului este 10; în schimb dacă dimensiunea sistemului este 100, pot fi generate 10 thread-uri ce vor rula, în funcție de disponibilitățile hardware, pe un număr de procesoare mai mic sau mai mare de 10. Intrucât încărcările procesoarelor care lucrează în paralel nu sunt egale, nu este obligatoriu ca sistemul să distribuie uniform secvențele de

program generate la un moment dat spre a rula în paralel, de unde posibile întârzieri suplimentare.

Admnistrarea activității thread-urilor se efectuează de regulă prin variabile globale la care au acces toate secvențele de cod generate dinamic la un moment dat; în plus, fiecare dintre aceste secvențe poate dispune de setul propriu, privat de date.

Porțiunile critice de cod tipice pentru acest model de software paralel sunt acelea în care mai multe procese pot avea acces de citire/scriere (R/W) la o zonă de memorie comună, generându-se așa numitele "*race conditions*". Aceste situații pot fi tratate prin tehnici ce se pot grupa în trei mari clase ([124], cap.4):

- rulare în așteptare (busy-waiting) caracterizat de inducerea temporară a execuției unui cod într-o buclă închisă până în momentul în care accesul acelui cod la zona comună de memorie este permisă; deși codul generat prin programarea acestei condiții este foarte restrâns, totuși tehnica este foarte consumatoare de timp, întrucât procesul în sine nu este efectiv oprit ci consumă în mod constant din resursele de timp ale procesorului (cicli mașină) rulând în mod continuu un ciclu închis; este motivul pentru care este o tehnică de sincronizare a thread-urilor paralele puțin folosită și anume doar în cazurile în care numărul de thread-uri paralele este mai mic decât numărul de procesoare;
- interblocarea exclusivă a proceselor ce încearcă să acceseze o zonă de memorie comună, prin obiecte software denumite mutex-uri: un thread care dorește să acceseze o zonă de memorie face mai întâi o solicitare de blocare exclusivă a zonei respective prin apelarea unei funcții denumite pthread_mutex_lock; dacă zona de memorie este deja blocată printr-un alt mutex, atunci secvența rămâne în așteptare, fără să încarce suplimentar procesorul pe care rulează. După ce primește accesul la datele partajate, în final, thread-ul anunță sistemul că a executat porțiunea de cod critică, iar zona de memorie asociată devine accesabilă de către alte procese, în urma apelării funcției pthread_mutex_unlock. Dezavantajul acestei tehnici rezidă în faptul că, ea nu este aplicabilă unui set de procese care ar trebui să prelucreze datele comune într-o anumită ordine (de ex. în cazul unor operații necomutative cum ar fi înmulțirea matricelor);
- semafoarele sunt variabile globale, întregi, fără semn, care, spre deosebire de mutex-uri, pot fi inițializate și controlate direct din programul principal, în afara thread-urilor; un semafor nul asociat unei zone de memorie indică faptul că acea zonă este blocată oricărui acces. Astfel, acest acces poate fi controlat după o schemă de sincronizare a proceselor definită după o anumită logică; în

cazul în care este util, accesul la anumite semafoare poate deveni privat unui anumit thread. De altfel funcțiile dedicate pentru manipularea semafoarelor (de ex. sem_post/sem_wait pentru ridicarea respectiv impunerea unei resticții) nu aparțin bibliotecii Pthreads, ci bibliotecii C. Folosirea semafoarelor permite modele de sincronizare extrem de utile tehnicilor software paralele:

- bariere un punct dintr-un program pe care niciun proces nu-l poate depăși decât dacă toate procesele care rulează în paralel îndeplinesc simultan o anumită condiție;
- blocaj R/W care permite ca un set de date să poată fi citit de către oricare proces, dar să nu poată fi modificat decât de către un singur proces la un moment dat;

In final, nu trebuie uitate acele funcții C care folosesc pentru decizii, variabile de tip static, care, într-un program executat serial, au rolul de selector al operațiilor efectuate de funcția respectivă; într-o implementare paralelă cu memorie partajată, toate thread-urile au acces la aceeași bibliotecă de funcții, ceea ce face ca o aceeași funcție să poată fi accesată, în mod asincron, de diferite procese; astfel variabilele statice menționate sunt modificate în mod necontrolat. Efectul global, este unul de greu de detectat deoarece, datorită fenomenului descris mai sus, același cod poate produce rezultate diferite pentru același set de date, în funcție de numărul de procese paralele care pot fi generate.

1.6.2 Biblioteca OpenMP

OpenMP reprezintă o colecție de funcții ce permit implementarea algoritmilor paraleli pe sisteme de calcul cu memorie partajată. Față de *Pthreads*, un mare avantaj oferit de OpenMP([14]), este acela că, pentru a genera secvențe de cod care vor rula în paralel, se pot utiliza directive de compilare (**#pragma**) dedicate acestui scop; acesta este motivul pentru care OpenMP necesită un compilator ce poate genera automat codul necesar inițierii statice sau dinamice a thread-urilor ce vor rula în paralel. Din acest motiv OpenMP nu este privit ca o simplă bibliotecă, ci ca un standard în domeniul programării paralele ([151]), pentru care la ora există suport inclus în compilatoarele de C, C++, și Fortran asociate versiunilor Gnu, Intel, IBM, Cray. Practic, OpenMP reprezintă un pas important în sensul migrării efortului de programare din zona utilizării exclusive a funcțiilor disponibile într-o bibliotecă (ca în cazul *Pthreads*), în zona programării asistate de calculator, în care, o bună parte din "bucătăria" de programare a unui cod multithread este preluată de compilator, acesta din urmă fiind instructat numai prin directivele de tip **#pragma**. In plus, prin folosirea de directive de compilare, nu este necesară modificarea structurii codului serial inițial

Directiva tipică utilizată de *OpenMP* este **# pragma omp parallel**, urmată de un bloc de program; ca urmare, între punctul de intrare și cel de ieșire din blocul de program respectiv, compilatorul va genera cod pentru inițierea și terminarea unui set de thread-uri ce vor executa în paralel funcțiile impuse de acel bloc; această grupare de thread-uri, este cunoscută sub denumirea de *echipă* (team)([124]). Administratorul echipei este secvența de cod care rula înainte de începerea execuției blocului de program, administrator ce este responsabil și de închiderea tuturor membrilor echipei înainte de ieșirea din blocul respectiv. Directivele de compilare OpenMP, pot fi însoțite de parametri, cunoscuți sub numele de *clauze*.

Cu toate că nivelul compilatoarelor ce oferă suport *OpenMP* este înalt, ele nu pot rezolva în mod automat *porțiunile critice de cod* tipice modelului de programare pe sisteme hardware cu memorie partajată, porțiuni critice de program întâlnite de altfel și în cazul folosirii bibliotecii *Pthreads*. Zonele critice de program sunt cele în care mai multe procese doresc să partajeze simultan același set de date putând genera, ca și în cazul *Pthreads*, "race conditions". Pentru tratarea acestora *OpenMP* dispune de un set de tehnici de blocare exclusivă constând din:

- directive de compilare **#pragma omp critical** care asigură faptul că un set de date nu este accesat la un moment dat decât de o unică secvență de date dintr-o echipă; dezavantajul acesttei tehnici constă în faptul că interblocajele trebuiesc stabilite de la început și nu pot fi alocate dinamic;
- directiva de compilare atomic care poate asigura exclusiv numai protecția instrucțiunilor simple - tipice operatorilor binari (de tip +, *, -, /, &, ...); aceasta deoarece acțiunea or se bazează pe protecția intrucțiunilor de procesor ce se ocupă cu preluarea datelor, iar apoi stocarea rezultatului în zona de memorie vizată, prelucrarea efectivă a datelor fiind realizată numai în procesor;
- folosirea unor funcții de interblocare (*lock*); cu ajutorul acestor funcții, blocările pot fi alocate dinamic, după necesități;

Folosirea tehnicilor de mai sus prezintă un risc major de blocare pe termen nedefinit a unora dintre procesele paralele implicate ([124]-Cap.5).

O altă metodă a *OpenMP* de tratare a porțiunilor de cod critice, este aceea oferită de utilizarea directivelor de tip **#pragma omp parallel for**; ele dau posibilitatea distribuirii (automate) a iterațiilor unui bloc de program iterativ între membrii echipei ce execută segmentul de program respectiv, astfel încât fiecare membru al echipe să se ocupe exclusiv de un anumit interval de itarații din mulțimea totală ce trebuie executate. Deși procedura este una automată și relativ simplu de implementat prin directiva susaminitită, ea nu este lipsită de riscul de a genera rezultate greșite: setul de calcule ce aparțin fiecărei iterații trebuie să fie inițiate și să dea rezultatul corespunzător în interiorul *acelei* iterații; cu alte cuvinte algoritmilor iterativi de tipul a[k] = f(a[k-1]), unde k este contorul de iterații, nu li se poate aplica această tehnică, deoarece există situația în care a[k-1] poate fi calculat de un thread, iar a[k] va fi calculat de thread-ul următor - care are un set de date privat, independent de primul menționat. Dacă algoritmul nu poate fi exprimat în alt fel atunci acesta nu poate fi paralelizat prin această metodă.

Asemănător tehnicilor utilizate cu funcțiile *Pthreads*, există directive de compilare *OpenMP* de tip **# pragma omp barrier** cu care se pot stabili bariere, în sensul de a suspenda execuția unor thread-uri până ce toate îndeplinesc o anume condiție.

Și tot ca în cazul bibliotecii *Pthreads*, întrucât și în cazul standardului *OpenMP* toate thread-urile au acces la aceeași bibliotecă de funcții, precum și la variabilele globale utilizate de acestea, același cod poate produce rezultate diferite pentru același set de date, în funcție de procesele paralele care accesează în mod asincron, necontrolat funcțiile și variabilele lor gobale (publice).

1.6.3 Medii software pentru sisteme cu memorie distribuită. Modelul MPI

MPI (*Message-Passing Interface*) este o bibliotecă de funcții dedicate programelor ce pot rula în paralel pe sisteme de calcul cu memoria distribuită (v.cap.1.3.2, 1.3.3). Funcțiile sunt posibil de apelat în programe scrise în C, C++, sau Fortran. In esență, MPI este un *protocol de comunicație* utilizat de procesele ce lucrează în paralel pe mai multe nuclee de procesor ([106]). Procesele comunică între ele prin *mesaje*; astfel întotdeauna va exista un *emițător* care inițiază un mesaj către un *receptor*.

La fel ca și în cazul *OpenMP*, o bună parte din "bucătăria" proceselor ce rulează în paralel este generată în faza de compilare. Mulțimea proceselor ce vor rula în paralel formează un *comunicator* și sunt inițiate prin apelarea funcției MPI_Init, iar pe parcursul execuției, comunicatorul poate fi adresat prin variabila standard MPI_COMM_WORLD. În majoritatea cazurilor, un comunicator conține procese ce execută același set de operații pe partiții diferite ale mulțimii de date de intrare, după modelul SPMD (v.fig.1.18). Incheierea acestor procese se realizează prin

funcția MPI_Finalize.

Pe de altă parte MPI, în formă în care a fost conceput inițial, nu este potrivit pentru sistemele hardware cu memorie partajată, fiind considerat ca un model de programare paralelă complementar cu *Pthreads* și *OpenMP*; mai nou însă, standardul MPI 3.0 conține și extensia MPI SHM, dedicată comunicării între procesele ce rulează pe sisteme cu memorie partajată ([44]).

Aspectele critice în programele generate cu biblioteca MPI pot fi împărțite în două mari clase:

- aspecte legate de comunicația dintre procesele paralele;
- aspecte legate de partiționarea datelor între procesele paralele;

Comunicația poate fi atât exclusiv între două procese (unu la unu), dar poate fi și colectivă: de la un proces - la toate, de la toate la unul singur, sau de la toate - către toate procesele.

Principala funcție de transmisie este MPI_Send, iar cea de recepție MPI_Recv. Odată apelată, funcția MPI_Recv blochează procesul până la momentul recepționării complete a mesajului. În schimb, funcția MPI_Send poate acționa în două moduri diferite:

- 1. blochează procesul până la momentul găsirii receptorului specificat și începerii transmiterii mesajului către acesta;
- 2. folosește un buffer de transmisie, caz în care funcția blochează procesul doar până când copiază mesajul de transmis în acest buffer, urmând ca identificarea receptorului și transmisia efectivă către acesta să se efectueze ulterior, fără a mai ține blocat procesul.

Din această cauză, chiar dacă toate procesele unui comunicator, practic, pornesc simultan, iar acțiunea lor este identică, dar pe seturi diferite de date, după un anumit interval de timp procesele se află în stadii diferite de execuție în funcție de modul de partiționare a datelor de intrare, precum și de încărcarea fiecărui nucleu de procesor pe care aceste procese rulează. Prin urmare culegerea completă a datelor de ieșire nu se poate efectua decât după o sincronizare a proceselor amintite. In plus, un program paralel MPI, este considerat nesigur, dacă are un comportament dependent de alegerea modelului de funcție MPI_Send.

Comunicația colectivă se poate efectua prin mai multe clase de funcții și anume ([124]-cap.3, [106]-cap.7):

- MPI_Bcast comunicație de la un singur proces la toate din același comunicator;
- MPI_Scatter distribuie elementele unei mulțimi de date între procese, pe cât posibil în mod uniform;
- MPI_Gather restaurează o mulțimea de date din partiții ale sale distribuite în procesele comunicatorului (este funcția inversă funcției MPI_Scatter);
- MPI_Barrier se ocupă de sincronizarea proceselor; procesele implicate sunt blocate până ce toate au apelat funcția MPI_Barrier.

Tot datorită comunicațiilor este posibilă atingerea unei stări de blocaj total a programului (*deadlock*), corespunzătoar situației în care procesele din setul Aal unui comunicator sunt în așteptare de mesaje de la procesele din setul B al aceluiași comunicator și invers, procesele din setul B sunt în așteptare de mesaje de la procesele din setul A.

Partiționarea datelor în programele generate cu MPI poate influența mult viteza de execuție a acestora din urmă. Există mai multe tehnici de a partiționa mulțimea datelor de intrare:

- a) în blocuri de date: dacă mulțimea conține n elemente, iar comunicatorul inițiază p procese, atunci fiecare bloc de date va conține n/p elemente ce vor fi prelucrate de câte un proces din comunicator;
- b) cele n elemente ale setului de date de intrare vor fi disribuite circular între procesele comunicatorului, astfel încât, elementul 1 va fi preluat de procesul 1, elementul p va fi preluat de către procesul p, elementul p+1 de către procesul 1, ș.a.m.d.;
- c) mulțimea datelor de intrare va fi partiționată în m blocuri de date care vor fi distribuite circular între procesele comunicatorului, după modelul descris la punctul b).

Modul de partiționare a datelor influențează puternic numărul de mesaje necesar a fi interschimbate între procese. Generarea și transmisia/recepția mesajelor reprezintă factorii hotărâtori ce determină efortul de timp solicitat de implementarea paralelă cu MPI. *Gradul de accelerare* obținut la trecerea unui algoritm de la o implementare serială la una paralelă se determină ca raportul dintre efortul de timp necesar rulării variantei seriale și cel necesar rulării variantei paralele; folosind notațiile de mai sus, în mod ideal acest raport este egal cu *p. Eficiența* trecerii de la serial la paralel se determină ca raportul dintre gradul de accelerare și p, în mod ideal fiind 1. Folosirea unui număr mare de procese pentru o dimensiune mică a problemei (n mic), de cele ai multe ori implică o comunicație mai intensivă între procese, față de cazurile în care raportul n/p este mare.

1.7 Serverul HPC Atlas al LMN. Concluzii



Figura 1.19: Structura hardware a serverului ATLAS al Laboratorului e Modelare Numerică din UPB; un rack conține 56 CPU quad core fabricație INTEL/model NEHALEM X5550 ([88]), iar celălalt rack conține 56 CPU quad core fabricație AMD/model BARCELONA ([87]), în total 112 nuclee (core) de procesare.

Concluzia acestui capitol este legată în mod direct de problemele științifice și inginerești actuale, care au o complexitate foarte mare necesitând în consecință, pentru a fi rezolvate, o foarte mare putere de calcul. La ora actuală, această putere de calcul se obține prin utilizarea a tot mai multe procesoare care operează în paralel. Pentru programarea acestora, în vederea rezolvării problemelor de natură științifică, s-a dezvoltat o nouă disciplină numită "Calcule Științifice de Inaltă Performanță" -în limba enleză High Performance Scientific Computing - de unde prescurtarea HPSC. Caracteristica esențială a HPSC este interdisciplinaritatea: abordarea ei presupune cunoștințe de fizică, inginerie, matematică aplicată împreună cu metodele numerice asociate, precum și de tehnologia informației - referitor în principal la sistemele de caclul - atât hardware (arhitecturi de procesoare, respectiv sisteme paralele), precum și software (programare de aplicații pentru procesare paralelă). De aceea, în capitolele ce urmează, vor fi abordate câteva aspecte legate atât de legile fizice fundamentale ce guvernează funcționarea dispozitivelor MEMS, precum și de aspectele esențiale matematice și de calcul numeric ce intervin în simularea acestor dispozitive. In același context, deloc lipsită de importanță este descrierea tehnicii de calcul folosite pentru simulările practice efectuate pentru elaborarea prezentei lucrări: aici mă refer la structura și caracteristicile principale ale serverului HPC al Laboratorului de Modelare Numerică (LMN) - al UPB, cunoscut sub numele de ATLAS (v. fig.1.19).

Cele 112 nuclee de procesare pot executa simultan 224 secvențe de program (thread-uri), întrucât fiecare nucleu are implementată proprietatea de *multi-threading* fiind astfel "văzut" de către sistemul de operare ca două procesoare logice diferite (care, în realitate, partajează aceeași resursă fizică oferită de nucleu). Capacitatea de calcul este estimată la 1 Tflop/sec. Cele două grupări de câte 56 CPU (v. fig.1.19), comunică între ele prin două modalități:

- la nivel de administrare prin interfețe ethernet grupate prin câte un switch CISCO (SW1 și SW2) de mare viteză, model CATALYST 2960-X ([2]);
- la nivel de procesare paralelă prin interfețe cu protocol InfiniBand ([1])- tipic comunicației interservere dedicate HPC, interfețe conectate la un switch de protocol InfiniBand cu 24 de porturi (IB).

Capacitatea de memorare totală a serverului ATLAS este de 270Gb memorie volatilă (RAM), respectiv 10Tb pe Hard disk-uri.

Capitolul 1. Calcul științific de înaltă performanță

Capitolul 2

Aspecte fizice

2.1 Modelare multifizică

Modelarea echipamentelor moderne necesită simularea cuplajului dintre mai multe fenomene fizice care interacționează pe parcursul funcționării respectivului dispozitiv. De regulă, sunt cuplate fenomene electromagnetice, mecanice, termice, de curgerea fluidelor. Procesul de modelare presupune deci și cuplarea cunoștințelor referitoare la fenomenele fizice pomenite mai sus. Prin urmare, este necesară buna înțelegere a legăturilor dintre:

- fenomenele fizice;
- aparatele matematice utilizate pentru modelarea acestor fenomene fizice;
- aparatele computaționale ce permit tratarea numerică a formulărilor matematice de mai sus.

Pentru a genera un model multifizic sunt necesare parcurgerea următoarelor etape:

 Modelarea conceptuală: etapa presupune stabilirea ipotezelor simplificatoare și aspectelor fizice neglijabile ce conduc atât la concepția unei descrieri geometrice a dispozitivului de simulat cât și la descrierea proprietăților de material ale entităților ce compun acest dispozitiv. Modelul geometric folosit trebuie redus astfel încât să reflecte numai acele elemente esențiale pentru funcționarea corectă a echipamentului de simulat; se poate renunța la toate detaliile de natură exclusiv tehnologică. Este necesară, pe cât posibil, asimilarea formelor elementelor selectate cu forme geometrice simple, astfel încât ele să poată fi ușor de descris în format electronic, dar să reprezinte și din punct de vedere matematic, suprafețe netede, cu un număr finit, de puncte de discontinuitate (colțuri, muchii, tăieturi); Tot în această etapă, trebuie reținute toate acele aproximări geometrice care influențează precizia de simulare, aceste aproximări ducând de fapt la o trunchiere a modelului real; după validarea modelului rezultat din prima aproximație, se pot efectua ajustări fine ale modelului geometric în scopul îmbunătățirii preciziei de simulare;

- Modelarea matematică:
 - în primul rând, se generează așa zisa formulare tare a problemei prin scrierea ecuațiilor caracteristice legilor (de obicei în forma lor locală) ce guvernează fizica funcționării dispozitivului simulat; acestor ecuații li se adaugă cele constitutive care reflectă influența proprietăților fizice al materialelor din construcția dispozitivului; setul de ecuații se completează cu relațiile corespunzătoare surselor fenomenului, care pot fi fie interne, fie reprezentabile prin condiții ce se impun pe frontierele dispozitivului modelat. În majoritatea cazurilor sistemul de ecuații rezultat este unul cu derivate parțiale. Este foarte important apoi de a preciza și spațiile vectoriale în care se caută soluțiile sistemului de ecuații ce caracterizează această formulare matematică.

Este foarte util a demonstra că problema matematică descrisă de ecuațiile de mai sus este bine formulată, această demonstrație fiind de multe ori dificil de efectuat;

- în al doilea rând, prin proiectarea ecuațiilor din formularea tare, pe un spațiu de funcții de test, se obține formularea slabă - în sens generalizat (distribuțional) a problemei; unul dintre avantajele majore ale formulării slabe este că reduce cu un ordin gradul sistemului de ecuații diferențiale asociat formulării matematice tari. Totodată soluțiile formulării slabe nu trebuie să fie netedete peste tot domeniul ci netede pe porțiuni (aproape peste tot); în acest caz nu mai este necesară, pentru funcțiile soluție, demonstrarea derivabilității de ordin k în fiecare punct din spațiu, ci este suficientă demonstrarea integrabilității pe porțiuni a derivatelor de ordin k-1. Și pentru formularea slabă este necesară precizarea spațiilor vectoriale Hilbert în care se caută soluțiile, astfel încât să poată fi posibilă demonstrarea existenței și a unicității soluțiilor în sens generalizat.

De altfel, de alegerea acestor spații va depinde și definirea cadrului funcțional pe care se va genera modelul numeric asociat problemei de rezolvat.

- Modelarea analitic-aproximativă: se determină relațiile între mărimile de intrare și cele de ieșire, in formă analitică, căutându-se o aproximare suficient de simplă pentru a putea rezolva analitic ecuațiile modelului; cu toate că precizia de modelare este redusă (modelul nu poate fi luat în considerare pentru realizarea practică) etapa este foarte importantă deoarece creează o imagine a limitelor între care evoluează fenomenele fizice ce guvernează funcționarea dispozitivului respectiv;
- Modelarea numerică: se construiește un algoritm numeric dedicat rezolvării ecuațiilor generate de către modelul matematic; în amajoritatea cazurilor metoda constă în discretizarea spațială a domeniului / domeniilor problemei, urmând ca soluția să fie reprezentată ca o combinație liniară de funcții de bază ce generează pe aceste domenii, spațiile vectoriale Hilbert precizate în cadrul formulării matematice slabe. Funcțiile de bază sunt polinoame ortogonale de diferite tipuri (Lagrange, Jacobi, Laguerre, Hermite) și grade, astfel încât orice funcție aparținând spațiilor Hilbert precizate în cadrul formulărilor matematice, poate fi aproximată cu o combinație liniară de astfel de polinoame ([139],[96]). Finețea de discretizare, precum și gradele polinoamelor menționate stabilesc robustețea și precizia algoritmului numeric.
- Verificarea și validarea modelului: se implementează algoritmul de rezolvare numerică pe un sistem de calcul și se realizează o serie de simulări, ale căror rezultate sunt folosite pentru a valida modelul elaborat prin comparație cu simulări similare, sau cu măsurători practice efectuate pe dipozitive similare existente.

Este de mare interes totodată, optimizarea întregului proces de simulare multifizică: se caută metode de reducere a efortului de calcul și timp pentru simulări, în condițiile păstrării, sau îmbunătățirii preciziei de simulare, respectiv de stabilitate a algoritmilor numerici utilizați. În acest sens, există mai multe abordări, printre care una dintre cele mai utilizate este cea a reducerii ordinului de complexitate a modelului (Model Order Reduction - MOR) [83, 140, 160, 82, 65]. Una dintre cele mai interesante lucrări din lista de mai sus este [83] în care sunt prezentate într- o manieră coerentă etapele modelării electromagnetice și ale etapelor de reducere a complexității. Este accentuat faptul că *reducerea complexității* modelului nu se rezumă doar la o etapă finală a procesului de simulare ci, pentru a avea eficiență maximă, ea *trebuie aplicată în toate etapele modelării*.

2.2 Mărimi fizice

Fenomenele fizice evoluează în spațiu și/sau în timp astfel încât mărimile fizice cu ajutorul cărora se descriu diverse domenii fizice se pot clasifica astfel:

- Locale atunci când sunt asociate unui punct din spațiu; aceste mărimi se încadrează în clasa formelor diferențiale [101], [37], [121], iar ordinul lor este în funcție de dimensiunile spațiului (în sens larg - a varietății) pe care ele se integrează (de ex. pe spații lineice, de suprafață, de volum, ele sunt de ordinul unu, doi, respectiv trei);
- Globale atunci când pot fi asociate unei mulțimi infinite de puncte din spațiu; mărimile globale se pot determina univoc prin integrarea mărimilor locale; operația inversă însă, nu este întotdeauna univocă [101], întrucât derivarea mărimii integrale nu presupune întotdeauna determinarea tuturor componentelor formei diferențiale corespondente;
- Instantanee dacă sunt asociate unui moment de timp;
- De proces dacă sunt definite pe un anumit interval de timp.

In general, mărimile fizice locale sunt mărimi primitive, în timp ce mărimile fizice globale sunt mărimi derivate. Mărimile cu ajutorul cărora se pot determina efectele evoluției fenomenului fizic (măsuri) sunt mărimi globale.

2.3 Câmp electromagnetic

2.3.1 Mărimile caracteristice locale ale câmpului electromagnetic

Un punct în spațiu este caracterizat complet din punct de vedere electromagnetic de către următorul set de vectori:

- E intensitatea câmpului electric;
- D inducția electrică;
- H intensitatea câmpului magnetic;
- **B** inducția magnetică;



Figura 2.1: Mărimile fizice locale ale câmpului electromagnetic.

În fig. 2.1 punctul din spațiu este identificat prin vectorul de poziție $\mathbf{r} = r_x \mathbf{i} + r_y \mathbf{j} + r_z \mathbf{k}$. Vectorii \mathbf{E} , \mathbf{D} , \mathbf{H} , \mathbf{B} sunt tridimensionali, având în coordonate carteziene o expresie de forma $\mathbf{V} = V_x \mathbf{i} + V_y \mathbf{j} + V_z \mathbf{k}$. Mărimile locale de mai sus reprezintă funcții cu argument spațio-temporal, spațiul fiind un subdomeniu Ω al \mathbb{R}^3 . Forma generală a acestor funcții poate fi definită astfel:

$$\mathbf{V} = \mathbf{f}(\mathbf{r}, t) \, cu \, \mathbf{f} : \Omega \times [t_i, t_f] \to \mathbb{R}^3, \Omega \subset \mathbb{R}^3.$$
(2.1)

unde t_i, t_f reprezintă momentele inițial, respectiv final ale intervalului de timp pe care se reprezintă mărimea locală.

Intensitățile câmpului electric, respectiv a câmpului magnetic fac parte din clasa formelor diferențiale de ordin unu, iar inducțiile electrică, respectiv magnetică fac parte din clasa formelor diferențiale de ordin doi. Acest fapt este pus în evidență la definirea mărimilor globale rezultate prin integrarea celor locale.

2.3.2 Mărimile locale ale corpurilor în interacțiune cu câmpul electromagnetic

Schimbarea stării electromagnetice a corpurilor în urma interacțiunii dintre acestea și câmpul electromagnetic sunt puse în evidență prin următoarele mărimi locale:

- densitatea de sarcină ρ care reflectă starea de electrizare;
- densitatea de curent ${\bf J}$ care reflectă starea electrocinetică.

Ambele sunt funcții de spațiu și timp - ρ o funcție scalară, iar **J** o funcție vectorială. In funcție de tipul de domeniu spațial pe care îl caracterizează, densitatea de sarcină reprezintă o formă diferențială de ordin unu, doi sau trei, pe când densitatea de curent reprezintă o formă diferențială de odin doi (v fig. 2.2):

$$\rho = \rho(\mathbf{r}, t) \, cu \, \rho : \Omega \times [t_i, t_f] \to \mathbb{R}$$
(2.2)

$$\mathbf{J} = \mathbf{J}(\mathbf{r}, t) \, cu \, \mathbf{J} : \Omega \times [t_i, t_f] \to \mathbb{R}^3$$
(2.3)

$$s_i \mathbf{J} = J_x \mathbf{i} + J_y \mathbf{j} + J_z \mathbf{k} \tag{2.4}$$



Figura 2.2: Mărimile fizice locale ale corpurilor în interacțiune cu câmpul electromagnetic.

2.3.3 Mărimile globale ale câmpului electromagnetic

Mărimile globale ale câmpului electromagnetic, se obțin prin integrarea mărimilor locale prezentate mai sus:

• tensiunea electrică (Fig. 2.3 - (a)):

$$u(t) = \int_C \mathbf{E}(\mathbf{r}, t) \cdot d\mathbf{r}$$
(2.5)

• fluxul electric (Fig. 2.3 - (b)):

$$\psi(t) = \int_{S} \mathbf{D}(\mathbf{r}, t) \cdot \mathbf{n} \, dA \tag{2.6}$$

• tensiunea magnetică (Fig. 2.3 - (a)):

$$u_m(t) = \int_C \mathbf{H}(\mathbf{r}, t) \cdot d\mathbf{r}$$
(2.7)

• fluxul magnetic (Fig. 2.3 - (b)):

$$\phi(t) = \int_{S} \mathbf{B}(\mathbf{r}, t) \cdot \mathbf{n} \, dA \tag{2.8}$$

• sarcina electrică - obtinută prin integrare pe domeniul spatial pe care este definită densitatea de sarcină (Fig. 2.3 - (c)):

$$q(t) = \int_C \rho_l(\mathbf{r}, t) \, dr \tag{2.9}$$

$$q(t) = \int_{S} \rho_{A}(\mathbf{r}, t) \, dA \tag{2.10}$$

$$q(t) = \int_{V} \rho_{V}(\mathbf{r}, t) \, dV \tag{2.11}$$

unde ρ_l, ρ_A, ρ_V reprezintă densitățile de sarcină lineică, de suprafață, respectiv de volum.

• intensitatea curentului electric (Fig. 2.3 - (b)):

$$i(t) = \int_{S} \mathbf{J}(\mathbf{r}, t) \cdot \mathbf{n} \, dA. \tag{2.12}$$

Se poate observa că toate mărimile globale ale câmpului electromagnetic sunt funcții scalare ce caracterizează un întreg domeniu spațial, în fiecare moment de timp, de la un moment inițial (t_i) până la unul final (t_f) :

$$f(t), cuf : [t_i, t_f] \to \mathbb{R}.$$
(2.13)

2.3.4Legile generale ale câmpului electromagnetic

Vor fi prezentate în continuare - pe scurt - legile generale ale câmpului electromagnetic pentru medii imobile și pentru materiale cu caracteristici electrice afine. Formulările complete pot fi găsite în [56], [113], [148], [130].

Legea 2.3.1. - a fluxului electric. Fluxul electric $\psi_{\partial\Omega}$ prin orice suprafață închisă $\partial \Omega$ este egal cu sarcina electrică q_{Ω} inclusă în domeniul Ω , închis de acea suprafață.

$$\psi_{\partial\Omega} = q_{\Omega} \Rightarrow \oint_{\partial\Omega} \mathbf{D} \cdot \mathbf{n} \, dA = \int_{\Omega} \rho \, dv \, \text{-forma integral} \breve{a}; \qquad (2.14)$$
$$\nabla \cdot \mathbf{D} = \rho \quad \text{-forma local} \breve{a}. \qquad (2.15)$$

$$\mathbf{V} \cdot \mathbf{D} = \rho$$
 - forma locală. (2.15)



Figura 2.3: Mărimile fizice globale ale câmpului electromagnetic obținute prin integrarea mărimilor locale pe varietăți de ordin 1 (a), (c), respectiv pe varietăți de ordin 2 (b), (c) și de ordin 3 (c).

Legea arată că electrizarea corpurilor (densitatea sarcinii electrice ρ) este una din sursele câmpului electric. Tot de aici, rezultă conservarea componentelor normale (transversale) ale inducției electrice **D** la trecerea printr-o suprafață de discontinuitate ne-electrizată.

Legea 2.3.2. - a fluxului magnetic. Fluxul magnetic $\phi_{\partial\Omega}$ prin orice suprafață închisă $\partial\Omega$ este nul.

$$\phi_{\partial\Omega} = 0 \Rightarrow \oint_{\partial\Omega} \mathbf{B} \cdot \mathbf{n} \, dA = 0 - forma \ integral \ddot{a}; \tag{2.16}$$

$$\nabla \cdot \mathbf{B} = 0 - forma \ local \breve{a}. \tag{2.17}$$

Rezultă inexistența - pentru câmpul magnetic - a unei surse (sarcină magnetică adevărată) echivalentă cu sarcina electrică, dar este pusă în evidență conservarea

componentelor normale (transversale) ale inducției magnetice \mathbf{B} la trecerea printr-o suprafață de discontinuitate.

Legea 2.3.3. - a inducției electromagnetice. Tensiunea electrică în lungul unei curbe închise ∂S este egală cu viteza de scădere a fluxului magnetic prin orice suprafață S mărginită de această curbă.

$$u_{\partial S} = -\frac{d\phi_S}{dt} \Rightarrow \oint_{\partial S} \mathbf{E} \cdot d\mathbf{r} = -\frac{d}{dt} \int_S \mathbf{B} \cdot \mathbf{n} \, dA - forma \; integral \breve{a}; \quad (2.18)$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} - forma \ local \breve{a}.$$
 (2.19)

Este evidențiată astfel o a doua sursă de câmp electric: un câmp magnetic variabil (în timp); din această lege rezultă totodată conservarea componentelor tangențiale (longitudinale) ale intensității câmpului electric \mathbf{E} la trecerea printr-o suprafață de discontinuitate.

Legea 2.3.4. - a circuitului magnetic. Tensiunea magnetică în lungul unei curbe închise ∂S este egală cu suma dintre intensitatea curentului ce traversează o suprafață oarecare S mărginită de curba ∂S și viteza de creștere a fluxului electric prin acea suprafață.

$$u_{m\partial S} = i_S + \frac{d\psi_S}{dt} \Rightarrow \tag{2.20}$$

$$\oint_{\partial S} \mathbf{H} \cdot d\mathbf{r} = \int_{S} \mathbf{J} \cdot \mathbf{n} \, dA + \frac{d}{dt} \int_{S} \mathbf{D} \cdot \mathbf{n} \, dA - forma \ integral \breve{a}; \qquad (2.21)$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t} - forma \ locală \ \hat{i}n \ medii \ imobile.$$
 (2.22)

Din această lege rezultă ca surse ale câmpului magnetic curentul electric de conducție și cel de deplasare (viteza de variație a inducției electrice); în același timp, este evidențiată conservarea componentelor tangențiale (longitudinale) ale intensității câmpului magnetic \mathbf{H} la trecerea printr-o suprafață de discontinuitate.

Sistemul formelor locale ale legilor câmpului electromagnetic în medii imobile poartă numele de ecuațiile lui Maxwell. Acestea sunt ecuațiile fundamentale ale electromagnetismului, dar în aplicații se folosesc forme simplificate ale lor, valabile în diferite regimuri: electrostatică, magnetostatică, electrocinetică, magnetic staționar, electrocvasistaționar, magnetocvasistaționar, etc.

2.3.5 Teoremele de conservare ale câmpului electromagnetic

Legătura dintre sarcina electrică și intensitatea curentului electric este pusă în evidență de următoarea teoremă:

Teorema 2.3.1. a conservării sarcinii electrice. Curentul electric ce iese dintrun domeniu Ω prin frontiera $\partial \Omega$ ce închide acest domeniu, este egal cu viteza de scădere a sarcinii electrice din Ω :

$$i_{\partial\Omega} = -\frac{dq_{\Omega}}{dt} \text{ în formă globală, respectiv}$$
$$\int_{\partial\Omega} \mathbf{J} \cdot \mathbf{n} \, dA = -\frac{d}{dt} \int_{\Omega} \rho \, dV \text{ în formă integrală.}$$

Pentru medii mobile se consideră că $\partial\Omega$, este antrenată de corpurile, conținute în Ω , aflate în mișcare.

In cazul mediilor imobile, este pusă în evidență numai componenta de conducție a curentului, din forma integrală rezultând:

$$\nabla \mathbf{J} = -\frac{\partial \rho}{\partial t}.$$
(2.23)

(2.23) atrage după sine conservarea componentei normale a densității de curent **J** la trecerea printr-o suprafață de discontinuitate neîncărcată electric. Ca urmare, în regim staționar, rezultă *continuitatea liniilor de curent electric de conducție*, din care derivă și prima lege a lui Kirchhoff.

In cele ce urmează, considerăm un sistem fizic format din corpuri ce interacționează între ele exclusiv prin intermediul câmpului electromagnetic. Intreg sistemul este conținut într-un domeniu Ω , închis de suprafața $\partial\Omega$. Suprafața $\partial\Omega$ este suficient de mare astfel încât mișcarea corpurilor din interiorul său să nu o antreneze și pe aceasta, astfel încât vectorul viteză în orice punct pe $\partial\Omega$ este nul. Interacțiunile exclusiv electromagnetice dintre corpurile din Ω , iar apoi dintre sistemul Ω și exterior, pot fi caracterizate în mai multe moduri și anume:

1. prin forțe generalizate X_k reprezentând mărimi de natură mecanică (forțe, cupluri, presiuni,...) care, prin efectuarea unui lucru mecanic, pot modifica poziția relativă a corpurilor ce constituie sistemul fizic din Ω ; pozițiile corpurilor sunt definite cu ajutorul unor mărimi de natură geometrică numite coordonate generalizate, notate cu x_k (distanțe, unghiuri, suprafețe, volume). Lucrul mecanic elementar efectuat de o forță generalizată este $dL_k = X_k \cdot dx_k$. 2. prin *forțe de volum* ce caracterizează local acțiunile de natură electromagnetică, prin *densitatea de volum a forței*:

$$\mathbf{f} = \lim_{\Delta V \to 0} \frac{\Delta \mathbf{F}}{\Delta V} = \frac{\mathrm{d} \mathbf{F}}{\mathrm{d} V};$$

3. prin tensiuni maxwelliene ce se pot determina presupunând că interacțiunea electromagnetică, a sistemului fizic din Ω cu exteriorul, se realizează exclusiv prin suprafața închisă $\partial\Omega$. Tensiunile maxweliene apar ca aplicații liniare, ce asociază normalei **n**, într-un punct, la suprafața $\partial\Omega$ componenta normală a forței ce acționează în acel punct:

$$\mathbf{T}_n = \overline{\mathbf{T}} \cdot \mathbf{n},$$

fiind deci un tensor de ordin 2. Echivalența dintre acțiunea forțelor de volum \mathbf{F} și cea a tensiunilor maxwelliene este dată de egalitatea dintre \mathbf{F} și fluxul tensorului $\overline{\overline{\mathbf{T}}}$ pe suprafața închisă $\partial \Omega$ ([149]):

$$\mathbf{F} = \int_{\Omega} \mathbf{f} \, dV = \int_{\partial \Omega} \overline{\overline{\mathbf{T}}} \cdot \mathbf{n} \, dA. \tag{2.24}$$

Tratarea detaliată a celor trei tipuri de acțiuni electromagnetice, precum și modul de exprimare a balanței energetice a sistemului fizic Ω se pot găsi în [149], [56], [113].

Pentru cazul corpurilor imobile constituite din materiale fără histerezis, balanța energetică, menționată mai sus, este dată de următoarea:

Teorema 2.3.2. a energiei electromagnetice. Puterea transferată de câmpul electromagnetic unui domeniu Ω prin frontiera acestuia $\partial\Omega$ este egală cu suma dintre puterea transferată corpurilor și viteza de creștere a energiei câmpului electromagnetic din Ω :

$$P_{\partial\Omega} = P_{\Omega} + \frac{\partial W_{em}}{\partial t}, \qquad (2.25)$$

In fiecare punct din domeniul Ω :

$$-\nabla(\mathbf{E} \times \mathbf{H}) = \mathbf{E} \cdot \mathbf{J} + \frac{\partial}{\partial t} \left(\frac{\mathbf{D} \cdot \mathbf{E}}{2} + \frac{\mathbf{B} \cdot \mathbf{H}}{2} \right).$$
(2.26)

unde pentru medii liniare imobile:

• $\mathbf{S} = \mathbf{E} \times \mathbf{H}$ - este vectorul Poynting, iar puterea transferată prin frontiera $\partial \Omega$ este $P_{\partial \Omega} = \int_{\partial \Omega} \mathbf{S} \cdot \mathbf{n} \, dA;$

- $p = \mathbf{E} \cdot \mathbf{J}$ este densitatea de volum a puterii transferată de câmpul electromagnetic corpurilor și $P_{\Omega} = \int_{\Omega} p \, dv;$
- $w_{em} = w_e + w_m = \mathbf{D} \cdot \mathbf{E}/2 + \mathbf{B} \cdot \mathbf{H}/2$ este densitatea de volum a energiei electromagnetice reprezentată de suma densităților de energie electrică (w_e) , respectiv magnetică (w_m) , cu $W_{em} = \int_{\Omega} w_{em} dv$.

Teorema 2.3.2 permite punerea în evidență a expresiilor forțelor generalizate (forță, moment, sau presiune) de natură electrică și magnetică ce se exercită asupra unui sistem cu n grade de libertate, forțe care pentru medii liniare au următoarele forme:

$$X_{ke} = -\frac{\partial W_e}{\partial x_k}\Big|_{q=ct.} \quad \text{respectiv} \quad X_{ke} = \frac{\partial W_e}{\partial x_k}\Big|_{U=ct.}$$
(2.27)

$$X_{km} = -\frac{\partial W_m}{\partial x_k}\Big|_{\Phi=ct.} \quad \text{respectiv} \quad X_{km} = \frac{\partial W_m}{\partial x_k}\Big|_{I=ct.}$$
(2.28)

unde x_k reprezită coordonata generalizată de ordin $k = 1, 2, \dots, n$.

In câmpuri electromagnetice variabile ciclic, corpurile ce conțin materiale caracterizate de *histerezis electric* și/sau *histerezis magnetic* sunt caracterizate trebuiesc energetic, electromagnetic, de următoarea teoremă:

Teorema 2.3.3. Teorema lui Warburg. Densitatea energiei w_h transferată de la câmpul electromagnetic către corpuri odată cu parcurgerea unui ciclu de histerezis (magnetic și/sau electric) este egală cu aria mărginită de curba ce descrie ciclul de histerezis respectiv:

$$w_h = w_{hm} + w_{he} = \oint_{ciclu_{hm}} \mathbf{H} \cdot d\mathbf{B} + \oint_{ciclu_{he}} \mathbf{E} \cdot d\mathbf{D}.$$

Forțele de volum pot fi evaluate în corpuri aflate în mișcare în interiorul domeniului Ω , dar fără a antrena și suprafața $\partial \Omega$. Această mișcare nu este obligatoriu de corp rigid, ci poate caracteriza punctele unui corp care se deformează sub acțiunea câmpului electromagnetic. Pot exista deci porțiuni de material cu caracteristici constitutive dependente de vectorul poziție **r**, respectiv de densitatea de masă locală notată cu $\tau = \tau(\mathbf{r}, t)$, unde t este variabila timp, adică:

$$\varepsilon = \varepsilon(\tau(\mathbf{r}, t), \dots), \text{ respectiv}$$

 $\mu = \mu(\tau(\mathbf{r}, t), \dots).$

In acest caz, în expresiile forțelor de volum apar componente de magnetostricțiune, respectiv electrostricțiune, datorate influenței tensiunilor mecanice asupra parametrilor de material (ultimele din relațiile (2.30) de mai jos). Densitățile de volum ale forțelor magnetice, respectiv electrice se exprimă astfel ([149]):

$$\mathbf{f}_m = \mathbf{J} \times \mathbf{B} - \frac{H^2}{2} (\nabla \mu) + \nabla \left(\frac{H^2}{2} \tau \frac{\partial \mu}{\partial \tau} \right)$$
(2.30a)

$$\mathbf{f}_e = \rho \mathbf{E} - \frac{E^2}{2} (\nabla \varepsilon) + \nabla \left(\frac{E^2}{2} \tau \frac{\partial \varepsilon}{\partial \tau} \right).$$
(2.30b)

In (2.30a) primii doi termeni reprezintă densitatea de forță Laplace și un termen datorat neomogenității materialelor. In (2.30b) primii doi termeni reprezintă densitatea forței electrostatice coulombiene și un termen datorat neomogenității materialelor.

In virtutea relației (2.24), tensorul tensiune maxwelliană echivalent cu forțele (2.30) are patru componente și anume tensorii maxwellieni magnetic și electric, precum și tensorii corespunzători de magneto și electrostricțiune. Expresiile lor matematice se pot reprezenta cu operatorul produs diadic (notat aici cu :) - tipic calculului tensorial:

$$\overline{\overline{\mathbf{T}}}_{em} = \overline{\overline{\mathbf{T}}}_{e} + \overline{\overline{\mathbf{T}}}_{m} + \overline{\overline{\mathbf{T}}}_{es} + \overline{\overline{\mathbf{T}}}_{em}$$

$$\overline{\overline{\mathbf{T}}}_{e} = \mathbf{E} : \mathbf{D}^{T} - w_{e} \overline{\overline{\mathbf{I}}}$$

$$\overline{\overline{\mathbf{T}}}_{m} = \mathbf{H} : \mathbf{B}^{T} - w_{m} \overline{\overline{\mathbf{I}}}$$

$$\overline{\overline{\mathbf{T}}}_{ms} = \overline{\overline{\mathbf{I}}} \frac{H^{2}}{2} \tau \frac{\partial \mu}{\partial \tau}$$

$$\overline{\overline{\mathbf{T}}}_{es} = \overline{\overline{\mathbf{I}}} \frac{E^{2}}{2} \tau \frac{\partial \varepsilon}{\partial \tau}$$
(2.31)

unde w_e și w_m sunt densitățile de energie electrică, respectiv magnetică, iar $\overline{\overline{\mathbf{I}}}$ este matricea unitate de ordin 3. Prin urmare:

$$\overline{\overline{\mathbf{T}}}_{e} = \begin{pmatrix} E_{x}D_{x} - \frac{\mathbf{E}\cdot\mathbf{D}}{2} & E_{x}D_{y} & E_{x}D_{z} \\ E_{y}D_{x} & E_{y}D_{y} - \frac{\mathbf{E}\cdot\mathbf{D}}{2} & E_{y}D_{z} \\ E_{z}D_{x} & E_{z}D_{y} & E_{z}D_{z} - \frac{\mathbf{E}\cdot\mathbf{D}}{2} \end{pmatrix}$$
(2.32)

și

$$\overline{\overline{\mathbf{T}}}_{m} = \begin{pmatrix} H_{x}B_{x} - \frac{\mathbf{H}\cdot\mathbf{B}}{2} & H_{x}B_{y} & H_{x}B_{z} \\ H_{y}B_{x} & H_{y}B_{y} - \frac{\mathbf{H}\cdot\mathbf{B}}{2} & H_{y}B_{z} \\ H_{z}B_{x} & H_{z}B_{y} & H_{z}B_{z} - \frac{\mathbf{H}\cdot\mathbf{B}}{2} \end{pmatrix}$$
(2.33)

In cazul general, câmpul electromagnetic interacționează cu sistemul fizic din domeniul Ω atât prin suprafața închisă $\partial \Omega$ cât și prin forța electromagnetică totală \mathbf{F}_{em} , distribuită în volumul Ω . In absența acțiunilor mecanice ne-electromagnetice se poate pune în evidență componenta electromagnetică a impulsului și care se supune legii generale de conservare a acestuia (v. cap. 2.4.5). Forma electromagnetică a acestui prinicipiu fundamental al dinamicii este enunțat prin următoarea teoremă:

Teorema 2.3.4. Teorema impulsului electromagnetic. Viteza de variație a impulsului electromagnetic al unui domeniu fizic Ω este egală cu rezultanta tuturor forțelor externe de natură electromagetică ce acționează asupra domeniului. Această rezultantă este dată de fluxul tensiunilor lui Maxwell pe frontiera $\partial\Omega$, a domeniului, minus forța totală, cu care câmpul electromagnetic acționează asupra lui Ω . Forma sa integrală este dată de relația:

$$\frac{\partial \mathbf{G}_{em}}{\partial t} = \oint_{\partial \Omega} \overline{\overline{\mathbf{T}}}_{em} \cdot \mathbf{n} \, dA - \mathbf{F}_{em},$$

iar forma locală de:

$$\frac{\partial \mathbf{g}_{em}}{\partial t} = div(\overline{\overline{\mathbf{T}}}_{em}) - \mathbf{f}_{em},$$

unde:

$$\begin{cases} \mathbf{g}_{em} = \mathbf{D} \times \mathbf{B} & \text{este densitatea de impuls electromagnetic,} \\ \mathbf{G}_{em} = \int_{\Omega} \mathbf{g}_{em} \, dV & \text{este impulsul electromagnetic total.} \end{cases}$$

2.3.6 Ecuații constitutive

Pentru determinarea câmpului electromagnetic, ecuațiile date de cele patru legi de mai sus trebuie completate cu relațiile constitutive ce se stabilesc, între mărimile locale de câmp de ordin unu și doi, în funcție de proprietățile materialelor din fiecare punct al domeniul de calcul. Din acest punct de vedere, comportarea materialelor se clasifică în trei categorii: dielectrică, magnetică și conductoare. De aici cele trei legi de material caracteristice câmpului electromagnetic:

Legea 2.3.5. - a legăturii între D și E. Inducția electrică într-un punct din spațiu este o funcție de intensitatea câmpului electric din acel punct:

$$\mathbf{D} = f(\mathbf{E}, \mathbf{r}), \ f : \mathbb{R}^3 \times \Omega \to \mathbb{R}^3, \tag{2.34}$$

Funcția f poartă denumirea de caracteristică dielectrică. Dacă f nu depinde de **r** atunci mediul este un dielectric omogen. Pentru mediile izotrope și liniare:

$$\mathbf{D} = \varepsilon \mathbf{E} = \varepsilon_0 \varepsilon_r \mathbf{E},\tag{2.35}$$

unde ε reprezintă permitivitatea mediului și este exprimată ca produsul dintre constanta universală ε_0 (permitivitatea vidului) și mărimea fizică scalară, adimensională ε_r (permitivitatea relativă a mediului respectiv). Pentru vid $\varepsilon_r = 1$.

In cazul mediilor anizotrope, în reprezentare carteziană, permitivitatea este dependentă de coordonate după cele trei direcții ale sistemului de referință, fiind exprimată printr-un tensor de ordin 2, simetric și pozitiv definit:

$$\overline{\overline{\varepsilon}} = \begin{pmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{21} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{31} & \varepsilon_{32} & \varepsilon_{33} \end{pmatrix}; \text{ iar } \mathbf{D} = \overline{\overline{\varepsilon}} \mathbf{E}.$$
(2.36)

In dielectricii afini, relația constitutivă are forma:

$$\mathbf{D} = \overline{\overline{\varepsilon}} \mathbf{E} + \mathbf{P}_p, \tag{2.37}$$

în care \mathbf{P}_p este polarizația permanentă. Astfel de relație este folosită pentru modelarea electreților (dielectrici polarizați permanent).

Legea 2.3.6. - a legăturii între B și H. Inducția magnetică într-un punct din spațiu este o funcție de intensitatea câmpului magnetic din acel punct:

$$\mathbf{B} = f(\mathbf{H}, \mathbf{r}), \ f : \mathbb{R}^3 \times \Omega \to \mathbb{R}^3, \tag{2.38}$$

Funcția f poartă denumirea de caracteristică de magnetizare. Dacă f nu depinde de **r** atunci mediul este magnetic omogen. Pentru mediile izotrope și liniare:

$$\mathbf{B} = \mu \mathbf{H} = \mu_0 \mu_r \mathbf{H},\tag{2.39}$$

unde μ reprezintă permeabilitatea mediului și este exprimată ca produsul dintre constanta universală μ_0 (permeabilitatea magnetică a vidului) și mărimea fizică scalară, adimensională μ_r (permeabilitatea magnetică a mediului respectiv). Pentru medii nemagnetice (inclusiv vid) $\mu_r = 1$.

In cazul mediilor anizotrope, în reprezentare carteziană, permeabilitatea magnetică este dependentă de coordonate după cele trei direcții ale sistemului de referință, fiind exprimată printr-un tensor de ordin 2, simetric și pozitiv definit:

$$\overline{\overline{\mu}} = \begin{pmatrix} \mu_{11} & \mu_{12} & \mu_{13} \\ \mu_{21} & \mu_{22} & \mu_{23} \\ \mu_{31} & \mu_{32} & \mu_{33} \end{pmatrix}; \text{ iar } \mathbf{B} = \overline{\overline{\mu}} \mathbf{H}.$$
 (2.40)

In mediile magnetice afine relația constitutivă are forma:

$$\mathbf{B} = \overline{\mu} \mathbf{H} + \mu_0 \mathbf{M}_p, \tag{2.41}$$

în care \mathbf{M}_p este magnetizația permanentă și $\mu_0 \mathbf{M}_p = \mathbf{B}_r$ este inducția remanentă. Astfel de relație este folosită pentru modelarea mediilor feromagnetice dure (magneți permanenți).

Legea 2.3.7. - a legăturii între J și E. Densitatea de curent într-un punct din spațiu este o funcție de intensitatea câmpului electric din acel punct:

$$\mathbf{J} = f(\mathbf{E}, \mathbf{r}), \ f : \mathbb{R}^3 \times \Omega \to \mathbb{R}^3, \tag{2.42}$$

Pentru mediile izotrope și liniare:

$$\mathbf{J} = \sigma \mathbf{E}, \text{ respectiv } \mathbf{E} = \rho \mathbf{J}, \tag{2.43}$$

unde σ reprezintă conductivitatea electrică - mărime fizică scalară - exprimată în $\frac{S}{m}$, iar ρ reprezintă rezistivitatea electrică - mărime fizică scalară - exprimată în Ωm . In vid și izolatoare perfecte $\mathbf{J} = 0$ și $\sigma = 0$, iar în conductoare perfecte $\mathbf{E} = 0$ și $\rho = 0$.

In cazul mediilor anizotrope, în reprezentare carteziană, conductivitatea electrică este dependentă de coordonate după cele trei direcții ale sistemului de referință, fiind exprimată printr-un tensor de ordin 2, simetric și pozitiv definit:

$$\overline{\overline{\sigma}} = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{pmatrix}; \text{ iar } \mathbf{J} = \overline{\overline{\sigma}} \mathbf{E}.$$
 (2.44)

In mediile conductoare afine relația constitutivă are forma:

$$\mathbf{J} = \overline{\overline{\sigma}}(\mathbf{E} + \mathbf{E}_i),\tag{2.45}$$

în care \mathbf{E}_i este intensitatea câmpului electric imprimat, iar $\mathbf{J}_i = \sigma \mathbf{E}_i$ este densitatea curentului imprimat. Astfel de relație este folosită pentru modelarea surselor electrochimice (baterii și acumulatoare), dar și a diferitelor feluri de senzori și traductoare.

2.4 Mecanica mediilor continue. Teoria elasticității liniare

Mecanica mediilor continue studiază tensiunile mecanice, deformațiile, sau curgerea ce apar în solide și respectiv fluide atunci când asupra lor sunt aplicate diverse forțe (solicitări) din exterior. Materialele de studiu sunt presupuse a avea o distribuție spațială continuă (v. ipoteza (ME0)). Solidele sunt caracterizate de faptul ca au o formă de repaus stabilă, respectiv pot suporta solicitări de forfecare (paralele cu o secțiune transversală), pe când fluidele (gaze sau lichide) preiau forma vasului în care se află și nu pot prelua forțe de forfecare.

Din punctul de vedere al reacției la acțiuni exterioare, pentru solide se pot identifica două categorii de caracteristici :

- de elasticitate care descriu capacitatea unui material de a reveni la forma inițială odată cu dispariția forțelor exterioare; în acest caz, prin deformare, în structura materialului se acumulează o energie potențială de deformație ce reușește să "refacă" forma materialului după îndepărtarea solicitărilor exterioare;
- de plasticitate care descriu deformațiile permanente care apar în solid, odată cu exercitarea din afară a unor forțe suficient de mari pentru a produce aceste deformații.

Teoria elasticității liniare este aplicabilă materialelor solide ale căror caracteristici mecanice se încadrează în următoarele ipoteze:

- (ME0) a mediului continuu materialele pot fi divizate continuu în elemente de dimensiuni infinitezimale [147]. Proprietățile fizice ale acestora (de ex. densitatea, elasticitatea) sunt reprezentabile prin funcții continue și derivabile cel puțin pe porțiuni, fiind posibilă scrierea sistemelor de ecuații cu derivate parțiale;
- (ME1) a mediului omogen proprietățile fizce ale materialului sunt constante pe întreg volumul său;
- (ME2) a mediului izotrop caracteristicile mecanice sunt independente de direcție;
- (ME3) ipoteza micilor deformații deformațiile corpurilor sunt neglijabile în raport cu dmensiunile lor. De aici rezultă faptul că schimbarea sistemului de referință față de care este analizat corpul respectiv (configurație) lasă invariante direcțiile forțelor și distanțele dintre punctele lor de aplicație. Totodată această ipoteză permite neglijarea tuturor mărimilor ale căror expresii matematice, depind de puterile acestor mici deformații;
- (ME4) proprietăți elastice de material liniare ce presupun o dependență liniară între tensiuni și deformații (v. subcap. 2.4.6);

(ME5) absența tensiunilor în lipsa sarcinilor mecanice în corpurile solide, ipoteză care presupune lipsa în aceste corpuri, a oricăror tensiuni remanente în urma acțiunii unor solicitări mecanice, termice; această ipoteză asigură determinarea în mod unic a stării de deformare atunci când se cunosc geometria configurației, respectiv sarcinile mecanice la care aceasta este supusă.

Abordările mecanice se pot face din două puncte de vedere ([98], [85], [153], [36]):

- Newtonian conform căruia mărimile primitive sunt reprezentate de forțe, iar energiile (potențială și cinetică) sunt mărimi derivate; conform acestui punct de vedere, determinările necesare pentru caracterizarea solidelor se efectuează pe baza calculului vectorial;
- Hamiltonian conform căruia mărimile primitive sunt reprezentate de energii (potențială și cinetică), iar forțele sunt mărimi derivate; acest mod de abordare stă la baza calculului analitic și numeric pentru problemele de mecanică (mecanica analitică - [36]), fiind avantajos în principal din următoarele motive [86]:
 - 1. permite generarea formulărilor variaționale (formulări matematice slabe) care acceptă soluții care sunt inadmisibile pentru sistemele formulate cu ecuații locale (formulări matematice tari)(v. cap. următor);
 - 2. în contextul puctului de mai sus, permit demonstrarea existenței soluției unice pentru formulările matematice respective.

2.4.1 Configurații

Mecanica mediilor continue folosește pentru descrierea mișcărilor și deformațiilor corpurilor noțiunea de *configurație*, fie ea de *referință* pentru reflectarea unei stări inițiale, fie oarecare pentru reflectarea unor stări ulterioare celei de referință [67][152]. Notăm mulțimea punctelor ce compun un anumit corp, raportate la sistemul de referință a configurației inițiale, cu \mathcal{B} . Membrii acestei mulțimi poartă denumirea de *puncte materiale*, iar într-un sistem de referință tridimensional putem scrie:

$$\mathcal{B} = \{ X = (X_1, X_2, X_3) \, | \, X_i \in \mathbb{R} \}$$
(2.46)
Devine astfel posibilă exprimarea oricărei alte configurații a corpului respectiv, de exemplu la alt moment de timp, ca o aplicație bijectivă [152]:

$$\Phi: \mathcal{B} \to \mathbb{R}^3$$
, a. î. pentru $\forall X \in \mathcal{B} \exists x = \Phi(X) = (x_1, x_2, x_3) \in \mathbb{R}^3$. (2.47)

Punctele x din (2.47) poartă denumirea de *puncte spațiale*, iar cu ajutorul lor pot fi definite familii de configurații, sau *deformații*. În cazul în care care în funcțiile de tip (2.47) apare și variabila timp, pentru a exprima evoluția în timp a formei corpului, atunci obținem o familie de *deplasări* exprimate prin prin :

$$\Phi: \mathcal{B} \times t \to \mathbb{R}^3; \quad x = \Phi(X, t). \tag{2.48}$$



Figura 2.4: Configurații pentru o bară rectangulară solidă: (a) - de referință, (b) - după aplicarea unor acțiuni asupra barei; în cazurile staționar sau cvasistaționar, vectorul deplasare \vec{d} este nul, iar vectorul deformație \vec{u} reflectă doar diferența între vectorii de poziție \vec{x} și \vec{X} .

Fig. 2.4 conține configurația de referință \mathcal{B} pentru o bară rectangulară (a), respectiv (b)- configurația aceleiași bare în urma unei deplasări și deformații \mathcal{B}' . Funcția (2.48) are forma:

$$\Phi(X, t) = \mathbf{u} + \mathbf{X} - \mathbf{d} \tag{2.49}$$

unde **d** reprezintă vectorul deplasare, iar **u** vectorul deformație. În cazurile (static, sau cvasistaționar) corpurile nu se află în mișcare, adică $\mathbf{d} = 0$, prin urmare funcția (2.48) are forma:

$$\Phi(X, t) = \mathbf{u} + \mathbf{X} \tag{2.50}$$

Modificările configurației de referință apar datorită aplicării unor forțe (acțiuni, sau eforturi). Forța este o mărime vectorială care acționează în fiecare moment de timp asupra fiecărui punct al unei configurații. Efectele acțiunii forței sunt descrise prin determinarea stărilor de:

- de tensiune;
- de deformație;
- de deplasare.

ce vor fi prezentate, sintetic, în continuare. Tratări detaliate ale acestor aspecte pot fi găsite în [86], [107], [67], [126], [152].

2.4.2 Forțe și cupluri

In mecanică, noțiunea de forță implică întotdeauna existența a două corpuri care, conform principiului acțiunii și reacțiunii ("a treia lege a lui Newton") ([57]) acționează unul asupra celulilalt cu forțe egale și de sens contrar. Cele două elemente pot fi și două părți ale aceluiași corp, obținute - de exemplu - prin secționarea imaginară a acestuia cu o suprafață de separație. Astfel, cele două forțe, care privite în ansamblu sunt interne întregului corp, pot fi privite ca acțiuni externe ale unei părți asupra celeilalte. In mecanica mediilor continue, forțele externe pot fi grupate în două mari clase:

- forțe de volum, reprezentând efecte ce pun în evidență acțiunea unor câmpuri cum ar fi câmpul gravitațional, sau cel electromagetic;
- forțe de suprafață reprezentate de totalitatea acțiunilor exercitate prin contact direct asupra respectivului corp pe totalitatea suprafaței sale externe, sau pe porțiuni din aceasta.

Din punct de vedere matematic, ipoteza (ME0) semnifică faptul că orice punct din domeniul spațial respectiv, reprezină limita oricărui șir de unități elementare de spațiu, centrate în jurul acelui punct, cu dimensiuni din ce în ce mai mici, fie ele de volum, sau de suprafață. Devine astfel posibilă definirea noțiunilor de densitate de forță de volum, respectiv de suprafață ca mărimi vectoriale astfel:

$$\mathbf{b} = \lim_{\Delta V \to 0} \frac{\Delta \mathbf{F}_V}{\Delta V} \tag{2.51}$$

$$\mathbf{t} = \lim_{\Delta A \to 0} \frac{\Delta \mathbf{F}_S}{\Delta A},\tag{2.52}$$

unde **b** reprezintă forța de volum unitară, iar **t** reprezină forța de suprafață unitară, purtând denumirea de *efort unitar* sau *tensiune*. Vectorii **b**, **t**, sunt tridimensionali, având în coordonate carteziene expresii de forma $\mathbf{f} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$. Aceste **mărimi** **locale** sunt reprezentate de funcții cu argument spațio-temporal, spațiul fiind un subdomeniu Ω al \mathbb{R}^3 . Forma generală a acestor funcții poate fi definită cu ajutorul relației (2.53) (v. fig. 2.5).

$$\mathbf{f} = \mathbf{f}(\mathbf{r}, t) \, cu \, \mathbf{f} : \Omega \times [t_i, t_f] \to \mathbb{R}^3, \Omega \subset \mathbb{R}^3.$$
(2.53)

unde t_i, t_f reprezintă momentele inițial, respectiv final ale intervalului de timp în care se studiază acțiunea forței respective.



Figura 2.5: Forța de volum cu densitate **b** și forța de suprafață cu tensiunea **t**. In cazul în care suprafața Σ este o suprafață de separație între două părți Ω_1 , Ω_2 ale aceluiași corp, **t** reprezintă reacția cu care partea Ω_2 acționează asupra părții Ω_1 - egală și de sens contrar cu forța cu care partea Ω_1 acționează asupra părții Ω_2 .

De remarcat însă, că dacă se aleg prin același punct P, două suprafațe Σ_1 , respectiv Σ_2 care au asociate normale \mathbf{n}_1 și \mathbf{n}_2 cu orientări diferite, atunci și tensiunile \mathbf{t}_1 , respectiv \mathbf{t}_2 vor fi diferite (v. fig. 2.6).



Figura 2.6: Două suprafețe cu normale diferite, \mathbf{n}_1 și \mathbf{n}_2 în același punct P, implică tensiuni diferite \mathbf{t}_1 , respectiv \mathbf{t}_2 asociate aceluiași punct.

Forța de volum, respectiv forța de suprafață totale ce acționează asupra unui corp de dimensiuni finite sunt date de suma rezultantă a tuturor forțelor elementare, rezultante care se obțin prin integrare pe componente pe întreg domeniul Ω , respectiv frontiera $\partial \Omega$, după cum urmează:

$$\int_{\Omega} \mathbf{b} \, dV = \mathbf{i} \int_{\Omega} b_x \, dV + \mathbf{j} \int_{\Omega} b_y \, dV + \mathbf{k} \int_{\Omega} b_k \, dV \tag{2.54}$$

$$\int_{\partial\Omega} \mathbf{t} \cdot \mathbf{n} \, dA = \mathbf{i} \int_{\partial\Omega} t_x \cdot \mathbf{n} \, dA + \mathbf{j} \int_{\partial\Omega} t_y \cdot \mathbf{n} \, dA + \mathbf{k} \int_{\partial\Omega} t_z \cdot \mathbf{n} \, dA \qquad (2.55)$$

Cuplul, este definit ca produsul vectorial dintre forță și brațul forței ([57], [66]). Sensul cuplului este definit de asocierea între sensul forței și sensul de referință a axei cuplului prin regula burghiului drept. Dacă pentru un corp oarecare alegem brațele forțelor elementare față de origine, atunci componentele cuplurilor elementare pe cele trei direcții sunt (v. fig. 2.7):

$$m_{x} = (b_{z}r_{Vy} - b_{y}r_{Vz})dV + (t_{z}r_{Ay} - t_{y}r_{Az})dA$$

$$m_{y} = (b_{x}r_{Vz} - b_{z}r_{Vx})dV + (t_{x}r_{Az} - t_{z}r_{Ax})dA$$

$$m_{z} = (b_{y}r_{Vx} - b_{x}r_{Vy})dV + (t_{y}r_{Ax} - t_{x}r_{Ay})dA$$
(2.56)



Figura 2.7: Componentele cuplurilor unitare datorate forțelor unitare de volum respectiv de suprafață. Brațele forțelor unt reprezentate de către componentele după axele de referință ale vectorilor de poziție \mathbf{r}_V respectiv \mathbf{r}_A

Egalitățile (2.56) se pot scrie condensat, în formulare vectorială:

$$\mathbf{m} = (\mathbf{r}_V \times \mathbf{b}) \, dV + (\mathbf{r}_A \times \mathbf{t}) \, dA \tag{2.57}$$

Ca și în cazul forțelor, valorile globale ale cuplurilor forțelor de volum, respectiv ale cuplurilor forțelor de suprafață se obțin prin integrarea pe spațiile respective:

$$\mathbf{M}_{V} = \int_{\Omega} \left(\mathbf{r}_{V} \times \mathbf{b} \right) dV \tag{2.58}$$

$$\mathbf{M}_A = \int_{\partial\Omega} \left(\mathbf{r}_A \times \mathbf{t} \right) \cdot \mathbf{n} \, dA \tag{2.59}$$

2.4.3 Starea de tensiune

In analiza solicitărilor mecanice care apar într-un punct P, în raport cu un sistem de referință ortogonal tridimensional, un rol deosebit îl au tensiunile \mathbf{t}_1 , \mathbf{t}_2 și \mathbf{t}_3 ce acționează pe trei plane ortogonale, care se intersectează în acest punct. Orientarea sistemului de referință este dată de tripletul de versori (\mathbf{i} , \mathbf{j} , \mathbf{k}), iar cele trei plane ortogonale au normalele \mathbf{n}_1 , \mathbf{n}_2 și \mathbf{n}_3 paralele cu axele sistemului de referință considerat (v. fig. 2.8).



Figura 2.8: Tetraedrul elementar Cauchy. Conform principiului al II-lea al mecanicii, forța totală ce acționează asupra tetraedrului elementar de înălțime *h* este egală cu viteza de variație a impulsului: $\mathbf{t}_n \cdot \Delta A - \mathbf{t}_1 \cdot \Delta A_1 - \mathbf{t}_2 \cdot \Delta A_2 - \mathbf{t}_3 \cdot \Delta A_3 - \mathbf{b} \cdot \Delta V = \rho \Delta V \frac{\mathrm{d}v}{\mathrm{d}t}$, unde ρ este densitatea mediului. Odată cu trecerea la limită, în sensul reducerii tetraedrului elementar la punctul P ($h \rightarrow 0$), rezultă expresia (2.60)

Se demonstrează ([107]) că tensiunea ce acționează pe un plan oarecare, ce trece prin punctul P, având în acest punct normala \mathbf{n} , se poate exprima în funcție de tripletul de tensiuni (\mathbf{t}_1 , \mathbf{t}_2 , \mathbf{t}_3) astfel:

$$\mathbf{t}_n = l \cdot \mathbf{t}_1 + m \cdot \mathbf{t}_2 + p \cdot \mathbf{t}_3, \tag{2.60}$$

unde l, m, n reprezintă cosinusurile directoare ale direcției normalei **n** în raport cu sistemul de referință:

$$\mathbf{n} = l\mathbf{i} + m\mathbf{j} + p\mathbf{k}.\tag{2.61}$$

Egalitatea (2.60) nu este influențată de acțiunea forței de volum în punctul P și nici nu este restricționată de vreo condiție de echilibru extern, prin urmare ea este valabilă pentru orice tip de mediu continuu.

Relația (2.60) poate fi privită ca o aplicație liniară $T : \mathcal{N}_P \to \mathcal{T}_P$ unde \mathcal{N}_P este spațiul vectorial al normalelor la suprafețele Σ_P ce trec prin punctul P, iar \mathcal{T}_P este spațiul vectorial al forțelor ce acționează pe suprafețele Σ_P în punctul P. Această aplicație este reprezentată de un tensor de ordin doi, care conține pe fiecare linie componentele tensiunilor $\mathbf{t}_1, \mathbf{t}_2$ și \mathbf{t}_3 :

$$\overline{\overline{\mathbf{T}}} = \begin{pmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{pmatrix}$$
(2.62)

Starea de tensiune dintr-un punct oarecare al unei configurații este determinată dacă se cunosc tensiunile pe cele trei plane reciproc ortogonale ce trec prin acel punct. Dacă descriem în jurul punctului P de mai sus un paralelipiped drept elementar cu dimensiunile dx, dy, dz (fig. 2.9), pentru fața normală la axa *i* se pot defini următoarele tensiuni:

- tensiunea normală notată cu σ_{ii} ;
- tensiunea tangențială de forfecare, cu componentele sale după celelalte două direcții normale, notate cu τ_{ij} respectiv τ_{ik} .

In figura 2.9 sunt reprezentate cele nouă componente care definesc starea de tensiune în punctul P, ele fiind reprezentabile matematic cu ajutorul unui tensor de ordin doi denumit **tensorul tensiunilor**:

$$\overline{\overline{\sigma}} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} = \begin{pmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{pmatrix}$$
(2.63)

Cu notațiile de mai sus, relația (2.60) se poate scrie astfel:

$$\mathbf{t}_n = \overline{\overline{\sigma}} \cdot \mathbf{n}. \tag{2.64}$$

In absența distribuțiilor de cuplu de volum sau de suprafață, tensorul $\overline{\overline{\sigma}}$ este simetric, fiind deci necesară cunoașterea numai a 6 componente.



Figura 2.9: Definirea stării de tensiuni în punctul P; pe fiecare plan tensiunea se descompune într-o componentă normală - σ , respectiv una tangențială - τ , care, la rândul său, se descompune după direcțiile care definesc planul pe care acționează tensiunea; primul indice indică normala la fața pe care acționează tensiunea, iar al doilea axa cu care aceasta este paralelă.

In fiecare punct spațial al unei configurații există un set unic de trei plane ortogonale pe care tensiunile tangențiale τ_{ij} se anulează. Axele rezultate din intersecția acestor plane poartă denumirea de direcții principale ale tensiunilor, iar tensiunile paralele cu aceste axe reprezintă setul de tensiuni principale.

2.4.4 Starea de deformare

Acțiunea unei forțe exterioare asupra unui corp poate imprima acestuia:

- o deplasare simultană a tuturor punctelor materiale ce compun corpul, numită deplasare materială, sau deplasare de corp rigid; în acest caz sunt de interes variația în timp a pozițiilor acestor puncte în raport cu un sistem de referință;
- o deplasare relativă a punctelor materiale unele față de altele numită deformare; în acest caz se urmăresc atât modificările relative de distanță (*alungiri*) și unghi (*alunecări*) între pozițiile acestor puncte, precum și modificările relative de viteză ale punctelor unele față de altele, ca măsură a vitezei de deformare a corpului compus din aceste puncte materiale. Sistemul de referință folosit poate fi atât cel al configurației de bază (abordare în sens Lagrange v. relația (2.47)), cât și cel al configurațiilor curente, rezultate în urma acțiunii forței (abordare Euleriană);

Starea de deformare se poate caracteriza complet prin punerea în evidență a alungirilor și alunecărilor relative ce apar odată cu acțiunea deformatoare.

Reluând cofigurația imobilă ($\mathbf{d} = 0$) reprezentată în fig. 2.4 drept configurație de referință, se pot urmări deplasările punctelor A și B în urma deformării acestei configurații. Pozițiile punctelor A și B sunt date, față de referențialul configurației inițiale, de vectorii \mathbf{r}_A și \mathbf{r}_B , iar distanța - considerată infinetizimal mică - dintre aceste puncte este dată de vectorul \mathbf{dS} (v. fig. 2.10).



Figura 2.10: Deformații într-o bară imobilă: liniare - $\overline{AB} \rightarrow \overline{ab}$, respectiv unghiulare (alunecări) - $\angle OAB \rightarrow \angle Oab$

In urma deformării, noile poziții ale punctelor sunt a, respectiv b, iar noua distanță dintre ele este **ds**. Față de același sistem de referință (abordare în sens Lagrange) noile poziții sunt date de vectorii \mathbf{r}_a și \mathbf{r}_b .

Conform fig. (2.10), deplasarea relativă a punctului A față de B este dată de vectorul:

$$\vec{du} = \vec{ds} - \vec{dS} = du_x \mathbf{i} + du_y \mathbf{j} + du_z \mathbf{k}, \qquad (2.65)$$

iar deplasarea relativă raportată la distanța inițială dS, numită și deplasare relativă specifică, poate fi descrisă pe fiecare componentă în parte:

$$\frac{\mathrm{d}u_x}{\mathrm{d}S} = \frac{\partial u_x}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}S} + \frac{\partial u_x}{\partial Y}\frac{\mathrm{d}Y}{\mathrm{d}S} + \frac{\partial u_x}{\partial Z}\frac{\mathrm{d}Z}{\mathrm{d}S}
\frac{\mathrm{d}u_y}{\mathrm{d}S} = \frac{\partial u_y}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}S} + \frac{\partial u_y}{\partial Y}\frac{\mathrm{d}Y}{\mathrm{d}S} + \frac{\partial u_y}{\partial Z}\frac{\mathrm{d}Z}{\mathrm{d}S}
\frac{\mathrm{d}u_z}{\mathrm{d}S} = \frac{\partial u_z}{\partial X}\frac{\mathrm{d}X}{\mathrm{d}S} + \frac{\partial u_z}{\partial Y}\frac{\mathrm{d}Y}{\mathrm{d}S} + \frac{\partial u_z}{\partial Z}\frac{\mathrm{d}Z}{\mathrm{d}S},$$
(2.66)

sau matriceal:

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}S} = \begin{pmatrix} \frac{\partial u_x}{\partial X} & \frac{\partial u_x}{\partial Y} & \frac{\partial u_x}{\partial Z} \\ \frac{\partial u_y}{\partial X} & \frac{\partial u_y}{\partial Y} & \frac{\partial u_y}{\partial Z} \\ \frac{\partial u_z}{\partial X} & \frac{\partial u_z}{\partial Y} & \frac{\partial u_z}{\partial Z} \end{pmatrix} \cdot \begin{pmatrix} \frac{\mathrm{d}X}{\mathrm{d}S} \\ \frac{\mathrm{d}Y}{\mathrm{d}S} \\ \frac{\mathrm{d}Z}{\mathrm{d}S} \end{pmatrix} = [\mathbf{J}_u] \cdot [\mathbf{1}_{AB}], \qquad (2.67)$$

unde matricea Jacobian \mathbf{J}_u este matricea gradientului de deplasare. Aceasta poate fi privită ca un operator care aplicat componentelor direcției poziției inițiale $(\frac{\mathrm{d}X}{\mathrm{d}S}\mathbf{i} + \frac{\mathrm{d}Y}{\mathrm{d}S}\mathbf{j} + \frac{\mathrm{d}Z}{\mathrm{d}S}\mathbf{k})$ a elementului infinit mic AB, determină componentele ortogonale ale deplasării relative specifice a punctului A față de B, în raport cu poziția inițială.

Matricea \mathbf{J}_u conține atât deplasarea materială a punctelor A și B, dar și deplasarea datorată deformațiilor lineice și unghiulare. \mathbf{J}_u se descompune într-o sumă de două matrice, una simetrică notată cu \mathcal{D} , respectiv una antisimetrică notată cu \mathcal{W} :

$$[\mathbf{J}_u] = [\mathcal{D}] + [\mathcal{W}] \tag{2.68}$$

unde:

$$\left[\mathcal{D}\right] = \begin{pmatrix} \frac{\partial u_x}{\partial X} & \frac{1}{2} \left(\frac{\partial u_x}{\partial Y} + \frac{\partial u_y}{\partial X} \right) & \frac{1}{2} \left(\frac{\partial u_x}{\partial Z} + \frac{\partial u_z}{\partial X} \right) \\ \frac{1}{2} \left(\frac{\partial u_y}{\partial X} + \frac{\partial u_x}{\partial Y} \right) & \frac{\partial u_y}{\partial Y} & \frac{1}{2} \left(\frac{\partial u_y}{\partial Z} + \frac{\partial u_z}{\partial Y} \right) \\ \frac{1}{2} \left(\frac{\partial u_z}{\partial X} + \frac{\partial u_x}{\partial Z} \right) & \frac{1}{2} \left(\frac{\partial u_z}{\partial Y} + \frac{\partial u_y}{\partial Z} \right) & \frac{\partial u_z}{\partial Z} \end{pmatrix} = \frac{1}{2} \left(\left[\nabla \cdot \mathbf{u} \right] + \left[\nabla \cdot \mathbf{u} \right]^T \right)$$

$$(2.69)$$

și

$$[\mathcal{W}] = \begin{pmatrix} 0 & \frac{1}{2} \left(\frac{\partial u_x}{\partial Y} - \frac{\partial u_y}{\partial X} \right) & \frac{1}{2} \left(\frac{\partial u_x}{\partial z} - \frac{\partial u_z}{\partial X} \right) \\ -\frac{1}{2} \left(\frac{\partial u_x}{\partial Y} - \frac{\partial u_y}{\partial X} \right) & 0 & \frac{1}{2} \left(\frac{\partial u_y}{\partial Z} - \frac{\partial u_z}{\partial Y} \right) \\ -\frac{1}{2} \left(\frac{\partial u_x}{\partial z} - \frac{\partial u_z}{\partial X} \right) & -\frac{1}{2} \left(\frac{\partial u_y}{\partial Z} - \frac{\partial u_z}{\partial Y} \right) & 0 \end{pmatrix} = \frac{1}{2} \left([\nabla \cdot \mathbf{u}] - [\nabla \cdot \mathbf{u}]^T \right)$$

$$(2.70)$$

Se demonstrează ([107], [45]) că, în ipoteza micilor deformații, deplasările punctelor datorate deformațiilor pot fi evaluate exclusiv cu ajutorul matricei \mathcal{D} ,

rolul matricei \mathcal{W} fiind doar de a pune în evidență deplasările punctelor datorate rotației rigide (fără deformații) a corpului. Mai mult, se poate arăta ([152]) că gradientul deformației în configurația de referință \mathcal{B} (puncte materiale) este practic egal cu cel din orice altă configurație de puncte spațiale \mathcal{B}' :

$$\frac{\partial \mathbf{u}}{\partial X} = \frac{\partial \mathbf{u}}{\partial x}.$$
(2.71)

Este posibilă astfel, definirea unei aplicații liniare $\epsilon : S \to U$, unde S este spațiul vectorial al vectorilor asociați cu o direcție **dS**, iar U este spațiul vectorial al vectorilor deplasare relativă specifică **du/dS**. Această aplicație este un tensor de ordin doi, ce poartă numele de **tensor al deformațiilor** specifice:

$$\bar{\bar{\epsilon}} = \begin{pmatrix} \varepsilon_{xx} & \varepsilon_{xy} & \varepsilon_{xz} \\ \varepsilon_{yx} & \varepsilon_{yy} & \varepsilon_{yz} \\ \varepsilon_{zx} & \varepsilon_{zy} & \varepsilon_{zz} \end{pmatrix} = \begin{pmatrix} \varepsilon_{xx} & \gamma_{xy}/2 & \gamma_{xz}/2 \\ \gamma_{yx}/2 & \varepsilon_{yy} & \gamma_{yz}/2 \\ \gamma_{zx}/2 & \gamma_{zy}/2 & \varepsilon_{zz} \end{pmatrix}$$
(2.72)

unde:

- termenii de pe diagonala principală reprezintă alungirile specifice ε_i pe direcția unei axe de versor \vec{i} ;
- termenii nediagonali, reprezintă lunecările specifice γ_{ij} în planul definit de versorii \vec{i} și \vec{j} .

Cu notatiile de mai sus, relatia (2.72) se poate scrie astfel:

$$\mathbf{du} = \overline{\overline{\epsilon}} \cdot \mathbf{dS}. \tag{2.73}$$

Pentru o exemplificare simplă, în fig. 2.11 sunt reprezentate deformațiile de mai sus într-un sistem de coordonate carteziene 2D.

Starea de deformare punctuală a unui corp este cunoscută integral dacă se cunosc toate componentele tensorului deformațiilor specifice în acel punct: In cazul materialelor izotrope, lunecările specifice nu depind de sensul de rotație (de ex. indici xy sau yx), astfel încât, în acest caz, tensorul (2.72) este simetric. Totodată, la acest tip de materiale nu există restricții datorate legăturilor dintre alungirile ε și lunecările specifice γ .

In fiecare punct spațial al unei configurații există un set unic de trei axe ortogonale pentru care alunecările specifice γ_{ij} se anulează. Aceste axe poartă denumirea de axe ale deformațiilor principale, iar unghiurile dintre aceste axe nu se



Figura 2.11: In urma unei deformații, un punct A suferă deplasări liniare u pe direcția Ox, respectiv v pe direcția Oy. Pătratul elementar $dx \times dy$ construit în vecinătatea punctului A, suferă o lunecare simetrică $\varepsilon_{xy} = \varepsilon_{yx} = \frac{1}{2}\gamma_{xy} = \frac{1}{2}(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y})$

modifică în urma deformării. Cele trei plane perpendiculare definite de aceste axe se numesc planele principale ale deformației.

Ecuația de compatibilitate geometrică în conformitate cu relația (2.69), între tensorul deformațiilor specifice și vectorul deplasărilor **u** are forma:

$$\overline{\overline{\epsilon}} = [\mathcal{D}] = \frac{1}{2} \left([\nabla \mathbf{u}] + [\nabla \mathbf{u}]^T \right)$$
(2.74)

2.4.5 Legile de conservare ale mecanicii mediilor continue

Mecanica mediilor continue se bazează pe patru legi de conservare ce vor fi prezentate în continuare, pe scurt. Formulări complete și aplicații pot fi găsite în [107], [86], [57], [45].

Legea 2.4.1. - de conservare a masei. Fluxul de masă ce iese un domeniu Ω închis de o suprafață $\partial \Omega$, este egal cu viteza de scădere a masei din acel domeiu. Forma sa integrală este dată de relația:

$$\int_{\partial\Omega} (\rho \mathbf{v}) \cdot \mathbf{n} \, dA = -\frac{dm_{\Omega}}{dt},\tag{2.75}$$

unde $\mathbf{v}_n = \mathbf{v} \cdot \mathbf{n}$ este componenta vitezei după normala exterioară a suprafeței $\partial \Omega$ (v. fig. 2.12). ρ reprezintă densitatea mediului și este o funcție continuă (v. ipoteza (ME0)) de punct și timp:

$$\rho = \rho(\mathbf{r}, t) \, cu \, \rho : \Omega \times [t_i, t_f] \to \mathbb{R}$$
(2.76)



Figura 2.12: Referitoare la legea conservării masei

Forma locală a legii de conservare a masei reprezintă ecuația de continuitate:

$$\nabla(\rho \mathbf{v}) + \frac{\partial \rho}{\partial t} = 0, \qquad (2.77)$$

De aici rezultă faptul că divergența vectorului viteză, indică viteza de scădere a densității de material din jurul unui punct. Pentru corpurile necompresibile, a căror densitate locală rămâne constantă pe parcusrul deplasării corpului, această viteză de scădere este nulă:

$$div(\mathbf{v}) = 0; \tag{2.78}$$

Ecuația (2.78) reprezintă condiția de necompresibilitate a unui material.

Legea 2.4.2. - de conservare a impulsului. Viteza de variație a impulsului unui corp (set rigid de corpuri) este egală cu rezultanta tuturor forțelor externe ce acționează asupra corpului (setului rigid de corpuri). Forma sa integrală este dată de relația:

$$\int_{\partial\Omega} \mathbf{t} \cdot \mathbf{n} \, dA + \int_{\Omega} \mathbf{b} \, dV = \frac{d}{dt} \int_{\Omega} \rho \mathbf{v} \, dV, \qquad (2.79)$$

unde **t** este tensiunea externă, **b** este densitatea volumică a forței externe (v. relațiile (2.51)), iar ρ reprezintă densitatea materialului.

Dacă exprimăm tensiunea \mathbf{t} cu ajutorul relației (2.64), iar apoi aplicăm integralei de suprafață teorema divergenței, forma globală a legii (2.4.2) devine:

$$\int_{\Omega} \left[\mathbf{b} + \nabla(\overline{\overline{\sigma}}) \right] \, dV = \int_{\Omega} \rho \frac{dv}{dt} \, dV, \tag{2.80}$$



Figura 2.13: Referitoare la legea conservării impulsului

reprezentând ecuația de mișcare a corpului (v. fig. 2.80).

Forma locală a ecuației de mișcare (2.80) este cunoscută sub numele de "prima lege de mișcare a lui Cauchy", având expresia de mai jos:

$$\mathbf{b} + \nabla(\overline{\overline{\sigma}}) = \rho \frac{dv}{dt},\tag{2.81}$$

iar pentru corpurile imobile, devenind ecuația de echilibru:

$$\mathbf{b} + \nabla(\overline{\overline{\sigma}}) = 0. \tag{2.82}$$

Legea 2.4.3. - de conservare a cuplului impulsului. Viteza de variație a cuplului impulsului unui corp este egală cu cuplul rezultant al tuturor forțelor externe ce acționează asupra corpului. Forma sa integrală este dată de relația:

$$\int_{\partial\Omega} \left(\mathbf{r} \times \mathbf{t} \right) \cdot \mathbf{n} \, dA + \int_{\Omega} \left(\mathbf{r} \times \mathbf{b} \right) dV = \frac{d}{dt} \int_{\Omega} \left(\mathbf{r} \times \rho \mathbf{v} \right) dV, \tag{2.83}$$

Cuplul impulsului descrie inerția rotațională a unui corp aflat în mișcare de rotație în jurul unei axe. Ca mărime vectorială este dată de produsul vectorial dintre brațul mișcării de rotație și impuls.

In cazul în care asupra corpului acționează câmpuri de cupluri externe fie de volum, fie de suprafață (datorate de exemplu acțiunii unui câmp electromagnetic), forma globală a legii 2.4.3 se scrie [107]:

$$\int_{\partial\Omega} \left[\mathbf{r} \times \mathbf{t} + \mathbf{m} \right] \cdot \mathbf{n} \, dA + \int_{\Omega} \left[\mathbf{r} \times \mathbf{b} + \mathbf{c} \right] dV = \frac{d}{dt} \int_{\Omega} \left[\mathbf{r} \times \rho \mathbf{v} + \mathbf{l} \right] dV, \qquad (2.84)$$

unde **c** este densitatea de volum a cuplului extern, **m** este densitatea de suprafață a cuplului extern, iar **l** este densitatea impulsului unghiular al corpului. In cazul lipsei cuplurilor externe, forma locală a legii 2.4.3 pune în evidență simetria tensorului de tensiuni (2.63) și anume:

$$\mathbf{r} \times \overline{\overline{\sigma}} \cdot \mathbf{n} = 0; \tag{2.85}$$

Legea 2.4.4. - prima lege a termodinamicii. [57], [107] Energia totală a unui volum dintr-un corp crește pe seama aportului de energie termică primit de acel volum și a lucrului mecanic efectuat asupra acestuia de către forțele de volum, respectiv de suprafață.

$$\Delta E = \Delta Q + \Delta W \tag{2.86}$$

unde E este energia totală a volumului considerat, Q este energia calorică acumulată în sistem, respectiv W este lucrul mecanic efectuat asupra acestuia.

Prin raportare la unitatea de timp, relația (2.86) devine:

$$\frac{dE}{dt} = Q_t + P \tag{2.87}$$

unde Q_t este energia calorică acumulată în sistem în unitatea de timp, respectiv P este lucrul mecanic efectuat asupra acestuia în unitatea de timp. Energia totală E este dată de suma:

$$E = U + K, \tag{2.88}$$

unde U este energia internă, ia
r $K=\int_{\Omega}\frac{1}{2}\rho\,{\bf v}\cdot{\bf v}\,dV$ energia cinetică.

In cazul în care lucrul mecanic este efectuat **de către** corp și **nu asupra sa**, atunci P își schimbă semnul, reprezentând o energie cedată și nu una acumulată. Preprezintă puterea mecanică transferată spre corp prin forțele de volum cu densitatea **b**, respectiv prin suprafață cu tensiunea mecanică **t** :

$$P = \int_{\partial\Omega} \mathbf{t} \cdot \mathbf{v} \cdot \mathbf{n} \, dA + \int_{\Omega} \mathbf{b} \cdot \mathbf{v} \, dV = \int_{\partial\Omega} \overline{\overline{\sigma}} \cdot \mathbf{v} \cdot \mathbf{n} \, dA + \int_{\Omega} \left(b_i \, v_i + b_j \, v_j + b_k \, v_k \right) dV,$$
(2.89)

unde $\mathbf{v} = v_i \mathbf{i} + v_j \mathbf{j} + v_k \mathbf{k}$ este câmpul vectorilor de viteză. Aportul de căldură poate avea două componente ([107] - cap 5.4, [86] - cap. 2):

- căldura transferată prin suprafața externă a corpului în unitatea de timp, denumită putere termică, sau flux termic, caracterizată de densitatea de flux termic q;
- căldura primită în întregul volum, caracterizată de densitatea pe unitatea de masă r = r(x, t); această componentă poate proveni, de exemplu, dintr-un efect de radiație a unui câmp, cum ar fi radiația electromagnetică.



Figura 2.14: Referitoare la prima lege a termodinamicii

$$Q_t = -\int_{\partial\Omega} \mathbf{q} \cdot \cdot \mathbf{n} \, dA + \int_{\Omega} \rho \, r \, dV \tag{2.90}$$

Notând cu u densitatea de volum a energiei interne, forma globală (2.87) a primei legi a termodinamicii devine:

$$\frac{d}{dt} \int_{\Omega} \left(u + \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} \right) dV = \int_{\Omega} \left(\mathbf{b} \cdot \mathbf{v} + \rho \, r \right) dV + \int_{\partial \Omega} \left[\left(\overline{\overline{\sigma}} : \nabla \overline{\mathbf{v}} \right) \cdot \mathbf{n} - \mathbf{q} \cdot \mathbf{n} \right] \, dA,$$
(2.91)

Dacă în (2.89) se aplică integralei de suprafață teorema divergenței, iar apoi în expresia obținută se folosește egalitatea (2.81), a ecuației Cauchy de mișcare, rezultă:

$$P = \frac{d}{dt} \int_{\Omega} \frac{1}{2} \rho \mathbf{v} \cdot \mathbf{v} \, dV + \int_{\Omega} \overline{\overline{\sigma}} : \nabla \overline{\overline{\mathbf{v}}} \, dV, \qquad (2.92)$$

unde:

- primul termen reprezintă derivata în raport cu timpul a energiei cinetice;
- al doilea termen reprezintă acea componentă de lucru mecanic în unitatea de timp ce se reflectă în energia internă a sistemului și nu din energia sa cinetică ([107] - cap 5.4) - practic o energie ce produce deformații.

Scalarul obținut din produsul tensorial diadic $\overline{\overline{\sigma}}: \nabla \overline{\overline{\mathbf{v}}}$ este densitatea de putere a

tensiunii mecanice, cu $\nabla \overline{\overline{\mathbf{v}}}$ tensorul gradient spațial al vitezei:

$$\nabla \overline{\mathbf{v}} = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_x}{\partial y} & \frac{\partial v_x}{\partial z} \\ \frac{\partial v_y}{\partial x} & \frac{\partial v_y}{\partial y} & \frac{\partial v_y}{\partial z} \\ \frac{\partial v_z}{\partial x} & \frac{\partial v_z}{\partial y} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$
(2.93)

fiind o măsură a vitezei de deformare a corpului respectiv. Inlocuind expresia (2.92) în forma integrală (2.91), rezultă forma locală a primei legi a termodinamicii:

$$\rho \frac{\partial u}{\partial t} = \overline{\overline{\sigma}} : \nabla \overline{\overline{\mathbf{v}}} + \rho \, r + \nabla \mathbf{q} \tag{2.94}$$

Energia totală, la nivel de element de volum, este reprezentată numai de energia internă u și este influențată numai de către acea componentă a lucrului mecanic care produce deformații, plus aportul de căldură.

In teoria ideală a elasticității, încălzirea este considerată neglijabilă, iar lucrul mecanic de deformare se presupune a fi înmagazinat în totalitate în energie internă, care va fi utilizată (tot sub formă de lucru mecanic) pentru revenirea corpului perfect elastic la forma sa inițială, imediat ce dispar forțele deformatoare. Fluxul termic **q** poate fi privit, în sens larg, ca un flux de energie nemecanică, astfel încât să fie posibilă luarea în considerare și a altor forme de energie, în afară de căldură, ce ar putea contribui la energia unui corp.

2.4.6 Ecuații constitutive

In mecanica mediilor continue, ecuațiile constitutive descriu comportamentul macroscopic al materialelor respectiv deformațiile în funcție de solicitările mecanice aplicate acestora. Prin natura lor, aceste comportamente sunt extrem de complexe, putând prezenta și fenomene de histetezis în cazul mediilor plastice. Pentru modelarea lor sunt necesare formulări matematice aproximative dependente de tipul și magnitudinea solicitărilor aplicate, respectiv gama de temperaturi de lucru. In teoria elasticității, cel mai des se folosesc modelele materialelor elastice ideale pentru care dependența deplasărilor punctelor materiale ce compun un corp, de solicitările aplicate este una liniară, stabilită de legea lui Hooke ([57], [158], [45], [66]). Ecuațiile constitutive stabilesc legături între fiecare componentă a tensorului tensiunilor (v. relația (2.63)) și **toate** componetele tensorului deformațiilor (v. relația (2.72)), legătură care este, în cazul mediilor liniare, tot una de tip tensorial ([107] - cap 6.2):

$$\sigma_{ij} = \sum_{r=1}^{3} \sum_{s=1}^{3} c_{ijrs} \varepsilon_{rs}$$
(2.95)

unde cele trei direcții ale sistemului de referință ortogonal sunt identificabile cu indicii 1, 2, respectiv 3. Relația (2.95), cunoscută sub numele de legea lui Hooke generalizată, este una liniară și implică scrierea a 9 ecuații și cunoașterea celor 81 de coeficienți ai tensorului de ordin patru \overline{c} , numit **tensor de rigiditate**. Ținând cont însă de simetria tensorilor $\overline{\sigma}$ (cap. 2.4.3) și $\overline{\epsilon}$ (cap. 2.4.4), egalitatea (2.95) se poate scrie sub formă matriceală astfel ([107], [67]):

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{31} \\ \sigma_{12} \end{pmatrix} = \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & d_{66} \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{23} \\ \varepsilon_{31} \\ \varepsilon_{12} \end{pmatrix},$$
(2.96)

sau, condensat:

$$[\sigma] = [d] \cdot [\epsilon], \qquad (2.97)$$

unde matricea [d] este **matricea de rigiditate**, care, la rândul ei este simetrică necesitând practic - pentru materialele anizotrope - cunoașterea numai a 21 din cei 36 de coeficienți. Expresia (2.96) pune în evidență faptul că în cazul materialelor anizotrope alungirile specifice (ε_{ii}) sunt influențate de către tensiunile tangențiale ($\sigma_{ij}, i \neq j$), respectiv lunecările ($\varepsilon_{ij}, i \neq j$) sunt influențate de către tensiunile normale (σ_{ii}).

In cazul materialelor izotrope, coeficienții matricei de rigiditate nu depind de direcție, prin urmare nici de sistemul de coordonate utilizat ca referință. Pentru astfel de materiale legea lui Hooke are următoarea formă matriceală, mult mai simplă:

$$\begin{pmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{31} \\ \sigma_{12} \end{pmatrix} = \begin{pmatrix} \lambda + 2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda + 2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda + 2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{pmatrix} \begin{pmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{23} \\ \varepsilon_{31} \\ \varepsilon_{12} \end{pmatrix},$$
(2.98)

unde μ și λ sunt coeficienții Lamé care se exprimă în funcție de modulele de elasticitate transversal G, respectiv longitudinal E (modulul lui Young) și coeficientul lui Poisson ν , parametri care sunt caracteristici pentru fiecare material în parte:

$$\lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}$$
(2.99)

$$\mu = G = \frac{E}{2(1+\nu)}.$$
(2.100)

(2.101)

Din expresia (2.98) rezultă, pentru materialele izotrope, atât independența deformațiilor transversale de tensiunile normale precum și cea a deformațiilor longitudinale de tensiunile tangențiale. Constatăm că pentru caracterizarea materialelor liniare și izotrope sunt suficiente doar două constante de material: modulul lui Young și coeficientul lui Poisson, sau echivalent, cei doi coeficienți Lamé.

2.5 Concluzii referitoare la aspectele fizice

Din cele prezentate în acest capitol rezultă că pentru descrierea fenomenelor fizice distribuite în spațiu se folosesc mărimi fizice locale, reprezentate de câmpuri scalare sau vectoriale ce satisfac ecuații diferențiale cu derivate parțiale, de regulă, de ordinul doi. Ecuațiile sunt obținute din relațiile fundamentale/generale ce caracterizează câmpul și din ecuațiile constitutive, care descriu comportarea materialelor caracteristice domeniului fizic respectiv. Membrul stâng al acestor ecuații este o reflectare a proprietăților de material din domeniul fizic considerat, conținând toate constantele de material. Membrul drept conține mărimile ce descriu sursele de câmp, fie interne, fie externe, caz în care sunt reprezentate prin condiții de frontieră.

Capitolul 3

Aspecte matematice

3.1 Considerațiuni preliminare

Formulările matematice au ca scop transpunerea ecuațiilor fizice ce caracterizează un anumit fenomen în limbaj matematic, care se referă la spațiul abstract al operatorilor și funcționalelor matematice, cu scopul de a formula corect, în termeni matematici, problema de rezolvat. Această transpunere se bazează pe faptul că, în multe cazuri, fenomene fizice diferite sunt caracterizate de mărimi cu proprietăți matematice similare, respectiv între care se pot stabili relații similare. In acest sens, formulările matematice tratează toate aceste fenomene diferite în mod unitar, lucrul fiind posibil prin transpunerea descrierii respectivelor fenomene în spații vectoriale abstracte în interiorul cărora semnificația fizică nu mai este evidentă. Acest fapt nu reduce însă importanța bunei cunoașteri a principiilor fizice, întrucât metoda matematică nu face decât să valideze corectitudinea unui set de ecuații, dar nu poate verifica modul în care aceste ecuații reflectă cu adevărat fenomenul fizic respectiv.

Cadrul matematic ce guvernează aceasta abordare este analiza funcțională ([38], [99], [97], [46], [138]). Conceptul fundamental al analizei funcționale este cel de spațiu abstract. Apartenența unui element la acest spațiu presupune respectarea unor anumite axiome matematice care nu au nicio legătură cu semnificația fizică reală a mulțimilor de elemente respective. Din această cauză spațiile abstracte au o aplicabilitate extrem de largă. Mai mult, seturi de axiome diferite generează spații abstracte diferite: se constată că aplicațiile (operatori) stabilite între două astfel de spații constituie ele însele niște spații abstracte. Devine astfel posibilă punerea în evidență a existenței unor relații între membrii unor spații abstracte, relații ce nu erau câtuși de puțin evidente inițial (respectiv în lumea reală). Această metodă axiomatic - abstractă a analizei funcționale dezvoltă un set coerent și consistent

de teoreme ce pot fi aplicate oricăror mulțimi de elemente ce respectă axiomele inițiale ([58], [84], [98]). Nu există vreo restricție asupra dimensiunii mulțimilor de obiecte matematice (coordonate generalizate) ce reflectă un anumit fenomen fizic, prin urmare el poate fi descris și dimensionat ca atare.

In esență, formularea matematică presupune găsirea unei aplicații $\mathcal{A} : X \to Y$ definită pe spațiul soluțiilor X, cu valori în spațiul Y al datelor de intrare ale problemei:

$$\mathcal{A}(x) = y. \tag{3.1}$$

Pentru ca problema să fie corect formulată, soluția trebuie să existe (\mathcal{A} este surjectivă), să fie unică (\mathcal{A} este și injectivă) și stabilă (\mathcal{A} este bine condiționată, deci continuă), adică variația ei să fie mărginită de variația datelor de intrare (excitații), amplificate eventual cu o constantă suficient de mică. În acest sens, domeniile Xși Y reprezintă spații abstracte, caracterizate axiomatic de proprietăți algebrice și topologice bine definite. Numai în baza acestor proprietăți se pot demonstra existența și unicitatea soluției ceea ce asigură buna formulare a problemei [84] [159], [72], [41].

In cele ce urmează vor fi folosite următoarele noțiuni tipice analizei funcționale [97], [46]:

(AF1) spațiu vectorial metric normat - spațiu abstract ai cărui elemente pot fi:

- vectori, elemente dintr-un spațiu n-dimensional V notat cu [V]ⁿ; în lucrare va fi utilizat frecvent spațiul tridimensional [V]³ de coordonate scalare reale din R respectiv complexe din C;
- funcții scalare cu valori într-un domeniu scalar U din ℝ sau C, respectiv vectoriale cu valori într-un domeniu [V]ⁿ;

Structura acestor spații este descrisă de operațiile algebrice de adunare și înmulțire cu un scalar precum și de noțiunile topologice de distanță (metrică), respectiv normă și seminormă în baza cărora se pot defini proprietățile de convergență, continuitate și diferențiabilitate; cele mai importante clase de spații vectoriale normate utilizate în această lucrare sunt:

- clasa spațiilor *Banach* ([99],[97],[46])- spații metrice vectoriale normate complete, în care orice șir Cauchy (cu distanța între două elemente succesive convergentă către zero) este convergent;
- clasa spațiilor *Hilbert* ([99], [97], [46], [138])- spații Banach pe care este definită și noțiunea de produs scalar pe baza căruia se definesc norma spațiului respectiv, precum și noțiunea de ortogonalitate;

- (AF2) izomor fism un concept care reflectă identitatea structurală a două spații abstracte bazate pe elemente de aceeași natură; de exemplu, un izomorifism între două spații vectoriale conservă structura reprezentată de operațiile algebrice, distanța și norma ce caracterizează cele două spații, chiar dacă semnificația elementelor celor două spații nu este identică în lumea concretă [97], [46].
- (AF3) operatori liniari aplicații liniare $\mathbf{T} : X \to Y$, unde X și Y sunt spații vectoriale normate, respectiv funcționale liniare - aplicații liniare $\varphi : U \to \mathbb{K}$, unde U este un spațiu vectorial normat, iar \mathbb{K} este un câmp de scalari din \mathbb{R} sau \mathbb{C} . Pe un spațiu normat, totalitatea funcționalelor φ mărginite formează la rândul lor, un spațiu vectorial normat, numit spațiu dual și care va primi notația U' ([97]-Def.2.10-3); spațiul dual este izomorf cu spațiul inițial și este un spațiu Banach chiar dacă cel inițial nu are această proprietate.

Scrierea unei formulări matematice a problemei de rezolvat presupune parcurgerea următoarelor etape:

- 1. descrierea domeniului de calcul pe care să se poată defini corect operatorii și funcționalele asociate aplicației \mathcal{A} ;
- 2. descrierea condițiilor pe care trebuie să le îndeplinească elementele vectoriale și scalare asociate problemei(date, soluție) astfel încât ele să poată constitui spații vectoriale normate - cf. (AF1) pe întreg domeniul de calcul și definirea acestor spații;
- 3. stabilirea de operatori și funcționale între spațiile care definesc operatorul \mathcal{A} al problemei (pct. 1) astfel încât, să se poată descrie spațiile X, respectiv Y din (3.1) și să se poată demonstra existența, unicitatea și stabilitatea soluției x;
- 4. spațiile vectoriale nou formate se aleg astfel încât funcționalele/operatorii definiți să păstreze descrierea aspectelor fizice ale problemei, materializate prin proprietăți de convergență, mărginire, și continuitate ale operatorilor inițiali și exprimate prin restricții matematice impuse de:
 - proprietăți de material;
 - excitații interne;
 - excitații externe (condiții de froniteră);

• condiții inițiale - pentru fenomenele tranzitorii.

Descrierile de mai sus formează cadrul funcțional al problemei de rezolvat. Ele vor fi tratate în continuare la modul general, urmând a fi aplicate ulterior pe probleme concrete.

3.2 Formulări slabe.

Sistemul de ecuații 3.1 este, de regulă, unul de ecuații diferențiale cu derivate pațiale, ce reprezintă forma tare a problemei. Formulările slabe sunt asociate (nu în mod unic) unei clase de metode, dedicate rezolvării ecuațiilor diferențiale, care presupun testarea modului în care o soluție aproximativă prestabilită, satisface soluția ecuațiilor în anumite condiții. Membrii acestei clase sunt cunoscuți sub denumirea de *metode ansatz* ([73] [84] [159]). Conform acestor metode soluția se aproximează cu ajutorul unei combinații liniare de funcții de bază (numite și funcții de încercare) alese astfel încât să respecte anumite condiții (fac parte dintr-un anumit spațiu funcțional). Problema constă în determinarea coeficienților acestei combinații liniare.

Pentru ecuații diferențiale de ordin k formulările matematice tari impun căutarea unor soluții din spațiul funcțiilor de k ori derivabile $(C^k(\Omega))$. Formulările slabe admit ca soluții funcții ce nu sunt netede peste tot domeniul ci *netede aproape peste tot* domeniul de calcul, fiind suficientă demonstrarea integrabilității derivatelor de ordin k - 1. Soluția este determinată prin minimizarea, sau chiar anularea, proiecției reziduului ecuației de rezolvat, ponderat cu o serie de funcții, numite *de test*, pe o mulțimea de funcțiilor de test. Spațiul funcțiilor de test, care se aleg identic cu funcțiile de încercare, este izomorf cu cel al soluțiilor. Această abordare, datorată lui Galerkin permite modelări mai apropiate de realitatea fizică decât cele oferite de către formulările matematice *tari* ([98] [73]). Apropierea se reflectă și în faptul că majoritatea fenomenelor fizice evoluează conform unor legi caracterizate de formulări energetice de maxim sau minim, care corespund minimului reziduului ponderat.

De fiecare dată însă, trebuie definit spațiul funcțional care să asigure existența și unicitatea soluțiilor slabe. În continuare vor fi prezentate aspectele cele mai importante necesare definirii cadrului funcțional adecvat formulărilor matematice variaționale slabe asociate modelelor fizice descrise în capitolul 2.

3.2.1 Distribuții

Formulările slabe au la bază noțiunea de distribuție care generalizează noțiunea clasică de funcție, astfel încât obiectul matematic obținut să poată fi derivat fără restricții. Distibuțiile sunt definite cu ajutorul unor funcții de test φ , indefinit derivabile și cu suport compact pe un domeniu $\Omega \subset \mathbb{R}^n$. Dacă acestor funcții din spațiul $\mathcal{C}_0^{\infty}(\Omega)$ li se asociază o topologie care reflectă noțiunea naturală de convergență (convergență uniformă, împreună cu toate derivatele parțiale de orice ordin ale funcției respective) se poate defini un spațiu vectorial, $\mathcal{D}(\Omega)$, al funcțiilor de test ([137], [100] - Teorema 9.8):

$$\mathcal{D}(\Omega) = \{ \varphi \in \mathcal{C}_0^\infty(\Omega) \mid \exists \, \Omega' \Subset \Omega \text{ a. î. supp } \varphi_n \subset \Omega', \, \forall n \in \mathbb{N};$$
(3.2)

$$\exists \text{ pe } \Omega \lim_{n \to \infty} \partial^{\alpha} \varphi_n = \partial^{\alpha} \varphi, \, \forall \alpha = (\alpha_1, \cdots, \alpha_m) \in \mathbb{N}^m. \}$$
(3.3)

unde $\partial^{\alpha}\varphi$ reprezintă derivata de ordin α a funcției φ . Pe spațiul $\mathcal{D}(\Omega) \subset \mathcal{C}_{0}^{\infty}(\Omega)$ se poate defini noțiunea de funcție generalizată (sau distribuție) ca o funcțională liniară [100] - Teorema 9.10:

$$T: \mathcal{D}(\Omega) \to \mathbb{R} \ cu \ T(\varphi) \stackrel{\text{not}}{=} < T, \ \varphi > .$$
 (3.4)

Funcția(3.4) este generalizată în sensul că argumentul său nu este reprezentat de un set de coordonate din \mathbb{R}^n ci de un set de *n* funcții de test din $\mathcal{D}(\Omega)$. Mulțimea funcționalelor $\langle T, \cdot \rangle$ formează un spațiu vectorial, *dual* spațiului $\mathcal{D}(\Omega)$ și este notat cu $\mathcal{D}'(\Omega)$. Spațiul distribuțiilor, $\mathcal{D}'(\Omega) \subset (\mathcal{C}_0^{\infty}(\Omega))'$, este dotat cu o topolgie de tip (3.2) - (3.3), adică un șir $\{T_n\}$ converge către distribuția T, dacă $\langle T_n, \varphi \rangle \rightarrow \langle T, \varphi \rangle$ pentru $\forall \varphi \in \mathcal{D}(\Omega), n \in \mathbb{N}$.

Distribuțiile au avantajul, față de funcțiile definite de analiza matematică, că sunt întotdeauna diferențiabile. O clasă importantă de distribuții, este cea asociată funcțiilor f, local integrabile pe domeniul Ω ($f \in L^1_{LOC}(\Omega)$) și definite ca:

$$\langle T_f, \varphi \rangle = \int_{\Omega} f \varphi \, dx \ \forall \varphi \in \mathcal{D}(\Omega)$$
 (3.5)

și cunoscută ca clasa distribuțiilor standard, iar notația $\langle T, \cdot \rangle$ ca produsul de dualitate standard. Spre deosebire de formulările care atașează fiecărui punct $x \in$ $M \subset \mathbb{R}$ o funcție f(x) cu $f : M \to \mathbb{R}$, distribuțiile standard T atașează fiecărei funcții de test φ din Ω , integrala $\int f\varphi$. Distribuția astfel definită reprezintă o medie a efectului pe care funcția f îl are asupra funcțiilor de test. Asocierea este potrivită funcțiilor ce reflectă evoluția unui fenomen fizic, determinată prin medierea ponderată a informațiilor obținute cu ajutorul unui set de "dispozitive" de testare (funcții de test)([58], [137]). Derivata parțială a unei funcționale T în raport cu o variabilă x_i se definește ca:

$$\frac{\partial T}{\partial x_i}(\varphi) = \langle \frac{\partial T}{\partial x_i}, \varphi \rangle \stackrel{\text{def}}{=} -\langle T, \frac{\partial \varphi}{\partial x_i} \rangle, \ \forall \varphi \in \mathcal{D}(\Omega)$$
(3.6)

Pentru distribuțiile standard, pentru $\forall f \in L^1_{LOC}(\Omega)$, (3.6) se scrie:

$$\left\langle \frac{\partial T_f}{\partial x_i}, \varphi \right\rangle = -\int_{\Omega} f \frac{\partial \varphi}{\partial x_i} \, dx, \qquad (3.7)$$

iar dacă presupunem că derivata (3.7) este tot o distribuție de tip standard, atunci $\exists g_i \in L^1_{LOC}(\Omega)$, a.î.:

$$-\int_{\Omega} f \frac{\partial \varphi}{\partial x_i} dx = \langle T_{g_i}, \varphi \rangle = \int_{\Omega} g_i \varphi \, dx.$$
(3.8)

Se spune atunci că f este diferențiabilă în sens slab, iar derivata parțială generalizată (sau în sens slab)([73], [46]), în raport cu variabila x_i este chiar funcția g_i și:

$$\int_{\Omega} f \frac{\partial \varphi}{\partial x_i} dx = -\int_{\Omega} g_i \varphi \, dx, \ \forall \varphi \in \mathcal{D}(\Omega).$$
(3.9)

Spre deosebire de derivata clasică ce descrie punctual tendința de evoluție a funcției f, derivata generalizată indică *efectul* pe care această tendință îl are asupra oricărui membru al unui set de funcții de test. Acest efect este relevat prin integrala din membrul drept al (3.9), efect egal cu cel al funcției f asupra derivatei analoage a funcției de test (membrul stâng al (3.9)). In acest mod, condiția restrictivă, de continuitate a funcției f în fiecare punct al domeniului Ω , se relaxează la aceea de continuitate pe porțiuni pe același domeniu; de unde și denumirea de formă slabă a derivatei funcției f. Cu ajutorul relației (3.9) se pune în evidență nu numai evoluția funcției f pe Ω , dar și modul de propagare a componentelor ei pe frontierele de separație între subdomeniile (porțiunile) pe care aceasta este continuă.

Similar relației (3.9) se scrie și derivata generalizată de ordin α dacă $\exists g_{\alpha} \in L^{p}(\Omega)$ astfel încât:

$$\int_{\Omega} v \frac{\partial^{\alpha} \varphi}{\partial x^{\alpha}} dx = (-1)^{\alpha} \int_{\Omega} g_{\alpha} \varphi dx, \ \forall \alpha = (\alpha_1, \cdots, \alpha_m) \in \mathbb{N}^m \text{ si } \forall \varphi \in \mathcal{D}(\Omega).$$
(3.10)

Distribuțiile standard sunt asociate unor funcții clasice, continui sau nu, iar distribuția asociată este mereu derivabilă; derivata (3.10) nu mai este mereu o distribuție standard, ci poate fi una de tip Dirac, care nu mai este asociată unei funcții clasice.

3.2.2 Spații vectoriale Sobolev.

3.2.2.1 Definiții

In contextul celor prezentate în cap. 3.1, spațiile vectoriale Sobolev pot fi definite astfel ([35]):

$$W^{m,p} = \{ v \in L^p(\Omega); \, \partial^{\alpha} \, v \in L^p(\Omega), \, \forall \alpha \le m \}, \ 1 \le p < \infty, \, m \in \mathbb{N}, \tag{3.11}$$

unde L^p este clasa spațiilor de funcții integrabile Lebesgue ([97],[46]) - spații Banach de funcții definite pe un domeniu deschis și mărginit Ω din \mathbb{R}^n , de putere p integrabilă Lebesgue:

$$L^{p}(\Omega) = \left\{ v : \int_{\Omega} |v(x)|^{p} dx < +\infty \right\} \text{ cu } 1 \le p < \infty.$$
(3.12)

cu norma
$$||v||_{L^p} = \left[\int_{\Omega} |v(x)|^p dx\right]^{1/p}$$
. (3.13)

Se poate afirma că spațiile Sobolev $W^{m,p}(\Omega)$ conțin toate funcțiile din $L^p(\Omega)$ pentru care toate derivatele slabe până la ordinul m (cf. (3.10)) sunt tot din $L^p(\Omega)$, sau, întro altă exprimare, spațiile Sobolev conțin toate funcțiile cu grad de derivabilitate m și de integrabilitate p. Practic, spațiile descrise de (3.11) conțin funcțiile ce sunt soluții ale sistemelor de ecuații cu derivate parțiale de ordin maxim m. Se demonstrează ([100] - Teorema 10.5 și Ex.10.9) că spațiul $W^{m,p}(\Omega)$ este un spațiu Banach, înzestrat cu norma:

$$\|u\|_{W^{m,p}} \stackrel{\text{not}}{=} \|u\|_{m,p} = \left[\int_{\Omega} \sum_{0 \le |\alpha| \le m} |\partial^{\alpha} u|^{p} dx\right]^{1/p} = \left[\|u\|_{L_{p}}^{p} + \int_{\Omega} \sum_{1 \le |\alpha| \le m} |\partial^{\alpha} u|^{p} dx\right]^{1/p} 1 \le p < \infty,$$

$$(3.14)$$

respectiv seminorma:

$$|u|_{m,p} = \left[\int_{\Omega} \sum_{|\alpha|=m} |\partial^{\alpha} u|^p \, dx\right]^{1/p}.$$
(3.15)

Ținând cont de definirea și proprietățile de convergență ale spațiului $\mathcal{D}(\Omega)$ (cf. (3.2), respectiv (3.3)), putem concluziona că, pentru $m \geq 1$, toate limitele șirurilor de distribuții sunt incluse într-un spațiu Sobolev, notat cu $W_0^{m,p}(\Omega)$, reprezentând, deci, închiderea spațiului $\mathcal{D}(\Omega)$ în raport cu norma $\|\cdot\|_{m,p}$:

$$W_0^{m,p}(\Omega) = \overline{\mathcal{D}(\Omega)}^{\|\cdot\|_{m,p}},\tag{3.16}$$

iar $W_0^{m,p}(\Omega) \subset W^{m,p}(\Omega)$ sunt funcțiile ce satisfac condiții omogene pe întreaga frontieră $\partial\Omega$. In cazul particular m = 0, se poate observa că închiderea spațiului $\mathcal{D}(\Omega)$ este $L^p(\Omega)$ - spațiul funcțiilor integrabile, dar nederivabile. Aceste relații de închidere, denumite și proprietăți de densitate, reprezintă punctul de plecare pentru stabilirea cadrului funcțional care asigură existența și unicitatea soluțiilor formulărilor slabe ale problemelor multifizice (v. și proprietățile de densitate date de teorema Meyers și Serrin - (3.30)).

Mulțimea funcționalelor definite pe spațiul $W_0^{m,p}$, cu $1 \leq p < \infty$, formează spațiul vectorial dual, notat în mod uzual cu $W^{-m,p'}$, și este format numai din distribuții care se pot descrie prin ([110]):

$$f: W_0^{m,p} \longrightarrow W^{-m,p'} \quad 1 \le p < \infty, \text{ cu}$$

$$f = \sum_{\alpha} \partial^{\alpha} f_{\alpha} \quad (f_0, \cdots, f_m) \in L^{p'} \text{ si}$$
(3.17)

$$f_{|\alpha| \le m} = \int_{\Omega} \left(\sum_{|\alpha| \le m} f_{\alpha} \partial^{\alpha} u \right) dx, \qquad (3.18)$$

unde p' este complementul Hölder al lui p ([97]-inegalitatea Hölder):

$$\frac{1}{p} + \frac{1}{p'} = 1, \tag{3.19}$$

Spațiul $W^{-m,p'}$ este înzestrat cu norma tipică pentru spațiile duale ([97] - Definiția 2.10-3), derivată din relația de dualitate standard (3.18):

$$||f||_{-m,p'} = \sup_{\substack{u \in W_0^{m,p} \\ u \neq 0}} \frac{\langle f, u \rangle}{||u||_{m,p}},$$
(3.20)

Pentru a putea studia comportamentul funcțiilor din $W^{m,p}(\Omega)$, pe frontiera $\partial\Omega$, este utilă definirea spațiilor Sobolev fracționare [72], notate cu $W^{s,p}$, unde m < s < m+1și dotate cu seminorma:

$$|u|_{s,p} = \int_{\Omega} \int_{\Omega} \left(\frac{|u(x) - u(y)|}{\|x - y\|^{N/p + s - m}} \right)^p dx \, dy \tag{3.21}$$

Astfel, exprimând $s = m + \sigma$, cu $0 < \sigma < 1$, $m \in \mathbb{N}$, un spațiu Sobolev fracționar pote fi definit ca:

$$W^{s,p} = \{ u \in W^{m,p}(\Omega) \mid |\partial^{\alpha} u|_{s,p} < +\infty, \,\forall \, |\alpha| = m, \, \Omega \subset \mathbb{R}^N \},$$
(3.22)

înzestrat cu norma:

$$||u||_{s,p} = \left[||u||_{m,p}^{p} + \sum_{|\alpha|=r} |\partial^{\alpha}u|_{s,p}^{p} \right]^{1/p}$$
(3.23)

O tratare detaliată referitoare la spațiile Sobolev fracționare, cunoscute și ca spații Sobolev - Slobodeckij, se găsește în [39]. Similar cu spațiile $W^{m,p}$, pentru spațiile $W^{s,p}$ se definesc dualele $W^{-s,p'}$. In ceea ce privește proprietățile de densitate, nu se poate arăta că spațiul $\mathcal{D}(\Omega)$ este dens în $W^{s,p}$ pentru orice s ([100]). Rezultă deci că:

$$W_0^{m,p}(\Omega) \subset W^{s,p}(\Omega). \tag{3.24}$$

In rest, se demonstrează ([110]-pp.96-99) că proprietățile spațiilor Sobolev întregi se extind și pentru spațiile Sobolev fracționare. Mai înainte însă de a discuta despre proprietățile esențiale ale spațiillor Sobolev, trebuie subliniat că, în cazul definirii pe un spațiu mărginit Ω , este necesar ca frontiera acestuia, $\partial\Omega$, să fie caracterizată de anumite proprietăți de regularitate (netezime) care sunt definite precis prin conceptul de domeniu Lipschitz, prezentat în continuare.

3.2.2.2 Domenii Lipschitz

Fie $\Omega \subset \mathbb{R}^3$ un domeniu, deschis, simplu conex și mărginit de frontiera $\partial\Omega$. Putem defini, pe această frontieră, funcții de clasă $\mathcal{C}^{\infty}(\partial\Omega)$ ifinit derivabile, dacă frontiera $\partial\Omega$ este continuă în sens Lipschitz [123], adică este caracterizată de următoarele proprietăți [72], [159], [55]:

1. Poate fi definită de un set finit de domenii w_i fiecare reprezentând câte o vecinătate a unui punct oarecare $y_i \in \partial \Omega$:

$$\partial \Omega \subset \cup w_i \tag{3.25}$$

Fiecare astfel de domeniu are asociat un set de coordonate ortogonale locale $(x_{1_i}, x_{2_i}, x_{3_i});$

2. Pe fiecare domeniu w_i se poate defini o funcție $f_i(x_{1_i}, x_{2_i})$, continuă în sens Lipschitz [123] astfel încât, f_i să reprezinte limita frontierei $\partial \Omega$ pe fiecare w_i :

$$\Omega \cap w_i = \{ (x_{1_i}, x_{2_i}, x_{3_i}) | x_{3_i} < f_i(x_{1_i}, x_{2_i}) \}$$

$$(3.26)$$

$$\partial \Omega \cap w_i = \{ (x_{1_i}, x_{2_i}, x_{3_i}) | x_{3_i} = f_i(x_{1_i}, x_{2_i}) \}$$
(3.27)

Conform celor de mai sus, frontiera $\partial\Omega$ este formată dintr-un set de subdomenii descrise parametric prin funcții definite pe \mathbb{R}^2 care în mod ideal ar trebui să fie netede peste toată $\partial\Omega$. Această exprimare este echivalentă afirmația că suprafața cu formă complexă $\partial\Omega \subset \mathbb{R}^3$ ar fi obținută prin proiecția continuă unui plan din \mathbb{R}^2 , al coordonatelor parametrice. De aici și necesitatea de derivabilitate, respectiv mărginire în sens Lipschitz, a funcției f din (3.26), (3.27). Generalizând, se poate spune că un spațiu mărginit $\Omega \in \mathbb{R}^N$ are o frontieră finită $\partial \Omega \in \mathbb{R}^N$, echivalentă cu un plan în N-1 dimensiuni - un *hiperplan*, proiectat pe $\partial \Omega$. $\partial \Omega$ este o frontieră Lipschitz, dacă aproape peste tot, a N-a coordonată a oricărui punct de pe $\partial \Omega$ este dată de o funcție mărginită în sens Lipschitz. Astfel, această funcție este fie netedă peste tot, $\partial \Omega$ putând fi în acest caz o sferă, fie $\partial \Omega$ este o reuniune disjunctă de tip (3.25) și pentru $\forall \omega_i$ din (3.25) există funcția Lipschitz $x_{N_i} = f_i(x_{1_i}, x_{2_i}, \cdots, x_{N-1_i})$ (v. fig.3.1). Așa cum se va vedea în capitolul următor,



Figura 3.1: (a)- domeniu mărginit de o suprafață netedă peste tot (sferică): coordonatele sunt definite parametric: $x_1(r, \varphi, \theta) = r \sin\theta\cos\varphi$, $x_2(r, \varphi, \theta) = r \sin\theta\sin\varphi$, $x_3(r, \varphi, \theta) = r \cos\theta$; pentru $\forall P \in \Omega$, $r(x) < f(\varphi, \theta)_{|\partial\Omega} = R$; (b) - domeniu Lipschitz sferic cu frontiera $\partial\Omega$, volum discretizat într-o reuniune disjunctă de volume tetraedrale, ce aproximează suprafața $\partial\Omega$ printr-o reuniune disjunctă de suprafețe triunghiulare $\{\omega_i\}$; poziția oricărui punct de pe suprafața ω_i poate fi exprimată față de poziția oricărui punct din Ω prin transformarea afină: $\vec{r}_{|\omega_i|} = \vec{r}_{\Omega} + \lambda \vec{v}, \lambda \in \mathbb{R}$., vectorul $\vec{r}_{|\omega_i|}$ find invariant, aproape peste toată frontiera $\partial\Omega$, față de poziția \vec{r}_{Ω} . Excepție fac punctele de pe muchiile comune ale suprafețelor $\{\omega_i\}$ pentru care câmpul vectorial al normalelor la aceste suprafețe nu mai este continuu.

pentru a genera formulări numerice, domeniile 3D din lumea reală, sunt aproximate prin discretizare într-o rețea de volume poliedrale. În consecință, frontiera unui astfel de domeniu aproximativ este o rețea finită de suprafețe poligonale. Echivalent, în 2D, discutăm de o rețea de discretizare formată din suprafețe poligonale, mărginită de o frontieră poligonală.

Spunem că aceste frontiere sunt continue Lipschitz, dacă pe aproape peste toată frontiera au sens definirea vectorului normală exterioară \vec{n} , respectiv a elementului de arie dA; excepțiile sunt reprezentate de către un set finit de puncte - anumite muchii, respectiv colțuri. Orice domeniu cu o astfel de frontieră este denumit domeniu Lipschitz. El este suficient de neted pentru a permite scrierea relațiilor de tip Gauss-Ostrogradski și Stokes și poate avea forme suficent de generale pentru toate aplicațiile practice, inginerești. Condiția de păstrare a uniformității "aproape peste tot" este aceea ca domeniul Ω să se afle pe aceeași parte a frontierei $\partial \Omega$. Astfel de geometrii permit existența colțurilor și a muchiilor, dar nu pot conține puncte care ar impune ca $\partial \Omega$ să lase tăieturi prin domeniul Ω .



Figura 3.2: Domeniu ce nu îndeplinește condițiile Lipschitz: pentru $\forall y_i \in [b,c], \exists$ o vecinătate w_i în care găsim o pereche (r_1, r_2) pentru care $f(r_1) < \pi/2$ și $f(r_2) > \pi/2$. Domeniul Ω se află de ambele părți ale porțiunii de frontieră BC

Un exemplu de domeniu care nu este Lipschitz este cel din fig. 3.2 în care, pentru orice punct de pe porțiunea BC există o vecinătate w_i astfel încât:

$$f_{i}: w_{i} = [b, c] \to \mathbb{R}, \ f_{i}(r) := \begin{cases} f_{i}(r_{1}) < \pi/2 \ \text{si} \ f_{i}(r_{2}) > \pi/2 & cu \ \{r_{1}, r_{2}\} \in w_{i} \cap \Omega \\ \pi/2 & pe \ w_{i} \cap \partial\Omega. \end{cases}$$
(3.28)

O discuție detaliată despre domeniile Lipschitz poate fi găsită în [110]-pp.89-93. În continuare vom presupune că toate domeniile de calcul, pe care sunt definite câmpurile vectoriale sunt domenii Lipschitz.

3.2.2.3 Proprietăți de densitate

In [35] și ([100]- Teorema10.15) este prezentată demonstrația unei importante teoreme a matematicienilor Meyers și Serrin a cărei principală concluzie este că funcțiile netede peste tot \mathbb{R}^N sunt dense în spațiul Sobolev $W^{m,p}(\Omega)$. Fie:

$$\mathcal{D}(\overline{\Omega}) = \left\{ \varphi \,|\, \varphi \in \mathcal{D}(\mathbb{R}^N) \right\},\tag{3.29}$$

extinderea spațiului distribuțiilor (3.2) la întreg \mathbb{R}^N . Cu această notație, se poate spune că teorema Meyers și Serrin afirmă că

spațiul
$$\mathcal{D}(\overline{\Omega})$$
 este dens în $W^{m,p}$, pentru $\forall m \ge 0$ și $1 \le p < \infty$. (3.30)

Consecința esențială a acestei teoreme este că elementele din spațiul Sobolev $W^{m,p}(\Omega)$, respectiv soluții ale sistemelor de ecuații cu derivate parțiale de ordin maxim m, pot fi aproximate oricât de bine cu funcții de test definite pe $\mathbb{R}^{N}([41], [38], [73], [84])$. Practic, proprietatea de densitate (3.30) arată echivalența sistemelor de ecuații cu derivate parțiale, asociate unei anumite formulări matematice tari, având ca soluții funcții clasice derivabile punctual, cu sistemele de ecuații ale formulării slabe corespondente, scrise cu funcții generalizate din spațiile Sobolev (v.(3.10)). Astfel de proprietăți de densitate stau la baza tuturor demonstrațiilor care verifică echivalența pomenită mai sus.

O altă consecință, utilă în tratarea condițiilor de frontieră asociate sistemelor de ecuații diferențiale cu derivate parțiale, se referă la faptul că spațiul $W_0^{m,p}(\Omega)$ definit în (3.16) reprezintă închiderea spațiului $C_0^m(\Omega)$:

$$W_0^{m,p}(\Omega) = \overline{\mathcal{C}_0^m(\Omega)}^{\|\cdot\|_{m,p}},\tag{3.31}$$

iar $W_0^{m,p}(\Omega)$ poate fi definit ca ([43] - Lema 8.9):

$$W_0^{m,p}(\Omega) = \{ v \in W^{m,p}(\Omega) \mid v = \partial^{\alpha} v = 0 \text{ pe } \partial\Omega, \forall \alpha \le m \}.$$
(3.32)

Notăm cu \tilde{v} extensia prin 0 în afara domeniului Ω a unei funcții $v \in W^{m,p}(\Omega)$:

$$\tilde{v}(x) = \begin{cases} v(x) & \text{pentru } x \in \Omega \\ 0 & \text{pentru } x \in \mathbb{R}^N \setminus \Omega \end{cases}$$
(3.33)

Relația (3.32) arată că dacă $\tilde{v} \in W^{m,p}(\Omega)$, atunci $v \in W_0^{m,p}(\Omega)$. Se poate demonstra astfel ([110]- Teorema A.4) că, dacă Ω este un domeniu mărginit și $m \ge 1$, se poate găsi un operator de extensie liniar și continuu T:

$$T: W^{m,p}(\Omega) \to W^{m,p}(\mathbb{R}^N), \text{ a.i. } Tv_{|_{\Omega}} = v, \ \forall v \in W^{m,p}(\Omega).$$
(3.34)

Extensiile definite în (3.33) și (3.34) permit demonstrarea proprietăților funcțiilor din $W^{m,p}(\Omega)$, pornind de la cazul general $\Omega = \mathbb{R}^N$. In virtutea celor de mai sus, această metodă se poate utiliza în cazul domeniilor Lipschitz. În lucrarea de față nu vor fi folosite decât spații Sobolev cu p = 2. Legate de acestea, o a treia importantă proprietate (demonstrată în [110]-Teorema 3.16) arată că dacă Ω este un domeniu Lipschitz și $s \geq 0$, atunci:

$$W^{s,2}(\Omega) = H^s(\Omega), \tag{3.35}$$

normele celor două spații fiind echivalente. Cazul p = 2, respectiv spațiile H^s , reprezintă singura clasă de spații Sobolev care sunt Hilbert, pe ele putându-se defini

produsul scalar a două funcții, respectiv norma indusă de acesta (v. [97]-Cap. 3.1-1). Ca și în cazul general, se face diferențierea între spațiile H^m cu $m \in \mathbb{N}$ și spațiul fracționar H^s cu $s = m + \sigma$ unde $0 < \sigma < 1$. In acest sens, se poate observa că pentru p = 2, norma (3.14) este indusă din produsul scalar:

$$(u,v)_m = \sum_{|\alpha| \le m} \int_{\Omega} \partial^{\alpha} u(x) \partial^{\alpha} v(x), \text{ pentru } x \in \mathbb{R}, \text{ respectiv}$$
$$(u,v)_m = \sum_{|\alpha| \le m} \int_{\Omega} \overline{\partial^{\alpha} u(x)} \partial^{\alpha} v(x), \text{ pentru } x \in \mathbb{C},$$
(3.36)

unde \overline{x} este conjugatul complex al numărului complex x. În mod similar, pentru spațiile H^s , norma (3.23), este indusă de produsul scalar:

$$(u,v)_s = (u,v)_m + \sum_{|\alpha|=m} \int_{\Omega} \int_{\Omega} \frac{\overline{[\partial^{\alpha} u(x) - \partial^{\alpha} u(y)]} [\partial^{\alpha} u(x) - \partial^{\alpha} u(y)]}{|x-y|^{N+2\sigma}} dx \, dy \quad (3.37)$$

Spațiul H^m poate fi definit ca:

$$H^{m}(\Omega) = \{ v \in L^{2}(\Omega); \, \partial^{\alpha} v \in L^{2}(\Omega), \, \forall \alpha \leq m \}, \quad \text{si } \alpha, \, m \in \mathbb{N},$$
(3.38)

pentru care produsul scalar (3.36) induce norma:

$$\|u\|_{H^m} \stackrel{\text{not}}{=} \|u\|_m = \left[\int_{\Omega} \sum_{0 \le |\alpha| \le m} |\partial^{\alpha} u|^2 \, dx\right]^{1/2}, \qquad (3.39)$$

respectiv seminorma:

$$|u|_m = \left[\int_{\Omega} \sum_{|\alpha|=m} |\partial^{\alpha} u|^2 dx\right]^{1/2}, \qquad (3.40)$$

Mulțimea funcționalelor definite pe spațiul H^m , formează spațiul vectorial dual, pentru care notația consacrată este $H^{-m} = (H^m)'$:

$$T: H^m \longrightarrow H^{-m} \tag{3.41}$$

înzestrat cu norma tipică spațiilor duale (v. [97]-Cap 2.7-1):

$$\| T \|_{-m} = \sup_{\substack{u \in H^m \\ u \neq 0}} \frac{\langle T, u \rangle}{\| u \|_m}$$
(3.42)

In legătură cu spațiile H^m este de menționat ([43]-Cap.9) relația de densitate valabilă pentru cazul m = 0, dacă Ω este un domeniu Lipschitz:

$$L^{2}(\Omega) = \overline{\mathcal{C}_{0}^{\infty}(\Omega)}^{\|\cdot\|_{0}}, \qquad (3.43)$$

deoarece pentru m = 0 se regăsește spațiul L^2 al funcțiilor de pătrat integrabil:

$$L^{2}(\Omega) := \{\varphi \mid \int_{\Omega} \varphi^{2} dx < \infty\}$$
(3.44)

înzestrat cu produsul scalar:

$$(u,v)_0 = \int_{\Omega} u \cdot v \, dx, \quad u,v \in [L^2(\Omega)]^k, \, k \in 1,3$$
 (3.45)

respectiv norma indusă de către produsul (3.45):

$$\|u\|_{0} = \left[\int_{\Omega} u^{2} dx\right]^{1/2}$$
(3.46)

Proprietățile de mai sus au fost descrise pentru funcții v, de argument scalar: $v : \Omega \to \mathbb{K}$, unde \mathbb{K} poate fi \mathbb{R} sau \mathbb{C} . Ele pot fi ușor extinse pentru funcții de argument vectorial cu d componente: $\mathbf{v} : \Omega \to \mathbb{K}^d$. In acest caz, notațiile folosite pentru spațiile Sobolev vor fi $[W^{m,p}(\Omega)]^d$, respectiv $[H^m(\Omega)]^d$ și $[L^2(\Omega)]^d$. Problemele fizice analizate în această lucrare, vor fi definite pe domenii 2D (d = 2), respectiv 3D (d = 3), cele mai utilizate spații Hilbert fiind cu $m \in \{0, 1, 2\}$, respectiv $s \in \{1/2, 3/2\}$.

In acest capitol au fost prezentate doar câteva dintre proprietățile de densitate (incluziune) ale spațiilor Sobolev, care ne arată cum elementele spațiilor Sobolev pot fi aproximate oricât de bine de funcții clasice netede. Acestea sunt descrise în formularea mai generală a celebrei teoreme de incluziune a lui Sobolev: sunt stabilite cu exactitate care sunt condițiile pe care trebuie să le îndeplinească parametrii $m \in \mathbb{N}$ și $1 \leq p \leq \infty$ astfel încât să se poată asigura echivalența dintre diferite spații Sobolev și spațiile funcțiilor netede. Enunțul si demonstrația teoremei sunt foarte tehnice și depășesc cadrul acestei lucrări. Ele pot fi găsite în mai multe cărți dedicate subiectului, printre care [35], [114], [110], [100].

Aceleași probleme se pun și pentru funcțiile ce descriu condițiile de frontieră din formulările matematice ale problemelor fizice. În acest caz însă, lucrurile sunt mai complicate, datorită neregularităților frontierelor ce descriu modelul fizic, fiind necesară apelarea la spațiile Sobolev fracționare, precum și la operatori diferențiali dedicați. De aceea pentru acest caz, am rezervat mai jos un subcapitol separat.

3.2.3 Operatori diferențiali pe domeniul Ω tridimensional

Prezentul capitol, enumeră principalii operatori utilizați în formulările matematice "tari" ale problemelor fizice studiate în această lucrare. Mai jos, sunt aplicații ale acestora pentru funcții scalare φ din $L^2(\Omega)$, respectiv funcții vectoriale \vec{u} din $[L^2(\Omega)]^3$, unde Ω este un domeniu $\Omega \subset \mathbb{R}^3$:

G - gradient:

$$\operatorname{\mathbf{grad}} \varphi = \nabla \varphi = \frac{\partial \varphi}{\partial x_1} \cdot \vec{1}_{x_1} + \frac{\partial \varphi}{\partial x_2} \cdot \vec{1}_{x_2} + \frac{\partial \varphi}{\partial x_3} \cdot \vec{1}_{x_3}$$
(3.47)

unde $\vec{1}_{x_i}$ reprezintă versorul coordonatei x_i ; i = 1, 2, 3,

R - rotor:

$$\operatorname{\mathbf{rot}} \vec{u} = \nabla \times \vec{u} = \begin{vmatrix} \vec{1}_{x_1} & \vec{1}_{x_2} & \vec{1}_{x_3} \\ u_1 & u_2 & u_3 \\ \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \frac{\partial}{\partial x_3} \end{vmatrix}$$
(3.48)

D - divergență:

$$\operatorname{\mathbf{div}} \vec{u} = \nabla \cdot \tilde{\mathbf{u}} = \frac{\partial u_1}{\partial x_1} + \frac{\partial u_2}{\partial x_2} + \frac{\partial u_3}{\partial x_3}, \qquad (3.49)$$

unde $\vec{u} = \sum_{i=1}^{3} u_i \cdot \vec{1}_{x_i}$,

Dacă pentru exprimarea operatorilor (3.47), (3.48) și (3.49) se folosesc derivate generalizate (cf. (3.9)) rezultă expresiile operatorilor generalizați corespunzători și anume ([73]):

1 Gradientul generalizat $\vec{g} = \nabla \varphi$ ([52]) al funcției scalare $\varphi \in L^2(\overline{\Omega})$ astfel încât:

$$\int_{\Omega} \varphi \nabla \vec{v} \, dx = -\int_{\Omega} \vec{g} \cdot \vec{v} \, dx, = -\int_{\Omega} (\nabla \varphi) \cdot \vec{v} \, dx, \ \forall \vec{v} \in [\mathcal{D}(\Omega)]^3 \tag{3.50}$$

2 Rotorul generalizat $\vec{r} = \nabla \times \vec{u}$ al funcției vectoriale $\vec{u} \in [L^2(\Omega)]^3$ astfel încât:

$$\int_{\Omega} \vec{u} \cdot (\nabla \times \vec{v}) \, dx = \int_{\Omega} \vec{r} \cdot \vec{v} \, dx, = \int_{\Omega} (\nabla \times \vec{u}) \cdot \vec{v} \, dx, \quad \forall \mathbf{v} \in [\mathcal{D}(\Omega)]^3 \quad (3.51)$$

3 Divergența generalizată $d = \nabla \cdot \vec{u}$ a funcției vectoriale $\vec{u} \in [L^2(\Omega)]^3$ astfel încât:

$$\int_{\Omega} \vec{u} \cdot (\nabla \varphi) \, dx = -\int_{\Omega} d\varphi \, dx, = -\int_{\Omega} (\nabla \cdot \vec{u}) \, \varphi \, dx, \quad \forall \varphi \in \mathcal{D}(\Omega) \tag{3.52}$$

Pe baza definițiior operatorilor generalizați de mai sus, se poate demonstra ușor că și pentru o distribuție scalară $\psi \in \mathcal{D}'(\Omega)$, respectiv vectorială $\vec{u} \in [\mathcal{D}'(\Omega)]^3$, există, în sens generalizat, relațiile:

$$\nabla \times (\nabla \psi) = 0, \tag{3.53}$$

$$\nabla \cdot (\nabla \times \vec{u}) = 0. \tag{3.54}$$

3.2.4 Spații Sobolev și operatori diferențiali pe frontiera $\partial \Omega$

Datorită particularităților frontierei $\partial\Omega$, a domeniului Ω , discutate în subcapitolul (3.2.2.2), se constată că funcțiile din spațiile Sobolev $W^{m,p}(\Omega)$, nu își mai păstrează proprietățile de derivabilitate pe toată această frontieră ([114], [41]). Așa cum am arătat în cap. 3.2.2.2 (exemplul din fig. 3.1) frontiera $\partial\Omega$ a domeniului Lipschitz OM este în 3D o reuniune disjunctă de suprafețe poligonale, respectiv în 2D un contur poligonal (reuniune disjunctă de segmente drepte sau curbe). Proprietatea de densitate (3.31), ce presupune convergența uniformă a tuturor șirurilor din $\mathcal{D}(\Omega)$ în $W_0^{m,p}(\Omega)$ nu mai este valabilă pe toate muchiile dintre suprafețele, respectiv în toate punctele dintre segmentele, ce formează frontiera $\partial\Omega$.

Necesitatea definirii unor spații Sobolev pentru condițiile de frontieră este justificată de formulările slabe derivate din formulările tari ale problemelor, prin proiecția, în sens distribuțional, a acestora din urmă pe funcții de test, scalare sau vectoriale, prin relații de tip (3.5). În mod uzual, dezvoltarea acestor proiecții se efectuează în baza teoremelor Gauss-Ostrogradski, sau Stokes ce se pot demonstra relativ ușor pentru un domeniu Ω mărginit de o suprafață Lipschitz $\partial \Omega$ ([110]-Teorema 3.34):

Teorema 3.2.1. Fie o funcție vectorială $\vec{u} \in [C^1(\Omega)]^d$, unde $\vec{u} : \mathbb{R}^d \to \mathbb{R}^d$, și $\Omega \subset \mathbb{R}^d$ este un domeniu Lipschitz, mărginit de frontiera $\partial\Omega$ pe care se poate defini vectorul normală exterioară de versor \vec{n} pentru d = 3, respectiv tangent la frontieră de versor $\vec{\tau}$, pentru d = 2. Atunci, pentru cele două cazuri, există relațiile:

$$\int_{\Omega} \nabla \cdot \vec{v} \, d\mathcal{V} = \int_{\partial \Omega} \vec{v} \cdot \vec{n} \, d\mathcal{A} \, (Gauss-Ostrogradski) \tag{3.55}$$

pentru un domeniu $\Omega \subset \mathbb{R}^3$, și $\vec{v} \in [C^1(\Omega)]^3$, respectiv:

$$\int_{\Omega} \nabla \times \vec{v} \cdot \vec{n} \, d\mathcal{A} = \int_{\partial \Omega} \vec{v} \cdot \vec{\tau} \, d\mathcal{S} \ (Stokes) \tag{3.56}$$

pentru un domeniu $\Omega \subset \mathbb{R}^2$, și $\vec{v} \in [C^1(\Omega)]^2$

De unde rezultă imediat relațiile Green, de integrare prin părți, în care $\varphi \in C^1(\Omega)$, iar $\vec{v}, \vec{u} \in [C^1(\Omega)]^3$:

$$\int_{\Omega} \vec{u} \cdot (\nabla \varphi) \, d\mathcal{V} = -\int_{\Omega} \varphi \cdot (\nabla \vec{u}) \, d\mathcal{V} + \int_{\partial \Omega} \varphi \vec{u} \cdot \vec{n} \, d\mathcal{A} \tag{3.57}$$

$$\int_{\Omega} (\nabla \times \vec{u}) \cdot \vec{v} \, d\mathcal{V} = \int_{\Omega} \vec{u} \cdot \nabla \times \vec{v} \, d\mathcal{V} - \int_{\partial \Omega} \vec{v} \cdot (\vec{u} \times \vec{n}) \, d\mathcal{A} \tag{3.58}$$

$$\int_{\Omega} \varphi \left(\nabla \cdot \vec{v} \right) d\mathcal{V} = -\int_{\Omega} \vec{v} \cdot \nabla \varphi \, d\mathcal{V} + \int_{\partial \Omega} \varphi \vec{v} \cdot \vec{n} \, d\mathcal{A} \tag{3.59}$$

Se poate observa cum identitățile integrale (3.57) - (3.59) se reduc la expresiile operatorilor generalizați (3.47) - (3.49), dacă impunem ca funcțiile de test φ și \vec{v} să aibă suport compact pe Ω . Ca urmare, pe $\partial\Omega$ ar trebui definit un spațiu Sobolev pentru care noțiunile de restricție $\varphi_{|\partial\Omega}$ și $\vec{v}_{|\partial\Omega}$ a unei funcții φ , respectiv \vec{v} definite pe Ω să aibă sens.

Un prim pas, în acest sens, ar fi definirea operatorilor gradient (∇) , rotor $(\nabla \times)$, respectiv divergență $(\nabla \cdot)$ pe frontiera $\partial \Omega$ a domeniului $\Omega \subset \mathbb{R}^3$. Reprezentarea parametrică a frontierei $\partial \Omega$ (v. fig. 3.1) este dată de un câmp vectorial:

$$\vec{r} = \vec{r}(p_1, p_2), \text{ cu } \vec{r} : \mathbb{R}^2 \to \mathbb{R}^3$$

si $\vec{r} = (r_1(p_1, p_2), r_2(p_1, p_2), r_3(p_1, p_2))^T.$
(3.60)

Putem defini pe această suprafață o funcție scalară $F(p_1, p_2)$, cu $F : \partial \Omega \to H^1(\partial \Omega)$, unde spațiul H^1 este spațiul Sobolev H^m cu m = 1 (v. cap.3.2.5). Definim gradientul de suprafață al funcției F prin:

$$\nabla_2 F = \frac{\partial F}{\partial p_1} \frac{\partial \vec{r}}{\partial p_1} + \frac{\partial F}{\partial p_2} \frac{\partial \vec{r}}{\partial p_2}, \qquad (3.61)$$

unde $\vec{\tau}_{p_1} = \frac{\partial \vec{r}}{\partial p_1}$, respectiv $\vec{\tau}_{p_2} = \frac{\partial \vec{r}}{\partial p_2}$ reprezintă vectori ortogonali, tangenți la suprafața $\partial \Omega$ în punctul de calcul, generând planul tangent la $\partial \Omega$ în acel punct. Prin urmare dacă definim spațiul câmpurilor vectoriale tangente la frontiera $\partial \Omega$ ca:

$$L_{\tau}^{2} = \left\{ \vec{u} \,|\, \vec{u} \in [L^{2}(\partial\Omega)]^{3}, \, \vec{n} \cdot \vec{u} = 0, \text{ aproape peste tot pe } \partial\Omega \right\},$$
(3.62)

putem exprima domeniul de definiție și codomeniul operatorului gradient de suprafață prin:

$$\nabla_2 : H^1(\partial\Omega) \to L^2_\tau(\partial\Omega). \tag{3.63}$$

Gradientul de suprafață este o măsură a variației funcției scalare F pe frontiera $\partial\Omega$, direcția sa indicând punctul de pe suprafață în care F are o valoare (local) extremă. Se pot remarca două relații utile pentru exprimarea componentelor tangențiale ale funcțiilor pe frontiera $\partial\Omega$ (v. fig. 3.3) și anume:

$$\nabla_2 F = -(\nabla F \times \vec{n}) \times \vec{n} \tag{3.64}$$

$$\nabla_2 F = \nabla F_{|\partial\Omega} - \frac{\partial F}{\partial n}\vec{n}$$
(3.65)

Din cele de mai sus rezultă, în mod natural, expresia divergenței de suprafață, operator notat cu ∇_2 , ce se poate aplica, vectorilor din planul tangent la $\partial\Omega$ întrun punct al acesteia. Altfel spus:

$$\nabla_2 \colon L^2_\tau(\partial\Omega) \to L^2(\partial\Omega), \tag{3.66}$$



Figura 3.3: Gradientul superficial al funcției F pe frontiera $\partial\Omega$; se pot observa cele două componente, tangențiale la suprafața $\partial\Omega$ în originea corespunzătoare vectorului $\nabla F: \nabla_2 F$, respectiv $\vec{n} \times \nabla F$, precum și compunerea vectorială (3.64). Este pus în evidență și rotorul superficial vectorial $\vec{\nabla}_2 \times$, ca o rotație cu $\pi/2$ a gradientului $\nabla_2 F$ (v. (3.69)).

Dacă $\vec{u} \in L^2_{\tau}(\partial\Omega)$, atunci se poate scrie divergența de suprafață în sens generalizat, ca $d_2 = \nabla_2 \cdot \vec{u}$, astfel încât:

$$\int_{\partial\Omega} \vec{u} \cdot (\nabla_2 \psi) \, d\mathcal{A} = -\int_{\partial\Omega} d_2 \psi \, d\mathcal{A} = -\int_{\partial\Omega} (\nabla_2 \cdot \vec{u}) \psi \, d\mathcal{A}, \quad \forall \, \psi \in H^1(\Omega) \quad (3.67)$$

Pentru o suprafață $\partial\Omega$ netedă, expresia clasică a divergenței superficiale (v. discuția și relațiile (4.110) din cap. 4.1.6) rezultă ca un caz particular al divergenței superficiale generalizate (3.67):

$$\nabla_2 \cdot \vec{u} = (\vec{u}_2 - \vec{u}_1) \cdot \vec{n}_{12}. \tag{3.68}$$

In sfârșit, pentru operatorul rotor ($\nabla \times$), fiind o formă diferențială de ordin 2, în cazul domeniilor bidimensionale se pot defini două variante, în funcție de operand (scalar sau vectorial) ([159],[114]):

1. Rotorul superficial vectorial, notat cu $\overrightarrow{\nabla}_2 \times$, sau **Rot** este aplicat unei distribuții scalare (formă diferențială de ordin 0) rezultând un vector cu două componente nenule(formă diferențială de ordin 2); el este determinat printr-o rotație cu $\pi/2$ a gradientului superficial (v. (3.63)) al unei funcții scalare din
$H^1(\partial\Omega)$ (v. fig. 3.3):

$$\overrightarrow{\nabla}_2 \times : H^1(\partial\Omega) \to L^2_\tau(\partial\Omega),$$
 (3.69)

$$\vec{\nabla}_2 \times F = -\vec{n} \times \nabla_2 F; \qquad (3.70)$$

Dacă $F = F(p_1, p_2)$ atunci se poate scrie:

$$\overrightarrow{\nabla}_2 \times F = \left(\frac{\partial F}{\partial x_{p_2}}, -\frac{\partial F}{\partial x_{p_1}}\right). \tag{3.71}$$

2. Rotorul superficial clasic ($\nabla_2 \times$, sau *rot*), aplicat distribuțiilor vectoriale cu două componente nenule, rezultând un vector cu o singură componentă, putând fi asimilat unei forme diferențiale de ordin 0, fapt care atrage operatorului numele de *rotor scalar*:

$$\nabla_2 \times : L^2_{\tau}(\partial \Omega) \to L^2(\partial \Omega), \text{ a.î. dacă } \vec{u} \in L^2_{\tau}(\partial \Omega) \text{ atunci:}$$
 (3.72)

$$\int_{\partial\Omega} (\nabla_2 \times \vec{u}) \psi \, d\mathcal{A} = \int_{\partial\Omega} \vec{u} \cdot \vec{\nabla} \times \psi \, d\mathcal{A}, \quad \forall \, \psi \in H^1(\partial\Omega); \tag{3.73}$$

Dacă $\vec{u} = (u_1(p_1, p_2), u_2(p_1, p_2))^T$, atunci de poate scrie:

$$\nabla_2 \times \vec{u} = \frac{\partial u_2}{\partial x_{p_1}} - \frac{\partial u_1}{\partial x_{p_2}}.$$
(3.74)

Pasul următor constă în stabilirea spațiilor Sobolev compatibile cu operatorii susmenționați. In [110]-(Teorema 3.38) este demonstrată următoarea:

Teorema 3.2.2. Fie $\Omega \in \mathbb{R}^n$ un domeniu Lipschitz cu o frontieră $\partial\Omega$ și fie $\frac{3}{2} > q > \frac{1}{2}$, pentru care definim un operator $tr_{\partial\Omega} : \mathcal{D}(\Omega) \to \mathcal{D}(\partial\Omega)$ astfel:

$$tr_{\partial\Omega}v(\Omega) = v_{|\partial\Omega} \tag{3.75}$$

In aceste condiții, operatorul $tr_{\partial\Omega}$ admite o unică definire ca operator liniar, continuu, mărginit și inversabil, și anume:

$$tr_{\partial\Omega} : H^q(\Omega) \to H^{q-\frac{1}{2}}(\partial\Omega)$$
 (3.76)

Operatorul (3.76) poartă denumirea de operator de urmă, aplicația $\operatorname{tr}_{\partial\Omega}(v)$ reprezentând urma funcției v pe frontiera $\partial\Omega$. Operatorul inverse ste:

$$\operatorname{tr}_{\partial\Omega}^{-1}: H^{q-\frac{1}{2}}(\partial\Omega) \to H^q(\Omega),$$
 (3.77)

cu:

$$(\operatorname{tr}_{\partial\Omega} \circ \operatorname{tr}_{\partial\Omega}^{-1})(v) = v \quad \forall v \in H^{q-\frac{1}{2}}(\partial\Omega),$$
(3.78)

In plus, deoarece $\mathcal{D}(\Omega)$ este dens în $H^q(\Omega)$ în virtutea relației (3.30), rezultă proprietatea de mărginire, deci și de continutate, a operatorului de urmă:

$$\exists C > 0, \ a.\hat{i}. \|\mathrm{tr}_{\partial\Omega}v\|_{H^{q-\frac{1}{2}}(\partial\Omega)} \le C \|v\|_{H^{q}(\Omega)}, \ \forall v \in \mathcal{D}(\overline{\Omega}).$$
(3.79)

Teorema (3.2.2) arată faptul că este posibilă stabilirea unui spațiu funcțional Sobolev pentru condițiile de frontieră satisfăcute de funcții aparținând unui spațiu Sobolev definit pe domeniul Lipschitz Ω . Legătura dintre cele două spații este asigurată de operatorul tr_{$\partial\Omega$} definit în teoremă. În particular, ținând cont de (3.32), pentru q = m = 1, p = 2, se poate defini spațiul funcțiilor Lipschitz cu condiții Dirichlet omogene pe $\partial\Omega$ ([110]-Lema 3.40):

$$H_0^1(\Omega) = \{ v \in H^1(\Omega) \, | \, \operatorname{tr}_{\partial\Omega}(v) = 0 \}.$$
(3.80)

In acest moment, dispunem de operatorii diferențiali implicați în formulările matematice slabe ale problemelor fizice, precum și de cadrul general al spațiilor Sobolev compatibile cu acești operatori atât în domeniul Ω , cât și pe frotiera sa. Dacă spațiile Sobolev fracționare sunt potrivite pentru reprezentarea funcțiilor pe frontiera $\partial\Omega$, respectiv pentru termenii de tip $\int_{\partial\Omega}$ din relațiile (3.57) - (3.59), este necesară particularizarea spațiilor Sobolev și pentru termenii integrali aplicați pe Ω pentru fiecare tip de operator din aceleași relații. In acest mod, se poate stabili cadrul funcțional necesar abordărilor numerice în mod consistent și coerent cu modul de comportare al tuturor funcțiilor implicate în formulările matematice definite pe Ω , respectiv $\partial\Omega$. Tot în baza acestui cadru funcțional, devine posibilă determinarea unor criterii de evaluare a preciziei de aproximare a modelului real, cu ajutorul modelelor numerice aproximative.

3.2.5 Spațiul H1

Spațiul vectorial Sobolev $H^1 = W^{1,2}$, rezultă în mod natural din relația (3.35) discutată în cap. 3.2.2.3, motiv pentru care se mai numește și spațiul funcțiilor grad-conforme. Spațiul Hilbert H^1 conține funcțiile L^2 cu gradient generalizat de pătrat integrabil. Funcțiile sunt scalare, definite pe un domeniu deschis $\Omega \subset \mathbb{R}^d$, cu $d \in \{2,3\}$ și mărginit de frontiera Lipschitz $\partial\Omega$:

$$H^{1}(\Omega) := \{ \varphi \in L^{2}(\Omega) \mid \nabla \varphi \in [L^{2}(\Omega)]^{d} \}$$

$$(3.81)$$

înzestrat cu produsul scalar:

$$(\varphi,\psi)_1 = \int_{\Omega} \left(\nabla \varphi \cdot \nabla \psi + \varphi \psi \right) dx \quad \varphi, \psi \in H^1(\Omega)$$
(3.82)

ce induce norma:

$$\|\varphi\|_{H^1} \stackrel{\text{not}}{=} \|\varphi\|_1 = \left[\int_{\Omega} \left[|\nabla\varphi|^2 + |\varphi|^2\right] dx\right]^{1/2} = \left[\|\nabla\varphi\|_0^2 + \|\varphi\|_0^2\right]^{1/2}, \qquad (3.83)$$

respectiv seminorma:

$$|\varphi|_{H^1} \stackrel{\text{not}}{=} |\varphi|_1 = \left[\int_{\Omega} |\nabla \varphi|^2 \, dx\right]^{1/2} = \|\nabla \varphi\|_0. \tag{3.84}$$

Pentru spațiile H^1 , proprietatea (3.30) se scrie astfel:

$$H^{1}(\Omega) = \overline{C^{\infty}(\overline{\Omega})}^{\|\cdot\|}$$
(3.85)

Relația (3.85) garantează că clasa funcțiilor din H^1 este echivalentă cu clasa soluțiilor formulărilor tari ale sistemelor de ecuații cu derivate parțiale, din care derivă formulări slabe reprezentate prin sisteme de ecuații cu derivate parțiale de ordin I. Spațiul condițiilor de frontieră Dirichlet asociate acestui tip de ecuații rezultă din Teorema (3.2.2) pentru q = 1 și este generat cu ajutorul operatorului de urmă tr_{$\partial\Omega$}:

$$\operatorname{tr}_{\partial\Omega} : H^1(\Omega) \to H^{1/2}(\partial\Omega), \text{ unde}$$
 (3.86)

$$H^{1/2}(\partial\Omega) := \{ g \in L^2(\partial\Omega) : \exists \varphi \in H^1(\Omega) \text{ pentru care } \operatorname{tr}_{\partial\Omega} \varphi = \varphi_{|_{\partial\Omega}} = g \} \quad (3.87)$$

Proprietatea de densitate (3.85) se extinde și asupra funcțiilor din $H^{1/2}$:

$$H^{1/2}(\partial\Omega) = \overline{C^{\infty}(\partial\Omega)}^{\|\cdot\|_{1/2}},$$
(3.88)

iar, conform Teoremei (3.2.2) rezultă și mărginirea operatorului de urmă din (3.86):

$$\exists C > 0, \ a.\hat{i}. \, \|\mathrm{tr}_{\partial\Omega}(v)\|_{1/2} \le C \, \|v\|_1, \forall v \in H^1(\Omega)$$
(3.89)

Norma în spațiul $H^{1/2}$ este definită conform relației (3.23) ca:

$$\|v\|_{1/2} = \left[\|v\|_0^2 + \int_{\partial\Omega} \int_{\partial\Omega} \frac{|v(x) - v(y)|^2}{|x - y|^d}\right]^{1/2},$$
(3.90)

unde *d* reprezintă dimensiunea spațiului în care este inclus domeniul Ω (2 sau 3 în cazul nostru). De exemplu, pentru o condiție Dirichlet $\alpha \in H^{1/2}(\partial\Omega)$ se poate scrie că:

 $\exists v \in H(\Omega) \text{ si } C > 0, \text{ a. î. } \operatorname{tr}_{\partial\Omega}(v) = \alpha \text{ si } \|\alpha\|_{1/2} \le C \|v\|_1$ (3.91)

Pentru rezolvarea ecuațiilor cu derivate parțiale de ordin unu, pentru care se dau condițiile pe frontiera $\partial\Omega$, este necesară și definirea spațiului funcțiilor de test, care se anulează pe $\partial\Omega$, conform proprietății (3.32):

$$H_0^1(\Omega) = \left\{ v \in H^1(\Omega) \, | \, \operatorname{tr}_{\partial\Omega}(v) = \frac{\partial v}{\partial n} |_{\partial\Omega} = 0 \right\}.$$
(3.92)

Pentru cazul particular al condițiilor de frontieră omogene Neumann pure $(\frac{\partial v}{\partial n} = 0$ pe tot $\partial \Omega$), pentru asigurarea unicității soluției trebuie adăugată și o condiție de etalonare (măsură nulă pe Ω), care presupune folosirea funcțiilor de test din *spațiul* cât de mai jos (v. (3.200) și discuția - problemei 3.3.3):

$$H^{1}(\Omega)/\mathbb{R} = \left\{ v \in H^{1}(\Omega) \mid \int_{\partial \Omega} v \, dx = 0 \right\}.$$
 (3.93)

3.2.6 Spațiul H(rot)

Spațiul $H(\mathbf{rot}, \Omega)$ este format din distribuții ce aparțin spațiului $[\mathcal{D}(\Omega)']^d$, $(d \in \{1, 2, 3\}$ (v. cap. 3.2.1) care admit un rotor de pătrat integrabil - de unde și denumirea de funcții *rot-conforme*. Aceste distribuții sunt funcții de clasă $[L^2(\Omega)]^d$, la care se adaugă câteva proprietăți de densitate specifice, care permit definirea spațiilor vectoriale ale soluțiilor problemelor cu formulări slabe bazate pe operatorul *rotor* ($\nabla \times$). În consecință, spațiul vectorial al distribuțiilor rot-conforme definite pe domeniul Ω , face parte din clasa de echivalență Sobolev pentru m = 1, p = 2 fiind deci un spațiu Hilbert. În lucrarea de față vom presupune că domeniul Ω este inclus în \mathbb{R}^N , cu N = 2, sau N = 3 și că este un domeniu Lipschitz, beneficiind de proprietățile specificate în cap. 3.2.2.2. Pentru $\Omega \subset \mathbb{R}^3$:

$$H(\mathbf{rot},\Omega) := \{ \vec{u} \in [L^2(\Omega)]^3 \, | \, \nabla \times \vec{u} \in [L^2(\Omega)]^3 \},$$
(3.94)

iar operatorul $\nabla \times$ este de forma (3.48). Pentru N = 2:

• operatorul rotor va primi notația $\overrightarrow{\nabla}_2 \times$ dacă este aplicat distribuțiilor scalare din $\mathcal{D}(\Omega)'$, expresia sa fiind dată de (3.69):

$$H(\mathbf{Rot},\Omega) := \{\varphi \in L^2(\Omega) \mid \overrightarrow{\nabla}_2 \times \varphi \in [L^2(\Omega)]^2\},\tag{3.95}$$

Spațiul (3.95) are proprietăți similare cu spațiul $H^1(\Omega)$ definit în (3.81).

• operatorul rotor va primi notația $\nabla_2 \times$ dacă este aplicat distribuțiilor vectoriale din $[\mathcal{D}(\Omega)']^2$, iar atunci expresia sa va fi dată de (3.72).

$$H(\mathbf{rot},\Omega) := \{ \vec{u} \in [L^2(\Omega)]^2 \mid \nabla_2 \times \vec{u} \in L^2(\Omega) \},$$
(3.96)

Spațiile (3.94), (3.96) sunt înzestrate cu produsul scalar:

$$(\vec{u}, \vec{v})_{\rm rot} = (\nabla \times \vec{u}, \nabla \times \vec{v})_0 + (\vec{u}, \vec{v})_0, \quad \vec{u}, \vec{v} \in H(\mathbf{rot}, \Omega), \tag{3.97}$$

ce induce norma:

$$\|\vec{u}\|_{\rm rot} = (\|\nabla \times \vec{u}\|_0^2 + \|\vec{u}\|_0^2)^{1/2}, \tag{3.98}$$

respectiv seminorma:

$$\|\vec{u}\|_{\text{rot}} = \|\nabla \times \vec{u}\|_0. \tag{3.99}$$

Notăm cu $H_0(\mathbf{rot}, \Omega)$ spațiul distribuțiilor rot-conforme pentru care integrala pe frontiera $\partial\Omega$ din formula Green (3.58) se anulează. În virtutea relației de densitate (3.31), putem afirma, că:

$$H_0(\mathbf{rot},\Omega) = \overline{[\mathcal{D}(\Omega)]^N}^{\|\cdot\|_{\mathbf{rot}}}, \text{ cu } N \in \{2,3\}.$$
(3.100)

Pornind de la această observație, în [72]-(Lema2.4.) este evidențiat un alt mod de definire a spațiului (3.100), prin demonstrarea faptului că dacă, și numai dacă, o funcție $\vec{u} \in H(\mathbf{rot}, \Omega)$, satisface egalitatea:

$$\int_{\Omega} \left(\nabla \times \vec{u} \right) \cdot \vec{v} \, dx - \int_{\Omega} \vec{u} \cdot \left(\nabla \times \vec{v} \right) \, dx = 0, \quad \forall \vec{v} \in [\mathcal{C}^{\infty}(\Omega)]^3, \tag{3.101}$$

atunci $\vec{u} \in H_0(\mathbf{rot}, \Omega)$. Lema (3.101) este o variantă a formulei Green (3.58) pentru spațiul funcțiilor rot-conforme. După un alt set de deducții destul de laborioase d.p.d.v. matematic, în aceeași lucrare se demonstrează o proprietate de densitate importantă pentru spațiul $H(\mathbf{rot}, \Omega)$ și anume:

$$H(\mathbf{rot},\Omega) = \overline{[\mathcal{D}(\overline{\Omega})]^N}^{\|\cdot\|_{\mathbf{rot}}}, \text{ cu } N \in \{2,3\}.$$
(3.102)

Relația (3.102) garantează că clasa de distribuții din $H(\mathbf{rot}, \Omega)$ este echivalentă cu clasa soluțiilor formulărilor tari ale problemelor fizice descrise cu ajutorul unui potențial vector, probleme din care derivă formulări slabe reprezentate de expresii liniare cu operatori de tip (3.48), (3.69), (3.72).

Pentru formulările reprezentate cu ajutorul funcțiilor din $H(\mathbf{rot}, \Omega)$, pe frontierele domeniilor este de interes comportarea componentelor tangențiale ale câmpurilor vectoriale (v. discuția din cap.4.1.5). Pentru caracterizarea spațiul acestor condiții de frontieră trebuiesc luate în considerare, separat, cazurile domeniilor $\Omega \subset \mathbb{R}^N$ bidimensionale (N = 2), respectiv în trei dimensiuni (N = 3). Pentru ambele situații, putem defini operatori de urmă (v. teorema (3.2.2)), care aplicați distribuțiilor vectoriale $\vec{v} \in [\mathcal{D}(\Omega)]^N$, dau componentele tangențiale ale urmelor acestora pe frontiera Lipschitz $\partial\Omega$:

$$\operatorname{tr}_{\tau}(\vec{v}) = \vec{v}_{|\partial\Omega} \cdot \vec{\tau} \text{ pentru } \Omega \subset \mathbb{R}^2; \tag{3.103}$$

$$\operatorname{tr}_{\tau}(\vec{v}) = \vec{v}_{|\partial\Omega} \times \vec{n} \text{ pentru } \Omega \subset \mathbb{R}^3; \tag{3.104}$$

$$\operatorname{tr}_{T}(\vec{v}) = (\vec{v}_{|\partial\Omega} \times \vec{n}) \times \vec{n} \text{ pentru } \Omega \subset \mathbb{R}^{3}; \qquad (3.105)$$

unde $\vec{\tau}$ este versorul tangentei, respectiv \vec{n} este versorul normalei în orice punct conținut de frontiera $\partial \Omega$ (v. fig. 3.4).



Figura 3.4: Urmele tangențiale \vec{v}_{τ} pentru un domeniu 2D ($\Omega \subset \mathbb{R}^2$) -(a), respectiv \vec{v}_{τ} și \vec{v}_T pentru un domeniu 3D ($\Omega \subset \mathbb{R}^3$)-(b). Definiția naturală, a componentei tangențiale la frontiera $\partial\Omega$, a vectorului \vec{v} este dată de relația $\vec{v}_T = \vec{v} - (\vec{v} \cdot \vec{n})\vec{n}$. Ca atare, pentru domeniile 2D ea este \vec{v}_{τ} dată de (3.103); pentru domeniile 3D această componentă este \vec{v}_T , dată de (3.105), cu toate că și vectorul $\vec{v} \times \vec{n}$ este tangent la $\partial\Omega$ și are același modul ca și \vec{v}_T .

In virtutea proprietății de densitate (3.102), operatorii (3.103) și (3.104) pot fi extinși, și peste spațiul $H(\mathbf{rot}, \Omega)$, astfel încât este posibilă scrierea formulei Green (3.58) pentru $\forall \vec{u} \in H(\mathbf{rot}, \Omega)$ și $\forall \vec{v} \in [H^1(\Omega)]^3$ ([114]-Teorema3.29):

pentru
$$\Omega \subset \mathbb{R}^3$$
, $\operatorname{tr}_{\tau} : H(\operatorname{\mathbf{rot}}, \Omega) \to [H^{-1/2}(\partial\Omega)]^3$,

$$\int_{\Omega} (\nabla \times \vec{u}) \cdot \vec{v} \, d\mathcal{V} = \int_{\Omega} \vec{u} \cdot (\nabla \times \vec{v}) \, d\mathcal{V} - \int_{\partial\Omega} \vec{v} \cdot \operatorname{tr}_{\tau}(\vec{u}) \, d\mathcal{A},$$
(3.106)

respectiv, pentru $\forall \vec{u} \in H(\mathbf{rot}, \Omega)$ și $\forall \varphi \in H^1(\Omega)$:

pentru
$$\Omega \subset \mathbb{R}^2$$
, $\operatorname{tr}_{\tau} : H(\operatorname{\mathbf{rot}}, \Omega) \to H^{-1/2}(\partial\Omega),$

$$\int_{\Omega} (\nabla_2 \times \vec{u}) \varphi \, d\mathcal{V} = \int_{\Omega} \vec{u} \cdot (\overrightarrow{\nabla}_2 \times \varphi) \, d\mathcal{V} - \int_{\partial\Omega} \varphi \operatorname{tr}_{\tau}(\vec{u}) \, d\mathcal{A},$$
(3.107)

Datorită faptului că spațiul $[H^{-1/2}(\partial\Omega)]^d$, dual al spațiului $[H^{1/2}(\partial\Omega)]^d$, poate conține vectori, care să nu fie tangenți la frontiera $\partial\Omega$, operatorii (3.106) și (3.107) nu sunt surjectivi pe domeniile precizate, prin urmare noțiunea de componentă tangentă a urmei oricărei funcții $\vec{u} \in H(\mathbf{rot}, \Omega)$ nu este bine definită. Pentru a îndeplini acest deziderat, ar trebui identificat un spațiu definit pe $\partial\Omega$, în care fiecare element al său să fie urma tangențială a unei funcții din $H(\mathbf{rot}, \Omega)$:

$$X(\partial\Omega) = \left\{ \vec{w} \in [H^{-1/2}(\Omega)]^3 \mid \exists \vec{u} \in H(\mathbf{rot}, \Omega), \text{ a.î. } \operatorname{tr}_{\tau}(\vec{u}) = \vec{u} \times \vec{n} = \vec{w} \right\}$$

$$X(\partial\Omega) = \left\{ \psi \in H^{-1/2}(\Omega) \mid \exists \vec{u} \in H(\mathbf{rot}, \Omega), \text{ a.î. } \operatorname{tr}_{\tau}(\vec{u}) = \vec{u} \cdot \vec{\tau} = \psi \right\}$$
(3.108)

In continuare, va fi tratat cazul general $\Omega \subset \mathbb{R}^3$.

Intr-o primă etapă, definim spațiul funcțiilor vectoriale ce au componentă normală nulă pe frontiera $\partial \Omega$:

$$H_t^{-1/2}(\partial\Omega) = \left\{ \vec{\omega} \in [H^{-1/2}(\partial\Omega)]^3 \, | \, \vec{\omega} \cdot \vec{n} = 0, \text{ aproape peste tot pe } \partial\Omega \right\}$$
(3.109)

Ca urmare, putem stabili o primă caracteristică a spațului $X(\partial \Omega)$ pe care îl căutăm:

$$X(\partial\Omega) \subset H_t^{-1/2}(\partial\Omega) \tag{3.110}$$

In a doua etapă alegem pentru \vec{v} orice funcție vectorială din $H(\mathbf{rot}, \Omega)$ derivată din gradientul unui câmp scalar ψ :

$$\vec{v} = \nabla \psi, \quad \psi \in H^1(\Omega)$$
 (3.111)

In acest caz, ținând cont și de (3.64), putem scrie integrala de suprafață din membrul drept al (3.106) astfel:

$$\vec{v}_{|\partial\Omega} = \nabla_2 \psi = (\nabla \psi \times \vec{n}) \times \vec{n} = \operatorname{tr}_T(\vec{v}), \qquad (3.112)$$

iar relația Green (3.106) devine:

$$\int_{\Omega} \left(\nabla \times \vec{u} \right) \cdot \nabla \psi \, d\mathcal{V} = - \int_{\partial \Omega} \operatorname{tr}_{T}(\vec{v}) \cdot \operatorname{tr}_{\tau}(\vec{u}) \, d\mathcal{A}.$$
(3.113)

Acum, intergând prin părți membrul stâng al egalității (3.113), similar cu relația Green (3.57) (în care înlocuim formal φ cu ψ și \vec{u} cu $\nabla \times \vec{u}$), obținem:

$$\int_{\partial\Omega} \psi(\nabla \times \vec{u}) \cdot \vec{n} \, d\mathcal{A} = -\int_{\partial\Omega} \operatorname{tr}_{\tau}(\vec{u}) \cdot \nabla_2 \varphi \, d\mathcal{A}.$$
(3.114)

Comparând cu (3.67), observăm că relația (3.114) este de fapt expresia divergenței superficiale generalizate a urmei tangențiale $tr_{\tau}(\vec{u})$:

$$\nabla_2 \operatorname{tr}_{\tau}(\vec{u}) = (\nabla \times \vec{u}_{|\partial\Omega}) \cdot \vec{n}.$$
(3.115)

Cum relația (3.115) este valabilă pentru orice $\vec{u} \in [H^{-1/2}(\partial\Omega)]^3$, rezultă că divergența superficială a urmei tr_{τ}(\vec{u}) este bine definită pe frontiera $\partial\Omega$. Ca urmare, spațiul $X(\partial\Omega)$ căutat este:

$$X(\partial\Omega) \stackrel{\text{not}}{=} H^{-1/2}(div,\partial\Omega) = \left\{ \vec{u} \in H_t^{-1/2}(\partial\Omega) \,|\, \nabla_2(\vec{u}) \in H^{-1/2}(\partial\Omega) \right\}.$$
(3.116)

Prin urmare, pentru orice funcție $g \in H^{-1/2}(div, \partial\Omega)$, există o funcție $\vec{w} \in H(\mathbf{rot}, \Omega)$, astfel încât $\operatorname{tr}_{\tau}(\vec{w}) = g$. Altfe spus, în problemele fizice caracterizate printr-un potențial vector, distribuția acestuia în interiorul domeniului de calcul este bine stabilită de către distribuția componentelor sale tangențiale pe frontiera domeniului, atâta timp cât aceste componente sunt funcții din $H^{-1/2}(div, \partial\Omega)$. Pentru rezolvarea acestui tip de problemă, este necesară și cunoașterea spațiului funcțiilor de test din $H(\mathbf{rot}, \Omega)$, respectiv a funcțiilor rot-conforme ce au componente tangențiale omogene pe frontiera $\partial\Omega$. Acest spațiu, definit inițial prin relația de densitate (3.100), respectiv condiția (3.101), este reprezentat sintetic prin definiția de mai jos:

$$H_0(\mathbf{rot},\Omega) = Ker(\mathrm{tr}_T) = \left\{ \vec{u} \in H(\mathbf{rot},\Omega) \,|\, (\vec{u}_{|\partial\Omega} \times \vec{n}) \times \vec{n} = 0 \right\}. \tag{3.117}$$

In cazul unui domeniu compus dintr-o reuniune disjunctă de subdomenii, posibil cu caracteristici fizice diferite, proprietățile de frontieră (3.116), (3.117) ale urmelor funcțiilor rot-conforme, garantează conservarea componentelor lor tangențiale la trecerea prin suprafețele de separație dintre subdomeniile vecine. Ca urmare, spațiile (3.94) și (3.116) sunt potrivite pentru tratarea numerică a problemelor fizice descrise de funcții vectoriale exprimate local prin rotorul lor, iar global prin circulația lor pe varietăți geometrice de speța I. Un exemplu tipic în electromagnetism este reprezentat de intensitățile câmpurilor electric, respectiv magnetic.

3.2.7 Spațiul H(div)

Spațiul $H(div, \Omega)$ este compus din distribuții din spațiul $[\mathcal{D}(\Omega)']^d$, $(d \in \{2, 3\}$ (v. cap. 3.2.1) a căror divergență este de pătrat integrabil - de unde și denumirea de funcții div-conforme. Aceste distribuții sunt funcții de clasă $[L^2(\Omega)]^d$, la care se adaugă câteva proprietăți de densitate specifice, care permit definirea spațiilor vectoriale ale soluțiilor problemelor cu formulări slabe bazate pe operatorul divergență ($\nabla \cdot$). In consecință, spațiul vectorial al distribuțiilor div-conforme definite pe domeniul Ω , face parte din clasa de echivalență Sobolev pentru m = 1, p = 2, fiind deci un spațiu Hilbert. Domeniul Ω este inclus în \mathbb{R}^d este presupus a fi un domeniu Lipschitz, conform cap. 3.2.2.2:

$$H(div, \Omega) := \{ \vec{u} \in [L^2(\Omega)]^d \mid \nabla \cdot \vec{u} \in L^2(\Omega) \}$$
(3.118)

Spațiul (3.118) este înzestrat cu produsul scalar:

$$(\vec{u}, \vec{v})_{\text{div}} = (\nabla \cdot \vec{u}, \nabla \cdot \vec{v})_0 + (\vec{u}, \vec{v})_0, \quad \vec{u}, \vec{v} \in H(div, \Omega),$$
(3.119)

produs ce, induce norma:

$$\|\vec{u}\|_{\rm div} = (\|\nabla \cdot \vec{u}\|_0^2 + \|\vec{u}\|_0^2)^{1/2}$$
(3.120)

precum și seminorma:

$$\|\vec{u}\|_{\rm div} = \|\nabla \cdot \vec{u}\|_0. \tag{3.121}$$

La fel ca și în cazul spațiilor funcțiilor rot-conforme, pentru spațiile funcțiilor divconforme este demonstrată proprietatea de densitate ([72]-Teorema 2.4):

$$H(div,\Omega) = \overline{[\mathcal{D}(\overline{\Omega})]^d}^{\|\cdot\|_{\operatorname{div}}}.$$
(3.122)

Proprietatea (3.122) garantează că clasa de distribuții din $H(div, \Omega)$ este echivalentă cu clasa soluțiilor formulărilor matematice tari ale problemelor fizice, din care derivă formulări slabe bazate pe expresii liniare cu operatori de tip (3.49), (3.67).

Pentru formulările reprezentate cu ajutorul funcțiilor din $H(div, \Omega)$, pe frontierele domeniilor este de interes comportarea componentelor normale ale câmpurilor vectoriale (v. discuția din cap.4.1.6). Putem defini un operator de urmă (v. teorema (3.2.2)), care aplicat unei distribuții vectoriale $\vec{v} \in [\mathcal{D}(\Omega)]^d$, dă componentele normale ale urmei acestei distribuții (urma normală) pe frontiera Lipschitz $\partial\Omega$:

$$\operatorname{tr}_{n}(\vec{v}) = \vec{v}_{\mid \partial \Omega} \cdot \vec{n}; \qquad (3.123)$$

unde \vec{n} este versorul normalei în orice punct conținut de frontiera $\partial \Omega$.

Plecând de la proprietatea de densitate (3.122), operatorul (3.123) poate fi extins și peste spațiul $H(div, \Omega)$, astfel încât este posibilă scrierea formulei Green (3.59) pentru $\forall \vec{u} \in H(div, \Omega)$ și $\forall \varphi \in H^1(\Omega)$ ([114]-Teorema3.24):

$$\operatorname{tr}_{n} : H(\operatorname{div}, \Omega) \to H^{-1/2}(\partial\Omega),$$

$$\int_{\Omega} \varphi \left(\nabla \cdot \vec{v}\right)' d\mathcal{V} = -\int_{\Omega} \vec{v} \cdot \nabla \varphi \, d\mathcal{V} + \int_{\partial\Omega} \varphi \operatorname{tr}_{n}(\vec{v}) \, d\mathcal{A}$$
(3.124)

Apoi, bazat pe existența și unicitatea soluției ecuației eliptice Neumann (v. (3.3.3) din cap.3.3.5), în aceeași teoremă de mai sus, este demostrată și surjectivitatea operatorului tr_n, astfel încât spațiul $H^{-1/2}(\partial\Omega)$ rezultă a fis codomeniul acestui operator (v. și [72]-Corolar 2.8) adică:

$$H^{-1/2}(\partial\Omega) = \{\varphi \mid \exists \vec{v} \in H(div, \Omega), \text{ a.i. } \operatorname{tr}_n(\vec{v}) = \varphi\}.$$
(3.125)

Notăm cu $H_0(div, \Omega)$ spațiul distribuțiilor div-conforme pentru care integrala pe frontiera $\partial\Omega$ din formula Green (3.59) se anulează. În virtutea relației de densitate (3.31), putem afirma, că:

$$H_0(div,\Omega) = \overline{[\mathcal{D}(\Omega)]^N}^{\|\cdot\|_{div}}, \text{ cu } N \in \{2,3\}.$$
(3.126)

Pornind de la această proprietate de densitate, se poate arăta mai departe ([72]-Teorema 2.6) că spațiul distribuțiilor $[\mathcal{D}(\Omega)]^N$ este dens în spațiul nucleu al operatorului tr_n, astfel încât spațiul $H_0(div, \Omega)$ poate fi caracterizat astfel:

$$H_0(div,\Omega) = Ker(\operatorname{tr}_n) = \left\{ \vec{v} \in H(div,\Omega) \,|\, \vec{v}_{|\partial\Omega} \cdot \vec{n} = 0 \right\}.$$
(3.127)

In cazul unui domeniu compus dintr-o reuniune disjunctă de subdomenii, posibil cu caracteristici fizice diferite, proprietatea (3.127), a urmelor funcțiilor div-conforme, garantează conservarea componentelor lor normale la trecerea prin suprafețele de separație dintre subdomeniile vecine. Ca urmare, spațiile (3.118) sunt potrivite pentru tratarea numerică a problemelor fizice descrise de funcții vectoriale exprimate local prin divergența lor și global prin fluxul lor pe varietăți geometrice de speța a II-a. Un exemplu tipic în electromagnetism este reprezentat de inducțiile electrică, respectiv magnetică.

3.2.8 Complexul De Rham

Spațiile funcționale (3.44), (3.81), (3.94), (3.118) definite pe un domeniu Ω simplu conex formează o *secvență exactă* ([145], [42]): spațiile vectoriale pot fi așezate

într-o secvență în care imaginea unui operator diferențial definit pe unul din spații formează nucleul spațiului imediat următor din acea secvență:

$$\mathbb{R} \to H^1(\Omega) \xrightarrow{\nabla} H(\mathbf{rot}, \Omega) \xrightarrow{\nabla \times} H(div, \Omega) \xrightarrow{\nabla \cdot} L^2(\Omega) \to \{0\}.$$
(3.128)

(3.128) este cunoscută ca secvența De Rham. Se pot observa următoarele relații:

1.

$$ker(\nabla H^{1}(\Omega)) = \left\{ \varphi \in H^{1}(\Omega) \, | \, \nabla \varphi = 0 \right\} =$$

$$\left\{ \varphi \in H^{1}(\Omega) \, | \, \varphi = \operatorname{ct} \in \mathbb{R} \right\} = Im\left(\mathbb{R} \to H^{1}(\Omega)\right).$$
(3.129)

2.

$$ker(\nabla \times H(\mathbf{rot}, \Omega)) = \{ \vec{u} \in H(\mathbf{rot}, \Omega) \mid \nabla \times \vec{u} = 0 \} =$$

$$\{ \vec{u} \in H(\mathbf{rot}, \Omega) \mid \vec{u} = \nabla \varphi, \ \varphi \in H^1(\Omega) \} = Im\left(\nabla H^1(\Omega)\right).$$
(3.130)

Dacă domeniul Ω este Lipschitz și simplu conex cu frontieră conexă, atunci se demonstrează ([42], [114]) că se poate defini un operator invers g^{-1} : $ker(\nabla \times H(\mathbf{rot}, \Omega)) \to H^1(\Omega)$ astfel încât:

dacă
$$\vec{u} \in ker(\nabla \times H(\mathbf{rot}, \Omega))$$
, atunci (3.131)

$$\exists \varphi \in H^1(\Omega), \text{ a.i. } g^{-1}\vec{u} = \varphi \Rightarrow \vec{u} = \nabla \varphi.$$
(3.132)

In cazul în care domeniul Ω nu este simplu conex, există posibilitatea ca existența condiției (3.131) să nu implice neapărat relația (3.132), caz în care (3.130) să se scrie ca o relație de incluziune ([42]-cap.5, [78]):

$$Im\left(\nabla H^{1}(\Omega)\right) \subset ker(\nabla \times H(\mathbf{rot}, \Omega)).$$
(3.133)

3.

$$ker(\nabla \cdot H(\operatorname{\mathbf{div}},\Omega)) = \{ \vec{v} \in H(\operatorname{\mathbf{div}},\Omega) \mid \nabla \cdot \vec{v} = 0 \} = \\ \{ \vec{v} \in H(\operatorname{\mathbf{div}},\Omega) \mid \vec{v} = \nabla \times \vec{u}, \ \vec{u} \in H(\operatorname{\mathbf{rot}},\Omega) \} = Im\left(\nabla \times H(\operatorname{\mathbf{rot}},\Omega)\right).$$
(3.134)

Dacă domeniul Ω este Lipschitz și simplu conex cu frontieră conexă, atunci se demonstrează ([42], [114]) că se poate defini un operator invers r^{-1} : $ker(\nabla \cdot H(\operatorname{\mathbf{div}}, \Omega)) \to H(\operatorname{\mathbf{rot}}, \Omega)$ astfel încât:

dacă
$$\vec{v} \in ker(\nabla \cdot H(\mathbf{div}, \Omega))$$
, atunci (3.135)

$$\exists \vec{u} \in H(\mathbf{rot}, \Omega), \text{ a.i. } r^{-1}\vec{v} = \vec{u} \Rightarrow \vec{v} = \nabla \times \vec{u}.$$
(3.136)

In cazul în care domeniul Ω nu este simplu conex, există posibilitatea ca existența condiției (3.135) să nu implice neapărat relația (3.136), caz în care (3.134) să se scrie ca o relație de incluziune ([42]-cap.5, [78]):

$$Im\left(\nabla \times H(\mathbf{rot}, \Omega)\right) \subset ker(\nabla \cdot H(\mathbf{div}, \Omega)).$$
(3.137)

4. În final:

$$ker(L^{2}(\Omega) \to \{0\}) = L^{2}(\Omega) = Im(\nabla \cdot H(\operatorname{div}, \Omega))$$
(3.138)

Secvența exactă (3.128) poate fi scrisă și pentru funcțiile ce satisfac condiții omogene pe întreaga frontieră $\partial \Omega$:

$$\mathbb{R} \to H_0^1(\Omega) \xrightarrow{\nabla} H_0(\operatorname{\mathbf{rot}}, \Omega) \xrightarrow{\nabla \times} H_0(\operatorname{div}, \Omega) \xrightarrow{\nabla} L_0^2(\Omega) \to \{0\},$$
(3.139)

unde spațiul $L_0^2 = \{f \in L_2(\Omega) | \int_{\Omega} f \, dx = 0\}$ reprezintă spațiul funcțiilor integrabile a căror medie este nulă pe domeniul Ω . În spațiul real 2D secvența (3.128) poate avea următoarele două variante:

$$\mathbb{R} \to H^1(\Omega) \xrightarrow{\nabla_2} H(\mathbf{rot}, \Omega) \xrightarrow{\nabla_2 \times} L^2(\Omega) \to \{0\}, \qquad (3.140)$$

$$\mathbb{R} \to H^1(\Omega) \xrightarrow{\nabla_2 \times} H(div, \Omega) \xrightarrow{\nabla_2 \cdot} L^2(\Omega) \to \{0\}.$$
(3.141)

Importanța secvențelor exacte de mai sus rezidă în faptul că ele reflectă izomorfismul existent între spațiile legate prin săgeți, precum și metoda de construcție a unei clase de funcții din alta. In ([42])-Cap.5, se demonstrează că secvența De Rham este valabilă atât pentru operatorii grad, div, rot, în formă "clasică" (sens tare), dar și pentru forma slabă a acestora, iar după cum se va vedea în capitolul următor, secvența (3.128) este utilă și pentru descrierea operatorilor de interpolare prin care se generează spațiile vectoriale corespunzătoare domeniului Ω discretizat în elemente spațiale finite. Tot cu ajutorul secvenței de Rham se poate pune în evidență ([114]-Teorema 3.45), într-o manieră foarte elegantă matematic, teorema de descompunere Helmholtz, fundamentală în analiza vectorială, care garantează posibilitatea descompunerii unice, a oricărui câmp vectorial $\vec{u} \in [L^2(\Omega)]^3$, întrun câmp irotațional, reprezentat prin gradientul unui potențial scalar și un câmp solenoidal, reprezentat prin rotorul unui potențial vector:

$$\vec{u} = \nabla \varphi + \nabla \times \vec{A},\tag{3.142}$$

unicitatea descompunerii fiind asigurată prin impunerea unor condiții de etalonare a potențialelor pe frontiera domeniului Ω .

3.2.9 Existența și unicitatea soluțiilor formulărilor slabe

Existența și unicitatea soluției unei probleme variaționale implică definirea unui set de proprietăți care trebuiesc satisfăcute de către formulările abstracte ale cadrului funcțional asociat problemei respective. Aceste formulări au la bază noțiunea de funcțională biliniară definită astfel ([38], [99], [53], [159]):

$$\mathcal{A}(\cdot, \cdot) : U \times V \to \mathbb{R}$$

$$\mathcal{A}(u, v) = \mathcal{F}(v) \text{ cu } u \in U, \ v \in V,$$

iar U, V sunt spații Hilbert cu normele $\|\cdot\|_U, \|\cdot\|_V$
(3.143)

și care satisface relațiile definitorii ale unei aplicații liniare:

$$\mathcal{A}(\alpha \cdot u_1 + \beta \cdot u_2, v) = \alpha \cdot \mathcal{A}(u_1, v) + \beta \cdot \mathcal{A}(u_2, v)$$

$$\mathcal{A}(u, \alpha \cdot v_1 + \beta \cdot v_2) = \alpha \cdot \mathcal{A}(u, v_1) + \beta \cdot \mathcal{A}(u, v_2)$$
pentru $\forall \alpha, \beta \in \mathbb{R} \text{ si } u_1, u_2 \in U, v_1, v_2 \in V$
(3.144)

In cazul în care $\alpha, \beta \in \mathbb{C}$ funcționala (3.143) se numește de tip sesquiliniar, iar relațiile (3.144) se scriu ținând cont de proprietățile produsului scalar între mărimi complexe:

$$\mathcal{A}(\alpha \cdot u_1 + \beta \cdot u_2, v) = \alpha \cdot \mathcal{A}(u_1, v) + \beta \cdot \mathcal{A}(u_2, v)$$

$$\mathcal{A}(u, \alpha \cdot v_1 + \beta \cdot v_2) = \overline{\alpha} \cdot \mathcal{A}(u, v_1) + \overline{\beta} \cdot \mathcal{A}(u, v_2)$$
pentru $\forall \alpha, \beta \in \mathbb{C} \text{ si } u_1, u_2 \in U, v_1, v_2 \in V$

$$(3.145)$$

unde $\overline{\alpha}$ este conjugatul numărului complex α . Totodată se observă că funcția \mathcal{F} face parte din mulțimea operatorilor definiți pe V, deci $\mathcal{F} \in V'$, unde V' reprezintă dualul spațiului V.

Demonstrarea existenței soluției unice a formulării slabe a unei probleme este strâns legată de punerea în evidență, pentru (3.143), a următoarelor proprietăți:

• *Mărginire*:

 $\exists \text{ constanta } M > 0 \text{ a.i. } |\mathcal{A}(u, v)| \le M \|u\|_U \|v\|_V \quad \forall u \in U, \forall v \in V, \quad (3.146)$

• Coercivitate:

$$\exists \text{ constanta } \gamma > 0 \text{ a.i. } \mathcal{A}(u, u) \ge \gamma \|u\|_U^2, \quad \forall u \in U.$$
(3.147)

Teorema Lax-Milgram ([99], [159], [73] [84]) stabilește criteriile de garantare a existenței soluției unice pentru următoarea problemă:

Se dă
$$\mathcal{F} \in V'$$

Se cere să se determine $u \in V$ a.î.: (3.148)
 $\mathcal{A}(u, v) = \mathcal{F}(v) \quad \forall v \in V$

Teorema 3.2.3 (Lax - Milgram). Fie $a : V \times V \to \mathbb{C}$ o aplicație sesquiliniară (v. (3.145)), mărginită (v. 3.146) și coercivă (v. 3.147). Atunci, pentru orice funcțională liniară continuă $\mathcal{F} \in V'$, există o soluție unică $u \in V$ care satisface egalitatea:

$$\mathcal{A}(u, v) = \mathcal{F}(v) \quad \forall v \in V \tag{3.149}$$

In plus, soulția u satisface următoarea restricție (condiție de stabilitate):

$$\|u\|_{V} \le \frac{M}{\gamma} \|f\|_{V^{*}} \tag{3.150}$$

unde M și γ sunt constantele din 3.146 și 3.147

O aplicație particulară ce se bazează pe teorema 3.2.3 este cazul biliniarelor simetrice \mathcal{A} , pentru care se poate enunța următoarea teoremă ([73] [84]):

Teorema 3.2.4. Fie $a: V \times V \to \mathbb{C}$ o aplicație biliniară și care îndeplinește condițiile teoremei 3.2.3. In plus $\mathcal{A}(\cdot, \cdot)$ este simetrică, adică:

$$\mathcal{A}(u,v) = \mathcal{A}(v,u) \quad \forall u, v \in V$$

In aceste condiții $u \in V$ este o soluție a problemei de găsire a minimului funcționalei energetice:

$$W(v) := \frac{1}{2} \cdot \mathcal{A}(v, v) - \mathcal{F}(v) \quad \forall v \in V \ adic \breve{a}$$
(3.151)

$$W(u) = \min_{v \in V} W(v).$$
 (3.152)

dacă și numai dacă u este o soluție a ecuației (3.149).

Această teoremă reprezintă formularea variațională Ritz și poate fi aplicată direct ecuației Poisson cu condiții Dirichlet omogene.

3.3 Formulări matematice pentru o ecuație eliptică

In majoritatea cazurilor, clasa ecuațiilor eliptice este caracteristică modelelor matematice asociate regimurilor staționare ale fenomenelor fizice ([73], [143], [150], [76]). In cele ce urmează, voi pune în evidență formulările tare, slabă, respectiv numerică pentru o problemă generală ce se poate modela cu o astfel de ecuație Poisson generalizată, de tip div-grad. Ea este ecuația fundamentală a regimului electrostatic, satisfacută de potențialul electrostatic V, intensitatea câmpului electrostatic \vec{E} fiind gradientul cu semn schimbat, al potențialului V. Ecuații similare se întâlnesc în regimul magnetostatic și electrocinetic staționar.

3.3.1 Formulare matematică tare

Fie $\Omega \subset \mathbb{R}^d$ cu d = 2, 3, un domeniu deschis, mărginit de frontiera de tip Lipschitz $\partial \Omega$, cu $\overline{\Omega} = \Omega \cup \partial \Omega$ (fig. 3.5). Forma tare a problemei presupune rezolvarea unei ecuații eliptice de ordin doi de forma:

$$-\nabla \cdot (a \cdot \nabla u) + b \cdot u = f \ pe \ \Omega, \qquad (3.153)$$
$$u = g \ pe \ \partial \Omega_D,$$
$$a \ \frac{\partial u}{\partial n} + q \cdot u = r \ pe \ \partial \Omega_N,$$

și presupune găsirea soluției $u \in [C^2(\Omega)]^d$. Condițiile de frontieră sunt de tip



Figura 3.5: Domeniul Ω pe care se rezolvă ecuația eliptică (3.153) este de tip Lipschitz. Frontiera $\partial\Omega$ este o curbă pentru d = 2, respectiv suprafață pentru d = 3

Dirichlet pe porțiunea de forntieră $\partial \Omega_D$, respectiv Neumann (pentru q = 0), sau Robin (pentru $q \neq 0$) pe porțiunea de forntieră $\partial \Omega_N$. Totodată, porțiunile de frontieră îndeplinesc condițiile:

$$\partial \Omega_D \cap \partial \Omega_N = \emptyset$$

$$\partial \Omega_D \cup \partial \Omega_N = \partial \Omega.$$
(3.154)

a = a(x) este un tensor de ordin 0 sau 2, continuu și pozitiv definit pe Ω : $\exists a_0, a_1 \in \mathbb{R}$, cu $0 < a_0 \leq a_1$, astfel încât:

$$a_0|\alpha|^2 \le \alpha^T \cdot a(x) \cdot \alpha \le a_1|\alpha|^2, \, \forall x \in \Omega, \, \alpha \in \mathbb{R}^d,$$
(3.155)

iar b, f, q și r sunt funcții definite pe $\overline{\Omega}$ și satisfac următoarele condiții:

$$b = b(x) \ge 0 \text{ si } q = q(x) \ge 0, \forall x \in \Omega$$

$$(3.156)$$

$$b, g, q, r \in C^{0}(\Omega),$$

$$f \in L^2(\Omega). \tag{3.157}$$

3.3.2 Unicitatea soluției în formulare tare

Rezovarea ecuației (3.153) presupune găsirea unei funcții potențial, caracteristice fenomenului fizic modelat de aceasta [150]. Existența soluției va fi analizată în subcapitolul următor. Distribuția de potențial u se poate determina în mod unic dacă sunt cunoscute:

- în orice punct din domeniu sursele de câmp date de distribuții interne domeniului Ω (funcțiile b și f), sau de surse externe de câmp date prin cel puțin de una dintre condițiile de frontieră pe $\partial \Omega_D$ (funcția g) și/sau $\partial \Omega_N$ (funcțiile q și r);
- tensorul *a*, în orice punct din domeniu, reprezentând caracteristicile locale ale materialului;

Metoda de demonstrare a unicității soluției se bazează pe determinarea **funcționalei** energetice asociate relației diferențiale (3.153), cu ajutorul căreia se pune în evidență imposibilitatea de a avea două soluții diferite pentru aceeași ecuație [84]. Pentru aceasta, proiectăm egalitatea (3.153) pe setul de funcții soluție u, folosind produsul scalar în sens L^2 :

$$\int_{\Omega} -u \cdot (\nabla \cdot (a \cdot \nabla u)) \, dV + \int_{\Omega} u \cdot b \cdot u \, dV = \int_{\Omega} u \cdot f \, dV. \tag{3.158}$$

După integrarea prin părți a primului termen din membrul stâng și aplicarea teoremei divergenței, putem separa (tot în membrul stâng) expresia funcționalei energetice în raport cu funcția potențial u:

$$\int_{\Omega} \left(a \cdot (\nabla u)^2 + b \cdot u^2 \right) dV + \int_{\partial \Omega_N} q \cdot u^2 dA = \int_{\Omega} u \cdot f \, dV + \int_{\partial \Omega_D} u \cdot (a \, \nabla u) \cdot \mathbf{n} \, dA + \int_{\partial \Omega_N} r \cdot u \, dA \tag{3.159}$$

Acum presupunem existența a două soluții diferite pentru ecuația (3.153) și anume u', respectiv u'' și definim diferența lor ca fiind w = u' - u''. In virtutea liniarității operatorului diferențial ce construiește (3.153), se poate afirma ca w satisface ecuația de mai jos, cu condiții de frontieră omogene:

$$-\nabla \cdot (a \cdot \nabla w) + b \cdot w = 0 \ pe \ \Omega, \qquad (3.160)$$
$$w = 0 \ pe \ \partial \Omega_D,$$
$$a \ \frac{\partial w}{\partial n} + q \cdot w = 0 \ pe \ \partial \Omega_N.$$

Aplicând același tratament prin care am obțiut egalitatea (3.159), dar ținând cont de condițiile de frontieră omogene rezultă:

$$\int_{\Omega} \left(a \cdot (\nabla w)^2 + b \cdot w^2 \right) dV + \int_{\partial \Omega_N} a \, w^2 \cdot q \, dA = 0, \qquad (3.161)$$

ținând cont de (3.155) și (3.156) rezultă că:

$$a \cdot (\nabla w)^2 + b \cdot w^2 = 0 \text{ si}$$

$$q \cdot w^2 = 0,$$

$$(3.162)$$

adică w = u' - u'' = 0, deci cele două presupuse soluții diferite se confundă într-o unică soluție.

In cazul în care b(x) = 0 pe Ω și q(x) = 0 pe $\partial \Omega_N$ (condiții de frontieră pur Dirichlet), atunci relația (3.162) indică faptul că w este constantă pe tot domeniul $\overline{\Omega}$. Din condițiile de frontieră Dirichlet omogene rezultă că această constantă nu poate fi decât 0, deci și în acest caz soluția este unică.

Există o situație specială, și anume aceea în care b(x) = 0 și nu există decât condiții de frontieră Neumann, adică:

$$\partial\Omega_D = \emptyset \text{ si } q(x) = 0. \tag{3.163}$$

In acest caz din egalitatea (3.162) rezultă:

$$(\nabla w)^2 = 0, \tag{3.164}$$

garantând că soluția diferență w este constantă, dar, neavând condiții Dirichlet, această constantă rămâne nedeterminată. Cu alte cuvinte, în cazul b(x) = 0, cu condițiile de frontieră (3.163) (condiții pur Neumann) soluția ecuației (3.153) este determinabilă până la o constantă arbitrară. Soluția problemei Neumann devine unică în cazul aplicării unei condiții de etalonare - valoarea medie a potențialului este impusă nulă. Pentru garantarea existenței, soluției este necesară și suficientă, îndeplinirea unei condiții de compatibilitate între sursele de câmp f(x) și r(x), condiție ce rezultă prin impunerea unei valori particulare constantă pentru soluția u în expresia funcționalei energetice (3.159) ([108], [118]):

$$\int_{\Omega} f \, dV + \int_{\partial \Omega_N} r \, dA = 0. \tag{3.165}$$

Cu această condiție, ecuația (3.153), admite ca soluție o familie de funcții ce diferă între ele printr-o constantă aditivă. O discuție referitoare la această situație va fi efectuată în subcapitolul 3.3.5.

3.3.3 Formulare matematică slabă

Formularea matematică slabă a ecuației (3.153) se obține prin proiectarea ecuației diferențiale pe spațiul generat de o bază de funcții test $v \in H_0^1(\Omega)$, - definit în (3.92).

In sens slab - se rezolvă următoarea problemă:

Problema 3.3.1. Pentru ecuația:

$$(-\nabla(a\nabla u), v) + (bu, v) = (f, v)$$
(3.166)

cu condițiile de frontieră:

$$\begin{cases} u = g & pe \ \partial \Omega_D \\ a \frac{\partial u}{\partial n} + qu = r & pe \ \partial \Omega_N \end{cases}$$
(3.167)

și în condițiile (3.155) pentru a, respectiv:

$$\begin{cases} b, f \in [H^1(\Omega)]^* = H^{-1}(\Omega), \\ g \in H^{\frac{1}{2}}(\partial \Omega_D), \\ q, r \in H^{-\frac{1}{2}}(\partial \Omega_N), \end{cases}$$

se caută soluția:

$$u \in H_D^1(\Omega) = \{ x \in H^1(\Omega) | tr_{\partial \Omega_D}(x) = g \} \qquad \forall v \in H_0^1.$$
(3.168)

In (3.166) operatorul (\cdot, \cdot) reprezintă produsul scalar în sens L^2 . Expresia rezultată prin proiecție, se integrează apoi prin părți, obținându-se în final o egalitate de tip (3.143). Rezolvarea problemei (3.3.1) presupune determinarea soluției u, în condițiile (3.168), pentru ecuația de mai jos:

$$\mathcal{A}(u, v) = \mathcal{F}(v) \operatorname{cu}, \qquad (3.169)$$
$$\mathcal{A}: H^{1}(\Omega) \times H^{1}_{0}(\Omega) \to \mathbb{R} \text{ si } \mathcal{F}: H^{1}(\Omega) \to \mathbb{R},$$

ai cărei membri sunt următorii:

$$\mathcal{A}(u, v) = \int_{\Omega} (a \nabla u \cdot \nabla v + b \cdot u \cdot v) \, dV + \int_{\partial \Omega_N} q \cdot u \cdot v \, dA$$

$$\mathcal{F}(v) = \int_{\Omega} f \, v \, dV + \int_{\partial \Omega_N} r \, v \, dA.$$

Membrul drept al (3.169) poate fi exprimat cu ajutorul produsului de dualitate:

$$\langle F, v \rangle = \int_{\Omega} f v \, dV + \int_{\partial \Omega_N} r v \, dA = \langle f, v \rangle |_{\Omega} + \langle tr_{\partial \Omega_N}(u), v \rangle |_{\partial \Omega_N}, \quad (3.170)$$

de unde:

$$\|\mathcal{F}\|_{H^{-1},\Omega} \le C\left(\|f\|_{H_D^{-1},\Omega} + \|tr_{\partial\Omega_N}(u)\|_{H^{-1/2},\partial\Omega_N}\right).$$
(3.171)

Ecuația (3.166) se poate scrie condensat:

$$\mathcal{A}(u, v) = \langle F, v \rangle, \text{ cu}$$

$$u \in H_D^1(\Omega) \text{ si } \forall v \in H_0^1.$$
 (3.172)

3.3.4 Existența și unicitatea soluției în formulare slabă

Existența și unicitatea soluției pentru problema (3.3.1) sunt garantate de teorema Lax-Milgram (3.2.3) ce presupune îndeplinirea condițiilor (3.146), respectiv (3.147) pentru operatorul biliniar $\mathcal{A}(\cdot, \cdot)$.

1. Mărginirea $\mathcal{A}(\cdot, \cdot)$

$$|\mathcal{A}(u, v)| \le |\int_{\Omega} a \,\nabla u \cdot \nabla v \, dV| + |\int_{\Omega} b \cdot u \cdot v \, dV| + |\int_{\partial \Omega_N} q \cdot u \cdot v \, dA|.$$
(3.173)

Ținând cont de mărginirea tensorului a(x), precum și de inegalitățile Cauchy-Schwarz ([97], [99]), Poincaré ([72], [73], [97]) și de poprietatea de continuitate (3.79) a operatorului de urmă, pentru fiecare din termenii de mai sus se poate scrie:

$$\begin{aligned} |\int_{\Omega} a \,\nabla u \cdot \nabla v \, dV| &\leq a_1 |(u, v)|_{1,\Omega} \leq a_1 C_1 ||u||_{1,\Omega} ||v||_{1,\Omega}, \\ |\int_{OM} b \cdot u \cdot v \, dV| &= |(bu, v)|_{0,\Omega} \leq b ||u||_{0,\Omega} ||v||_{0,\Omega} \leq b C_1 ||u||_{1,\Omega} ||v||_{1,\Omega}, \end{aligned}$$
(3.174)
(3.175)

$$\left|\int_{\partial\Omega_{N}} q \cdot u \cdot v \, dA\right| \le q \|u\|_{0,\partial\Omega} \|v\|_{0,\partial\Omega} \le C_{3} \|u\|_{1,\Omega} \|v\|_{1,\Omega}, \tag{3.176}$$

unde $C_i \in \mathbb{R}$, i = 1, 2, 3. Funcțiile b și c fiind pozitiv definite, se poate găsi un interval $W \subset \Omega$ pentru care $b(x) > b_0$, respectiv $c(x) > c_0$, $\forall x \in W$. Prin urmare se poate spune că $\mathcal{A}(\cdot, \cdot)$ este un operator mărginit deoarece:

$$\exists M(\Omega) = a_1 C_1 + b_0 C_2 + c_0 C_3, \text{ a. î. } |\mathcal{A}(u, v)| \le M ||u||_{1,\Omega} ||v||_{1,\Omega}.$$
(3.177)

2. Coercivitatea $\mathcal{A}(\cdot, \cdot)$ Presupunem că \mathcal{A} nu este coerciv. Acest fapt ar implica faptul că, în virtutea proprietății de compactitate ([97]) a domeniului Ω , să

existe un șir mărginit $\{x_n\}$ în $H^1(\Omega)$ (fie, de exemplu, $||x||_{1,\Omega} = k \in \mathbb{R}, k > 0$), pentru care:

$$\lim_{n \to \infty} \mathcal{A}(x_n, x_n) = 0 \qquad \text{ în metrica } H^1.$$
(3.178)

Tot în virtutea proprietății de compactitate, orice subșir $\{x_{n_i}\}$ al șirului $\{x_n\}$ va avea limita x tot în $H^1(\overline{\Omega})$:

$$\lim_{i \to \infty} \|x_{n_i} - x\|_{1,\Omega} = 0.$$
(3.179)

Limita (3.179) este valabilă și în sens slab ([73]), exprimată cu ajutorul produsului de dualitate:

$$<\cdot,\cdot>: H^1(\Omega) \to H^{-1}(\Omega),$$

astfel pentru $\forall w \in H^1(\Omega)$:

$$\lim_{i \to \infty} \langle (x_{n_i} - x), w \rangle_{1,\Omega} = \lim_{i \to \infty} \left[\int_{\Omega} \nabla (x_{n_i} - x) \cdot \nabla w \, dV + \int_{\Omega} (x_{n_i} - x) \cdot w \, dV \right] = 0,$$
(3.180)

care pentru w = x, se scrie:

$$-\int_{\Omega} (\nabla x)^2 \, dV + \lim_{i \to \infty} \int_{\Omega} \nabla x_{n_i} \nabla x \, dV + \lim_{i \to \infty} \int_{\Omega} (x_{n_i} - x) \cdot x \, dV = 0, \quad (3.181)$$

de unde, folosind inegalitatea Hölder [97]:

$$\int_{\Omega} (\nabla x)^2 dV = |x|_{1,\Omega}$$

$$= \lim_{i \to \infty} \int_{OM} \nabla x_{n_i} \nabla x \, dV \le \lim_{i \to \infty} \left[\int_{\Omega} |\nabla x_{n_i}|^2 \, dV \right]^{1/2} \left[\int_{\Omega} |\nabla x|^2 \, dV \right]^{1/2}$$

$$= \lim_{i \to \infty} \int_{\Omega} |\nabla x_{n_i}|^2 \, dV, \qquad (3.182)$$

adică:

$$|x|_{1,\Omega} \le \lim_{i \to \infty} |x_{n_i}|_{1,\Omega}.$$
 (3.183)

Acum, din ipoteza (3.178), rezultă:

$$\lim_{n \to \infty} \mathcal{A}(x_n, x_n) = \lim_{n \to \infty} \left[\int_{\Omega} a(\nabla x_n)^2 \, dV + \int_{\Omega} bx_n^2 \, dV + \int_{\partial \Omega_N} qx_n^2 \, dA \right] = 0$$
(3.184)

Ținând cont de pozitiv definirea funcțiilor a, b, q, rezultă:

$$\lim_{n \to \infty} \int_{\Omega} \left(\nabla x_n \right)^2 dV = 0, \qquad (3.185)$$

$$\lim_{n \to \infty} \int_{\Omega} x_n^2 \, dV = 0, \qquad (3.186)$$

$$\lim_{n \to \infty} \int_{\partial \Omega} x_n^2 \, dA = 0. \tag{3.187}$$

Din (3.185) și (3.183), rezultă:

$$|x|_{1,\Omega} \le \lim_{n_i \to \infty} |x_{n_i}|_{1,\Omega} = \lim_{n \to \infty} \int_{\Omega} (\nabla x_n)^2 \, dV = 0, \qquad (3.188)$$

deci:

$$||x||_{1,\Omega} = ||x||_{0,\Omega} + |x|_{1,\Omega} = ||x||_{0,\Omega} = k > 0.$$
(3.189)

Totodată:

$$\begin{cases} \dim (3.185), \text{ rezultă că } x \text{ este constant pe } \Omega, \\ \dim (3.186), \text{ rezultă că } x = 0 \text{ pe } \Omega, \\ \dim (3.189), \text{ rezultă că } x > 0 \text{ pe } \Omega. \end{cases}$$
(3.190)

apărând o contradicție care neagă ipoteza (3.178), fiind posibilă validarea proprietății de coercivitate (3.147) prin următoarea formulare:

Pentru
$$\forall \{x_n\} \subset H^1(\Omega)$$
 mărginit în sens $\|\cdot\|_{1,\Omega}$, operatorul $\mathcal{A}(\cdot, \cdot)$
este mărginit inferior de valori nenule, prin urmare: (3.191)
 $\exists \gamma(\Omega) > 0 \text{ a.î. } \mathcal{A}(u, u) \geq \gamma \|u\|_{1,\Omega}^2, \quad \forall u \in H^1(\Omega).$

In această formulare, existența șirurilor $\{x_n\}$ mărginite și cu limita în $H^1(\Omega)$ este garantată de faptul că spațiul $H^1(\Omega)$ este complet.

(3.177) și (3.191) demonstrează existența și unicitatea soluției problemei (3.3.1), conform teoremei Lax-Millgram, care garantează și continua dependență a soluției de forma domeniului Ω precum și de excitații:

$$\|u\|_{1,\Omega} \le C(\Omega) \|\mathcal{F}\|_{-1,\Omega}$$

Dacă regrupăm termenii ecuației (3.166) astfel:

$$\int_{\Omega} \left(a \, \nabla u \cdot \nabla v + b \cdot u \cdot v \right) dV + \int_{\partial \Omega_N} \left(q \cdot u - r \right) \cdot v \, dA = \int_{\Omega} f \, v \, dV,$$

și, în membrul stâng, integrăm prin părți primul termen din integrala de volum, ținând cont de faptul că $v \in H_0^1(\Omega)$, obținem:

$$\int_{\Omega} \left[-\nabla(a\,\nabla u) + b \cdot u \right] \, v \, dV + \int_{\partial \Omega_N} \left(a \, \frac{\partial u}{\partial n} + q \cdot u - r \right) \, v \, dA = \int_{\Omega} f \, v \, dV,$$

relație în care

$$a\frac{\partial u}{\partial n} + q \cdot u - r = 0$$

reprezentă condiția pe frontieră $\partial \Omega_N$ din ecuația (3.153) și este satifăcută automat de ecuația:

$$-\nabla \cdot (a \cdot \nabla u) + b \cdot u = f,$$

fiind deci o **condiție de frontieră naturală**. În consecință, pentru rezolvarea (3.153) este necesară numai precizarea condiției de frontieră Dirichlet, pe $\partial\Omega_D$, aceasta fiind una **esențială**. Se pot desprinde câteva caracteristici principale ale formulării slabe (3.166), caracteristici ce evidențiază avantajele utilizării acestui tip de formulare față de cea tare ([84]):

- Nivelul de derivabilitate necesar pentru funcțiile ce descriu formularea slabă, scade cu un ordin față de cel impus de ecuația diferențială tare;
- Condițiile Neumann sau Robin sunt satisfăcute în mod natural, expresiile acestora fiind incluse în ecuația caracteristică formulării slabe. Condiția esențială ce trebuie impusă este numai cea Dirichlet; lipsa acesteia duce însă la necesitatea aplicării unor condiții suplimentare de compatibilitate și etalonare;
- Este suficientă continuitatea pe porțiuni a funcției a(x) ce descrie caracteristica de material, nefiind necesară derivarea ei pe parcursul calculelor;
- Soluția formulării slabe verifică ecuația la nivel funcțional, prin proiecție pe un set de funcții de test ce constituie o bază a spațiului vectorial asociat domeniului pe care este definită problema, spre deosebire de forma tare ce impune găsirea soluției în fiecare punct al acestui domeniu.

3.3.5 Forme particulare de ecuații eliptice

Putem particulariza problema (3.3.1) pentru cazuri des întâlnite în aplicațiile practice de modelare multifizică. În problemele de mai jos domeniul Ω este considerat a fi de tip Lipschitz, deschis și mărginit de forntiera simplu conexă $\partial \Omega$. Problema 3.3.2 (Dirichlet). Pentru ecuația:

$$(-\nabla(a\nabla), v) = < f, v >$$

cu condițiile de frontieră:

$$u = g_D \ pe \ \partial \Omega \ cu,$$

și $f \in H^{-1}(\Omega)$, respectiv condițiile (3.155) pentru a, se caută soluția:

$$u \in H^1_D(\Omega) = \{ x \in H^1(\Omega) | tr_{\partial\Omega} = g_D \} \qquad \forall v \in H^1_0.$$

In urma proiecției pe spațiul funcțiilor de test, operatorul \mathcal{A} are forma:

$$\mathcal{A}(u, v) = \int_{\Omega} a \nabla u \nabla v \, dV$$

 $g_D = 0$: Din relația (3.174) rezultă mărginirea acestuia (condiția (3.146)). Pentru $g_D = 0$, adică $u \in H_0^1(\Omega)$, din proprietățile (3.155), și inegalitatea Poincaré - Friedrichs rezultă coercivitatea aceluiași operator (condiția (3.147)). Prin urmare soluția $u \in H_0^1(\Omega)$ a problemei (3.3.1) există, este unică și urmărește continuu variațiile de domeniu și surse:

$$\|\mathcal{F}\|_{-1,\Omega} \le C(\Omega) \|f\|_{-1,\Omega}.$$

 $g_D \neq 0$: Pentru acest caz, fie $u \in H^1(\Omega)$ și $u_D \in H^1(\Omega)$. Putem defini funcția:

$$\begin{cases} \varphi : H^{1}(\overline{\Omega}) \to H^{-1}(\Omega) \cup H^{1/2}(\Omega), \text{ cu proprietatea} \\ \varphi = u - u_{D} \text{ pe } \overline{\Omega}. \end{cases}$$
(3.192)

Pentru ecuația:

$$\begin{cases} -\nabla(a\nabla\varphi) = f & \text{pe } \Omega\\ \varphi = 0 & \text{pe } \partial\Omega. \end{cases}$$
(3.193)

cu formularea slabă corespondentă:

$$\begin{cases} \mathcal{A}(\varphi, v) = \langle f, v \rangle & \text{pe } \Omega \\ tr_{\partial\Omega}(\varphi) = 0 \end{cases}$$
(3.194)

căutăm soluția $\varphi = (u - u_D) \in H_0^1$ pentru $\forall v \in H_0^1$. Fiind similară cu cea din cazul precedent, putem afirma că ecuația (3.194) are soluție unică. Această afirmație este echivalentă cu următoarea: spațiul vectorial

$$H_D^1 = \{ x \in H^1(\Omega) | tr_{\partial\Omega}(x - g_D) \in H_0^1(\Omega) \}$$

reprezintă spațiul soluțiilor unice pentru o ecuație cu forma slabă:

$$\begin{cases} \mathcal{A}(u, v) = \langle f, v \rangle & \text{pe } \Omega \\ tr_{\partial\Omega}(u) = g_D \end{cases}$$
(3.195)

care nu este alta decât ecuația din problema (3.3.2) cu condiții de frontieră Dirichlet neomogene.

In ambele cazuri de mai sus soluția ecuației (3.3.2) urmărește contiuu variațiile domeniului Ω , respectiv a surselor de câmp reprezentate de f și g_D :

$$||u||_{H^{1},\Omega} \leq C(\Omega) \left(||f||_{H^{-1},\Omega} + ||tr_{\partial\Omega}(u)||_{H^{1/2},\partial\Omega} \right).$$
(3.196)

In cazul în care frontiera pe care sunt puse condițiile Dirichlet este o submulțime $\partial \Omega_D$ a frontierei $\partial \Omega \supset \partial \Omega_D$ atunci se definește un spațiu vectorial de extensie prin zero a spațiului condițiilor de frontieră Dirichlet $H^{1/2}(\partial \Omega_D)$ ([72], [157]):

$$H_{00}^{1/2}(\partial\Omega_D) = \left\{ u \in H^{1/2}(\partial\Omega_D) | \mathcal{E}(u) \in H^{1/2}(\partial\Omega) \right\}.$$
 (3.197)

In concluzie se poate afirma că: Problema (3.3.2) admite o unică soluție în $H^1(\Omega)$, pentru care există constanta $C(\Omega)$ astfel încât:

$$\|u\|_{H^{1},\Omega} \le C(\Omega) \left(\|f\|_{H^{-1},\Omega} + \|g_{D}\|_{H^{1/2}_{00},\partial\Omega_{D}} \right).$$
(3.198)

Netezimea soluției u depinde deci, de regularitatea domeniului de calcul, precum și de regularitatea funcțiilor de excitație f și g_D .

Problema 3.3.3 (Neumann). Pentru ecuația:

$$(-\nabla(a\nabla), v) = < f, v >$$

cu condițiile de frontieră:

$$a\frac{\partial u}{\partial n} = g_N \ pe \ \partial \Omega \ cu,$$

 $f \in H^{-1}(\Omega), g_N \in H^{-1/2}(\Omega), \text{ condițiile (3.155) pentru a și îndeplinirea condiției de compatibilitate: (3.165)}$

$$\int_{\Omega} f \, dV + \int_{\partial \Omega_N} g_N \, dA = 0. \tag{3.199}$$

se caută soluția într-u spațiu vectorial definit astfel încât existența și unicitatea acesteia să fie asigurată. De observat că în expresiile problemei (3.3.3) apar numai derivatele funcției necunoscute, fiind evident faptul că, încercarea de a determina soluții direct în spațiul $H^1(\Omega)$, duce la găsirea unei familii de funcții ce diferă între ele prin constante aditive. Prin urmare, încercăm găsirea soluției tocmai într-un astfel de spațiu, tradițional numit *spațiul cât* [97], definit astfel:

$$H_{\mathcal{K}} \stackrel{\text{not}}{=} H^1(\Omega)/\mathbb{R} = \{ x \in H^1(\Omega) | \exists y \in H^1(\Omega), \text{ si } \mathcal{K} \in \mathbb{R} \text{ a.i. } x = y + \mathcal{K} \}.$$
(3.200)

Constantele \mathcal{K} generează astfel clase de echivalență, cum ar fi, de exemplu, locul geometric al punctelor cu același potențial față de unul ales ca referință (v. și relația (3.93)). Ca funcție (potențial) de referință pentru fiecare astfel de clasă se alege aceea pentru care se poate impune o condiție de valoare medie nulă pe Ω ([84], [157]), adică:

$$\int_{\Omega} \varphi \, dV = 0$$

denumită condiție de etalonare. Existența acestei condiții este legată de posibilitatea de definire pe $H^1(\Omega)$ și $H_{\mathcal{K}}(\Omega)$ a funcțiilor potențial scalar φ , provenite din câmpuri vectoriale irotaționale, adică de condiția ca $ker(rot, H^1(\Omega)) = \nabla(H^1(\Omega))$.

In [118] este demonstrată teorema care afirmă că pentru $\Omega \subset \mathbb{R}^d$, mărginit, simplu conex și Lipschitzian, spațiul definit de (3.200) este un spațiu Hilbert, înzestrat cu norma:

$$\|u\|_{H^1/\mathbb{R}(\Omega)} \stackrel{\text{not}}{=} \|u\|_{\mathcal{K}} = \inf_{\mathcal{K}\in\mathbb{R}} \|u - \mathcal{K}\|_{H^1(\Omega)}.$$
(3.201)

Mai mult, seminorma $|u|_{1,\Omega}$ este echivalentă cu norma (3.201). Legat de spațiile cât, precum și de teorema de mai sus, în aceeași lucrare, se demonstrează faptul că existența și unicitatea soluției problemei (3.3.3) sunt date, pe lângă condițiile teoremei Lax-Milgram, de existența condiției de compatibilitate (3.165), respectiv de etalonare, de tip valoare medie nulă.

In urma proiecției pe spațiul funcțiilor de test, putem stabili pentru problema (3.3.3) următoarea formulare slabă echivalentă:

$$\int_{\Omega} a \nabla u_{\mathcal{K}} \cdot \nabla v_{\mathcal{K}} \, dV = \int_{\Omega} f v_{\mathcal{K}} \, dV + \int_{\partial \Omega} g_n v_{\mathcal{K}} dA, \qquad (3.202)$$

scrisă în notație compactă:

$$\begin{cases} \mathcal{A}(u_{\mathcal{K}}, v_{\mathcal{K}}) = \langle F, v_{\mathcal{K}} \rangle, \\ \text{unde } F : H_{\mathcal{K}}(\Omega) \times H_{\mathcal{K}}(\Omega) \to H^{-1}(\Omega), \\ \text{cu } \langle F, v_{\mathcal{K}} \rangle = \int_{\Omega} f v_{\mathcal{K}} \, dV + \int_{\partial \Omega} g_n v_{\mathcal{K}} dA. \end{cases}$$
(3.203)

Pentru (3.202) căutăm soluția $u_{\mathcal{K}} \in H_{\mathcal{K}}(\Omega)$ pentru $\forall v \in H_{\mathcal{K}}(\Omega)$. Mărginirea și coercivitatea operatorului $\mathcal{A}(u_{\mathcal{K}}, v_{\mathcal{K}})$ se demonstrează similar cu problemele de mai sus, folosind definiția (3.201), echivalența normelor $||u||_{H^1/\mathbb{R}(\Omega)}$ și $|u|_{1,\Omega}$, precum și inegalitatea Poincaré.

In final, în baza teoremei Lax-Milgram și în condițiile existenței funcțiilor potențial scalar pe $H^1(\Omega)$, se poate afirma că Problema (3.3.3) admite o unică soluție în $H^1(\Omega)$, pentru care există constanta $C(\Omega)$ astfel încât:

$$\|u\|_{H^{1},\Omega} \leq C(\Omega) \left(\|f\|_{H^{-1},\Omega} + \|g_{N}\|_{H^{-1/2},\partial\Omega} \right).$$
(3.204)

Netezimea soluției u depinde deci, de regularitatea domeniului de calcul, precum și de regularitatea funcțiilor de excitație f și g_N .

3.4 Ecuații ale elasticității liniare

In cele ce urmează, luăm în considerare un corp perfect elastic, aflat în repaus (vectorul viteză este nul). Materialul din care este alcătuit corpul este omogen și izotrop. Asupra corpului acționează umătoarele forțe (v. fig. 3.6):

- o forță de volum cu densitate **b**;
- pe o porțiune de frontieră a corpului, o forță de suprafață cu tensiunea \mathbf{t} .

Pentru caracterizarea stării elastice a corpului trebuiesc scrise următoarele tipuri de ecuații fundamentale:

- ecuații de echilibru;
- ecuații de compatibilitate geometrică ;
- ecuații constitutive.

3.4.1 Forma tare

Din punct de vedere matematic, corpul ocupă un domeniu conex $\Omega \subset \mathbb{R}^3$, considerat Lipschitzian, fiind mărginit de frontiera $\partial \Omega = \partial \Omega_D \cup \partial \Omega_t$, cu $\partial \Omega_D \cap \partial \Omega_t = \emptyset$.

Conform celor de mai sus, se obține formularea tare a problemei de elasticitate liniară astfel [122]:

1. ecuațiile de echilibru:

$$\begin{cases} -\nabla \overline{\sigma(u)} = \mathbf{b} \quad \text{pe volumul } \Omega \text{ - cf. legii } (2.4.2), \\ \overline{\sigma} \mathbf{n} = \mathbf{t} \quad \text{pe frontiera } \partial \Omega_t \text{ - cf. legii } (2.4.2), \end{cases}$$
(3.205)



Figura 3.6: Corp omogen cu caractreistici elastice liniare și izotrope, asupra căruia acționează atât o forță de volum cât și una de suprafață pe porțiunea de frontieră $\partial \Omega_t$

2. ecuațiile de compatibilitate geometrică cf. relației (2.69), între tensorul deformațiilor $\overline{\overline{\epsilon}}$ și vectorul deplasare **u**:

$$\overline{\overline{\epsilon(u)}} = \frac{1}{2} \left(\left[\nabla \cdot \mathbf{u} \right] + \left[\nabla \cdot \mathbf{u} \right]^T \right), \qquad (3.206)$$

adică:

$$\epsilon_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_j} \right). \tag{3.207}$$

3. ecuațiile liniare de material conform legii lui Hooke pentru medii omogene și izotrope:

$$[\sigma(u)] = [d] \cdot [\epsilon(u)], \qquad (3.208)$$

unde $[\mathbf{d}]$ este matricea de rigiditate pentru medii liniare, omogene și izotrope specificată în (2.98), de unde ([86], [76]):

$$\sigma_{ij}(u) = \lambda \delta_{ij} \nabla \cdot \mathbf{u} + 2\mu \epsilon_{ij}(u), \qquad (3.209)$$

unde δ_{ij} este simbolul Kroneker, $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$ și $\mu = \frac{E}{2(1+\nu)} = G$ sunt parametrii Lamé, E este modulul Young de elasticitate longitudinală, G este modulul de elasticitate transversală, iar ν este coeficientul Poisson pentru materialul respectiv ([107]).

Ținând cont de cele de mai sus, se poate formula următoarea problemă:

Problema 3.4.1. Pentru ecuația:

$$-\nabla \left[\lambda \nabla(\mathbf{u})[\mathbf{I}] + 2\mu \left[\epsilon(\mathbf{u})\right]\right] = \mathbf{b} \qquad pe \ \Omega, \tag{3.210}$$

cu condițiile de frontieră:

$$\begin{cases} \mathbf{u} = \mathbf{u}_D & pe \ \partial \Omega_D \\ \sigma \mathbf{n} = \mathbf{t} & pe \ \partial \Omega_t \end{cases}$$
(3.211)

si:

 $\begin{cases} \lambda \ \text{$$i$ μ numere reale strict pozitive,} \\ [\mathbf{I}] \ este \ matricea \ unitate \ de \ dimensiune \ corspunzătoare,} \end{cases}$ (3.212)

se caută soluția:

$$u \in [\mathcal{C}^2(\Omega)]^3. \tag{3.213}$$

Pentru fiecare componentă i a forței de volum **b** ecuația (3.210) se poate scrie:

$$-\sum_{j=1}^{3} \frac{\partial}{\partial x_j} \left(\lambda \delta_{ij} \nabla \cdot \mathbf{u} + 2\mu \epsilon_{ij}(\mathbf{u}) \right) = b_i.$$

3.4.2Forma slabă

Formularea slabă pentru problema (3.210) se obține, ca și în cazul problemei eliptice Poisson, prin proiectarea ei pe spațiul funcțiilor de test $v \in [H_0^1(\Omega)]^3$, iar apoi prin integrare prin părți. În formă slabă enunțul problemei (3.4.1) devine:

Problema 3.4.2. Pentru ecuația:

$$\int_{\Omega} \left(\lambda \nabla \cdot \mathbf{u} \nabla \cdot \mathbf{v} + 2\mu \epsilon(\mathbf{u}) : \epsilon(\mathbf{v}) \right) dV = \int_{\Omega} \mathbf{b} \cdot \mathbf{v} \, dV + \int_{\partial \Omega_t} \mathbf{t} \cdot \mathbf{v} \, dA \tag{3.214}$$

cu condițiile de frontieră:

$$\begin{cases} \mathbf{u} = \mathbf{u}_D & pe \ \partial \Omega_D \\ \sigma \mathbf{n} = \mathbf{t} & pe \ \partial \Omega_t \end{cases}$$
(3.215)

cu:

$$\begin{cases} \mathbf{b} \in [H^{-1}(\Omega)]^3, \\ \mathbf{u}_D \in [H^{\frac{1}{2}}(\partial \Omega_D)]^3, \\ \mathbf{t} \in [H^{-\frac{1}{2}}(\partial \Omega_t)]^3, \end{cases}$$

se caută soluția:

$$u \in [H_D^1(\Omega)]^3 = \{ x \in [H^1(\Omega)]^3 | tr_{\partial \Omega_D}(x) = u_D \} \qquad \forall v \in [H_0^1(\Omega)]^3.$$
(3.216)

Ecuația (3.214) pune în evidență operatorul biliniar $\mathcal{A}(\mathbf{u}, \mathbf{v})$, respectiv produsul de dualitate $\langle \mathbf{F}, \mathbf{v} \rangle$, scriindu-se în notație compactă astfel:

$$\begin{cases} \mathcal{A}(\mathbf{u}, \mathbf{v}) = \langle \mathbf{F}, \mathbf{v} \rangle, \\ \operatorname{cu} \mathcal{A}(\mathbf{u}, \mathbf{v}) = \int_{\Omega} (\lambda \nabla \cdot \mathbf{u} \nabla \cdot \mathbf{v} + 2\mu \epsilon(\mathbf{u}) : \epsilon(\mathbf{v})) \, dV \\ \operatorname{si} \langle \mathbf{F}, \mathbf{v} \rangle = \int_{\Omega} \mathbf{b} \cdot \mathbf{v} \, dV + \int_{\partial \Omega_t} \mathbf{t} \cdot \mathbf{v} \, dA. \operatorname{cu} \mathbf{F} : [L^2(\Omega)]^3 \times [H_0^1(\Omega)]^3 \to [H^{-1}(\Omega)]^3, \end{cases}$$

$$(3.217)$$

Produsul $\epsilon(\mathbf{u}) : \epsilon(\mathbf{v})$ este un produs scalar tensorial:

$$\epsilon(\mathbf{u}): \epsilon(\mathbf{v}) = \sum_{i,j=1}^{3} \epsilon_{ij}(\mathbf{u}) \epsilon_{ij}(\mathbf{v}).$$
(3.218)

Existența și unicitatea soluției (3.216) se demonstrează prin demonstrarea îndeplinirii condițiilor de mărginire și coercivitate impuse de teorema Lax-Milgram, după modelul demonstrației din cap. 3.3.4. Acea demonstrație folosea inegalitățile Poincaré-Friedrichs aplicabile pentru funcții scalare. Analoagele lor pentru funcțiile tensoriale $\bar{\epsilon}(\mathbf{u})$ sunt inegalitățile lui Korn ale căror demonstrații pot fi găsite în [79] și afirmă că într-un domeniu mărginit Lipschitzian Ω există relațiile:

$$|u|_{1,\Omega^n}^2 \leq \int_{\Omega} |\epsilon(\mathbf{u})|^2 dV = 2\sum_{i,j=1}^n \int_{\Omega} \epsilon_{ij}(\mathbf{u})^2 dV, \qquad \mathbf{u} \in [H_0^1(\Omega)]^n \qquad (3.219)$$

$$\|u\|_{1,\Omega^n}^2 \leq C \int_{\Omega} |\epsilon(\mathbf{u})|^2 dV, \ \mathbf{u} \in H = [H^1(\Omega)]^n / \mathcal{RB} \text{ si } C = C(\Omega) \in \mathbb{R}(3.220)$$

Spațiul cât H din (3.220) este spațiul soluțiilor care diferă între ele printr-o funcție $\mathbf{r} = (\mathbf{A} + \mathbf{B}\mathbf{x}) \in \mathcal{RB}$, și care reprezintă translația ($\mathbf{A} \neq 0$ și $\mathbf{B} = 0$), respectiv rotația ($\mathbf{A} = 0$ și $\mathbf{B} \neq 0$) corpului fără deformare (corp rigid), fiind echivalentul spațiului cât definit, pentru funcții scalare (v. (3.200)) ce diferă printr-o constantă aditivă (v. tratarea problemei (3.3.3)). Mărginirea și coercivitatea operatorului $\mathcal{A}(\mathbf{u}, \mathbf{v})$ din (3.217) se demonstrează folosind inegalitățile Cauchy-Schwarz, Friedrichs (pentru scalari) și Korn pentru n = 3 ((3.219) și (3.220)). O abordare aprofundată asupra acestei demonstrații și discuții pentru cazuri particulare se pot găsi în [86] și [79].

In concluzie, se poate afirma că problema (3.4.2) admite soluție unică. In plus, din teorema Lax-Milgram și (3.217) rezultă și dependența variației soluțiilor de variațiile datelor de intrare:

$$\|u\|_{H^{1},\Omega^{3}} \leq C(\Omega) \left(\|\mathbf{b}\|_{H^{-1}_{D},\Omega^{3}} + \|\mathbf{t}\|_{H^{-1/2},(\partial\Omega)^{3}} \right).$$
(3.221)

3.4.3 Concluzii referitoare la aspectele matematice

Concluzia generală a acestui capitol este că, pentru o corectă formulare matematică se utilizează forma slabă a problemei. Conform acestei formulări, se caută o soluție u, într-un spațiu U al funcțiilor de încercare cu condiții de frontieră esențiale, care este un spațiu Sobolev; soluția u trebuie să satisfacă egalitatea:

$$\mathcal{A}(u,v) = \mathcal{F}(v) \text{ pentru } \forall v \in V, \tag{3.222}$$

unde V este spațiul Sobolev al funcțiilor de test cu condiții esențiale nule.

Problema slabă este deci definită de funcționala biliniară $\mathcal{A}(u, v)$ și de funcționala liniara $\mathcal{F}(v)$, dar și de cele două spații Sobolev, ale funcțiilor de test, respectiv de încercare. Un aspect esențial în formularea matematică este identificarea cadrului funcțional al problemei, adică a celor două spații de funcții.

După cum se va vedea în continuare, în funcție de problema fizică, aceste spații Sobolev reprezintă unul din spațiile secvenței De Rham, respectiv ale funcțiilor grad-, rot-, sau div-conforme. Formularea slabă permite demonstrarea condițiilor de bună formulare și identificarea condițiilor de frontieră esențiale și a celor naturale, ultimele fiind satisfăcute în sens slab.

Capitolul 4

Aspecte numerice

4.1 Discretizarea cu elemente finite

Subcapitolul de față începe printr-o scurtă prezentare a metodei elementului finit, bazată pe formularea slabă expusă în subcapitolul precedent, cu accent pe una din cele mai utilizate variante cunoscută sub numele de "metoda Galerkin". Sunt discutate aspecte geometrice, funcționale și fizice ale metodei. In final, este pus în evidență modul de generare și partiționare în blocuri matriceale a sistemului de ecuații rezultat, putându-se astfel vorbi de o imagine matriceală a domeniului fizic modelat.

In cele ce urmează Ω este un domeniu deschis din \mathbb{R}^d , d = 1, 2, 3, iar $U(\Omega)$ reprezintă un spațiu vectorial Hilbert, definit pe Ω . Conform celor expuse în capitolul 3.2.9 formularea slabă a unei probleme de câmp fizic, se poate enunța astfel:

Problema 4.1.1. Pentru ecuația:

$$\mathcal{A}(u, v) = \mathcal{F}(v), \tag{4.1}$$

se caută soluția:

 $|\mathcal{A}|$

$$u \in U_D(\Omega) \text{ pentru } \forall v \in U_0, \text{ unde } U_D(\Omega) = \{ x \in U(\Omega) | tr_{\partial \Omega_D}(x) = g_D \}$$
(4.2)
$$si \ U_0 = \{ x \in U(\Omega) | tr_{\partial \Omega_D}(x) = 0 \} \text{ - spatial function de test,}$$

presupunând îndeplinite următoarele condiții:

$$|\mathcal{A}(u, v)| \le M \|u\|_U \cdot \|v\|_U, \qquad \forall u, v \in U(\Omega), \ M > 0$$
(4.3)

$$|(u, u)| \ge \alpha ||u||_U^2, \qquad \forall u \in U(\Omega), \ \alpha > 0.$$

$$(4.4)$$

Din punct de vedere matematic, aplicarea metodei elementului finit pentru problema (4.1.1) presupune definirea unor aspecte atât geometrice cât și funcționale. Totodată, oridecâteori este posibil, este foarte utilă și evidențierea semnificațiilor fizice ale aspectelor de mai sus, ca criteriu de evaluare și înțelegere a modului de aplicare a metodei.

Domeniul Ω se aproximează cu un subdomeniu finit Ω_h , obținut prin reuniunea disjunctă a unui număr de N(h) subdomenii Ω_k astfel:

$$\begin{cases} \overline{\Omega}_{h} = \bigcup_{1 \leq k \leq N(h)} \overline{\Omega}_{k}; \\ \overline{\Omega}_{k} = \Omega_{k} \cup \partial \Omega_{k}, \text{ fiecare subdomeniu este închis de frontiera } \partial \Omega_{k}, \text{ netedă pe porțiuni;} \\ \Omega_{i} \cap \Omega_{j} = \emptyset, \text{ pentru } \forall 1 \leq i, j \leq N(h) \text{ şi } i \neq j; \\ \text{ pentru } \forall i, \exists j \neq i, \text{ a.î. frontiera comună } \Gamma_{ij} = \overline{\Omega}_{i} \cap \overline{\Omega}_{j} \neq \emptyset; \\ (\forall \Omega_{i}, \text{ are cel puțin un vecin } \Omega_{j}, \text{ şi în acest caz frontiera comună între cele două subdomenii este nevidă).} \end{cases}$$

Indicele h se referă la finețea de aproximare a domeniului Ω . Din punct de vedere geometric, h permite definirea unei familii de domenii Ω_h astfel încât:

$$\lim_{h \to 0} \Omega_h = \Omega. \tag{4.6}$$

(4.5)

Din punct de vedere funcțional, pe domeniul $\overline{\Omega}_h$ se definește un spațiu vectorial închis, cu dimensiune finită, $U_h(\Omega_h)$. In ipoteza că există o familie de operatori $r_h: U \to U_h$ cu $r_h v = v_h$, astfel încât:

$$\lim_{h \to 0} r_h v = v, \text{ limită în sens tare, } \forall v \in U,$$
(4.7)

se poate enunța următoarea formă discretă pentru problema (4.1.1)([42], [48]):

Problema 4.1.2. Pentru ecuația:

$$\mathcal{A}(u_h, v_h) = \mathcal{F}(v_h), \tag{4.8}$$

se caută soluția:

$$u_{h} \in U_{D_{h}}(\Omega_{h}) = \{ x \in U_{h}(\Omega_{h}) | tr_{\partial \Omega_{D_{h}}}(x) = g_{D} \} = U_{D} \cap U_{h}, \ \forall v_{h} \in U_{0_{f}},$$
(4.9)
$$cu \ U_{0_{h}} = U_{0} \cap U_{h}.$$

presupunând îndeplinite condițiile (4.4) și (4.3).

De regulă, operatorul r_h este un polinom de interpolare ce determină valorile soluției aproximative $u_h(x_{h_i}), x_{h_i} \in \Omega_h$ în funcție de valorile $u(x_i), x_i \in \Omega$ într-un set de N(h) elemente spațiale din Ω .

Se poate demonstra ușor ([48]) că relațiile (4.4) și (4.3) sunt valabile și pentru forma discretă (4.1.2), și anume:

$$|\mathcal{A}(u_h, v_h)| \le M ||u_h||_{U_h} \cdot ||v_h||_{U_h}, \qquad \forall u_h, v_h \in U_h(\Omega_h), \ M > 0$$
(4.10)

$$|\mathcal{A}(u_h, u_h)| \ge \alpha ||u_h||_{U_h}^2, \qquad \forall u_h \in U_h(\Omega_h), \ \alpha > 0, \qquad (4.11)$$

de unde rezultă existența și unicitatea soluției problemei (4.9). De remarcat că prin *soluție* trebuie înțelese valorile determinate în punctele din domeniul Ω (respectiv Ω_h), exclusiv cele de pe porțiunea de frontiera $\partial\Omega_D$ (respectiv $\partial\Omega_h$), aceastea din urmă contribuind direct la termenul liber al ecuației. În baza relației de mărginire (4.10) se poate scrie că:

$$\lim_{h \to 0} \left[\mathcal{A}(u_h, r_h v) - \mathcal{A}(u_h, v) \right] = 0, \ \forall v \in U_{0_h}$$

$$(4.12)$$

Totodată din (4.7) rezultă că:

$$\lim_{h \to 0} \mathcal{F}(r_h v) = \mathcal{F}(v) \tag{4.13}$$

In final, din relațiile (4.12) și (4.13), se poate concluziona ([48]) că soluția problemei discrete este o aproximare, dependentă de h, a soluției problemei (4.1.1) astfel încât:

$$\lim_{h \to 0} u_h = u, \text{ limită în sens tare pe domeniul U}(\Omega).$$
(4.14)

Mai departe, se pot scrie următoarele egalități:

$$\mathcal{A}(u, v) = \mathcal{F}(v), \ \forall v \in U_h$$
$$\mathcal{A}(u_h, v) = \mathcal{F}(v), \ \forall v \in U_h, \ \text{deci}$$
$$\mathcal{A}(u - u_h, v) = 0 \ \forall v \in U_h$$
(4.15)

In termeni geometrici, relația de ortogonalitate (4.15) arată că proiecția distanței $|u - u_h|$ pe spațiul funcțiilor de test este nulă (v. fig. 4.1). Soluția aproximativă u_h , determinată prin metoda elementelor finite este, deci, cea mai apropiată de soluția exactă u, dintre toate soluțiile aproximative din spațiul $U_h(\Omega)$, întrucât segmentul $|u - u_h|$ este perpendicular pe spațiul U_{D_h} al funcțiilor care satisfac problema (4.1.2).

Distanța $|u - u_h|$ este evaluată în sensul normei $\mathcal{A}(u, u)$ care, din punct de vedere fizic, reprezintă energia câmpului vectorial u în domeniul Ω . Interpretarea



Figura 4.1: Interpretarea geometrică a soluției date de metoda Galerkin ([42]): proiecțiile punctelor, u_h din spațiul U_{D_h} și u din U_D , pe spațiul U_0 coincid. In consecință $|u - u_h| \perp U_{h_D}$, deci $||u - u_h||_{U_D} = \inf_{v \in U_{h_0}} ||u - r_h v||_{U_D}$

fizică a (4.15) ar fi deci, că soluția aproximativă u_h , determinată prin metoda elementelor finite reprezintă acel set de funcții ce minimizează energia sistemului fizic modelat:

$$\mathcal{A}(u_h, u_h) \le \mathcal{A}(w, w), \, \forall \, w = r_h v, \, v \in U.$$

$$(4.16)$$

In lucrarea sa, dedicată tehnicilor de aproximare variațională a problemelor la limită ([48]), Jean Céa demonstrează o relație de estimare a preciziei cu care u_h aproximează soluția exactă u, pentru operatori coercivi (probleme eliptice) și autoadjuncți, prin urmăroarea teoremă:

Teorema 4.1.1. Dacă $u \, si \, u_h$ sunt soluțiile problemelor (4.1.1), respectiv (4.1.2), aproximarea cea mai bună, pentru u dintre toate funcțiile din U_h este u_h , iar eroarea de aproximare satisface relația:

$$\|u - u_h\|_U \le \sqrt{\frac{M}{\alpha}} \|u - r_h v\|_U = \sqrt{\frac{M}{\alpha}} \|u - v_h\|_U, \text{ pentru } \forall v_h \in U_h, \qquad (4.17)$$

unde $M \not i \alpha$ sunt constantele de mărginire (continuitate) din (4.3), respectiv de coercivitate din (4.4).

O estimare a preciziei de aproximare poate fi dată și de relația ([42], [159]):

$$||u - u_h||_U \le C(u)h^{\alpha},$$
 (4.18)

unde C(u) este o constantă ce depinde exclusiv de soluția u și de regularitatea acesteia, iar α este o constantă pozitivă ce reflectă gradul p al polinoamelor de interpolare r_h . Relațiile (4.12) si (4.18) reprezintă ideile ce stau la baza tehnicilor de reducere a erorii de determinare a soluției problemei (4.1.1), precum și de îmbunătățire a vitezei de convergență și anume:

- rafinarea rețelei de discretizare în special în jurul punctelor singulare (v. cap. 3.2.2.2) cunoscută ca metodă de tip h;
- mărirea gradului polinoamelor de interpolare cunoscută ca metodă de tip p.

Evaluarea preciziei prin (4.18) nu este deloc simplă, întrucât constanta C nu poate fi determinată înainte de rezolvare, regularitatea (netezimea) soluției fiind influențată atât de regularitatea excitației \mathcal{F} , precum și de netezimea condițiilor de frontieră și de continuitatea proprietăților de material ([84], [42])(v. și relațiile (3.198), (3.204)). De aceea, pentru o primă estimare, se folosesc estimatori de ordin 1 ($\alpha = 1$), ce implică polinoame de interpolare de ordin 1, urmând apoi a se lua decizii asupra tehnicilor optime de îmbunătățire a preciziei, respectiv vitezei de convergență.¹

Spațiul finit U_h este generat de un set de N(h) funcții de bază $\{\varphi_i\}$ numite **funcții de formă, sau de interpolare**, aproximările $u_h \in U_h$ fiind combinații liniare de aceste funcții:

$$u_h(x) = \sum_{j=1}^N u_j \varphi_j(x),$$
 (4.19)

unde N = N(h) este dimensiunea spațiului U_h . Cei N coeficienți u_j din expresia (4.19) poartă denumirea de **grade de libertate**. Ca urmare, ecuația de rezolvat în problema (4.1.2) se poate scrie astfel:

$$\mathcal{A}\left(\sum_{i=1}^{N} u_i \varphi_i(x), v\right) = \mathcal{F}(v), \qquad (4.20)$$

Metoda Galerkin, propune exprimarea atât a funcțiilor necunoscute u cât și a funcțiilor de test cu ajutorul aceluiași set de funcții de bază $\{\varphi_i\}$ ce generează spațiul V al funcțiilor de test. Prin urmare poiecția (4.20) capătă următoarea formă:

$$\mathcal{A}\left(\sum_{j=1}^{N} u_j \varphi_j(x), \sum_{j=1}^{N} v_j \varphi_j(x)\right) = \mathcal{F}\left(\sum_{j=1}^{N} v_j \varphi_j(x)\right), \quad (4.21)$$

¹In contextul de mai sus, notația corectă pentru spațiul U_h al soluțiilor aproximative ar trebui să fie U_{hp} pentru a evidenția dependența soluției de valorile h și p ([60]).

iar relația (4.21) trebuie să fie adevărată pentru orice set de coeficienți $\{v_j\}$. Dacă alegem ca $v_j = \delta_{ij}$, cu δ_{ij} simbolul Kronecker, atunci se obține un sistem liniar de N ecuații cu N necunoscute, de forma:

$$\mathcal{A}\left(\sum_{j=1}^{N} u_{j} \varphi_{j}(x), \varphi_{i}(x)\right) = \mathcal{F}(\varphi_{i}(x)), \text{ respectiv}$$
$$\sum_{j=1}^{N} u_{j} \mathcal{A}(\varphi_{j}(x), \varphi_{i}(x)) = \mathcal{F}(\varphi_{i}(x)). \tag{4.22}$$

In cele prezentate până acum, spațiul V_h este o aproximare finită a spațiului infinit Val funcțiilor de test, oricare dintre bazele φ_i fiind incluse în V. O astfel de aproximare poartă denumirea de *aproximare internă* și face posibilă generarea elementelor finite conforme. În cazuri mai generale, se pot folosi și baze ale unor spații de funcții de test *în care spațiul V este inclus*, caz în care vorbim de *aproximații externe* pentru care elementele finite asociate sunt neconforme, dar care permit modelarea cu o bună precizie a problemelor neeliptice, și/sau a celor definite pe geometrii complexe [84]. Cele două tipuri de proximații sus pomenite au fost definite prima dată de Jean Céa ([48]). **Dacă soluția problemei** (4.1.2) **este unică**, rezultă că și sistemul de ecuații (4.22) este de tip Cramer, admițând tot o soluție unică. În consecință, gradele de libertate reprezentate de elementele vectorului soluție u_h sunt liniar independente, formând, la rândul lor, un set de funcții de bază ce generează spațiul vectorial al funcțiilor $\mathcal{F}(\varphi_i)$, spațiu dual celui generat de funcțiile de formă, fiind și izomorf cu acesta.

Aici trebuie remarcat, că există situații în care modelul matematic al sistemului fizic nu permite independența oricărui tip de grade de libertate considerate în expresia (4.20). De exemplu, gradele de libertate reprezentate de valori nodale (v. cap 4.1.4) nu sunt independente pe frontiera $\partial\Omega$ pe care se impune condiția de flux magnetic constant F: $\int_{\partial\Omega_D} \mathbf{B} \cdot \mathbf{n} \, dA = \int_{\partial\Omega_D} \nabla \times \mathbf{A} \cdot \mathbf{n} \, dA = F$, [42]. Ca urmare, proprietatea de independență trebuie demonstrată pentru orice set de funcții alese ca grade de libertate.

Putem acum să enumerăm cele trei entități matematice care caracterizează un element finit, și anume ([49]):

- 1. suportul spațial al elementului reprezentat de subdomeniul Ω_k , nevid, mărginit și închis de o frontieră nedetă pe porțiuni, notată cu $\partial \Omega_k$:
- 2. spațiul \mathcal{P}_k al funcțiilor de formă definit pe fiecare din subdomeniile Ω_k , spațiu de dimensiune finită m:

$$\mathcal{P}_{k} = \left\{ \varphi_{1}^{k}, \dots, \varphi_{m}^{k} | \varphi_{i}^{k} : \Omega_{k} \to \mathbb{R}, 1 \le k \le m \right\},$$
(4.23)
3. spațiul gradelor de libertate (notate în (4.19) cu u_j) \mathcal{N}_k , de dimensiune m, definit pe elementul Ω_k , care determină în mod unic de setul de m funcții de formă locale ale spațiului \mathcal{P}_k :

$$\mathcal{N}_k = \left\{ N_i^k \,|\, 1 \le i \le m \right\}. \tag{4.24}$$

Unicitatea soluției $\{u_h\}$ implică dualitatea spațiilor \mathcal{P}_k și \mathcal{N}_k .

Odată alese subdomeniile Ω_k , trebuiesc alese spațiile \mathcal{P}_k , respectiv \mathcal{N}_k , astfel încât elementul finit reprezentat de tripletul { $\Omega_k, \mathcal{P}_k, \mathcal{N}_k$ } să îndeplinească următoarele proprietăți:

(P1) **proprietatea de conformitate**, care se referă la faptul că soluțiile u_h sunt funcții continue pe U_h , iar restricțiile lor pe fiecare din porțiunile Ω_k , aparțin spațiului funcțional \mathcal{P}_k . În consecință, spațiul U_h , al funcțiilor ce aproximează soluția exactă u în fiecare punct determinat de vectorul de coordonate \mathbf{x} din Ω , se poate descrie astfel:

$$U_{h} = \{ u_{h}(\mathbf{x}), \text{ cu } u_{h} : \overline{\Omega}_{h} \to \mathbb{R} \mid u_{h} \text{ continuă pe } \Omega_{h} \text{ și } u_{h}^{k}(\mathbf{x}) \in \mathcal{P}_{k}, \forall \Omega_{k} \in \Omega_{h} \},$$

$$(4.25)$$

unde
$$u_h^k(\mathbf{x}) = u_{h|\Omega_k}$$

Pe fiecare element Ω_k funcția necunoscută este reprezentată local printr-o combinație liniară de cele *m* funcții de bază locale $\varphi_i^k(\mathbf{x})$, din spațiul \mathcal{P}_k (v. (4.19)), iar $\mathbf{x} = (x_1, x_2)^T$ pentru domenii 2D, respectiv $\mathbf{x} = (x_1, x_2, x_3)^T$ pentru domenii 3D:

$$u_h^k(\mathbf{x}) = \sum_{i=1}^m N_i^k(\mathbf{x})\varphi_i^k(\mathbf{x}), \qquad (4.26)$$

unde k este indicele elementului spațial Ω_k , iar i este indicele local al variabilei nodale căreia îi este asociată funcția de formă respectivă. Aici prin nod se înțelege o noțiune abstractă prin care se identifică domeniul spațial (vertex, latură, față, sau volum) asociat funcției $\varphi_i^k(\mathbf{x})$, domeniu căruia i se asociază și funcția variabilă nodală $N_i^k(\mathbf{x})$. În această accepțiune, setul de funcții de bază ce generează spațiul \mathcal{P}_k poartă denumirea de bază nodală a acestui spațiu (([60], [159]). Funcțiile u_h sunt continue global pe întreg spațiul funcțional U_h , ele fiind combinații liniare de funcțiile de formă $\{\varphi_j(\mathbf{x})\}$ descrise mai sus, indexate după j ca indice global. Continuitatea se înțelege în sens slab, conform normei induse în spațiul funcțional $U_h \subset U$. Prin urmare este necesar ca și spațiile \mathcal{P}_k să fie subspații ale spațiului U. Funcțiile de formă $\varphi_i^k(\mathbf{x})$ trebuie astfel alese încât să asigure continuitatea restricțiilor funcțiilor u_h^k la trecerea prin frontierele dintre subdomeniile Ω_k , respectiv continuitatea globală pe U_h a funcțiilor u_h . În consecință proprietatea de conformitate (4.25) se poate scrie ca:

$$U_{h} = \left\{ u_{h}(\mathbf{x}) \mid u_{h}^{k}(\mathbf{x}) \in U_{h}(\Omega_{k}), \, \forall \Omega_{k} \in \Omega_{h} \text{ si } u_{h} \in C^{\infty}(\Omega)^{\|\cdot\|_{U}} \cap U(\Omega) \right\}.$$

$$(4.27)$$

Funcțiile de formă se pot împărți în două mari clase ([60]):

- a. definite pe domenii ce constituie frontiera comună dintre elemente vecine (vertexuri, laturi, fețe - v. cap. 4.1.1); în acest caz, suportul acestor funcții de formă este reprezentat de reuniunea subdomeniilor vecine Ω_k care conțin această frontieră comună;
- b. definite pe un domeniu interior al elementului, suportul acestor funcții fiind inclus numai în elementul Ω_k .
- (P2) proprietatea de unisolvență care reflectă faptul că unicitatea soluției implică dualitatea spațiilor \mathcal{P}_k și \mathcal{N}_k . Operatorul specificat în relația (4.7) este un operator de interpolare, definit global prin:

$$\Pi^U: U(\Omega) \to U_h(\Omega), \tag{4.28}$$

și care crează în spațiul finit U_h o imagine discretă a spațiului Sobolev $U(\Omega)$. Acesta din urmă, poate fi inclus într-unul din spațiile $H^1(\Omega)$, $H(\mathbf{rot}, \Omega)$, $H(\mathbf{div}, \Omega)$, respectiv $L^2(\Omega)$, descrise în Cap. 3, iar forma generală a operatorului de intrepolare este:

$$v_h = \Pi^U(v) = \sum_j N_j^U(v(\mathbf{x}))\varphi_j^U(\mathbf{x}), \qquad (4.29)$$

unde $\{N_j^U\}$ este setul de grade de libertate (valori nodale) asociate, după indexul global j, bazei nodale globale $\{\varphi_j^U\}$. Gradele de libertate pot fi privite ca funcții care extrag din spațiul soluțiilor, setul de valori nodale asociate bazei $\{\varphi_j^U\}$ alese; ele se pot defini, deci, ca funcționale pe spațiul U_h al funcțiilor soluție generat de baza nodală $\{\varphi_j^U\}$, cu valori în spațiul dual al acestuia:

$$N_j^U : v \to N_j^U(v) \subset U_h'(\Omega_h), \text{ cu } v \in U_h(\Omega_h) \subset U(\Omega).$$
(4.30)

Local, pentru fiecare din subdomeniile Ω_k , operatorul (4.28) poate fi definit astfel:

$$\Pi_k^U : U(\Omega_k) \to U_h(\Omega_k), \tag{4.31}$$

cu expresia cf. (4.26):

$$v_h^k = \Pi_k^U(v) = \sum_{i=1}^m N_i^{k_U}(v)\varphi_i^{k_U}(\mathbf{x}),$$

unde $m = \dim(\mathcal{P}_k) = \dim(\mathcal{N}_k),$ (4.32)

astfel încât:

$$\Pi^U(v)|_{\Omega_k} = \Pi^U_k(v). \tag{4.33}$$

Din (4.30) și (4.32) deducem că spațiul \mathcal{N}_k este dualul spațiului \mathcal{P}_k . Deci aplicația oricărui operator $T : \mathcal{P}_k \to \mathcal{N}_k$ poate fi reprezentata ca o combinație liniară de elementele bazei $\{N_i^{k_U}\}$:

$$T\varphi^k = \sum_{i=1}^m \psi_i^k N_i^k, \qquad (4.34)$$

între elementele bazelor din cele două spații duale existând relația ([97] - p.114):

$$N_i^k(\varphi_j^k) = \delta_{ij}.\tag{4.35}$$

unde δ_{ij} este simbolul Kronecker.

In concluzie, ținând cont de (4.29), (4.30) și (4.31), putem defini un element finit și prin tripletul { Ω_k , \mathcal{P}_k , Π_U^k }, mai potrivit elementelor finite reprezentate prin funcții de formă polinomiale de grad mai mare decât 1, respectiv pentru cazul optimizărilor de tip p (relația (4.18) și [159], [117], [60]).

Observații 4.1.1. 1. Odată aleasă o bază locală de funcții de formă \mathcal{P}_k , atunci pentru demonstrarea proprietății de conformitate a unui element finit și anume:

$$\mathcal{P}_k \subset U \Leftrightarrow U_h \subset U, \tag{4.36}$$

este suficientă demonstrarea continuității, în sensul normei $\|\cdot\|_U$, a funcțiilor de formă, la trecerea prin frontierele de separație dintre subdomeniile Ω_k ([132], [159], [80]):

(a) Funcțiile u din spațiul $H^1(\Omega)$ trebuie să fie continue pe fiecare interfață comună $\Gamma_{ij} = \partial \Omega_i \cap \partial \Omega_j$:

$$tr_{\partial\Omega_i}(u^i) = tr_{\partial\Omega_j}(u^j) \ pe \ \Gamma_{ij};$$

$$(4.37)$$

(b) Funcțiile u din spațiul $H(rot, \Omega)$ trebuie să conserve componenta tangențială la trecerea prin fiecare interfață comună $\Gamma_{ij} = \partial \Omega_i \cap \partial \Omega_j$:

$$tr_{t_i|\Gamma_{ij}}(u^i) = tr_{t_i|\Gamma_{ij}}(u^j) \ pe \ \Gamma_{ij}; \tag{4.38}$$

(c) Funcțiile u din spațiul $H(div, \Omega)$ trebuie să conserve componenta normală la trecerea prin fiecare interfață comună $\Gamma_{ij} = \partial \Omega_i \cap \partial \Omega_j$:

$$tr_{n_i|\Gamma_{ij}}(u^i) = tr_{n_i|\Gamma_{ij}}(u^j) \ pe \ \Gamma_{ij}; \tag{4.39}$$

2. Unisolvența unui element finit, conform în spațiul funcțional U, poate fi demonstrată dacă funcțiile de formă φ_i^k asociate subdomeniului Ω_k sunt independente, fiind unic determinate de baza de grade de libertate $\mathcal{N}_k^U = \{N_1^{k_U}, \ldots, N_m^{k_U}\}$. Ca atare, trebuie arătată existența implicației:

$$\sum_{i}^{m} N_{i}^{k_{U}} \varphi_{i}^{k_{U}} = 0, \Leftrightarrow N_{i}^{k_{U}} \equiv 0, \forall 1 \le i \le m.$$

$$(4.40)$$

Se spune despre un astfel de element finit că este **izoparametric**, m de grade de libertate asociate acestuia (valori nodale), determinând în mod unic cele m funcții ale bazei nodale a elementului finit respectiv.

4.1.1 Discretizarea spațială

Suportul spațial pentru U_h este reprezentat de împărțirea domeniului $\Omega \subset \mathbb{R}^d$ într-o mulțime finită, $\mathcal{T}_h(\Omega)$, de elemente geometrice disjuncte Ω_k , numită triangulație. Fiecare element geometric este mărginit de frontiera $\partial \Omega_k$. Pentru d = 1, aceste elemente sunt segmente de dreaptă mărginite de câte două puncte numite vertexuri. Pentru d = 2, elementele sunt poligonale, mărginite fiecare de câte un set de laturi și noduri. Pentru d = 3 elementele sunt poliedrale, fiind mărginite în acest caz de câte un set de noduri, laturi, respectiv fețe.

In continuare voi folosi următoarele notații:

- \mathcal{V}_h pentru mulțimea vertexurilor din $\mathcal{T}_h(\Omega)$, respectiv V pentru dimensiunea acestei mulțimi;
- \mathcal{V}_k pentru mulțimea vertexurilor din elementul Ω_k , respectiv V_k pentru dimensiunea acestei mulțimi;
- \mathcal{E}_h pentru mulțimea laturilor din $\mathcal{T}_h(\Omega)$, respectiv E pentru dimensiunea acestei mulțimi;
- \mathcal{E}_k pentru mulțimea laturilor din Ω_k , respectiv \mathbf{E}_k pentru dimensiunea acestei mulțimi;
- \mathcal{F}_h pentru mulțimea fețelor din $\mathcal{T}_h(\Omega)$, respectiv F pentru dimensiunea acestei mulțimi;

- \mathcal{F}_k pentru mulțimea fețelor din Ω_k , respectiv \mathbf{F}_k pentru dimensiunea acestei mulțimi;
- T² pentru dimensiunea triangulației $\mathcal{T}_h(\Omega)$.

Practic $\mathcal{T}_h(\Omega)$ acoperă o aproximație a domeniului Ω , notată aici cu Ω_h , astfel încât:

$$\bigcup_{k \in \mathcal{T}_{h}(\Omega)} \Omega_{k} = \overline{\Omega}_{h},$$

$$\forall i \neq j, \ \Omega_{i} \cap \Omega_{j} = \begin{cases} \emptyset & \Omega_{i} \neq \Omega_{j} \text{ nu sunt vecine,} \\ \text{toate nodurile, laturile comune} & \Omega_{i} \neq \Omega_{j} \text{ vecine } \neq 0, \\ \text{toate nodurile, laturile, fetele comune} & \Omega_{i} \neq \Omega_{j} \text{ vecine } \neq 0, \\ \Omega_{i} \neq 0, \quad \Omega_{j} \neq 0, \end{cases}$$

$$(4.41)$$

Rețeaua cu proprietățile (4.41) nu permite existența de noduri, sau laturi necuplate la *toate* elementele învecinate (elemente "agățate") (v. fig. 4.2). Indicele h reprezintă



Figura 4.2: Rețea de discretizare conformă (a), respectiv neconformă (b)

o condiție de etalonare dimensională a elementelor T_k [132]. De exemplu, aceasta poate fi marginea superioară a mulțimii formate din cele mai mari dimensiuni ale elemetelor Ω_k . Condiția menționată nu este însă suficientă, fiind necesare și restricții de formă - limitări ale valorilor unghiurilor dintre laturile poligoanelor (caz 2D), sau dintre fețele poliedrelor (caz 3D) ([42]). Astfel, se poate defini o măsură a fineței rețelei de discretizare, ce face posibilă generarea unei familii de domenii Ω_h (v. subcap. anterior) cu

$$\lim_{h \to 0} \Omega_h = \Omega$$

²Ar trebui ca dimensiunile legate de triangulația \mathcal{T}_h să poarte și indicele inferior h, specificând astfel dependența dimensiunii respective de finețea de discretizare; întrucât însă, pe parcursul lucrării, nu se vor naște confuzii, pentru simplitatea scrierii, voi folosi simbolurile acestor dimensiuni fără indicele inferior h.

4.1.2 Coordonate baricentrice. Elementul master

Coordonatele baricentrice au un rol important în definirea funcțiilor de formă. Intrun spațiu 2D se pot defini pe poligoane (de ex. triunghi, patrulater), iar în spațiul 3D le putem defini pe poliedre (de ex. tetraedru, paralelipiped). In continuare ne vom referi la triunghiuri, respectiv tetraedre. Considerăm un triunghi Δ ale cărui vârfuri (vertexuri) sunt poziționate față de un punct de referință O prin vectorii \mathbf{r}_1 , \mathbf{r}_2 și \mathbf{r}_3 . Pentru orice punct \mathbf{P} din interiorul acestui triunghi, există un triplet unic de parametri [$\lambda_1, \lambda_2, \lambda_3$], astfel încât poziția acestui punct este dată de vectorul ([141],[47], [120]):

$$\mathbf{r} = \lambda_1 \mathbf{r}_1 + \lambda_2 \mathbf{r}_2 + \lambda_3 \mathbf{r}_3$$

cu $\lambda_i = \frac{\mathcal{A}_{\Delta_i}}{\mathcal{A}_{\Delta}}, i \in \{1, 2, 3\}$
de unde $\lambda_1 + \lambda_2 + \lambda_3 = 1, \lambda_i \in \mathbb{R}^+$
(4.42)

 \mathcal{A}_{Δ_i} reprezintă aria triunghiului Δ_i (v. fig. 4.3). Tripletul $[\lambda_i]$ reprezintă coordonatele baricentrice ale punctului **P** față de triunghiul Δ . Prin urmare, orice punct **P** din acest triunghi are următoarele coordonate carteziene:

$$\begin{aligned} x &= \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 \\ y &= \lambda_1 y_1 + \lambda_2 y_2 + \lambda_3 y_3, \end{aligned}$$

sau folosind (4.42):

$$x = \lambda_1 x_1 + \lambda_2 x_2 + (1 - \lambda_1 - \lambda_2) x_3$$

$$y = \lambda_1 y_1 + \lambda_2 y_2 + (1 - \lambda_1 - \lambda_2) y_3,$$
(4.43)

Se poate constata că vertexurile triunghiului Δ au coordonatele baricentrice cu următoarea proprietate:

$$\lambda_i(P_j) = \delta_{ij}, \ \forall \ 1 \le j \le d+1, \tag{4.44}$$

unde δ este simbolul Kroneker.

Dacă asociem laturilor triunghiului Δ sensuri de referință, le putem privi ca pe niște vectori, caz în care coordonatele baricentrice (4.42) pot fi exprimate folosind expresiile vectoriale ale ariilor respective (ariile sunt pozitive dacă sensul produsului vectorial este determinat de o rotație în sens trigonometric a vectorilor implicați):

$$\lambda_i = \frac{\left(\overrightarrow{PP_{i+1}} \times \overrightarrow{PP_{i+2}}\right) \cdot \vec{k}}{\det_{\mathcal{A}}}, \ 1 \le i \le 3,$$
(4.45)



Figura 4.3: Coordonate baricentrice. (a) - Raportate la un triunghi (2D): toate punctele conținute în latura \mathcal{L}_i , opusă vârfului P_i , sunt caracterizate de relația $\lambda_i = 0$; punctul median M_i al laturii *i*, opuse vârfului *i* are $\lambda_i = 0$ și celelate două coordonate baricentrice egale cu 1/2; centrul de greutate P_g al triunghiului Δ are coordonatele baricentrice [1/3, 1/3, 1/3]. (b) - Raportate la un tetraedru (3D); toate punctele conținute în fața \mathcal{F}_i , opusă vârfului P_i , sunt caracterizate de relația $\lambda_i = 0$; centrul de greutate G_i al feței \mathcal{F}_i are $\lambda_i = 0$ și celelate trei coordonate baricentrice egale cu 1/3; centrul de greutate P_g al tetraedrului are coordonatele baricentrice [1/4, 1/4, 1/4].

unde indicii strict mai mari decât 3 se înțeleg în sens modulo(3) (de ex. pentru i = 3, i+1 = 1); det_A este determinantul egal cu dublul ariei triunghiului Δ :

$$\det_{\mathcal{A}} = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = 2\mathcal{A}_{\Delta}.$$
 (4.46)

In consecuță, gradientul coordonatei baricentrice $\lambda_i(\mathbf{x})$ este constant:

$$\nabla \lambda_i = -\frac{\overrightarrow{P_{i+1}P_{i+2}} \times \vec{k}}{\det_{\mathcal{A}}} = -\frac{|P_{i+1}P_{i+2}| \cdot \vec{n_i}}{\det_{\mathcal{A}}} = -\frac{\vec{n_i}}{h_i}, \qquad (4.47)$$

unde $\vec{n_i}$ este normala exterioară triunghiului, la latura opusă vertexului i (v. fig. 4.4), iar h_i este înălțimea triunghiului Δ , corespunzătoare vârfului P_i (fig. 4.4). In concluzie, gradientul funcției coordonată baricentrică $\lambda_i(\mathbf{x})$ față de triunghiul Δ este un câmp vectorial uniform de modul $\frac{1}{|h_i|}$, direcție normală pe latura opusă vârfului i și sens înspre interiorul triunghiului.

In cazul 3D, al unui tetraedru Th, fiecare punct din acesta este caracterizat de



Figura 4.4: Gradienții coordonatelor baricentrice pentru un triunghi (spațiu 2D)

câte patru coordonate baricentrice:

$$\mathbf{r} = \lambda_1 \mathbf{r}_1 + \lambda_2 \mathbf{r}_2 + \lambda_3 \mathbf{r}_3 + \lambda_4 \mathbf{r}_4$$

cu $\lambda_i = \frac{\mathcal{V}_{Th_i}}{\mathcal{V}_{Th}}, i \in \{1, 2, 3, 4\}$
și $\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 = 1, \lambda_i \in \mathbb{R}^+$
(4.48)

 \mathcal{V}_{Th_i} este volumul tetraedrului definit de punctul interior P și fața \mathcal{F}_i , opusă vertexului P_i . Pentru vârfuri coordonatele satisfac relația (4.44). Dacă asociem sensuri de referință laturilor tetraedrului, se poate defini o orientare a acestuia în funcție de semnul determinantului generat de coordonatele vertexurilor P_1, \dots, P_4 , determinant egal în modul cu de 6 ori volumul \mathcal{V}_{Th} :

$$\det_{\mathcal{V}_{Th}} = \begin{vmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{vmatrix} = 6\mathcal{V}_{Th}.$$
(4.49)

Pentru exemplul din fig. 4.5, cu det_{V_{Th}} dat de relația (4.49), orientarea este pozitivă dacă burghiul drept, rotit în sensul de parcurgere al feței { P_2, P_3, P_4 }, se deplasează către al patrulea vârf P_1 , neconținut în acea față.

Astfel expresia coordonatei λ_1 din relația (4.48) se poate scrie:

$$\lambda_{1} = \frac{\mathcal{V}_{Th_{1}}}{\mathcal{V}_{Th}} = \frac{\frac{1}{3}\mathcal{A}_{1}\vec{n}_{1}\cdot\overrightarrow{PP_{2}}}{V_{Th}} = \frac{1}{3\mathcal{V}_{Th}}\mathcal{A}_{1}\left[(x_{2}-x)\cos(\alpha_{1})+(y_{2}-y)\cos(\beta_{1})+(z_{2}-z)\cos(\gamma_{1})\right] = \frac{1}{h_{1}}\left[(x_{2}-x)\cos(\alpha_{1})+(y_{2}-y)\cos(\beta_{1})+(z_{2}-z)\cos(\gamma_{1})\right],$$
(4.50)



Figura 4.5: In tetraedrul $\{P_1, P_2, P_3, P_4\}$, vectorii luați în ordinea $\{P_1P_2, P_1P_3, P_1P_4\}$ generează o valoare pozitivă pentru determinantul det $_{\mathcal{V}_{Th}}$; de remarcat că setul de vertexuri $\{P_2, P_3, P_4\}$ luați în aceeași ordine, generează o orientare în sens negativ față de suprafața orientată deoarece, conform regulii burghiului drept, rotația în această ordine, determină deplasarea burghiului în sens invers normalei exterioare a feței. Gradienții coordonatelor baricentrice $\lambda_i(\mathbf{x})$ sunt constanți și orientați înspre interiorul tetraedrului.

unde $h_{T1} = \vec{n}_1 \cdot \overrightarrow{PP_2}$ este înălțimea dusă din vârful P pe fața \mathcal{F}_1 , respectiv proiecția muchiei orientate $\overrightarrow{PP_2}$ pe normala \vec{n}_1 , iar $\cos(\alpha_1)$, $\cos(\beta_1)$, $\cos(\gamma_1)$ sunt cosinusurile directoare ale acesteia din urmă. Se poate constata că și în cazul tetraedrului, gradientul funcției baricentrice λ_1 este constant și anume:

$$\nabla \lambda_1 = -\frac{\vec{n}_1}{h_1},\tag{4.51}$$

și generalizând:

$$\nabla \lambda_i = -\frac{\vec{n}_i}{h_i}.\tag{4.52}$$

In concluzie, gradientul funcției coordonată baricentrică $\lambda_i(\mathbf{x})$ față de tetraedrul Th este un câmp vectorial uniform de modul $1/|h_i|$, direcție normală pe fața opusă vârfului i și sens înspre interiorul tetraderului.

Mai departe, rezultă imediat expresia ariei feței \mathcal{F}_i în funcție de λ_i :

$$\mathcal{A}_i = 3\mathcal{V}_{Th} |\nabla \lambda_i|. \tag{4.53}$$

Observații 4.1.2. Putem menționa două consecințe, utile în descrierea elementelor finite, rezultate din cele de mai sus ([42], [156]):

1. Circulația funcției $\nabla \lambda_i(\mathbf{x})$ pe un segment P_1P_2 este egală cu diferența dintre valorile funcției $\nabla_i(\mathbf{x})$ în cele două puncte:

$$\int_{P_1}^{P_2} \nabla \lambda_i(\mathbf{x}) = \lambda_i(P_2) - \lambda_i(P_1); \qquad (4.54)$$

2. Pentru oricare latură $P_k P_j$ dintr-un tetraedru $(1 \le k, j \le 4)$ se poate scrie:

$$\overrightarrow{P_k P_j} \cdot \nabla \lambda_i(\mathbf{x}) = \begin{cases} -1 & dac \breve{a} \ i = k, \\ 0 & dac \breve{a} \ i \neq k \ si \ i \neq j \\ 1 & dac \breve{a} \ i = j \end{cases}$$
(4.55)

Este avantajos ca, mai întâi, analiza și reprezentarea discretizărilor cu elemente triunghiulare (2D), respectiv tetraedrale (3D) să se efectueze pe elemente de referință ("master" - [60], [159], [84]), descrise în coordonate $\xi_i \in [0, 1]$ ca în fig. 4.6 și pe care le notăm cu \hat{T} . Elementele reprezentate în "coordonate fizice reale" urmează a fi obținute prin transformări afine (translații și rotații). Din (4.43), rezultă imediat



Figura 4.6: (a) - Triunghi de referință (2D); (b) - tetraedru de referință; în paranteze rotunde sunt coordonatele carteziene, iar între parantezele pătrate sunt coordonatele baricentrice.

expresiile coordonatelor baricentrice ca funcții de ξ pentru triunghiul de referință:

$$\lambda_{1}(\xi) = 1 - \xi_{1} - \xi_{2},$$

$$\lambda_{2}(\xi) = \xi_{1},$$

$$\lambda_{3}(\xi) = \xi_{2},$$
cu $\hat{T} \subset \mathbb{R}^{2} = \{\xi \mid 0 \le \xi_{i} \le 1, 0 \le \xi_{1} + \xi_{2} \le 1\},$
(4.56)

respectiv pentru tetraedrul de referință:

$$\lambda_{1}(\xi) = 1 - \xi_{1} - \xi_{2} - \xi_{3},$$

$$\lambda_{2}(\xi) = \xi_{1},$$

$$\lambda_{3}(\xi) = \xi_{2},$$

$$\lambda_{4}(\xi) = \xi_{3},$$
cu $\hat{T} \subset \mathbb{R}^{3} = \{\xi \mid 0 \le \xi_{i} \le 1, 0 \le \xi_{1} + \xi_{2} + \xi_{3} \le 1\}.$
(4.57)

In mod similar, se pot determina coordonatele baricentrice, respectiv funcțiile λ_i și pentru alte elemente geometrice utilizate pentru discretizarea spațială a domeniului Ω , cum ar fi elemente patrulatere drepte, derivate dintr-un pătrat de referință (2D), respectiv prisme pentru domeniile 3D ([60], [132]). Așa cum se va vedea și în continuare, reprezentarea spațială în coordonate baricentrice este avantajoasă pentru descrierea funcțiilor de formă asociate elementelor finite conforme. Totodată transformarea cartezian -> baricentric permite calcularea mult mai simplă a integralelor pe suprafețele triunghiurilor, respectiv pe volumele tetraedrelor ([116]).

4.1.3 Transformări afine

Coordonatele elementelor ce descriu domeniul fizic pe care se rezolvă sistemul PDE se obțin din coordonatele elementelor de referință prin transformări afine (translații și rotații). Putem defini un operator de transformare Φ_k din domeniul \hat{T} , în domeniul Ω_k :

$$\Phi_k : \hat{T} \to \Omega_k,$$

$$\mathbf{x} = \Phi_k(\xi) = \mathbf{B}_k \xi + \mathbf{b}_k, \ \forall \xi \in \hat{T},$$

$$(4.58)$$

unde \mathbf{B}_k reprezintă o rotație, fiind o matrice reală de ordin d, iar \mathbf{b}_k , este matricea coloană de dimensiune d a vectorului de translație. Transformarea Φ_k este bijectivă, continuă și derivabilă. Elementele matricelor \mathbf{B}_k și \mathbf{b}_k se pot determina ușor din condițiile de transformare de la coordonatele carteziene ale vertex-urilor elementului de referință la coordonatele (numite și coordonate globale) elementului real corespondent ([84]). De exemplu, pentru d = 3, pentru fiecare vertex $1 \le i \le 4$ (v. fig. 4.7):

$$\begin{pmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{pmatrix} \cdot \begin{pmatrix} \xi_{1_i} \\ \xi_{2_i} \\ \xi_{3_i} \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} x_{1_i} \\ x_{2_i} \\ x_{3_i} \end{pmatrix}$$
(4.59)



Figura 4.7: Transformarea unui tetraedru de referință \hat{T} într-unul din tetraedrele (Ω_k) rețelei de discretizare a modelului fizic.

Operatorul definit de (4.58) este bijectiv, pentru fiecare k imaginea sa reprezentând imaginea elementului master \hat{T} pe elementul fizic Ω_k . Funcțiile de formă asociate elementului Ω_k pot fi determinate din funcțiile de formă ale elementului de referință prin relația:

$$\varphi_i(\mathbf{x}) = \hat{\varphi}_i(\xi), \text{ unde } \mathbf{x} = \Phi_k(\xi)$$

deci $\varphi_i(\mathbf{x}) = \hat{\varphi}_i \circ \Phi_k^{-1}(\mathbf{x})$ (4.60)

Prin urmare, spațiul funcțiilor de formă locale (v.(4.23)) pentru orice element Ω_k poate fi descris ca imagine a spațiului funcțiilor de formă definite pe elementul master astfel:

$$\mathcal{P}_{k} = \left\{ \varphi_{i} = \hat{\varphi}_{i} \circ \Phi_{k}^{-1} : \hat{\varphi}_{i} \in \hat{\mathcal{P}}_{k} \right\}$$

$$(4.61)$$

unde:

$$\hat{\mathcal{P}}_k = \left\{ \hat{\varphi}_1, \dots, \hat{\varphi}_m | \, \hat{\varphi}_i : \hat{T} \to \mathbb{R}, 1 \le k \le m \right\}, \tag{4.62}$$

este spațiul funcțiilor de formă asociate elementului master T. Matricea Jacobian a transformării afine (4.58) notată cu $\mathbf{F}_k(\xi)$ este chiar \mathbf{B}_k , iar Jacobianul asociat este $J_k(\xi) = \det(B_k)$, ambele fiind invariante pe \hat{T} . In consecință, putem scrie transformarea (4.60) aplicată operatorului ∇ astfel:

$$\nabla \varphi_i(\mathbf{x}) = \left[\mathbf{F}_k^{-1}\right]^T \nabla \left(\hat{\varphi}_i \circ \Phi_k^{-1}(\mathbf{x})\right) = \left[\mathbf{F}_k^{-1}\right]^T \nabla \hat{\varphi}_i(\xi).$$
(4.63)

O măsură locală a indicelui h_k , poate fi reprezentată de norma matricei Jacobian \mathbf{F}_k , Jacobianul fiind și o măsură a gradului de deformare a elementului spațial real (Ω_k) față de elementul de referință \hat{T} :

$$h_k(x) = \parallel \mathbf{F}_k(x) \parallel \tag{4.64}$$

Astfel indicele de rafinare h pentru întreaga rețea de discretizare este:

$$h = \sup_{1 \le k \le \dim(\mathcal{T}_h)} h_k(x). \tag{4.65}$$

Pentru cea mai bună condiționare, în mod ideal, numărul de condiționare al matricei \mathbf{F} ar trebui să fie unitar:

$$\|\mathbf{F}_k\| \cdot \|\mathbf{F}_k^{-1}\| = 1. \tag{4.66}$$

Dacă forma elementului spațial real este însă degenerată (unghiuri ascuțite foarte mici, respectiv obtuze foarte mari) atunci produsul celor două norme poate fi departe de unitate. Acesta este un semnal că geometria spațiului discretizat Ω_h va genera erori numerice importante (v. condiția lui Zlamal în [42]), fapt care poate duce la blocarea rezolvării numerice a sistemului de ecuații (4.22). Prin urmare, este de preferat ca numărul de condiționare (4.66) să fie mărginit între valori cât mai apropiate de unitate pentru toate elementele triangulației \mathcal{T}_h , caz în care se spune că aceasta este cvasiuniformă ([159]). Această situație are loc atunci când celulule reale sunt aproape asemanatoare (au unghiuri aproape egale) cu celula master.

Pentru majoritatea tipurilor de element finit, funcțiile de formă sunt polinoame. Transformarea afină (4.58) are avantajul că păstrează gradele polinoamelor de interpolare utilizate pe elementul de referință și pentru elementul real. Estimările erorilor de interpolare în funcție de gradul polinoamelor de interpolare, ce se efectuează pe elementul de referință, pot fi preluate și pentru elementele fizic reale ale întregii rețele de discretizare.

4.1.4 Elementul finit conform în H^1

Elementele finite conforme în spațiul H^1 sunt potrivite pentru rezolvarea problemelor în care necunoscutele sunt funcții potențial scalar din H^1 , de regulă soluții ale unui sistem de ecuații cu derivate parțiale de tip div - grad.

Pe un element master (descris în subcap. 4.1.2), se poate defini spațiul de funcții de formă:

$$\hat{\mathcal{P}}^{H_1} = \left\{ \hat{\varphi}_i \, | \, \hat{\varphi}_i : \hat{T} \to \mathbb{R}, \, \hat{\varphi}_i^1(\xi) = \lambda_i(\xi), \, 1 \le i \le \mathcal{V}_k = d+1 \right\}$$

cu dim $(\hat{\mathcal{P}}^{H_1}) = 3$ pentru triunghiul master (spațiu 2D), respectiv (4.67)
dim $(\hat{\mathcal{P}}^{H_1}) = 4$ pentru tetraedrul master (spațiu 3D),

de unde rezultă imediat că $\hat{\varphi}_i(\xi_j) = \delta_{ij}$: funcțiile de formă $\hat{\varphi}_i$ sunt unitare în fiecare vertex al elementului master care poartă același index. În cazul în care funcțiile $\hat{\varphi}_i$ sunt polinoame de ordin 1, elementul este de liniar (de ordin I). Dacă spațiul $\hat{\mathcal{P}}^{H1}$ din (4.67) este format de seturile de polinoame (4.56), respectiv (4.57), atunci $\hat{\mathcal{P}}^{H1}(\hat{T}) \subset P_1(\hat{T}) \subset H^1(\hat{T})$, unde $P_1(\hat{T})$ este spațiul polinoamelor de ordin 1, de variabilă scalară, definite pe \hat{T} . Aplicând transformarea (4.60) rezultă spațiul funcțiilor de formă definite pe un element local Ω_k :

$$\mathcal{P}^{H1_k} = \left\{ \varphi_i^k \,|\, \varphi_i^k \in P_1(\Omega_k), \, \varphi_i^k(\mathbf{x}) = \lambda_i(\mathbf{x}) \right\}$$
(4.68)

cu proprietatea că în fiecare vertex funcțiile de formă sunt unitare: $\varphi_i^k(\mathbf{x}_j) = \delta_{ij}$. Ținând cont că transformarea (4.60) nu modifică gradul polinoamelor se poate afirma că:

$$\mathcal{P}^{H_{1_k}}(\Omega_k) \subset H^1(\Omega_k) \tag{4.69}$$

Datorită faptului că funcțiile $\varphi_i^k(\mathbf{x})$ au valori 1 sau 0 în vertexuri și o variație liniară pe laturi, se poate arăta foarte ușor continuitatea acestora la trecerea prin frontiera comună (vertexuri, laturi, fețe) dintre două elemente vecine. Prin urmare condiția, (4.37), de conformitate H^1 este îndeplinită. Pe de altă parte, se poate remarca faptul că $\nabla \varphi_i(\mathbf{x})$ este constantă pe orice subspațiu Ω_k , nefiind continuă la trecerea de la un element la altul: $\nabla \varphi_i \in P_0(\Omega_k)$.

Pentru elementele liniare conforme în H^1 , gradele de libertate N_j^{H1} folosite reprezintă potențialele scalare asociate vertexurilor elementelor Ω_k . Ținând cont de (4.30) se poate scrie:

$$N_j^{H1}: v \to v(\mathbf{x}_j), \text{ cu } 1 \le j \le \mathcal{V}_h, \tag{4.70}$$

unde j este indexul global al vertexurilor din Ω_h . Soluția aproximativă locală este dată de operatorul de interpolare local reprezentat de o combinație liniară de funcții de formă independente ([42]):

$$u_h^k(\mathbf{x}) = \Pi_1^{H1_k}(v) = \sum_{i=1}^{V_k} N_i^{H1_k}(v)\varphi_i^k(\mathbf{x}), \qquad (4.71)$$

unde *i* este indexul local al vertexurilor din Ω_k . Independența funcțiilor φ_i^k este echivalentă cu proprietatea de unisolvență (4.40) a elementului finit H^1 conform descris de următorul triplet:

- 1. Domeniul spațial Ω_k ;
- 2. Spațiul funcțiilor de formă \mathcal{P}^{H1_k} :

$$\mathcal{P}^{H_{1_k}} = \left\{ \varphi_i^k \,|\, \varphi_i^k \in P_1(\Omega_k), \, \varphi_i^k(\mathbf{x}) = \lambda_i(x) \right\}, \tag{4.72}$$

unde $P_1(\Omega_k)$ este spațiul polinoamelor de ordin 1, de variabilă scalară, definite pe Ω_k .

3. Spațiul gradelor de libertate \mathcal{N}^{H1_k} :

$$\mathcal{N}^{H_{1_k}} = \{ N_i^{H_{1_k}} \mid N_i^{H_{1_k}}(\mathbf{x}_i) = v(\mathbf{x}_i), \ 1 \le i \le V_k \}.$$
(4.73)

Soluția aproximativă globală este dată de operatorul de interpolare global descris în (4.28):

$$\Pi^{H_1} : H^1(\Omega) \to \mathcal{W}_h^1(\Omega_h),$$

$$u_h = \Pi^{H_1}(u) = \sum_{j=1}^{\mathcal{V}} N_j^{H_1}(u) \varphi_j^1(\mathbf{x}),$$

$$\operatorname{cu} \Pi^{H_1}(u)|_{\Omega_k} = \Pi^{H_{1_k}}(u).$$
(4.74)

unde \mathcal{W}_h^1 reprezintă spațiul soluțiilor formei discrete a problemei (4.1.1), corespunzătoare căutării soluției în spațiul $U_0 \subset H_0^1(\Omega)$:

$$\mathcal{W}_{h}^{1} = \left\{ u_{h}(\mathbf{x}) \in H^{1}(\Omega) \, | \, u_{h}^{k}(\mathbf{x}) \in \mathcal{P}^{1}(\Omega_{k}) \subset H^{1}(\Omega_{k}), \, \forall \Omega_{k} \in \mathcal{T}_{h} \right\},$$
(4.75)

4.1.5 Elementul finit conform în H(rot)

In cazul problemelor în care necunoscutele sunt funcții de potențial vector (de exemplu probleme de tip *rot-rot*) definite pe un domeniu Ω , soluția este căutată în spațiul $H(\mathbf{rot}, \Omega)$. Este cazul regimului magnetic staționar, pentru care este avantajoasă folosirea unor elemente finite conforme în spațiul $H(\mathbf{rot}, \Omega_h)$. Aceasta presupune ca funcțiile necunoscute să fie continue pe porțiuni, în sens $\|\cdot\|_{\mathbf{rot}}$, pe tot Ω_h . Pentru astfel de funcții, se urmărește, saltul componentei tangențiale a câmpului vectorial la trecerea prin forntiera dintre două subdomenii incluse în Ω_h , datorită în principal, interpretărilor fizice asociate acestui salt (de ex. una din consecințele legii lui Ampére din electromagnetism - v.(2.3.4)).

Dacă notăm cu $[\vec{v}_t]_{\Sigma_{ij}}$ saltul componentei tangențiale a câmpului vectorial **v** la terecea prin interfața Σ_{ij} dintre două subdomenii Ω_i și Ω_j (v. fig. 4.8), luând ca referință sensul de trecere de la Ω_i înspre Ω_j se poate scrie:

$$\begin{bmatrix} \vec{v}_t \end{bmatrix}_{\Sigma_{ij}} = \vec{v}_{t_i} - \vec{v}_{t_j} = -\vec{n}_{ij} \times \begin{bmatrix} \vec{n}_{ij} \times \vec{v} \end{bmatrix}_{\Sigma_{ij}} = -\vec{n}_{ij} \times (\vec{n}_i \times \vec{v}_i + \vec{n}_j \times \vec{v}_j) .$$
(4.76)

Prin urmare, la tercerea prin Σ_{ij} , funcțiile din $H(\mathbf{rot}, \Omega_h)$ pot fi continue în sens $\|\cdot\|_{\mathbf{rot}}$, caz în care:

$$\left[\vec{n}_{ij} \times \vec{v}\right]_{\Sigma_{ii}} = 0, \tag{4.77}$$

respectiv pot avea un salt finit:

$$\left[\vec{n}_{ij} \times \vec{v}\right]_{\Sigma_{ij}} = \vec{S}, \text{ cu } |\vec{S}| < \infty.$$

$$(4.78)$$



Figura 4.8: Interpretarea fizică a saltului componentei tangențiale al unei funcții vectoriale la trecerea prin frontiera Σ_{ij} este legată de circulația funcției pe o curbă închisă, notată aici cu Γ_{ij} , având forma unui patrulater foarte îngust ($\delta \rightarrow 0$), cu laturile lungi γ_i și γ_j , paralele cu frontiera și situate în subdomeniile Ω_i , respectiv Ω_j . In acest caz, evaluarea integralei este aproximată cu circulația componentelor tangențiale pe laturile γ , respectiv evaluarea fluxului rotorului superficial $\nabla_s \times \vec{v}$ pe suprafața S_{Γ} subântinsă de curba Γ_{ij} : $\oint_{\Gamma_{ij}} \vec{v} \cdot d\vec{r} = \int_{\gamma} (\vec{v}_i \cdot \vec{\tau}_i - \vec{v}_j \cdot \vec{\tau}_j) = - \iint_{S_{\Gamma}} \vec{n}_{ij} \times [\vec{n}_{ij} \times \vec{v}]$



Figura 4.9: Triunghi (a), respectiv tetraedru (b) de referință cu laturi orientate. Pentru fiecare latură se definesc un vertex inițial și unul final. De exemplu, (a): pentru latura \mathcal{L}_1 a triunghiului $V_{1_{in}} = \hat{P}_2$, respectiv $V_{1_{fin}} = \hat{P}_3$, iar funcția de formă asociată acestei laturi este $\hat{\omega} = \lambda_2 \nabla \lambda_3 - \lambda_3 \nabla \lambda_2$ și (b): pentru latura \mathcal{L}_1 a tetraedrului $V_{1_{in}} = \hat{P}_1$, respectiv $V_{1_{(fin}} = \hat{P}_2$, iar funcția de formă asociată laturii \mathcal{L}_1 este $\hat{\omega} = \lambda_1 \nabla \lambda_2 - \lambda_2 \nabla \lambda_1$

Conform celor de mai sus, pentru elementul finit conform în $H(\mathbf{rot})$, este naturală alegerea unor funcții de formă cu proprietăți legate de circulația, respectiv de comportamentul componentei tangențiale ale câmpului vectorial \mathbf{v} pe laturile (muchiile) triangulației \mathcal{T}_h . In acest caz este utilă alegerea sensului de referință asociat laturii *i* prin stabilirea (arbitrară) la capetele laturii a unui nod inițial, respectiv al unuia final (v. fig. 4.9). Elementele finite conforme în $H(\mathbf{rot})$ sunt derivate din cele conforme în H^1 prin aplicarea operatorului ∇ , urmărind secvența exactă de Rham, descrisă în subcap (3.2.8) ([60], [94]). Operatorul ∇ transformă spațiul $P_{k+1}(\hat{T}) \subset H^1(\hat{T})$, al polinoamelor de variabile scalare și de grad maxim k+1, în spațiul $[P_k]^d(\hat{T}) \subset H(\mathbf{rot}, \hat{T})$ al polinoamelor de grad maxim k, de argument vectorial (d = 2, 3), rezultând următoarea secvență exactă:

$$\mathbb{R} \to P_{k+1}(\hat{T}) \xrightarrow{\nabla} [P_k]^d(\hat{T}) \xrightarrow{\operatorname{rot}} [P_{k-1}]^d(\hat{T}).$$
(4.79)

Spațiul P_{k+1} poate fi privit ca sumă directă între subspațiul închis P_k și un complement algebric (nu este unic) ([97]) notat \tilde{P}_{k+1} , reprezentând spațiul polinoamelor omogene de ordin maxim k + 1 ([60]):

$$P_{k+1} = P_k \oplus \tilde{P}_{k+1}. \tag{4.80}$$

In ([114]) este demonstrată descompunerea Helmholtz generalizată (în sens slab) a spațiului $[P_k]^d$:

$$[P_k]^d = \nabla(\tilde{P}_{k+1}) \oplus R_k, \tag{4.81}$$

unde:

$$R_k = [P_{k-1}]^d \oplus S_k, (4.82)$$

cu $[S_k] = \{\mathbf{q} \mid \mathbf{q} \in [\tilde{P}_k]^d, \mathbf{x} \cdot \mathbf{q} \equiv 0\}$ fiind spațiul polinoamelor omogene de ordin maxim k, de agument din \mathbb{R}^d , ortogonale pe spațiul \hat{T} . Inlocuind (4.80) și (4.81) în (4.79) și eliminând din primii doi membri \tilde{P}_{k+1} , respectiv $\nabla(\tilde{P}_{k+1})$, secvența (4.79) devine:

$$\mathbb{R} \to P_k(\hat{T}) \xrightarrow{\nabla} R_k(\hat{T}) \xrightarrow{\nabla \times} [P_{k-1}]^d(\hat{T}).$$
(4.83)

Astfel se definește:

$$\hat{\mathcal{P}}_{k,I}^{\mathbf{rot}}(\hat{T}) = R_k(\hat{T}) \subset H(\mathbf{rot}, \hat{T}), \tag{4.84}$$

ca spațiu al funcțiilor de formă pentru elementul Nédélec de ordin k, de prima speță ([119]).

Cea mai restrânsă bază nodală conformă în spațiul $H(\mathbf{rot}, \hat{T})$, se obține pentru k = 1, pentru care secvența exactă (4.79) se scrie:

$$\mathbb{R} \to P_1(\hat{T}) \xrightarrow{\nabla} [P_0]^3(\hat{T}) \oplus [S_1]^3(\hat{T}) \xrightarrow{\nabla \times} \{0\}.$$
(4.85)

Pe elementul master, această bază nodală descrie spațiul $\hat{\mathcal{P}}_{1,I}^{\text{rot}}$, ai cărui membri corespund elementului Whitney de ordin 1, după numele matematicianului care lea formulat pentru prima dată într-un tratat dedicat geometriei diferențiale ([156], [42]); ulterior, elementul a fost reformulat de Nédélec ([119]), ca element de muchie de ordin 1:

$$\hat{\mathcal{P}}_{1,I}^{\mathbf{rot}} = [P_0]^d \oplus S_1 = R_1.$$
(4.86)

cu spații de funcții de forma:

$$\hat{\mathcal{P}}_{1,I}^{\mathbf{rot}} = \left\{ \mathbf{a} + (\xi_2, -\xi_1)^T \, | \, \mathbf{a} \in \mathbb{R}^2, \, b \in \mathbb{R} \right\} \text{ pentru domenii 2D,}
\hat{\mathcal{P}}_{1,I}^{\mathbf{rot}} = \left\{ \mathbf{a} + \mathbf{b} \times \mathbf{x} \, | \, \mathbf{a}, \mathbf{b} \in \mathbb{R}^3 \right\}, \text{ pentru domenii 3D.}$$
(4.87)

Funcțiile de formă definite pe triunghiul, respectiv tetraedrul de referință, sunt asociate laturilor acestora, iar reprezentarea lor în coordonate baricentrice $\lambda_i(\xi) \in P_1(\hat{T})$ este următoarea:

$$\hat{\mathcal{P}}_{1,I}^{\text{rot}} = \left\{ \hat{\omega}_{i}^{\text{rot}} \mid \hat{\omega}_{i}^{\text{rot}} \in [P_{0}]^{d} \oplus [S_{1}]^{d}, 1 \leq i \leq L_{k} \right\}$$

iar i este indexul local al laturilor elementului master.

Pentru claritate, următoarele proprietăți for fi evidențiate pe cazul particular al funcției de formă $\hat{\omega}_1^{\text{rot}}$ asociată laturii \mathcal{L}_1 din tetraedrul de referință, latură orientată de la nodul \hat{P}_1 la nodul \hat{P}_2 . De aici, se pot deduce foarte ușor, proprietățile analoage ale celorlalte funcții de formă: $\hat{\omega}_2^{\text{rot}}, \cdots, \hat{\omega}_6^{\text{rot}}$.

Cu ajutorul relației (4.55) și a faptului că $\hat{\lambda}_1 + \hat{\lambda}_2 = 1$ pe latura \mathcal{L}_1 , putem deduce expresia componentei tangențiale a funcției vectoriale $\hat{\omega}_1^{\mathbf{rot}_1}$ pe laceastă latură:

$$\hat{\omega}_{1}^{\mathbf{rot}} \cdot \vec{\tau}_{1} = \hat{\omega}_{1}^{\mathbf{rot}} \cdot \frac{\overrightarrow{\hat{P}_{1}\hat{P}_{2}}}{|\hat{P}_{1}\hat{P}_{2}|} =$$

$$\frac{\lambda_{1}}{|\hat{P}_{1}\hat{P}_{2}|} \left(\nabla\lambda_{2} \cdot \overrightarrow{\hat{P}_{1}\hat{P}_{2}}\right) - \frac{\lambda_{2}}{|\hat{P}_{1}\hat{P}_{2}|} \left(\nabla\lambda_{1} \cdot \overrightarrow{\hat{P}_{1}\hat{P}_{2}}\right) = \frac{1}{|\hat{P}_{1}\hat{P}_{2}|},$$

$$(4.89)$$

de unde rezultă că circulația funcției de formă $\hat{\omega}_1^{\mathbf{rot}}$ este unitară doar pe latura cu același indice, fiind nulă pe celelalte laturi. Generalizând:

$$\hat{\omega}_i^{\text{rot}} \cdot \vec{\tau}_j = \frac{1}{|\hat{\mathcal{L}}_i|} \delta_{ij},\tag{4.90}$$



Figura 4.10: Liniile de câmp pentru funcția de formă $\hat{\omega}_1^{\mathbf{rot}_1}$ pentru elementele master triunghi (a), respectiv tetraedru (b).

respectiv:

$$\int_{\hat{\mathcal{L}}_i} \hat{\omega}_j^{\text{rot}} \cdot \vec{\tau}_i = \delta_{ij}, \ 1 \le i, j \le m.$$
(4.91)

Se poate observa că rotorul funcție
i $\hat{\omega}_1^{{\bf rot}_1}$ este constant:

$$\nabla \times \hat{\omega}_1^{\mathbf{rot}_1} = 2 \left(\nabla \lambda_1 \times \nabla \lambda_2 \right), \tag{4.92}$$

iar pe fețele laterale laturii $\hat{\mathcal{L}}_i$ avem:

$$\hat{\omega}_{1}^{\mathbf{rot}} = -\frac{\lambda_{1}}{h_{2}}\vec{n}_{2} \text{ pe fața } \mathcal{F}_{2}$$

$$\hat{\omega}_{1}^{\mathbf{rot}} = +\frac{\lambda_{2}}{h_{2}}\vec{n}_{1} \text{ pe fața } \mathcal{F}_{1},$$
(4.93)

de unde deducem că funcția $\hat{\omega}_1^{\text{rot}}$ reprezintă un câmp vectorial ce se rotește în jurul laturii ce nu conține vertexurile \hat{P}_1 și \hat{P}_2 (muchia opusă muchiei $\hat{\mathcal{L}}_1$), fiind nulă pe acea latură. Rotirea în sensul câmpului deplasează burghiul drept către vertexul opus feței ce conține latura $\hat{\mathcal{L}}_1$ (asemănător regulii stabilite pentru orientarea tetraedrului) (v. fig, 4.10-(b)). Acest câmp vectorial este normal pe cele două fețe laterale \mathcal{F}_1 respectiv \mathcal{F}_2 și tangențial pe celelalte două fețe care se intersectează după latura $\hat{\mathcal{L}}_1$: \mathcal{F}_3 și \mathcal{F}_4 . Mai mult, pe aceste ultime două fețe componenta tangențială nu depinde decât de coordonatele baricentrice λ_1 și λ_2 , fiind astfel identică pe ambele părți ale acestor fețe.

Din cele de mai sus rezultă faptul că funcțiile de formă $\hat{\omega}_i^{\text{rot}}$, din spațiul $\hat{\mathcal{P}}_I^{\text{rot}}$ din (4.88), au componenta tangențială continuă la trecerea prin oricare față de frontieră dintre două tetradere vecine. O logică asemănătoare se aplică și pentru elementele 2D triunghiulare, acolo fiind vorba despre continuitatea componentelor tangențiale ale funcțiilor de formă la trecerea prin orice latură de frontieră dintre două triunghiuri vecine (v. fig, 4.10 - (a)). Prin urmare condiția (4.38) de conformitate $H(\mathbf{rot})$ este îndeplinită.

Ținând cont de (4.91), variabila nodală asociată elementului nodal $\hat{\omega}_i^{\text{rot}}$, este aleasă a fi în acest caz, circulația funcției vectoriale necunoscute \vec{v} pe latura *i*:

$$\hat{N}_{i}^{\mathbf{rot}} : \hat{\vec{v}} \to \int_{\hat{\mathcal{L}}_{i}} \vec{\hat{v}} \cdot d\vec{\tau}_{i}, \text{ cu } 1 \le i \le \mathcal{L}_{k}.$$

$$(4.94)$$

Putem deci exprima funcția necunoscută, la nivel de element master, prin combinația liniară:

$$\vec{\hat{v}} = \sum_{i=1}^{m} \hat{N}_{i}^{\text{rot}} \hat{\omega}_{i}^{\text{rot}}$$
(4.95)

Dacă impunem $\hat{v} = 0$, rezultă și anularea componentelor tangențiale ale \vec{v} pe oricare din laturile elementului master: $\vec{v} \cdot \vec{\tau}_j = 0$, $\forall 1 \leq j \leq m$; apoi, luând în considerare (4.90), rezultă:

$$\sum_{i=1}^{m} \hat{N}_{i}^{\mathbf{rot}} \hat{\omega}_{i}^{\mathbf{rot}} \cdot \vec{\tau}_{j} = \hat{N}_{j}^{\mathbf{rot}} \frac{1}{|\hat{\mathcal{L}}_{j}|} = 0 \Leftrightarrow \hat{N}_{j}^{\mathbf{rot}} = 0, \, \forall \, 1 \le j \le \mathcal{L}_{k}.$$
(4.96)

In consecință este demonstrată implicația de unisolvență (4.40).

In concluzie, tripletul de referință $\{\hat{T}, \hat{\mathcal{P}}_{1,I}^{\mathbf{rot}}, \hat{\mathcal{N}}_{1,I}^{\mathbf{rot}}\}$ reprezintă un element finit conform și unisolvent în $H(\mathbf{rot}, \hat{T})$.

Pentru determinarea elementului finit din domeniul fizic, trebuie aplicată elementelor tripletului de mai sus transformarea afină (4.58), adaptată la faptul că trebuie să transforme $\nabla \times \hat{\vec{u}}$ în $\nabla \times \vec{u}$. In [114] (Lema 3.57) se demonstrează că dacă funcției vectoriale $\vec{\vec{u}} \in H(\mathbf{rot}, \hat{T})$ i se aplică transformarea Φ definită în (4.63), rezultă funcția \vec{u} cu:

$$\vec{u} = \left[\mathbf{F}_k^{-1}\right]^T \hat{\vec{u}} \circ \Phi_k^{-1} \text{ si } \vec{u} \subset H(\mathbf{rot}, \Omega_k), \tag{4.97}$$

respectiv:

(a). pentru cazul 2D

$$\nabla \times \vec{u} = J_k^{-1} \nabla \times \left(\vec{\hat{u}} \circ \Phi_k^{-1} \right), \tag{4.98}$$

(b). pentru cazul 3D

$$\nabla \times \vec{u} = J_k^{-1} \mathbf{F}_k \nabla \times \left(\hat{\vec{u}} \circ \Phi_k^{-1} \right)$$
(4.99)

(c). componentele tangențiale pe laturi se transformă astfel:

$$(\vec{u} \cdot \vec{\tau_i}) \circ \Phi_k = \frac{\vec{\hat{u}} \cdot \vec{\hat{\tau_i}}}{\|\mathbf{F}_k^T \vec{\tau_i}\|}, \qquad (4.100)$$

(d). de unde rezultă și circulația acestora pe una din laturile elementului finit fizic:

$$\int_{\Phi_k(\hat{\mathcal{L}})} \vec{u} \cdot \vec{\tau} dx = \int_{\hat{\mathcal{L}}} \vec{\hat{u}} \cdot \vec{\hat{\tau}} d\xi.$$
(4.101)

Transformarea afină lasă neschimbată acțiunea operatorilor gradient și rotor (respectv componenta tangențială) și integrală la trecerea de la domeniul \hat{T} la Ω_k , păstrându-se astfel neschibată și secvența (4.85). Din (4.101) rezultă și modul de obținere a gradelor de libertate pe elementul finit fizic Ω_k , conform cu $H(\mathbf{rot})$, din cel definit pe elementul master:

$$N_i^{\mathbf{rot}_k}(\vec{u}) = \int_{\hat{\mathcal{L}}} \frac{\vec{\hat{u}} \cdot \vec{\hat{\tau}_i}}{\|\mathbf{F}_k^T \vec{\tau}_i\|} \|\mathbf{F}_k^T \vec{\tau}_i\| d\xi = \hat{N}_i^{\mathbf{rot}}(\vec{u}).$$
(4.102)

La nivel global, pe domeniul Ω_h :

$$N_j^{\mathbf{rot}} : \vec{v} \to \int_{\mathcal{L}_j} \vec{v} \cdot d\vec{\tau}_j, \qquad (4.103)$$

unde j este indexul global al laturilor (muchiilor) din Ω_h . Soluția aproximativă locală este descrisă de funcții independente, descrise de operatorul de interpolare local:

$$\vec{u}_h^k(\mathbf{x}) = \Pi_{1,I}^{\mathbf{rot}_k}(\vec{v}) = \sum_{i=1}^{L_k} N_i^{\mathbf{rot}_k}\vec{v})\omega_i^k\mathbf{x}, \qquad (4.104)$$

unde *i* este indexul local al laturilor din Ω_k , iar $\omega_i^k(\mathbf{x})$ este funcția de formă corespunzătoare muchiei *i* din Ω_k . Expresia acestei funcții de formă se determină din funcțiile de formă ale elementului de referință cu ajutorul transformării (4.63):

$$\omega_i^k(\mathbf{x}) = \lambda_{i_{in}}(\mathbf{x}) \nabla \lambda_{i_{fin}}(\mathbf{x}) - \lambda_{i_{fin}}(\mathbf{x}) \nabla \lambda_{i_{in}}(\mathbf{x}) = \left[\mathbf{F}_k^{-1}\right]^T (\mathbf{\hat{x}}) \hat{\omega}_i(\mathbf{\hat{x}})$$
(4.105)

In concluzie, putem descrie elementul finit Nédélec de ordin 1, de speța întâi, prin următorul triplet:

- 1. Domeniul spațial Ω_k
- 2. Spațiul funcțiilor de formă $\mathcal{P}_{1,I}^{\mathbf{rot}_k}$:

$$\mathcal{P}_{1,I}^{\mathbf{rot}_{k}} = \left\{ \omega_{i}^{k} \mid \omega_{i}^{k} \in R_{k}(\Omega_{k}), \\ \omega_{i}^{k}(\mathbf{x}) = \lambda_{i_{in}}(\mathbf{x}) \nabla \lambda_{i_{fin}}(\mathbf{x}) - \lambda_{i_{fin}}(\mathbf{x}) \nabla \lambda_{i_{in}}(\mathbf{x}), \ 1 \leq i \leq L_{k} \right\},$$

unde *i* este indexul local al laturilor elementului Ω_{k} .
(4.106)

3. Spațiul gradelor de libertate $\mathcal{N}_{1,I}^{\mathbf{rot}}$:

$$\mathcal{N}_{1,I}^{\mathbf{rot}_k} = \left\{ N_i^{\mathbf{rot}} \mid N_i^{\mathbf{rot}} = \int_{\mathcal{L}_i} \vec{v}(\mathbf{x}) \cdot d\vec{\tau}_i, \ 1 \le i \le \mathcal{L}_k \right\}.$$
(4.107)

Soluția aproximativă globală este dată de operatorul de interpolare global descris în (4.28):

$$\Pi_{1,I}^{\mathbf{rot}} : H(\mathbf{rot}, \Omega) \to \mathcal{V}_{h,I}^{1}(\Omega_{h}),$$

$$u_{h} = \Pi_{1,I}^{\mathbf{rot}}(\vec{u}) = \sum_{j=1}^{\mathrm{E}} N_{j}^{\mathbf{rot}}(\vec{u})\omega_{j}(\mathbf{x}),$$

$$\mathrm{cu} \ \Pi_{1,I}^{\mathbf{rot}}\vec{u}|_{\Omega_{k}} = \Pi_{1,I}^{\mathbf{rot}_{k}}\vec{u}.$$
(4.108)

unde $\mathcal{V}_{h,I}^1$ reprezintă spațiul funcțional al soluțiilor formei discrete a problemei (4.1.1), corespunzătoare căutării soluției în spațiul vectorial $U_0^{\text{rot}} \subset H_0(\text{rot}, \Omega)$:

$$\mathcal{V}_{h,I}^{1} = \left\{ \vec{u}_{h}(\mathbf{x}) \in H(\mathbf{rot},\Omega) \mid \vec{u}_{h}^{k}(\mathbf{x}) \in R_{k}(\Omega_{k}) \subset H(\mathbf{rot},\Omega_{k}), \forall \Omega_{k} \in \mathcal{T}_{h} \right\}.$$
(4.109)

4.1.6 Elementul finit conform în $H(\operatorname{div})$

Pentru problemele pentru care câmpul vectorial necunoscut este căutat în spațiul H(div) (de ex. inducția electrică, sau cea magnetică în problemele de electromagnetism), este avantajoasă folosirea unor elemente finite conforme în spațiul $H(div, \Omega_h)$. Aceasta presupune ca funcțiile necunoscute să fie continue pe porțiuni, în sens $\|\cdot\|_{div}$, pe tot Ω_h . Pentru astfel de funcții se urmărește saltul componentei normale a câmpului vectorial la trecerea prin frontiera dintre două subdomenii incluse în Ω_h , datorită interpretărilor fizice asociate acestui salt (de ex. una din consecințele legii fluxului magnetic - v. (2.3.2)).

Dacă notăm cu $[\vec{v}_n]$ saltul componentei normale a câmpului vectorial \vec{v} la trecerea prin interfața Σ_{ij} dintre două subdomenii Ω_i și Ω_j (v. fig. 4.11), luând ca referință sensul de trecere de la Ω_i înspre Ω_j se poate scrie:

$$[\vec{v}_n]_{\Sigma_{ij}} = \vec{v}_{n_i} - \vec{v}_{n_j} = \vec{n}_{ij} \cdot \vec{v}_i - \vec{n}_{ij} \cdot \vec{v}_j = \vec{n}_i \cdot \vec{v}_i + \vec{n}_j \cdot \vec{v}_j.$$
(4.110)

Prin urmare, la tercerea prin Σ_{ij} , funcțiile din $H(div, \Omega_h)$ pot fi continue în sens $\|\cdot\|_{div}$, caz în care:

$$[\vec{n}_{ij} \cdot \vec{v}]_{\Sigma_{ii}} = 0, \tag{4.111}$$

respectiv pot avea un salt finit:

$$\left[\vec{n}_{ij} \cdot \vec{v}\right]_{\Sigma_{ij}} = s, \text{ cu } |s| < \infty.$$

$$(4.112)$$



Figura 4.11: Interpretarea fizică a saltului componentei normale al unei funcții vectoriale la trecerea prin frontiera Σ_{ij} este legată de evaluarea fluxului funcției respective pe o suprafață cilindrică, notată aici cu S_{ij} , care la limită, tinde spre un punct situat pe Σ_{ij} $(\delta \to 0, \rho \to 0)$. Aplicând teorema lui Ostrogradski, rezultă că acest flux este egal cu integrala de volum a divergenței superficiale $\nabla_s \vec{v}$ pe volumul cilindric V_{Sij} închis de S_{ij} : $\oint_{S_{ij}\to 0} \vec{v} \cdot \vec{n} \, d\mathcal{A} = \int_{V_{Sij}\to 0} div_s \vec{v} \, d\mathcal{V} = \int_{V_{Sij}\to 0} [\vec{v}_n]_{\Sigma ij} \, d\mathcal{V}$

Pentru elementul finit conform în H(div), tridimensional, este potrivită alegerea unor funcții de formă cu proprietăți legate de comportamentul componentei normale a câmpului vectorial \vec{v} pe fețele tetraedrelor (fluxuri) triangulației \mathcal{T}_h pentru domenii 3D, respectiv pe laturile triunghiurilor pentru triangulații 2D. Fețele vor fi orientate, iar sensul de referință este dat de normala exterioară pe suprafața respectivă (v. fig. 4.12). Din acest motiv acest tip de elemente poartă și denumirea de elemente de față.

Elementele conforme în H(div) sunt derivate din cele conforme în $H(\mathbf{rot})$ prin aplicarea operatorului **rot** ($\nabla \times$), urmând secvența exactă de Rham, în variantele sale pentru domenii 2D (v. (3.140)), respectiv 3D (v. (3.139)). Pentru domeniile 2D funcțiile de formă $\omega_i^{div_1}$ sunt obținute prin rotirea cu $\pi/2$ a funcțiilor $\omega_i^{\mathbf{rot}_1}$. De exemplu pentru latura $\hat{\mathcal{L}}_1$ a triunghiului master din fig.4.12-(a), funcția de formă este:

$$\hat{\omega}_1^{div_1} = \lambda_2 \vec{\nabla}_2 \lambda_3 - \lambda_3 \vec{\nabla}_2 \lambda_2 = \lambda_2 \vec{k} \times \nabla \lambda_3 - \lambda_3 \vec{k} \times \nabla \lambda_2 = \vec{k} \times \hat{\omega}_1^{\mathbf{rot}_1}.$$
(4.113)

Pentru fața orientată $\hat{\mathcal{F}}_1$ a tetraedrului de referință cf. fig.4.12-(b):

$$\hat{\omega}_{1}^{div_{1}} = \lambda_{2} \nabla \times \hat{\omega}_{5}^{\mathbf{rot}_{1}} + \lambda_{4} \nabla \times \hat{\omega}_{4}^{\mathbf{rot}_{1}} + \lambda_{3} \nabla \times \hat{\omega}_{6}^{\mathbf{rot}_{1}} = 2\left[\lambda_{2}(\nabla\lambda_{4} \times \nabla\lambda_{3}) + \lambda_{4}(\nabla\lambda_{3} \times \nabla\lambda_{2}) + \lambda_{3}(\nabla\lambda_{2} \times \nabla\lambda_{4})\right]$$

$$(4.114)$$

Spațiul funcțional din care fac parte funcțiile de formă $\hat{\omega}_i^{div_1}$ este ([114]):

$$D_{k} = [P_{k-1}]^{d} \oplus \tilde{P}_{k-1}\mathbf{x}, \text{ cu } \mathbf{x} = \{x_{1}, \cdots, x_{d}\}^{T},$$
(4.115)



Figura 4.12: (a) Triunghi de referință; toate laturile sunt orientate la fel relativ la normalele exterioare proprii. (b)Tetraedru de referință cu fețe orientate. Sensul pozitiv de parcurgere a laturilor unei fețe este sensul de rotație al burghiului drept care îl deplasează în sensul normalei exterioare a feței respective.

unde d = 2 pentru domeniile 2D, respectiv d = 3 pentru domeniile 3D. In [114] -Lema 5.13 - se demonstrează că $\nabla \cdot D_k = P_{k-1}$, astfel încât secvența exactă de spații funcționale ale funcțiilor de formă conforme, corespunzătoare secvenței de Rham (3.128) este:

$$\mathbb{R} \to P_k \xrightarrow{\nabla} R_k \xrightarrow{\nabla \times} D_k \xrightarrow{\nabla} P_{k-1} \to \{0\}.$$
(4.116)

Funcțiile de formă (4.113) și (4.114) sunt asociate elementelor finite de față, cunoscute sub denumirea de Raviart-Thomas ([114], [94]). Cel mai restrâns spațiu al acestor funcții de formă este D_1 (k = 1), al cărui descriere în coordonate carteziene este:

$$\hat{\mathcal{P}}^{div_1} = \left\{ \mathbf{a} + b\mathbf{x} \mid \mathbf{a} \in \mathbb{R}^d, \ b \in \mathbb{R} \right\}$$
$$dim(\hat{\mathcal{P}}^{div_1}) = 3, \text{ pentru } d = 2$$
$$dim(\hat{\mathcal{P}}^{div_1}) = 4, \text{ pentru } d = 3,$$
$$(4.117)$$

iar în coordonate baricentrice:

$$\hat{\mathcal{P}}^{div_1} = \left\{ \hat{\omega}_i^{div_1} \mid \hat{\omega}_i^{div_1} \in [P_{k-1}]^d \oplus \tilde{P}_{k-1}\mathbf{x}, \ 1 \le i \le m \right\}$$

cu $\omega_i^{div_1}$ dat de (4.113) pentru $d = 2, \ m = 3$, respectiv
 $\omega_i^{div_1}$ dat de (4.114) pentru $d = 3, \ m = 4$
și *i* indexul local al laturilor, respectiv fețelor elementului master. (4.118)

Elementele din (4.117) corespund elementelor Whitney de ordin 2 ([156], [42]).

Ținând cont de faptul că:

$$\hat{\lambda}_{1} = 0 \text{ pe } \hat{\mathcal{F}}_{1} \text{ și}$$

$$\sum_{i=1}^{4} \lambda_{i} = 1 \text{ pe } \hat{T} \Rightarrow \sum_{i=2}^{4} \lambda_{i} = 1 \text{ pe } \hat{\mathcal{F}}_{1} \text{ și}$$

$$\nabla \lambda_{1} = -\frac{\vec{n}_{1}}{|h_{1}|},$$
(4.119)

și înlocuind relațiile (4.119) în (4.114), obținem pentru componenta funcției de formă $\omega_1^{div_1}$ după normala exterioară feței $\hat{\mathcal{F}}_1$:

$$\hat{\omega}_1^{div_1} \cdot \vec{n}_1 = \frac{2}{|\nabla \lambda_1|} (\nabla \lambda_2 \times \nabla \lambda_4) \cdot \nabla \lambda_3.$$
(4.120)

Deoarece direcția vectorului $\nabla \lambda_i$ este normală pe fața $\hat{\mathcal{F}}_i$, respectiv direcția $\nabla \lambda_j$ normală pe $\hat{\mathcal{F}}_j$, deducem că vectorul ($\nabla \lambda_i \times \nabla \lambda_j$) este paralel cu latura comună a fețelor $\hat{\mathcal{F}}_i$ și $\hat{\mathcal{F}}_j$ (fig. 4.13). Pentru exemplul nostru, cu orientările laturilor din fig. 4.12, sensul vectorului ($\nabla \lambda_2 \times \nabla \lambda_4$) este cel al vectorului $\overrightarrow{\hat{P}_1 \hat{P}_3}$. Se poate ușor demonstra că:

$$\nabla\lambda_2 \times \nabla\lambda_4 = \frac{\hat{P}_1 \hat{P}_3}{6\mathcal{V}_{\hat{T}}},\tag{4.121}$$

iar din (4.53) rezultă:

$$\frac{1}{|\nabla\lambda_1|} = \frac{3\mathcal{V}_{\hat{T}}}{\hat{\mathcal{A}}_1},\tag{4.122}$$

unde $\mathcal{V}_{\hat{T}}$ și $\hat{\mathcal{A}}_1$ sunt volumul tetraedrului \hat{T} , respectiv aria feței $\hat{\mathcal{F}}_1$. In plus, se poate observa că:

$$\overrightarrow{\hat{P}_1\hat{P}_3}\nabla\lambda_3 = \frac{\overrightarrow{\hat{P}_1\hat{P}_3}\cdot(-\vec{n}_3)}{h_3} = 1.$$
(4.123)

Inlocuind relațiile de mai sus în (4.120) rezultă:

$$\hat{\omega}_1^{div_1} \cdot \vec{n}_1 = \frac{6\mathcal{V}_{\hat{T}}}{\hat{\mathcal{A}}_1} \frac{\overrightarrow{\hat{P}_1\hat{P}_3}\nabla\lambda_3}{6\mathcal{V}_{\hat{T}}} = \frac{1}{\hat{\mathcal{A}}_1}.$$
(4.124)

Similar, pentru cazul 2D:

$$\omega_1^{div_1} \cdot \vec{n}_1 = (\nabla \lambda_3 \times \nabla \lambda_2) \frac{\vec{k}}{|\nabla \lambda_1|} = \frac{\vec{k}}{2\mathcal{A}_{\hat{T}}} \cdot \vec{k} \frac{2\mathcal{A}_{\hat{T}}}{|\vec{\hat{P}}_2 \vec{\hat{P}}_3|} = \frac{1}{|\hat{\mathcal{L}}_1|}.$$
(4.125)



Figura 4.13: (a)- Pe triunghiul master, vectorii câmpului $\omega_1^{div_1}$ sunt coliniari cu laturile vecine cu latura \mathcal{L}_1 , având componentă normală nulă pe acestea; în vertexurile \hat{P}_2 și \hat{P}_3 se poate observa relația de ortogonalitate între funcțiile de formă din $H(\mathbf{rot})$ și cele din H(div); divergența câmpului $\omega_1^{div_1}$ este constantă pe triunghi fiind egală cu $1/\hat{\mathcal{A}}$. (b) - Pe tetraedrul master, vectorii câmpului $\omega_1^{div_1}$ sunt coliniari cu laturile ce se sprijină pe vertexurile feței \mathcal{F}_1 , având componentă normală nulă pe fețele subântinse de acestea; divergența câmpului $\omega_1^{div_1}$ este constantă pe tetraedru fiind egală cu $1/3\mathcal{V}_{\hat{T}}$.

Deci fluxul funcției de formă $\omega_1^{div_1}$ pe fața $\hat{\mathcal{F}}_1$:

$$\iint_{\hat{\mathcal{F}}_1} \hat{\omega}_1^{div_1} \cdot \vec{n}_1 \, dA_1 = 1, \tag{4.126}$$

respectiv pe latura $\hat{\mathcal{L}}_1$ din triunghi:

$$\int_{\hat{\mathcal{L}}_1} \hat{\omega}_1^{div_1} \cdot \vec{n}_1 \, dl_1 = 1. \tag{4.127}$$

In general:

$$\iint_{\hat{\mathcal{F}}_j} \hat{\omega}_i^{div_1} \cdot \vec{n}_j \, dA_j = \delta_{ij},\tag{4.128}$$

$$\int_{\hat{\mathcal{L}}_j} \hat{\omega}_i^{div_1} \cdot \vec{n}_j \, dl_j = \delta_{ij}. \tag{4.129}$$

In concluzie, $\omega_i^{div_1}$ reprezintă un câmp vectorial cu componentă normală constantă și flux unitar pe fața $\hat{\mathcal{F}}_i$ și componente normale nule (flux nul) pe celelalte fețe ale tetraedrului de referință (v. fig. 4.13). Componenta normală de pe fața $\hat{\mathcal{F}}_i$ este continuă la trecerea prin această suprafață fiind egală cu inversul ariei acesteia.

Din cele de mai sus rezultă faptul că funcțiile de formă $\hat{\omega}_i^{div_1}$, din spațiul $\hat{\mathcal{P}}^{div_1}$ au componenta normală continuă la trecerea prin oricare față de frontieră dintre două tetradere vecine. O logică asemănătoare se aplică și pentru elementele 2D triunghiulare, acolo fiind vorba despre continuitatea componentelor normale ale funcțiilor de formă la trecerea prin orice latură de frontieră dintre două triunghiuri vecine (v. fig, 4.13 - (a)). Prin urmare condiția (4.39) de conformitate H(div) este îndeplinită.

Ținând cont de (4.128), variabila nodală asociată elementului nodal $\hat{\omega}_i^{div_1}$, este aleasă a fi în acest caz, fluxul funcției vectoriale necunoscute \vec{v} pe fața (pentru domenii 2D - latura) *i*:

$$\hat{N}_{i}^{div_{1}} : \vec{\hat{v}} \to \iint_{\hat{\mathcal{F}}_{i}} \vec{\hat{v}} \cdot \vec{n}_{i} \, dA_{i}, \text{ cu } 1 \le i \le m.$$

$$(4.130)$$

Putem deci exprima funcția necunoscută, la nivel de element master, prin combinația liniară:

$$\vec{\hat{v}} = \sum_{i=1}^{m} \hat{N}_{i}^{div_{1}} \hat{\omega}_{i}^{div_{1}}$$
(4.131)

Dacă impunem $\hat{v} = 0$, rezultă și anularea componentelor normale ale \vec{v} pe oricare din fețele elementului master: $\vec{v} \cdot \vec{n}_j = 0$, $\forall 1 \leq j \leq m$; apoi, luând în considerare (4.124), rezultă:

$$\sum_{i=1}^{m} \hat{N}_{i}^{div_{1}} \hat{\omega}_{i}^{div_{1}} \cdot \vec{n}_{j} = \hat{N}_{j}^{div_{1}} \frac{1}{\hat{\mathcal{A}}_{j}} = 0 \Leftrightarrow \hat{N}_{j}^{div_{1}} = 0, \, \forall \, 1 \le j \le m.$$
(4.132)

In consecință este demonstrată implicația de unisolvență (4.40).

In concluzie, tripletul de referință $\{\hat{T}, \hat{\mathcal{P}}^{div_1}, \hat{\mathcal{N}}_I^{div_1}\}$ reprezintă un element finit conform și unisolvent în $H(div, \hat{T})$.

La fel ca și în cazul elementelor finite conforme în $H(\mathbf{rot}, \Omega)$, pentru determinarea elementului finit din domeniul fizic, trebuie aplicată elementelor tripletului de mai sus transformarea afină (4.58), într-o formă la care operatorul ∇ · rămâne invariant. In [114] (Lema 3.59) se demonstrează că transformarea potrivită acestui scop este cea cunoscută sub numele de *transformarea Piola* ([136]):

$$\vec{u} = \frac{\mathbf{F}}{J_k} \vec{\hat{u}} \circ \Phi_k^{-1}, \tag{4.133}$$

unde notațiile sunt cele folosite în cap. 4.1.2, iar $\vec{u} \subset H(div, \Omega_k)$ și $\hat{\vec{u}} \in H(div, \hat{T})$. Rezultă următoarele relații de transformare care lasă invariante relațiile definite pe domeniul \hat{T} , al elementului master: Capitolul 4. Calcul numeric

(a). divergență:

$$\nabla \cdot \vec{u} = \frac{1}{J_k} \nabla \cdot \vec{\hat{u}} \circ \Phi_k^{-1}; \qquad (4.134)$$

(b). componenta normală pe o față de element:

$$(\vec{u} \cdot \vec{n}) = \frac{1}{J_k \parallel (\mathbf{F}_k^T)^{-1} \vec{\hat{n}} \parallel}_{|_{\mathcal{F}}} \left(\vec{\hat{u}} \cdot \vec{\hat{n}} \right) \circ \Phi_{k|_{\hat{\mathcal{F}}}}^{-1};$$
(4.135)

(c). Operatorul de integrare pe suprafață:

$$\iint_{\mathcal{F}} dA = \iint_{\hat{\mathcal{F}}} |J_k| \parallel (\mathbf{F}_k^T)^{-1} \hat{\vec{n}} \parallel dA;$$
(4.136)

(d). Operatorul produs vectorial, respectiv relația $\nabla \hat{\varphi} \times \nabla \hat{\psi}$ cu $\hat{\varphi}, \hat{\psi} \in H^1(\hat{T})$:

$$\nabla \times \vec{u} = \frac{\mathbf{F}}{J_k} \nabla \times \vec{\hat{u}} \circ \Phi_k^{-1}, \text{ respectiv}$$
(4.137)

$$\nabla \varphi \times \nabla \psi = \frac{\mathbf{F}}{J_k} (\nabla \hat{\varphi} \times \nabla \hat{\psi}) \circ \Phi_k^{-1}.$$
(4.138)

Din relațiile (4.137), rezultă prin aplicarea transformării (4.133), funcțiile de formă corespunzătoare elementului finit real, de ordin unu, H(div) conform. Pentru exemplul din (4.114):

$$\omega_1^{div_1} = \frac{\mathbf{F}}{J_k} \hat{\omega}_i^{div_1} \circ \Phi_k^{-1} =$$

$$2 \left[\lambda_2 (\nabla \lambda_4 \times \nabla \lambda_3) + \lambda_4 (\nabla \lambda_3 \times \nabla \lambda_2) + \lambda_3 (\nabla \lambda_2 \times \nabla \lambda_4) \right],$$
(4.139)

cu proprietatea similară cu (4.128):

$$\iint_{\mathcal{F}_j} \omega_i^{div_1} \cdot \vec{n}_j \, dA_j = \iint_{\hat{\mathcal{F}}_j} |J_k| \parallel (\mathbf{F}_k^T)^{-1} \frac{1}{J_k \parallel (\mathbf{F}_k^T)^{-1} \vec{\hat{n}} \parallel} \hat{\omega}_i^{div_1} \cdot \vec{n}_j \, dA_j = sign(J_k) \, \delta_{ij}$$

$$\tag{4.140}$$

La fel, în virtutea relației (4.136), rezultă invarianța gradelor de libertate ale elementelor div conforme alese pentru elementul de referință (cf. (4.130)):

$$N_j^{div_1} : \vec{v} \to \iint_{\mathcal{F}_j} \vec{v} \cdot \vec{n}_j, dA_i \text{ cu } 1 \le i \le m.$$
(4.141)

unde j este indexul global al fețelor din Ω_h . Soluția aproximativă locală este descrisă de funcții independente, descrise de operatorul de interpolare local:

$$\vec{u}_{h}^{k}(\mathbf{x}) = \Pi_{k}^{div}(\vec{v}) = \sum_{i=1}^{m} N_{i}^{k_{div_{1}}}(\vec{v})\omega_{i}^{div_{1}}(\mathbf{x}), \qquad (4.142)$$

unde *i* este indexul local al fețelor din Ω_k . In concluzie, putem descrie elementul finit Raviart-Tomas de ordin 1, prin următorul triplet:

- 1. Domeniul spațial Ω_k
- 2. Spațiul funcțiilor de formă \mathcal{P}^{div_1} :
 - $\mathcal{P}^{div_1} = \left\{ \omega_i^{div_1} \mid \omega_i^{div_1} \in \left([P_{k-1}]^d \oplus \tilde{P}_{k-1} \mathbf{x} \right) (\Omega_k), \ 1 \le i \le m \right\}$ cu $\omega_i^{div_1}$ dat de expresii de forma (4.113) pentru $d = 2, \ m = 3$, respectiv $\omega_i^{div_1}$ dat de expresii de forma (4.114) pentru $d = 3, \ m = 4$ (4.143)
- 3. Spațiul gradelor de libertate \mathcal{N}^{div_1} :

$$\mathcal{N}^{div_1} = \left\{ N_i^{k_{div_1}} \mid N_i^{k_{div_1}} = \int_{\mathcal{F}_i} \vec{v}(\mathbf{x}) \cdot \vec{n}_i \, dA_i, \, 1 \le i \le m \right\}.$$
(4.144)

Soluția aproximativă globală este dată de operatorul de interpolare descris în (4.28):

$$\vec{u}_h = \Pi^{div_1}(\vec{u}) = \sum_{j=1}^{\mathrm{F}} N_j^{div_1}(\vec{u})\omega_j^1(\mathbf{x}), \qquad (4.145)$$

cu

$$\Pi^{div_1} : H(div, \Omega) \to \mathcal{Q}_h^1(\Omega_h), \tag{4.146}$$

unde \mathcal{Q}_h^1 reprezintă spațiul discret al soluțiilor aproximative, obținute prin discretizarea problemei (4.1.1) cu elemente finite Raviart-Thomas de ordin 1:

$$\mathcal{Q}_{h}^{1} = \left\{ \vec{u}_{h}(\mathbf{x}) \in H(div, \Omega) \mid \vec{u}_{h}^{k}(\mathbf{x}) \in D_{k}(\Omega_{k}) \subset H(div, \Omega_{k}), \forall \Omega_{k} \in \mathcal{T}_{h} \right\}.$$
(4.147)

4.1.7 Elementul finit conform în L^2

Elementul finit conform în $L^2(\Omega)$, este definibil numai pentru domeniile $\Omega \subset \mathbb{R}^3$. Funcția de formă asociată, notată cu $\omega_i^{0_1}$ este derivată, din cea a elementului conform în H(div), prin aplicarea operatorului $div(\nabla \cdot)$, urmând secvența exactă de Rham. Pentru tetraedrul de referință \hat{T} se poate scrie, în funcție de orice funcție de formă definită pe o față $\hat{\mathcal{F}}_j$ (cf. (4.118)) a acestuia:

$$\hat{\omega}^{0_1} = \nabla \cdot \hat{\omega}_j^{div_1} = \frac{3}{3\hat{\mathcal{V}}_T} = \frac{1}{\hat{\mathcal{V}}_T}.$$
(4.148)

de unde proprietatea:

$$\iiint_{\hat{T}} \hat{\omega}^{0_1} \, dv_i = 1. \tag{4.149}$$

Prin urmare, funcția de formă $\hat{\omega}^{0_1}$ este o constantă reală, fiind deci un membru al mulțimii polinoamelor reale de ordin 0 - $P_0(\hat{T}) \subset L^2(\hat{T})$. Trecerea de la elementul master la cel fizic se poate face folosind transformarea Piola (4.133).

$$\omega^{0_1} = J_k^{-1} \hat{\omega}^{0_1} \circ \Phi^{-1}, \text{ cu } \omega^{0_1} \in P_0(\Omega_k) \subset L^2 \Omega_k.$$

$$(4.150)$$

Elementul fizic are următoarea proprietate:

$$\iiint_{\Omega_i} \omega_j^{0_1} dv_i = \iiint_{\Omega_i} \nabla \cdot \omega_j^{div_1} dv_i = \iint_{\partial\Omega_i} \omega_j^{div_1} dA_i = \sum_{m=1}^{F_i} \iint_{\mathcal{F}_m} \omega_j^{div_1} dA_i = \delta_{ij}.$$
(4.151)

Această funcție de formă corespunde elementului Whitney de ordin 3 ([156], [42]) și nu este continuă la trecerea de la un element la altul. Variabila nodală asociată elementului nodal este integrala de volum a funcției de formă interpolată cu ajutorul unei funcții polinomiale de grad l:

$$\mathcal{N}_{j}^{0_{l}}: \varphi \to \iiint_{\Omega_{k}} \varphi p_{l} dv, \text{ cu } l \in \mathbb{N}$$

$$(4.152)$$

Pentru ordinul 1, $p_0 = 1$, iar elementul finit conform în $L^2(\Omega_k)$ este definit de următorul triplet:

- 1. Domeniul spațial Ω_k ;
- 2. Spațiul funcțiilor de formă \mathcal{P}^{0_1} :

$$\mathcal{P}^{0_1} = \{ \omega^{0_1} \, | \, \omega_k^{0_1} \in P_0(\Omega_k), \, \omega_k^{0_1} = \frac{1}{\mathcal{V}_k} \}, \tag{4.153}$$

3. Spațiul gradelor de libertate \mathcal{N}^{0_1} :

$$\mathcal{N}^{0_1} = \left\{ N_k^{0_1} \,|\, N_k^{0_1} : \psi \to \iiint_{\mathcal{V}_k} \psi \, dv \right\}, \tag{4.154}$$

unde \mathcal{V}_k este volumul tetraedrului Ω_k .

Soluția aproximativă globală este dată de către funcția de interpolare:

$$\psi_h = \Pi^{0_1}(\varphi) = \sum_{j=1}^T N_j^{0_1}(\varphi) \omega_j^{0_1}, \qquad (4.155)$$

cu

$$\Pi^{0_1} : L^2(\Omega) \to \mathcal{S}_h^1(\Omega_h) \tag{4.156}$$

unde S_h^1 reprezintă spațiul discret al soluțiilor aproximative, obținute prin discretizarea problemei (4.1.1) cu elemente finite ordin 1 din L^2 :

$$\mathcal{S}_{h}^{1} = \left\{ \varphi \in L^{2}(\Omega) \mid \varphi_{h}^{k} \in P_{0}(\Omega_{k}) \subset L^{2}(\Omega_{k}), \forall \Omega_{k} \in \mathcal{T}_{h} \right\}.$$

$$(4.157)$$

4.1.8 Complexul De Rham discret

Din cele de mai sus, se poate demonstra ([42]-Cap.5), că spațiile funcționale ale elementelor conforme, (4.75), (4.109), (4.147), (4.157), pot fi așezate într-o secvență De Rham (v. (3.128)) exactă, în raport cu operatorii **grad**, **rot**, **div**, în condițiile în care domeniul Ω este simplu conex și mărginit de o frontieră conexă. Pentru domeniile în 3 dimensiuni, secvența este următoarea:

$$\mathbb{R} \to \mathcal{W}_h^1 \xrightarrow{\nabla} \mathcal{V}_{h,I}^1 \xrightarrow{\nabla \times} \mathcal{Q}_h^1 \xrightarrow{\nabla} \mathcal{S}_h^1 \to \{0\}.$$
(4.158)

Trebuie însă remarcat că operatorii de interpolare globală Π^{H1} (4.74), Π^{rot} (4.108), Π^{div} (4.145), respectiv Π^{0_1} (4.155) nu sunt bine definiți pe spațiile $H^1(\Omega)$, $H(\text{rot}, \Omega)$, $H(div, \Omega)$, $L^2(\Omega)$ ([114]), întrucât condițiile de frontieră își găsesec corespondentul funcțional în spații Sobolev fracționare (v. operatorii de urmă (3.86), (3.106), (3.124)). Ca urmare, în scopul de a reprezenta imaginea secvenței exacte De Rham pentru problema (4.1.1) discretizată, prin folosirea funcțiilor de interpolare specificate mai sus, trebuie pornit de la spații pe care aceste interpolări sunt bine definite:

a) Teorema 5.49 din [114] stabilește spațiul pe care operatorul de interpolare Π^{H_1} este bine definit, și anume:

$$W = H^{\frac{3}{2} + \delta}(\Omega) \subset H^1(\Omega), \quad \delta > 0, \tag{4.159}$$

$$\Pi^{H1}: W \to \mathcal{W}_b^1. \tag{4.160}$$

Conform secvenței exacte (4.158):

$$Im(\nabla \mathcal{W}_h^1) = ker(\nabla \times \mathcal{V}_{h,I}^1). \tag{4.161}$$

de unde:

$$\nabla \left(\Pi^{H1}(W) \right) = \Pi^{\mathbf{rot}} \nabla(W), \qquad (4.162)$$

cu $\Pi^{\mathbf{rot}}$ dat de (4.164).

b) Lema 5.38 din [114] stabilește spațiul pe care operatorul de interpolare Π^{rot} este bine definit, și anume:

$$V = \left\{ \vec{u} \in \left[H^{\frac{1}{2} + \delta}(\Omega) \right]^3 \subset H(\mathbf{rot}, \Omega), \ | \nabla \times \vec{u} \in \left[H^{\frac{1}{2} + \delta}(\Omega) \right]^3, \ \delta > 0 \right\},$$

$$(4.163)$$

$$\Pi^{\mathbf{rot}} : V \to \mathcal{V}^1_{h,I}.$$

$$(4.164)$$

Conform secvenței exacte (4.158):

$$Im\left(\nabla \times \mathcal{V}_{h,I}^{1}\right) = ker(\nabla \cdot \mathcal{Q}_{h}^{1}).$$
(4.165)

de unde:

$$\nabla \times \left(\Pi^{\mathbf{rot}}(V)\right) = \Pi^{div}(\nabla \times (V)), \qquad (4.166)$$

iar Π^{div} este dat de (4.168).

c) Lema 5.15 din [114] stabilește spațiul pe care operatorul de interpolar
e Π^{div} este bine definit, și anume:

$$Q = \left\{ \vec{u} \in \left[H^{\frac{1}{2} + \delta}(\Omega) \right]^3 \subset H(div, \Omega), \, | \, \nabla \cdot \vec{u} \in L^2(\Omega), \, \delta > 0 \right\}, \quad (4.167)$$
$$\Pi^{div} : \, Q \to \mathcal{Q}_b^1. \tag{4.168}$$

$$^{div}: Q \to \mathcal{Q}_h^1. \tag{4.168}$$

Conform secventei exacte (4.158):

$$Im\left(\nabla \cdot \mathcal{Q}_{h}^{1}\right) = ker(L^{2}(\Omega)).$$
(4.169)

din care:

$$\nabla \cdot \left(\Pi^{div}(Q)\right) = \Pi^{0_1}(\nabla \cdot Q),$$

si $\Pi^{0_1}: S \to \mathcal{S}_h^1$ este bine definit pe $S = L^2(\Omega)$ (v.(4.156)).
(4.170)

Ca urmare, secvența (4.158) se poate scrie astfel (pentru claritatea schemei, nu am mai mentionat capetele secvenței):

Pentru domeniile 2D disctretizate, secvența exactă (4.158) poate căpăta 2 forme, derivate din formele (3.140) cu aceleași limitări de mai sus, din cazul domeniilor 3D discretizate: ∇

$$\mathbb{R} \to \mathcal{W}_{h}^{1} \xrightarrow{\nabla_{2}} \mathcal{V}_{h,I}^{1} \xrightarrow{\nabla_{2}\times} \mathcal{S}_{h}^{1} \to \{0\},$$

$$\mathbb{R} \to \mathcal{W}_{h}^{1} \xrightarrow{\overrightarrow{\nabla}_{2}\times} \mathcal{Q}_{h,I}^{1} \xrightarrow{\overrightarrow{\nabla}_{2}\cdot} \mathcal{S}_{h}^{1} \to \{0\}.$$
(4.172)

In [159] este demonstrată o metodă de construcție a elementelor finite de grad superior, conforme cu spațiile $H^1(\Omega)$, $H(\mathbf{rot}, \Omega)$, $H(div, \Omega)$, $L^2(\Omega)$. Procedura pornește de la ideea alegerii spațiilor funcționale ale elementelor generate, astfel încât acestea să formeze o secvență exactă (4.171) atât la nivel local, de element de referință, dar și la nivel global al întregului spațiu discretizat. Stabilirea acestor spații funcționale polinomiale aflate în secvență exactă nu mai necesită apoi generarea separată a spațiilor funcțiilor de formă. Este suficientă generarea unor funcții polinomiale generate de o câte o bază din spațiile respective. In acest mod, se pot construi rețele de discretizare neuniforme, adaptate dinamic necesităților de îmbunătățire a performanțelor de timp și precizie ale algoritmului numeric de element finit.

Aceste optimizări se pot efectua atât prin mărirea ordinului polinoamelor de interpolare (metoda p), dar și prin rafinarea locală a rețelei de discretizare (metoda h) fără a avea obligativitatea de a efectua aceste rafinări în mod uniform. De exemplu, este posibilă utilizarea unei discretizări cu elemente triunghiulare, pentru care se pot defini elemente de muchie de ordin diferit pentru fiecare latură, în condițiile în care distribuția ordinului polinomial adoptat pentru laturi satisface anumite reguli derivate din ideea ca spațiile polinomiale folosite să păstreze proprietatea de secvență exactă ([159]-Teorema 5.32 și observația 5.33). Dezvoltări similare, se pot găsi și în [60], [59].

4.1.9 Concluzii referitoare la aspectele numerice

Se poate concluziona că pentru rezolvarea problemelelor fizice complexe, descrise de ecuații cu derivate parțiale este necesară folosirea metodelor numerice, dintre care, cea mai populară este Metoda Elementelor Finite. Aceasta se bazează pe formularea slabă a problemei matematice și pe aproximarea spațiului în care se calcuează soluția cu un spațiu discret de dimensiune finită, izomorf cu cel al funcțiilor de test, numit spațiul funcțiilor de formă.

Pentru a genera aceste spații se descompune domeniul de calcul într-o multime finită de celule de forme geometrice simple, cel mai frecvent triunghiuri sau tetraedre. Finețea discretizării spațiale este descrisă de parametrul h, numit norma rețelei, reprezentând dimensiunea maximă a celulelor. Soluția se aproximează în fiecare celulă cu un polinom de grad p, cel mai simplu fiind cazul în care p = 1. Valorile soluției numerice în cele N noduri ale rețelei de discretizare alcătuiesc gradele de libertate ale problemei, și determină în mod univoc soluția numerică, soluție ce păstrează proprietatea de continuitate pe întreg domeniul de calcul. Spațiul gradelor de libertate, dual al spațiului funcțiilor de formă, este un element esențial al metodei. Soluția numerică, reprezentată prin valorile gradelor de libertate, se obține prin rezolvarea unui sistem de N ecuații algebrice liniare, sistem generat din forma slabă discretă $\mathcal{A}(u_h, v_h) = \mathcal{F}(v_h)$. Fiecare element contribuie la matricea sistemului în pozițiile determinate de indecșii globali - după care se numără gradele de libertate - ai variabilelor nodale asociate fiecărui element, și de indexul global al elementului care îl identifică în mulțimea elementelor rezultate din discretizarea domeniului asociat problemei. Cu cât h este mai mic, cu atât N este mai mare și soluția numerică se apropie mai mult de cea exactă.

Pentru a standardiza implementarea metodei elementelor finite se folosește un element master, de referință, care este apoi mapat pe fiecare celulă din rețea folosind o transformare afină.

Funcționala biliniară $\mathcal{A}(.,.)$ este exprimată ca produs scalar al funcțiilor de test și de încercare, adică integrala, pe domeniul de calcul, din constanta de material înmulțită cu produsul a doi operatori diferențiali (*grad* în cazul problemelor scalare de tip div - grad, sau **rot** în cazul problemelor vectoriale de tip **rot-rot**) aplicați funcțiilor de formă de test, respectiv de încercare. Pentru calculul numeric al integralei se consideră fiecare celulă omogenă și se aplică metoda de integrare Gauss, care dă rezultate exacte în cazul polinoamelor. Fiecare element contribuie la termenii liberi, cu o integrală pe element estimată numeric tot cu metoda Gauss, din sursele de câmp proiectate pe funcțiile de test, surse asociate termenilor liberi corespnzători nodurilor elementului.

Prin alegerea corectă a cadrului funcțional, soluția numerică obținută este grad-, rot-, sau div-conformă, după cum trebuie să fie câmpul fizic ce se dorește a fi calculat. Forma diferențială (de ordin 1, 2 sau 3) caracteristică mărimii locale prin care se descrie problema fizică, reprezintă legătura esențială între aspectele fizice, cele matematice și cele numerice.

Capitolul 5

Aspecte computationale practice. Studiul unui dispozitiv MEMS

5.1 Dispozitive MEMS

Dispozitivele MEMS (Micro Electro Mechanical Systems) au apărut din necesitatea de a adăuga circuitelor integrate posibilitatea de a avea funcții de comutare și/sau de traductoare ale unor mărimi neelectrice (mișcare în 3D, vibrație, presiune, iluminare, temperatură) în mărimi electrice. Cele mai frecvente variante constructive presupun existența unor microelemente mobile, care se deformează sau se deplasează fie sub influența acțiunii mărimilor neelectrice, fie sub acțiunea unor forțe electrice - de regulă electrostatice. In mod curent, dispozitivele MEMS au aplicații în domeniile IT (imprimante, hard-discuri) [95], medical [31], telefonie mobilă [64], aplicații RF [134], aplicații militare [32].

Pentru proiectarea, apoi execuția dispozitivelor MEMS este necesară o înțelegere cât mai exactă a proceselor fizice ce se desfășoară în acestea. Chiar și pentru configurațiile simple, respectiva analiză nu este ușoară datorită în special dimesiunilor reduse ale domeniului de simulat (de ordinul $10^{-6} \div 10^{-4}m$), domeniu în care trebuie să coexiste mai multe fenomene fizice: electrostatic, elastic, termic, curgere a fluidelor, vibrații ale unor membrane elastice; în plus legile ce guvernează aceste fenomene au un mod particular de aplicare datorită interacțiunilor ce apar între fenomenele susamintite, într-un spațiu atât de restrâns: de exemplu forța de gravitație este neglijabilă față de cea de atracție electrostatică, sau față de influența fluidului în care se mișcă membranele mobile ale MEMS [134].

Ca urmare, tehnicile de modelare multifizică își găsesc un teren foarte bun de aplicare în domeniul simulării funcționării acestor microsisteme electromecanice. Sistemele de ecuații sunt în cele mai multe cazuri neliniare trebuind totodată să surprindă stânsele cuplaje dintre diferitele fenomenele fizice ce descriu stările statice sau dinamice ale unui dispozitiv MEMS. Modelările nu trebuie numai să descrie cu acuratețe funcționarea unui MEMS, dar ar trebui să îndeplinească și rolul de predictor al evoluției în timp a stării componentelor acestuia, cu precădere a celor mobile (membrane, lamele, etc.), stabilind astfel jaloanele duratelor de viață și de funcționare sigură ale dispozitivului [102].

Pentru modelarea multifizică a MEMS, pot fi abordate tehnici la nivel sistemic, pe baza cărora se pot face simulări la nivel funcțional în scopul de a determina comportamentul general al microsistemului în funcție de excitațiile tipice care pot fi aplicate acestuia pe durata funcționării sale: de exemplu, se pot determina limitele unor solicitări mecanice, electrice sau termice la care sunt supuse componentele unui MEMS. Mult mai precise sunt însă simulările care se efectuează la nivel fizic, și care pornesc de la scrierea ecuațiilor ce caracterizează fenomenele fizice ce descriu funcționarea dispozitivului, precum și a cuplajelor ce se stabilesc între acestea. Sistemele de ecuații rezulate sunt cu derivare parțiale, și, de regulă, pentru rezolvarea acestora pe domenii bi sau tridimensionale se utilizează metoda elementelor finite (MEF) [112].
5.2 Un model simplu de microcomutator capacitiv RF

Dispozitivele MEMS au căpătat o largă întrebuințare în transmisia și prelucrarea semnalelor de RF cu frecvențe de până la 100Ghz în special datorită pierderilor mici și dimensiunilor reduse pe care le au în comparație cu circuitele clasice cu componente discrete sau integrate hibrid [81],[112]. Unul dintre cele mai răspândite modele de microcomutator RF este ghidul de undă coplanar - prescurtat CPW (Coplanar Waveguide). In fig. 5.1 se regăsește imaginea de principiu a unui astfel de dispozitiv MEMS.



Figura 5.1: Dispozitiv MEMS de tip CPW. Semnalul RF trece prin bara centrală; aceasta este încadrată de două bare de masă (legate la potențialul de referință al întregului circuit). Câmpul electromagnetic se propagă prin conductorul central evoluând între frontierele laterale conectate la potențialul de referință. Barele conductoare se obțin prin depunerea de material conductor (de regulă Au) pe aceeași parte a unui substrat de Si. In scopul reducerii pierderilor în substrat, între acesta și straturile conductoare se depune un strat izolator foarte subțire.

Ghidul de undă CPW poate fi transformat într-un comutator RF odată ce între barele laterale se depune o lamelă conductoare elastică ce poate fi atrasă electrostatic spre bara centrală. Acest fapt se obține prin aplicarea unei diferențe de potențial continuă între conductorul central și barele laterale aflate în contact direct cu lamela mobilă (v. fig.5.2). Peste electrodul central este depus un strat dielectric (izolator). Practic, dacă tensiunea V_a nu este aplicată, capacitatea dintre lamelă și electrodul central este minimă (C_{min}), stabilind o anumită frecvență de tăiere pentru semnalele RF ce trec prin circuitul ce conține acest comutator. La aplicarea unei valori critice a tensiunii, numită tensiune de minimă acționare (în limba engleză "pull in") și notată în continuare cu V_{PI} , lamela atinge stratul dielectric depus peste electrodul central, capacitatea astfel formată fiind maximă, fapt ce duce la reducerea impedanței capacitive a dispozitivului, prin urmare la modificarea benzii de trecere a semnalelor RF prin circuit.



Figura 5.2: Principiul de funcționare al unui șunt capacitiv realizat cu un microcomutator integrat. Șuntul este deschis atunci când nu este aplicată tensiunea continuă de comandă V_a (a); Șuntul se închide complet dacă se aplică tensiunea critică $V_a = V_{PI}$; se poate observa că valoarea capacității (deci și a frecvenței de tăiere) poate fi controlată în tensiune atâta timp cât $0 < V_a \leq V_{PI}$.

In fig.5.3 sunt prezentate două variante de șunturi capacitive în construcție CPW.



Figura 5.3: (a) - șunt capacitiv produs de Raytheon (imagine preluată din [133]); (b) - șunt capacitiv produs de Analog Devices (imagine preluată din [81]).

Lățimea barelor laterale și a celei centrale este de $120\mu m$, iar distanța dintre bare este de câte $80\mu m$; membrana elastică este perforată cu găuri de $2\mu m$ utile atât în procesul tehnologic, dar și funcțional - ușurând deplasarea membranei în mediul de gaz inert în care se află întreg micromecanismul. Capacitatea minimă a unui astfel de șunt este de 20-50 fF, iar cea maximă de 3-5pF ([133]).



Figura 5.4: Reprezentarea de principiu a funcționării a unui dispozitiv MEMS cu acționare electrostatică. Figura de față a fost obținută prin prelucrarea unei imagini din [81].

Principial, un dispozitiv MEMS electrostatic poate fi reprezentat ca un condensator plan format dintr-o armătură fixă și una mobilă depărtată inițial de cea fixă cu distanța g_0 (v. fig.5.4). Armătura fixă este conectată la potențialul de referință (masă), iar cea mobilă la un potențial de comandă V_a față de referință. Dacă tensiunea de comandă este mai mică decât valoarea critică V_{PI} (tensiune de "pull-in") armătura efectuează o deplasare u după care rămâne stabil în acea poziție. Daca tensiunea de comandă este mai mare decît V_{PI} atunci armătura mobilă se lipește de cea fixă. Ideal, tensiunea de acționare ar trebui să fie egală cu V_{PI} .

5.3 Problema 1D pentru microșuntul electrostatic.

Problema este una cuplată, de electrostatică și elasticitate liniară, care pot fi descrise pe domeniile reprezentate în fig. 5.5. Sub influența forței electrostatice, armătura



Figura 5.5: Domeniile geometrice ale modelului 1D pentru un microșunt rezistiv. Ambele armături au aceeași arie $A = l_b \times w$.

mobilă este atrasă spre armătura fixă. Dacă cele două armături au suprafețe conductoare avem de a face cu un șunt rezistiv. În cazul în care pe suprafața conductoare a armăturii fixe este depus un strat dielectric izolator atunci șuntul este capacitiv. Pentru simplitate problema va fi studiată pentru cazul rezistiv.

Domeniul Ω_E a problemei electrostatice este descris de dreptunghiul format de armăturile plan paralele ale condensatorului. Armătura mobilă a acestuia evoluează de la distanța inițială g_0 față de armătura fixă, până la o distanță g determinată de tensiunea de comandă V_a aplicată între cele două armături:

$$\Omega_E = [0, l_b] \times [0, g]. \tag{5.1}$$

Problema electrostatică este una plan paralelă, în care potențialul electric are linii echipotențiale paralele cu armăturile. Dielectricul dintre cele două armături este aerul. Formularea matematică a acestei probleme este următoarea:

In domeniul Ω_E se caută soluția ecuației:

$$\frac{\partial^2 v}{\partial y^2} = 0, \text{ cu } v \in C^2(\Omega), \tag{5.2}$$

cu condițiile de frontieră:

$$v(y) = V_a, \text{ pentru } y = 0,$$

$$v(y) = 0, \text{ pentru } y = g,$$

$$\frac{\partial v}{\partial n} = 0, \text{ pentru } x = 0, x = l_b$$
(5.3)

Pentru un g fixat, ecuația (5.2) are soluția analitică:

$$v(y) = V_a(1 - \frac{y}{g})$$
 (5.4)

In consecință câmpul electrostatic are expresia:

$$\vec{E}(y) = \frac{V_a}{g}\vec{j}, \,\forall y \in (0,g).$$
(5.5)

Forța electrostatică ce acționează asupra suprafeței armăturii mobile poate fi determinată prin două metode:

1. evaluarea tensiunii maxwelliene într-un punct de pe suprafața armăturii:

$$T_e(g) = -\frac{1}{2}\varepsilon_0 \vec{E}(g)\vec{D}(g) = -\frac{1}{2}\varepsilon_0 \frac{V_a^2}{g^2},$$
(5.6)

unde \vec{D} este inducția electrică, iar ε_0 este permitivitatea electrică a aerului. Forța totală rezultă a fi:

$$\vec{F}_{e}(g) = -\frac{1}{2}\varepsilon_{0}\frac{V^{2}}{g^{2}}A\vec{n} = -\frac{1}{2}\varepsilon_{0}\frac{V^{2}_{a}}{g^{2}}A\vec{j},$$
(5.7)

unde $A = l_b \times w$ este aria suprafeței fiecăreia dintre armături.

2. evaluare energetică prin aplicarea primei legi a termodinamicii (2.86): forța este determinată din lucrul mecanic efectuat de către câmpul electrostatic pentru deplasarea armăturii pe o distanță suficient de mică - notată cu δy - pentru a putea considera această forță constantă:

$$\delta W_e = \delta L = F_e \, \delta y, \tag{5.8}$$

unde:

$$\delta W_e = W_e(g + \delta y) - W_e(g), \tag{5.9}$$

din care rezultă că:

$$\vec{F}_e(g) = -\frac{1}{2}\varepsilon_0 \frac{V_a^2}{g(g+\delta y)} A \vec{j}.$$
(5.10)

Pentru $\delta y \to 0$ relația (5.10) este echivalentă cu (5.7). La aceeași expresie se ajunge și prin aplicarea teoremei forțelor generalizate în câmp electrostatic (2.28) pentru potențial electric constant:

$$\vec{F}_e(g) = \frac{\partial W_e}{\partial g}\Big|_{V=V_a} \vec{j} = -\frac{1}{2}\varepsilon_0 \frac{V_a^2}{g^2} A \vec{j}, \qquad (5.11)$$

In problema mecanică ne interesează determinarea deplasării \vec{u} , a unui punct material asociat armăturii, deplasare datorată acțiunii forței electrostatice. Această mărime rezultă din scrierea ecuației de echilibru static al forțelor ce acționează asupra punctului material. Se poate arăta foarte simplu că în ecuația de echilibru static, forța gravitațională nu are nicio influență, ecuația de echilibru fiind:

$$\vec{F_m} + \vec{F_e} = 0, \tag{5.12}$$

adică,

$$-\frac{1}{2}\varepsilon_0 \frac{V_a^2}{(g_0 - u)^2} A + k u = 0, \qquad (5.13)$$

respectiv:

$$-\frac{1}{2}\varepsilon_0 \frac{V_a^2}{g^2} A + k \left(g_0 - g\right) = 0, \qquad (5.14)$$

unde k este constanta elastică a arcului care produce revenirea armăturii mobile, iar g_0 este distanța inițială dintre cele două armături. Problema de determinare a deplasării \vec{u} în funcție de V_a este una cuplată electromecanică - presupunând deci rezolvarea simultană sau succesivă a unei probleme electrice (PE), respectiv a uneia mecanice - de elasticitate (PM). Cu toate că fiecare dintre cele două probleme, tratată separat, este liniară, problema cuplată este neliniară din două motive:

- 1. Forța electrostatică $\vec{F_e}$ care reprezintă legătura dintre cele două probleme este dependentă de pătratul excitației V_a ;
- 2. Domeniul Ω_e pe care se rezolvă PE este dependent de distanța g, respectiv de deplasarea u, care se determină din PM, în funcție de F_e .

Problema se poate aborda fie prin rezolvarea unui unic sistem de ecuații rezultat din scrierea compactă a ecuațiilor PE, respectiv PM, fie prin rezolvarea succesiv separată a celor două probleme, conform organigramei din fig. 5.6. Ecuația (5.14) reprezintă legătura dintre tensiunea de acționare V_a și distanța g:

$$V_{a}(g) = \sqrt{\frac{2k}{\varepsilon_{0}A} g^{2}(g_{0} - g)}$$
(5.15)



Figura 5.6: Organigrama de rezolvare iterativă a problemei electromecanică neliniară, 2D, pentru o tensiune de acționare V_a .

Pe intervalul $[0, g_0]$ funcția $V_a(g)$ admite un maxim ce rezultă din anularea derivatei:

$$\frac{\partial V_a}{\partial g} = 0. \tag{5.16}$$

Ecuația (5.15), admite pentru o anumită tensiune V_a două soluții $g_{1,2} \in [0, g_0]$ [74],[112], [103]:

- 1. una instabilă, în care armătura mobilă nu poate rămâne în repaus, ci se deplasează venind în contact cu armătura fixă;
- 2. una stabilă, în care armătura mobilă rămâne în repaus.

Cea mai mică tensiune care aduce armătura în zona instabilă, producând contactul ei cu cea fixă este tensiunea minimă de acționare V_{PI} , se determină prin înlocuirea soluției ecuației (5.16) în relația (5.15):

$$V_{PI} = \sqrt{\frac{8kg_0^3}{27\varepsilon_0 A}} \tag{5.17}$$

și se obține pentru $g_{PI} = \frac{2}{3}g_0$. Cu ajutorul relațiilor (5.15) și (5.17) se poate determina o formă normalizată a legăturii dintre V_a și g, respectiv $u = g_0 - g$:

$$\left(\frac{V_a}{V_{PI}}\right)^2 = \frac{27}{4} \frac{g_0 - g}{g_0} \left(\frac{g_0 - g}{g_0} - 1\right)^2 \tag{5.18}$$

a cărei reprezentare este în fig. 5.7.

In cazul șuntului capacitiv (v. fig. 5.8) în problema electrostatică, apare un strat dielectric de grosime t_d , respectiv de permitivitate relativă $\varepsilon_r > 1$. In această situație, deplasarea maximă g_0 a armăturii mobile este considerată între poziția sa inițială și suprafața superioară a dielectricului menționat. Distanța maximă dintre cele două armături este $g_0 + t_d$, iar cea minimă este t_d .

Utilizând aceeași abordare, dar ținând cont de stratul dielectric de grosime t_d și permitivitatea electrică ε_r rezultă următoarele relații:

• Ecuația de echilibru static:

$$-\frac{1}{2}\varepsilon_0 \frac{V_a^2}{\left(g + \frac{t_d}{\varepsilon_r}\right)^2} A + k\left(g_0 - g\right) = 0, \qquad (5.19)$$

• Tensiunea de minimă acționare și distanța corespunzătoare:

$$V_{PI} = \sqrt{\frac{8k}{27\varepsilon_0 A} \left(g_0 + \frac{t_d}{\varepsilon_r}\right)^3}$$

$$g_{PI} = \left(\frac{2g_0}{3} - \frac{t_d}{3\varepsilon_r}\right).$$
 (5.20)



Figura 5.7: Caracteristica normalizată a tensiunii V_{PI} în raport cu deplasarea $u = g_0 - g$ a armăturii mobile. Curba desenată cu linie întreruptă reprezintă zona de instabilitate. Curba desenată cu linie plină reprezintă pe intervalul $[0, g_0/3]$ zona de stabilitate, iar pe intervalul $(g_0/3, g_0]$ reprezintă evoluția reală a armăturii mobile: pentru orice tensiune $V_a > V_{PI}$ armătura mobilă se lipește de cea fixă.



Figura 5.8: Domeniile geometrice ale modelului 1D pentru un microșunt capacitiv.

• Forma normalizată a legăturii dintre V_a și g, respectiv $u = g_0 - g$:

$$\left(\frac{V_a}{V_{PI}}\right)^2 = \frac{27}{4} \frac{g_0 - g}{\left(g_0 + \frac{t_d}{\varepsilon_r}\right)} \left(\frac{g_0 - g}{\left(g_0 + \frac{t_d}{\varepsilon_r}\right)} - 1\right)^2 \tag{5.21}$$

Caracteristica (5.21) este similară cu ce din fig.5.7.

Pentru modelul 1D se pot face următoarele observații, utile și pentru cazurile de modele 2D și 3D:

- teniunea V_{PI} depinde de proprietățile elastice ale membranei mobile, precum și de distanța inițială g_0 ;
- distanța dintre armături critică g_{PI} depinde numai de g_0 ; cu toate că acest fapt este teoretic valabil numai pentru modelul 1D, se va vedea în continuare că zona de instabilitate a armăturii mobile începe și la celelate modele pentru un spațiu dintre armături de aproximativ $\frac{2}{3}g_0$.

5.4 Un model 2D pentru șuntul capacitiv

In continuare va fi analizat un model 2D al unui șunt capacitiv simplu produs de firma Raytheon (v.fig.5.3). Dimensiunile sunt cele reale, preluate din lucrări în care au fost studiate diverse aspecte fizice ale acestui tip de dispozitiv MEMS [131], [135], [103]. Modelul este unul foarte des utilizat pentru circuitele utilizate în prelucrarea semnalelor cu frecvențe cuprinse pornind din gama megahertzilor până în cea a zecilor de gigahertzi. Schița 2D a dispozitivului este redată în fig.5.9, iar dimensiunile și proprietățile de material în tabelul 5.1.

Denumire	Valoare	Parametru
Lungime membrană mobilă	$280\mu m$	l_b
Grosime membrană mobilă	$0,4\mu m$	h_b
Lățime membrană mobilă	$120\mu m$	w_b
Lungime conductor central	$240 \mu m$	l_{el}
Grosime conductor central	$0,4\mu m$	h_{el}
Lățime conductor central	$120\mu m$	w_{el}
Grosime strat dielectric	$0, 1\mu m$	t_d
Inălțime conductoare laterale	$4\mu m$	h_{cpw}
Lățime conductor lateral	$120\mu m$	w_{cpw}
Lungime conductor lateral	$240\mu m$	$l_{cpw} = l_{el}$
Distanța inițială dintre armături	$3,5\mu m$	g_0

Tabela 5.1: Valorile asociate parametrilor specificați în modelul din fig.5.9.

5.4.1 Formulări tari

Ca și în cazul 1D, separăm problema neliniară cuplată, în două probleme liniare: una de electrostatică, iar alta de elastostatică.

Problema electrică este definită pe un domeniu Ω_E , care este de fapt spațiul curins între:

- sus fața inferioară a membranei elastice notată cu Γ_{E_b} ;
- jos suprafața conductoare a barei centrale peste care este depus startul de dielectric, notată cu Γ_{E_d} , împreună cu suprafațele de substrat dintre barele laterale și conductorul central, notate cu Γ_{E_s} ;

lateral - fețele barelor dinspre bara centrală - notate cu $\Gamma_{E_{cpw}}$;

Cu aceste notații (v. fig.5.10) frontiera $\partial \Omega_E$ se poate scrie astfel:

$$\partial \Omega_E = \Gamma_{E_b} \cup \Gamma_{E_{cpw}} \cup \Gamma_{E_d} \cup \Gamma_{E_s}.$$
(5.22)

Forma tare a problemei presupune rezolvarea ecuației eliptice (v. cap. 3.3) de ordin





Figura 5.9: Secțiune transversală și vedere de sus pentru microșuntul capacitiv. Secțiunea transversală **A-A** este folosită pentru modelul 2D al problemei plan-paralelă.

doi:

$$-\nabla \cdot (\varepsilon \nabla v) = 0 \ pe \ \Omega_E, \tag{5.23}$$

$$v = V_a \ pe \ \Gamma_{E_d}, \tag{5.24}$$

$$v = 0 \ pe \Gamma_{E_b} \cup \Gamma_{E_{cpw}}, \tag{5.25}$$

$$\frac{\partial v}{\partial n} = 0 \ pe \Gamma_{E_s},$$

impunând găsirea soluției $v \in C^2(\Omega_E)$ cu $\Omega_E \subset \mathbb{R}^2$. Din punctul de vedere al proprietăților de material, domeniul $\partial \Omega_E$ este compus din două zone:

1. startul dielectric de grosime t_d depus peste conductorul central: în mod uzual, acest strat este din nitridă de Si (Si₃Ni₄) cu permitivitate electrică relativă $\varepsilon_r = 7$.



Figura 5.10: Domeniile 2D ale problemelor plan paralele: electrostatică - (a); elastostatică - (b).

2. restul de domeniu, care în practică este umplut cu un gaz inert pentru care $\varepsilon_r = 1$.

Problema mecanică este definită pe un domeniu Ω_M , care reprezintă de fapt corpul lamelei elastice limitat de:

- sus fața superioară a membranei elastice notată cu $\Gamma_{M_t};$
- jos fața inferioară a membranei elastice notată cu Γ_{M_b} ;
- lateral fețele determinate de cele două secțiuni transversale prin lamelă în punctele de încastrare a acesteia în barele laterale notate cu Γ_{M_l} ;

Cu aceste notații (v. fig.5.10) frontiera $\partial \Omega_M$ se poate scrie astfel:

$$\partial\Omega_M = \Gamma_{M_b} \cup \Gamma_{M_l} \cup \Gamma_{M_t}. \tag{5.26}$$

Forma tare a problemei presupune rezolvarea ecuației caracteristice problemelor de elasticitate liniară (v. Problema 3.4.1):

$$-\nabla \left[\lambda \nabla(\vec{u})[\mathbf{I}] + 2\mu \left[\epsilon(\mathbf{u})\right]\right] = 0 \qquad \text{pe } \Omega_M, \tag{5.27}$$

unde [I] este matricea unitate de ordin 2, cu condițiile de frontieră:

$$\begin{cases} \vec{u} = 0 & \text{pe } \Gamma_{M_l}, \\ \overline{\sigma}\vec{n} = \vec{t_b} & \text{pe } \Gamma_{M_b}, \end{cases}$$
(5.28)

se caută soluția: $u \in [\mathcal{C}^2(\Omega_M)]^2$. Lamela elastică este din Al, material presupus a fi omogen și izotrop. Proprietățile de material de interes sunt:

 $\begin{cases} \text{Coeficientul Poisson } \nu = 0,35, \\ \text{Modulul Young de elasticitate longitudinală } E = 70 \text{GPa.} \end{cases}$ (5.29)

Cu ajutorul acestor coeficienți de material, se determină parametrii Lamé (v. cap.2.4.6), $\lambda \neq \mu$, ce intervin în ecuația (5.27):

$$\begin{cases} \lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \\ \mu = \frac{E}{2(1+\nu)}. \end{cases}$$
(5.30)

Cuplajul dintre cele două probleme este dat de forța electrostatică ce acționează asupra suprafeței inferioare a lamelei elastice și care rezultă din problema electrică. Forța electrostatică este una superficială și acționează pe frontiera comună (notată în continuare cu Γ_b dintre domeniile Ω_E și Ω_M :

$$\Gamma_b = \Gamma_{E_b} = \Gamma_{M_b}.\tag{5.31}$$

Asemănător celor discutate în cazul problemei 1D, această forță poate fi determinată prin două metode:

1. evaluarea tensorului tensiunii maxwelliene într-un punct de pe suprafața armăturii - conform (2.32):

$$\overline{\overline{\mathbf{T}}}_{e} = \varepsilon_0 \begin{pmatrix} \frac{1}{2} \begin{pmatrix} E_x^2 - E_y^2 \end{pmatrix} & E_x E_y \\ E_x E_y & \frac{1}{2} \begin{pmatrix} -E_x^2 + E_y^2 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}$$
(5.32)

Unde $E_x = -\frac{\partial v}{\partial x}$ și $E_y = -\frac{\partial v}{\partial y}$ sunt componentele vectorului intensitate a câmpului electrostatic. Din (5.32) rezultă tensorul tensiunilor mecanice întrun punct de pe frontiera Γ_b :

$$\begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix} = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix},$$
(5.33)

iar conform condiției de frontieră (5.28) pe Γ_b :

$$\begin{pmatrix} t_{b_x} \\ t_{b_y} \end{pmatrix} = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix} \cdot \begin{pmatrix} n_x \\ n_y \end{pmatrix}$$
(5.34)

2. evaluare energetică prin aplicarea teoremei forțelor generalizate în câmp electrostatic (2.28) pentru potențial electric constant:

$$\vec{t}_b = \frac{\partial W_e}{\partial x}\vec{i} + \frac{\partial W_e}{\partial y}\vec{j}.$$
(5.35)

5.4.2 Formulări slabe

Formularea matematică slabă a ecuației (5.23) se obține prin proiectarea acesteia pe spațiul generat de o bază de funcții test $\varphi \in H_0^1(\Omega_E)$, - definit în (3.92).

In sens slab - se rezolvă următoarea problemă: pentru ecuația:

$$(-\nabla(\varepsilon\nabla v), \varphi) = 0, \text{ si } \varepsilon \in \mathbb{R}^+, \tag{5.36}$$

cu condițiile de frontieră:

$$\begin{cases} v = V_a & \text{pe } \Gamma_{E_d}, \\ v = 0 & \text{pe } \Gamma_{E_b} \cup \Gamma_{E_{cpw}}, \\ \frac{\partial v}{\partial n} = 0 & \text{pe } \Gamma_{E_s}. \end{cases}$$
(5.37)

se caută soluția:

$$v \in H_D^1(\Omega_E) = \{ x \in H^1(\Omega_E) | tr_{\Gamma_{E_d}}(x) = V_a \} \qquad \forall \varphi \in H_0^1.$$
(5.38)

Expresia rezultată prin proiecție, se integrează apoi prin părți, obținându-se în final o egalitate de tip (3.143). Rezolvarea problemei electrice presupune determinarea soluției v, în condițiile (5.38), pentru ecuația de mai jos:

$$\begin{cases} \mathcal{A}(v,\,\varphi) = 0 \text{ cu}, \\ \mathcal{A}: H^1(\Omega_E) \times H^1_0(\Omega_E) \to \mathbb{R}, \end{cases}$$
(5.39)

unde:

$$\mathcal{A}(v,\,\varphi) = \int_{\Omega_E} \left(\varepsilon\,\nabla u \cdot \nabla\varphi\right)\,dA.\tag{5.40}$$

Existența și unicitatea soluției problemei electrostatice sunt discutate și demonstrate în cap.3.3.4.

Formularea slabă pentru problema de elasticitate se obține, ca și în cazul problemei electrice, prin proiectarea formulării tari (5.27) pe spațiul funcțiilor de test $w \in [H_0^1(\Omega_M)]^2$, iar apoi prin integrare prin părți. În formă slabă enunțul problemei mecanice devine: pentru ecuația:

$$\int_{\Omega_M} \left(\lambda \nabla \cdot \vec{u} \nabla \cdot \vec{w} + 2\mu \epsilon(\vec{u}) : \epsilon(\vec{w}) \right) dA = \int_{\Gamma_{M_b}} \vec{t_b} \cdot \vec{w} \, dr \tag{5.41}$$

cu condițiile de frontieră:

$$\begin{cases} \vec{u} = 0 & \text{pe } \Gamma_{M_l}, \\ \sigma \vec{n} = \vec{t}_b & \text{pe } \Gamma_{M_b} \text{ si } \vec{t}_b \in [H^{-\frac{1}{2}}(\Gamma_{M_b})]^2, \end{cases}$$
(5.42)

se caută soluția:

$$u \in [H_D^1(\Omega_M)]^2 = \{ x \in [H^1(\Omega_m)]^2 | tr_{\Gamma_{M_l}}(x) = 0 \} \qquad \forall w \in [H_0^1(\Omega_M)]^2.$$
(5.43)

Ecuația (5.41) pune în evidență operatorul biliniar $\mathcal{A}(\vec{u}, \vec{w})$, respectiv produsul de dualitate $\langle \vec{F}, \vec{w} \rangle$, scriindu-se în notație compactă astfel:

$$\begin{cases} \mathcal{A}(\vec{u}, \vec{w}) = \langle \vec{F}, \vec{w} \rangle, \\ \mathrm{cu} \ \mathcal{A}(\vec{u}, \vec{w}) = \int_{\Omega_M} (\lambda \nabla \cdot \vec{u} \nabla \cdot \vec{w} + 2\mu \epsilon(\vec{u}) : \epsilon(\vec{w})) \, dA \\ \mathrm{si} \ \langle \vec{F}, \vec{w} \rangle = \int_{\Gamma_{M_b}} \vec{t_b} \cdot \vec{w} \, dr, \\ \mathrm{cu} \ \vec{F} : [L^2(\Omega_M)]^2 \times [H^1_0(\Omega_M)]^2 \to [H^{-1}(\Omega_M)]^2, \end{cases}$$
(5.44)

Produsul $\epsilon(\vec{u}) : \epsilon(\vec{w})$ este un produs scalar tensorial:

$$\epsilon(\vec{u}):\epsilon(\vec{w}) = \sum_{i,j=1}^{3} \epsilon_{ij}(\vec{u})\epsilon_{ij}(\vec{w}).$$
(5.45)

Existența și unicitatea soluției problemei de elasticitate sunt discutate și demonstrate în cap.3.4.2.

Pentru formulările slabe, condiția de cuplaj este aceeași ca și în cazul formulărilor tari (v.(5.33), (5.35)).

5.5 Modelul 3D pentru șuntul capacitiv

In acest capitol se regăsesc formulările tari, respectiv slabă pentru modelul 3D pentu același șunt capacitiv prezentat în capitolele anterioare. In fig.5.11 este figurat domeniul problemei electrostatice. Pe figură apar notațiile utilizate la specificarea condițiilor de frontieră. Urmează apoi, fără explicații suplimentare, formulările tari și slabe din care reies și spațiile vectoriale în care se caută soluțiile. Problema neliniară cuplată, este separată în două probleme liniare: una de electrostatică, iar alta de elastostatică.

5.5.1 Formulări tari

Problema electrică este definită pe un domeniu Ω_E , reprezentând un paralelipiped delimitat de următoarele fețe:

- sus baza superioară în plan paralel cu planul xOz la cota $y = h_{cpw}$; în baza superioară este inclusă fața inferioară a membranei elastice - notată cu Ω_{E_b} ;
- jos baza inferioară în planul xOz; în același plan este plasat și stratul conductor al barei centrale, de înălțime h_{el} al cărui suprafață este notată cu Ω_{E_d} , împreună cu suprafațele de substrat izolator dintre barele laterale și conductorul central;
- lateral fețele barelor laterale dinspre bara centrală notate cu $\Omega_{E_{cpw}}$, perpendiculare pe axa Ox plasate la cotele $x = -l_b/2$, respectiv $x = l_b/2$; fețele laterale perpendiculare pe axa Oz plasate la cotele $z = -l_{cpw}/2$, respectiv $z = l_{cpw}/2$.

Reuniunea zonelor de suprafață laterală ce nu conțin material conductor este notată cu $\partial \Omega_{E_a}$. Cu aceste notații (v. fig.5.11) frontiera $\partial \Omega_E$ se poate scrie astfel:

$$\partial \Omega_E = \partial \Omega_{E_b} \cup \partial \Omega_{E_{crav}} \cup \partial \Omega_{E_d} \cup \partial \Omega_{E_d}.$$
(5.46)

Forma tare a problemei presupune rezolvarea ecuației eliptice (v. cap. 3.3) de ordin doi:

$$-\nabla \cdot (\varepsilon \nabla v) = 0 \ pe \ \Omega_E, \tag{5.47}$$

$$v = V_a \ pe \ \partial\Omega_{E_d}, \tag{5.48}$$

$$v = 0 \ pe \,\partial\Omega_{E_b} \cup \partial\Omega_{E_{cpw}}, \tag{5.49}$$

$$\frac{\partial v}{\partial n} = 0 \ pe \, \partial \Omega_{E_a},$$

impunând găsirea soluției $v \in C^2(\Omega_E)$ cu $\Omega_E \subset \mathbb{R}^3$. Din punctul de vedere al proprietăților de material, domeniul $\partial \Omega_E$ este compus din două zone:

- 1. startul dielectric de grosime t_d depus peste conductorul central cu permitivitate electrică relativă $\varepsilon_r = 7$.
- 2. restul de domeniu, pentru care $\varepsilon_r = 1$.

Problema mecanică este definită pe un domeniu Ω_M , care reprezintă corpul lamelei elastice, în formă de paralelipiped a cărui bază inferioară se suprapune peste suprafața $\partial \Omega_{E_b}$ definită în problema electrică:

- sus fața superioară a membranei elastice notată cu $\partial \Omega_{M_t}$;
- jos fața inferioară a membranei elastice notată cu $\partial \Omega_{M_b}$ (se supraune peste $\partial \Omega_{E_b}$);
- lateral fețele determinate de cele două secțiuni transversale prin lamelă în punctele de încastrare a acesteia în barele laterale - notate cu $\partial\Omega_{M_l}$;



Capitolul 5. Aspecte computationale



Figura 5.11: Modelul 3D al problemei de electrostatică.

Cu aceste notații frontiera $\partial \Omega_M$ se poate scrie astfel:

$$\partial\Omega_M = \partial\Omega_{M_b} \cup \partial\Omega_{M_l} \cup \partial\Omega_{M_t}.$$
(5.50)

 $\partial\Omega_{E_a}$

Forma tare a problemei presupune rezolvarea ecuației caracteristice problemelor de elasticitate liniară (v. Problema 3.4.1):

$$-\nabla \left[\lambda \nabla(\vec{u})[\mathbf{I}] + 2\mu \left[\epsilon(\mathbf{u})\right]\right] = 0 \qquad \text{pe } \Omega_M, \tag{5.51}$$

unde [I] este matricea unitate de ordin 3, cu condițiile de frontieră:

$$\begin{cases} \vec{u} = 0 & \text{pe } \partial \Omega_{M_l}, \\ \overline{\overline{\sigma}} \vec{n} = \vec{t_b} & \text{pe } \partial \Omega_{M_b}, \end{cases}$$
(5.52)

se caută soluția: $u \in [\mathcal{C}^2(\Omega_M)]^3$. Lamela elastică este din Al, material presupus a fi omogen și izotrop. Proprietățile de material de interes sunt cele specificate pentru modelul 2D.

Cuplajul dintre cele două probleme este dat de forța electrostatică ce acționează asupra suprafeței inferioare a lamelei elastice și care rezultă din problema electrică. Forța electrostatică este una superficială și acționează pe frontiera comună (notată în continuare cu $\partial\Omega_b$ dintre domeniile Ω_E și Ω_M :

$$\partial \Omega_b = \partial \Omega_{E_b} = \partial \Omega_{M_b}. \tag{5.53}$$

Asemănător celor discutate în cazul problemei 2D, această forță poate fi determinată prin două metode:

1. evaluarea tensorului tensiunii maxwelliene într-un punct de pe suprafața armăturii - conform (2.32), din care rezultă tensorul tensiunilor mecanice întrun punct de pe frontiera $\partial \Omega_b$:

$$\begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} = \frac{\varepsilon_0}{2} \begin{pmatrix} E_x^2 - E_y^2 - E_z^2 & 2E_x E_y & 2E_x E_z \\ 2E_y E_x & -E_x^2 + E_y^2 - E_z^2 & 2E_y E_z \\ 2E_z E_x & 2E_z E_y & -E_x^2 - E_y^2 + E_z^2 \end{pmatrix}$$
(5.54)

care înlocuită în condiția (5.52) determină \vec{t} pe frontiera $\partial \Omega_b$. $E_x = -\frac{\partial v}{\partial x}$, $E_y = -\frac{\partial v}{\partial y}$ și $E_z = -\frac{\partial v}{\partial z}$ sunt componentele vectorului intensitate a câmpului electrostatic.

2. evaluare energetică prin aplicarea teoremei forțelor generalizate în câmp electrostatic (2.28) pentru potențial electric constant:

$$\vec{t}_b = \frac{\partial W_e}{\partial x}\vec{i} + \frac{\partial W_e}{\partial y}\vec{j} + \frac{\partial W_e}{\partial z}\vec{k}.$$
(5.55)

5.5.2 Formulări slabe

Formularea matematică slabă a ecuației (5.47) se obține prin proiectarea acesteia pe spațiul generat de o bază de funcții test $\varphi \in H_0^1(\Omega_E)$,

In sens slab - se rezolvă următoarea problemă: pentru ecuația:

$$(-\nabla(\varepsilon\nabla v), \varphi) = 0, \text{ si } \varepsilon \in \mathbb{R}^+, \tag{5.56}$$

cu condițiile de frontieră:

$$\begin{cases} v = V_a & \text{pe } \partial \Omega_{E_d}, \\ v = 0 & \text{pe } \partial \Omega_{E_b} \cup \partial \Omega_{E_{cpw}}, \\ \frac{\partial v}{\partial n} = 0 & \text{pe } \partial \Omega_{E_a}. \end{cases}$$
(5.57)

se caută soluția:

$$v \in H_D^1(\Omega_E) = \{ x \in H^1(\Omega_E) | tr_{\partial \Omega_{E_d}} = V_a \} \qquad \forall \varphi \in H_0^1.$$
(5.58)

Expresia rezultată prin proiecție, se integrează apoi prin părți, obținându-se în final o egalitate de tip (3.143). Rezolvarea problemei electrice presupune determinarea soluției v, în condițiile (5.57), pentru ecuația de mai jos:

$$\begin{cases} \mathcal{A}(v,\,\varphi) = 0 \text{ cu},\\ \mathcal{A}: H^1(\Omega_E) \times H^1_0(\Omega_E) \to \mathbb{R}, \end{cases}$$
(5.59)

unde:

$$\mathcal{A}(v,\,\varphi) = \int_{\Omega_E} \left(\varepsilon\,\nabla u \cdot \nabla\varphi\right)\,dV.\tag{5.60}$$

Existența și unicitatea soluției problemei electrostatice sunt discutate și demonstrate în cap.3.3.4.

Formularea slabă pentru problema de elasticitate se obține, ca și în cazul problemei electrice, prin proiectarea flormulării tari (5.27) pe spațiul funcțiilor de test $w \in [H_0^1(\Omega_M)]^3$, iar apoi prin integrare prin părți. În formă slabă enunțul problemei mecanice devine: pentru ecuația:

$$\int_{\Omega_M} \left(\lambda \nabla \cdot \vec{u} \nabla \cdot \vec{w} + 2\mu \epsilon(\vec{u}) : \epsilon(\vec{w}) \right) dV = \int_{\partial \Omega_{M_b}} \vec{t}_b \cdot \vec{w} \, dA \tag{5.61}$$

cu condițiile de frontieră:

$$\begin{cases} \vec{u} = 0 & \text{pe } \partial \Omega_{M_l}, \\ \sigma \vec{n} = \vec{t}_b & \text{pe } \partial \Omega_{M_b} \text{ si } \vec{t}_b \in [H^{-\frac{1}{2}}(\partial \Omega_{M_b})]^3, \end{cases}$$
(5.62)

se caută soluția:

$$u \in [H_D^1(\Omega_M)]^3 = \{ x \in [H^1(\Omega_m)]^3 | tr_{\partial \Omega_{M_l}}(x) = 0 \} \qquad \forall w \in [H_0^1(\Omega_M)]^3.$$
 (5.63)

Ecuația (5.61) pune în evidență operatorul biliniar $\mathcal{A}(\vec{u}, \vec{w})$, respectiv produsul de dualitate $\langle \vec{F}, \vec{w} \rangle$, scriindu-se în notație compactă astfel:

$$\begin{cases} \mathcal{A}(\vec{u}, \vec{w}) = \langle \vec{F}, \vec{w} \rangle, \\ \operatorname{cu} \mathcal{A}(\vec{u}, \vec{w}) = \int_{\Omega_M} (\lambda \nabla \cdot \vec{u} \nabla \cdot \vec{w} + 2\mu \epsilon(\vec{u}) : \epsilon(\vec{w})) \, dV \\ \operatorname{si} \langle \vec{F}, \vec{w} \rangle = \int_{\partial \Omega_{M_b}} \vec{t_b} \cdot \vec{w} \, dA, \\ \operatorname{cu} \vec{F} : [L^2(\Omega_M)]^3 \times [H^1_0(\Omega_M)]^3 \to [H^{-1}(\Omega_M)]^3, \end{cases}$$
(5.64)

Existența și unicitatea soluției problemei de elasticitate sunt discutate și demonstrate în cap.3.4.2.

Pentru formulările slabe, condiția de cuplaj este aceeași ca și în cazul formulărilor tari (v.(5.54)).

5.6 Rezolvări numerice prin metoda elementelor finite

Acest capitol este dedicat rezolvării numerice a problemelor 2D și 3D pentru șuntul capacitiv Raytheon descris mai sus. În acest scop am folosit un pachet software denumit **FreeFEM**++ [75],[76]. După o scurtă introducere referitoare la **FreeFEM**++, sunt prezentate modurile de descriere și de rezolvare a problemelor decsrise în cap.5.4 și 5.5, prin metoda elementelor finite (MEF), folosind algoritmi seriali.

5.6.1 Pachetul software FreeFEM++

In contextul prezentei lucrări, alegerea acestui pachet software nu este întâmplătoare, întrucât **FreeFEM++** este un software dedicat rezolvării ecuațiior cu derivate parțiale pentru probleme multifizice, neliniare. Programul este scris in C++, iar descrierea problemelor și rezolvarea lor se face printr-un limbaj ce reprezentând un subset al C++.

In plus, așa cum sugerează și denumirea, este vorba de un software ce se poate prelua, instala și utiliza la întreaga sa capacitate în mod gratuit. Spre deosebire de softurile comerciale echivalente, **FreeFem++** nu are o interfață de preprocesare grafică a geometriilor asociate problemelor de rezolvat: descrierea acestora se efectuează exclusiv prin limbajul propriu. Pentru vizualizarea modelelor geometrice, precum și a rezultatelor problemelor, **FreeFem++** are câteva module dedicate, însă cu posibilități destul de reduse de evidențiere grafică a mărimilor fizice determinate/prelucrate. Echipa de dezvoltatori pune însă la dispoziția utilizatorilor atât funcții de interfațare cu programe specializate cu posibilități grafice mai multe (Matlab©, Octave, Gnuplot, Paraview) precum și modul de structurare a datelor de I/E.

FreeFem++ este scris de matematicieni, iar pentru utilizarea sa este absolut necesar ca utilizatorul să fie familiarizat, cel puțin la nivel mediu, cu analiza funcțională și modul de aplicare a acesteia în rezolvarea problemelor cu derivate parțiale prin cu ajutorul MEF:

• FreeFem++ are o bibliotecă de funcții matematice de nivel înalt, care permit descrierea directă a unei formulări slabe, la modul general, așa cum a fost descris și în cap.3.2.9 - relația (3.143). De exemplu formularea slabă (5.60), (5.37), pentru problema 2D de mai sus se descriu astfel:

```
solve Electro(v, fi) = // v-necunoscuta, fi-functii de test
int2d(The) ( EPS*grad(v)'*grad(fi) ) // biliniara
+ on(GammaEd, v=Va) //cond. de frontiera
+ on(GammaEb, v=0)
+ on(GammaEcpw1, v=0)
+ on(GammaEcpw2, v=0);
```

- pachetul FreeFem++ conține o varietate de biblioteci matematice dedicate algebrei liniare, multe dintre ele cu posibilitate de a rula pe procesoare paralele (BLAS, ScaAPACK, PETSc, ARPACK);
- în scopul rezolvării problemelor de optimizare în mod dinamic, sau de rezolvare a problemelor în mișcare este disponibi un set de biblioteci dedicate generării, deformării dinamice, respectiv optimizării rețelelor de discretizare cu elemente triunghiulare, sau tetraedrale (Mmg, mshmet, Gmm++, Ipopt);
- rețelelor de discretizare generate li se pot atașa elemente finite de diverse tipuri și ordine: nodale (v.cap.4.1.4), de muchie (v.cap.4.1.5), de suprafață (v.cap.4.1.6, de volum(v.cap.4.1.7);

5.6.2 Rezolvare numerică iterativă

Utilizarea MEF pentru rezolvarea numerică a problemelor din cap. 5.4 și 5.5 presupune mai întâi, cf. cap.4.1 stabilirea următoarelor entități matematice:

1. suportul spațial pe care se aplică metoda, în speță stabilirea rețelelor de discretizare ale domeniilor Ω_E , respectiv Ω_M , rețele care vor genera domeniile apoximative Ω_{E_h} pentru PE, respectiv Ω_{M_h} pentru PM; fiecare rețea este reprezentată respectiv de triangulațiile:

$$\mathcal{T}_{E_h} = \mathcal{T}_h(\Omega_{E_h})$$
 și (5.65)

$$\mathcal{T}_{M_h} = \mathcal{T}_h(\Omega_{M_h}),\tag{5.66}$$

După descrierea geometriei lor, discretizarea domeniilor va fi efectuată automat de **FreeFEM**++ folosind elemente triunghiulare pentru problema 2D, respectiv tetraedrale prntru problema 3D.

- 2. spațiul funcțiilor de formă ce se asociază elementelor geometrice ale rețelei de discretizare:
 - pentru funcțiile constante pe porțiuni, dar care pot fi discontinui de la un element la altul (de ex. permitivitatea electrică) vom folosi spațiul \mathcal{P}^{0_1} al funcțiilor de formă asociate elementului finit conform în L^2 (v. (4.153) din cap.4.1.7);
 - pentru funcțiile continui vom folosi spațiul $\mathcal{P}^{H_{1_k}}$ al funcțiilor de formă asociate elementului finit conform în H^1 (v. (4.72) din cap.4.1.4);
- 3. spațiul $\mathcal{N}^{H_{1_k}}$ al gradelor de libertate pentru elementul finit conform în H^1 (v. (4.73) din cap.4.1.4) și dual cu spațiul funcțiilor de formă $\mathcal{P}^{H_{1_k}}$;

Pentru început se vor folosi elemente nodale de ordin 1. Apoi, pentru îmbunătățirea acurateței calculelor vor fi folosite elemente nodale de ordin 2. Astfel, spațiile funcționale ale soluțiilor numerice pentru cele două probleme vor fi respectiv:

$$\mathcal{W}_{E_h}^m = \left\{ v_h(\mathbf{x}) \in H^1(\Omega_{E_h}) \, | \, v_h^k(\mathbf{x}) \in \mathcal{P}^m(\Omega_{E_k}) \subset H^1(\Omega_{E_k}), \, \forall \Omega_{E_k} \in \mathcal{T}_{E_h} \right\}, \quad (5.67)$$
$$\mathcal{W}_{M_h}^m = \left\{ v_h(\mathbf{x}) \in H^1(\Omega_{M_h}) \, | \, v_h^k(\mathbf{x}) \in \mathcal{P}^m(\Omega_{M_k}) \subset H^1(\Omega_{M_k}), \, \forall \Omega_{M_k} \in \mathcal{T}_{M_h} \right\}, \quad (5.68)$$

unde *m* reprezintă ordinul elementului, iar $P^m(\Omega)$ este spațiul polinoamelor de ordin *m* definite pe Ω . Trebuie precizat că (5.68) reprezintă spațiul fiecărei componente a vectorului soluție \vec{u} din fiecare nod al rețelei de discretizare.

Fie N_{E_h} , respectiv N_{M_h} numărul de grade de libertate asociate triangulațiilor \mathcal{T}_{E_h} , respectiv \mathcal{T}_{E_h} și fie seturile de funcții de bază φ^m , respectiv \vec{w}_i^m cu care se pot genera spațiile $\mathcal{W}_{E_h}^m$, respectiv $\mathcal{W}_{M_h}^m$. Atunci putem scrie că pentru PE soluția numerică aproximativă este:

$$v_h(\mathbf{x}) = \sum_{j=1}^{N_{E_h}} \mathcal{N}_{E_j} \varphi^m(\mathbf{x}), \qquad (5.69)$$

unde \mathcal{N}_{E_j} sunt potențialele nodale ce trebuiesc determinate.

Soluția aproximativă, reprezentând componentele vectorului deplasare \vec{u} din PM, sunt date de:

$$\vec{u_h}(\mathbf{x}) = \sum_{j=1}^{N_{M_h}} \vec{\mathcal{N}_{M_j}} \vec{w_{x_j}}^m(\mathbf{x}),$$
 (5.70)

unde \mathcal{N}_{M_j} sunt deplasările nodale (după 2 respectiv 3 direcții). In consecință, în urma discretizării Galerkin a ecuației (5.39), rezultă un sistem liniar de N_{E_h} ecuații cu tot atâtea necunoscute, ecuații de forma:

$$\sum_{j=1}^{N_{E_{j}}} \mathcal{N}_{E_{j}} \int_{\Omega_{E}} \varepsilon \nabla \varphi_{j}(\mathbf{x}) \cdot \nabla \varphi_{i}(\mathbf{x}) = 0, \qquad (5.71)$$

iar în urma discretizării Galerkin a ecuației (5.64), rezultă un sistem liniar de N_{M_h} ecuații cu tot atâtea necunoscute, ecuații de forma:

$$\sum_{j=1}^{N_{M_h}} \mathcal{N}_{M_j} \int_{\Omega_M} \left[\lambda \nabla \cdot \vec{w_j} \nabla \cdot \vec{w_i} + 2\mu \epsilon(\vec{w_j}) : \epsilon(\vec{w_i}) \right] dV = \int_{\partial \Omega_{M_b}} \vec{t_b} \cdot \vec{w_i} dA, \quad (5.72)$$

Dacă în ecuațiile (5.71) se înlocuiesc potențialele cunoscute ale nodurilor situate pe frontiera Dirichlet, în termenul liber este pusă în evidență sursa de câmp electric a PE. Rezultă un sistem de ecuații liniar ce se poate scrie compact:

$$\mathbf{A_E} \cdot \mathbf{v} = \mathbf{W},\tag{5.73}$$

cu dimensiunea $N_{E_h} - N_{D_E}$, unde N_{D_E} este numărul de grade de libertate de pe frontiera Dirichlet. **A**_E reprezintă matricea de rigiditate a PE.

Sistemul de ecuații ale PM se poate scrie compact:

$$\mathbf{A}_{\mathbf{M}} \cdot \mathbf{u} = \mathbf{f},\tag{5.74}$$

unde A_M este matricea de rigiditate a PM.

Conform celor prezentate pentru modelul 1D (v.cap.5.3), rezultă că \mathbf{W} depinde de tensiunea de comandă V_a , precum și de deplasarea \vec{u} deoarece rețeaua de discretizare se modifică odată cu modificarea frontierelor Γ_b (cazul 2D), respectiv $\partial\Omega_b$ (cazul 3D). La rândul ei, excitația problemei mecanice, reprezentată prin termenul liber \mathbf{f} reprezentând forța superficială ce acționează pe frontiera Γ_b , respectiv $\partial\Omega_b$ este forța electrostatică determinată în PE, deci depinde atât de setul de potențiale \mathbf{v} , cât și de setul de deplasări \mathbf{u} din iterația precedentă, set care a influențat distribuția de potențial electric în domeniul Ω_E . Prin urmare, fiecare iterație neliniară k poate fi scrisă compact astfel:

$$\begin{cases} \mathbf{v}_{k} = \left[\mathbf{A}_{\mathbf{E}_{k-1}}(\mathbf{u}_{k-1})\right]^{-1} \cdot \mathbf{W}_{k-1}(\mathbf{u}_{k-1}, V_{a}) \\ \mathbf{u}_{k} = \left[\mathbf{A}_{\mathbf{M}_{k-1}}\right]^{-1} \cdot \mathbf{f}(\mathbf{v}_{k}, V_{a}). \end{cases}$$
(5.75)

In consecință putem aplica următorul algoritm numeric iterativ pentru determinarea deplasării \vec{u} , a membranei mobile, în cazul aplicării unei tensiuni de comandă V_a :

Date intrare: ; V_a ; // tensiunea de comandă nit_{max} ; // număr maxim de iterații err_{adm} ; // eroarea admisă ; // domenii geometrice nedeformate pentru ambele probleme Ω_E ; Ω_M ; Inițializare k = 0; // contor iterații ut = 0; // deplasare maximă cumulată a lamei elastice g = g0; // distanța mimimă dintre armături (gap) $\mathbf{u}_k = 0$; // init. vector deplasare curentă

```
•
```

repetă

$$\begin{split} \mathbf{u}_k &\longrightarrow \Omega_{E_k} ; & // \text{ discretizare PE cu deplasarea } \vec{u} \text{ curentă} \\ \mathbf{u}_k &\longrightarrow \Omega_{M_k} ; & // \text{ discretizare PM cu deplasarea } \vec{u} \text{ curentă} \\ \mathbf{v}_{k+1} &= [\mathbf{A}_{\mathbf{E_k}}(\mathbf{u}_k)]^{-1} \cdot \mathbf{W}_k(\mathbf{u}_k, V_a) ; & // \text{ PE} \\ \mathbf{f}_{k+1}(\mathbf{v}_k, V_a) ; & // \text{ det. forța superficială pe lama elastică} \\ \mathbf{u}_{k+1} &= [\mathbf{A}_{\mathbf{M_k}}]^{-1} \cdot \mathbf{f}_{k+1}(\mathbf{v}_k, V_a) ; & // \text{ PM} \\ ut &= ut + max(\mathbf{u}_{k+1}) ; & // \text{ noua deplasare cumulată maximă} \\ g &= g0 - ut ; & // \text{ noul gap} \\ k &= k+1; \end{split}$$

până când $(||max(\mathbf{u}_k) - max(\mathbf{u}_{k-1})|| < err_{adm} \ sau \ k > nit_{max});$

Date ieșire: k - 1, ut, g;

Pseudocod 1: Pseudocodului algoritmului iterativ pentru determinarea deplasării lamei elastice pentru o anumită tensiune de comandă V_a .

5.6.3 Problema2D

Pentru analiza cu MEF a problemelor expuse mai sus, pentru fiecare dintre domeniile $\partial\Omega_E$, respectiv $\partial\Omega_M$, trebuie mai întâi stabilite atât finețea rețelelei de discretizare, precum și tipul de element finit folosit. Apoi trebuie selectat ordinul spațiilor polinomiale asociate elementelor finite folosite, spații cărora vor aparține soluțiile aproximative ale problemelor. Așa cum am menționat în cap.4.1, este indicat a porni cu o rețea de discretizare pe cât posibil uniformă și cu elemente de ordin întâi. După o primă estimare, urmează apoi luarea unei decizii referitoare la modalitățile de îmbunătățire ale preciziei, respectiv a vitezei de convergență. Ținând cont de faptul că formulările slabe (5.36) și (5.41) au soluții în spații H^1 , putem folosi elemente finite nodale - conforme cu acest tip de spațiu vectorial (v.cap.4.1.4). Deși are o geometrie simplă (v. fig.5.9), șuntul capacitiv de modelat pune probleme în alegerea unei rețele de discretizare din mai multe motive:

- 1. lungimea sa, dictată de lungimea lamelei elastice, este 70 de ori mai mare decât înălțimea, fapt ce impune algerea unei rețele de discretizare care să limiteze numărul elementelor triunghiulare cu forme dezavantajoase, ce pot introduce erori numerice de nedorit (v. cap.4.1.3, relațiile (4.64) - (4.66)); același fenomen afectează ambele domenii Ω_E și Ω_M ;
- 2. existența colțurilor datorate formei conductorului central, care reprezintă puncte de discontinuitate pentru normala frontierei $\partial \Omega_E$ (v. cap.3.2.2.2);
- 3. la fiecare iterație, conform algoritmului (1), zona de aer dintre lama elastică și suprafața dielectricului depus pe conductorul central se micșorează, necesitând re-discretizarea domeniului Ω_E , cu o probabilitate crescătoare de a fi generate triunghiuri cu un factor de formă dezavantajos;
- 4. nu în ultimul rând, trebuie luată în considerare neliniaritatea problemei și modalitatea de determinare și transmitere a mărimii de cuplaj între cele două domenii, respectiv forța electrostatică pe suprafața comună $\partial \Omega_b$: trebuie ținut cont de faptul că valorile numerice sunt determinate pe nodurile rețelei de discretizare asociată domeniului Ω_E transferându-se - sursă de câmp - pe nodurile frontierei domeniului Ω_M ; o interpolare incorectă, sau însoțită de erori numerice mari, a acestui set de valori poate duce la instabilități numerice importante.

In cazul în care simulările 2D vor fi urmate de cele efectuate pe un model echivalent 3D, mult mai complex, este recomandabil a se efectua mai întâi evaluări



Figura 5.12: Rețeaua de discretizare de referință pentru aproximativ1/2 din microșuntul capacitiv.

numerice pe modelul 2D care sunt mult mai puțin costisitoare din punctul de vedere aș efortului de calcul.

Pentru început am stabilit câte o rețea de discretizare de referință, pentru fiecare domeniu (v.fig.5.12 și 5.14), denumite în continuare "*rețeaua E0*", respectiv *"rețeaua M0*". Fiecărei rețele li s-a asociat câte un spațiu de elemente finite nodale de ordin unu (tip **P1** - în terminologia **FreeFEM++**). Rețelele de elemente finite astfel formate au următoarele caracteristici:

- rețeaua E0 pentru domeniul Ω_E : 3943 triunghiuri, 2347 vertexuri, 8636 grade de libertate (prescurtat DOF);
- rețeaua M0 pentru domeniul Ω_M : 2264 triunghiuri, 1701 vertexuri, 7930 DOF;

5.6.4 Evaluarea rețelei de elemente finite cu elemente de ordin I

Pentru a avea un control calitativ asupra simulărilor 2D, putem efectua mai întâi o evaluare a modelului 1D din cap.5.3. Acest control este util în primul rând din punct de vedere calitativ, iar numeric, poate oferi informații suficiente despre ordinele de mărime ale valorilor mărimilor implicate în modelare. In acest sens, putem dimensiona armăturile condensatorului din fig. 5.5 astfel:

o lungime medie:
$$l_{b_{med}} = \frac{l_b + l_{el}}{2} = 200 \mu m$$
,
lățimea armăturii egală cu cea a lamelei elastice: $w = wb$. (5.76)



Figura 5.13: Rețeaua de discretizare de referință pentru o zonă centrală a lamelei elastice (domeniul Ω_M). Ținând cont că acest domeniu este paralelipipedic, rețeaua de discretizare este uniformă peste tot. Poate fi observată distribuția vectorului deplasare \vec{u} corespunzătoare domeniului nedeformat.



Figura 5.14: Rețeaua de discretizare de referință pentru o zonă adomeniului Ω_M care conține și unul din colțurile conductorului central. Pot fi observate liniile echipotențiale ale potențialului electric, corespunzătoare domeniului nedeformat.



Figura 5.15: Caracteristica tensiunii de comandă V_a în raport cu deplasarea $u = g_0 - g$ a armăturii mobile. Curba desenată cu linie întreruptă reprezintă zona de instabilitate. Curba desenată cu linie plină reprezintă pe intervalul $[0, g_0/3]$ zona de stabilitate, iar pe intervalul $(g_0/3, g_0]$ reprezintă evoluția reală a armăturii mobile: pentru orice tensiune $V_a > V_{PI}$ armătura mobilă se lipește de cea fixă.

Notațiile pentru modelul 2D sunt cele din tab.5.1. Constanta elastică pentru contactul mobil, poate fi aproximată cu o relație practică, folosită în [135] și anume:

$$k = \frac{32Ehb^3w_b}{l_b^3} \simeq 9,5N/m.$$
(5.77)

Cu aceste valori, folosind relațiile (5.20) și putem detremina tensiunea $V_{PI} = 30, 8V$ și distanța $g_{PI} = 2, 34\mu m$, iar din (5.19) putem trasa caracteristica $V_{PI}(u) = V_{PI}(g_0 - g)$ (v.fig.5.15).

Din caracteristica tipică din fig.5.15 se poate observa că, pe măsură ce tensiunea de comandă se apropie de valoarea critică V_{PI} , algoritmul iterativ neliniar, va găsi din ce în ce mai greu soluția, putând deveni fie divergent, fie să nu mai reușească să atingă precizia solicitată. Atât rețeaua de disctretizare cât și ordinele elementelor finite folosite pot influența fenomenele numerice menționate mai sus. De aceea, am considerat că modelele folosite pentru evaluare ar trebui testate atât pentru o excitație medie, dar și pentru o excitație mai aproape de tensiunea critică.

Pentru evaluări au fost folosite trei variante de rețele de discretizare derivate din cea de referință menționată mai sus (fig.5.12 și 5.14), denumite în continuare " $Ref \times 0,7$ ", " $Ref \times 2$ ", respectiv " $Ref \times 3$ ". Multiplicatorii derivă din raportul numerelor gradelor de libertate asociate acestor rețele față de cel al rețelei de referință:

- Rețeaua $Ref \times 0,7$ este obținută prin reducerea fineței de discretizare aproximativ două ori pentru domeniul Ω_M , rețeaua pentru Ω_E ramânând neschimbată (v. fig. 5.17);
- Rețeaua *Ref*×3 este obținută prin divizarea în trei a tuturor triunghiurilor din rețeaua de referință din cele două domenii (v. fig. 5.17);
- Rețeaua $Ref \times 2$ este obțiuntă din $Ref \times 3$ în sensul reducerii rafinării rețelei domeniului Ω_E numai în jurul punctelor de singularitate reprezentate de cele două colțuri ale conductorului central (v. fig. 5.18).

		$Ref \times 0,$	7	$ \leftarrow l$	Referinț	$t \breve{a} \rightarrow$		Cef×3 –	÷		$Ref \times 2$	
	NT	NV	DOF	NT	NV	DOF	NT	NV	DOF	NT	NV	DOF
Ω_E	2943	2347	2347	3943	2347	3943	15772	8636	8636	4877	2841	2841
Ω_M	286	288	1148	568	568	2268	2264	1701	7930	2264	1701	7930

Tabela 5.2: Caracteristicile rețelelor de discretizare de referință, respectiv derivate din aceasta. Semnificația notațiilor este: NT - nr. de triunghiuri, NV - nr. de vertexuri, DOF - grade de libertate.

Primul set de evaluări a fost efectuat pentru o tensiune de comandă $V_a = 15V$, iar al doilea set pentru $V_a = 22V$. Au fost urmărite pe parcursul iterațiilor, evoluțiile următoarelor mărimi:

- depasarea u, respectiv eroarea relativă eps_u , a deplasării armăturii, care reprezintă mărimea de control a convergenței algoritmului (v.tab.5.3 și fig.5.19);
- viteza de convergență (număr de iterații) pentru o eroare relativă limită impusă de 0,03% (v.tab.5.4 și fig.5.20);
- Forța electrostatică rezultată din problema electrică și modul în care a răspuns simularea problemei mecanice la această excitație (v.tab.5.5 și fig.5.21);



Figura 5.16: Rețeaua de discretizare $Ref \times 0,7$ pentru o zonă centrală a dispozitivului din jurul unuia dintre cele două puncte de singularitate ale Ω_E .



Figura 5.17: Rețeaua de discretizare $Ref \times 3$.



Figura 5.18: Rețeaua de discretizare $Ref \times 2$.

kit	Referință	$Ref \times 0,7$	Ref×3	$\text{Ref} \times 2$
	[%]	[%]	[%]	[%]
1	0,0747	0,0424	0,368	0,366
2	0,0242	0,0404	0,306	0,308
3		0,0358	0,133	0,132
4		0,0341	0,051	0,049
5		0,0304	0,014	0,015
6		0,0286		
t[s]	0,226	0,450	2,1	1,7

• timpul de calcul pe întreg algoritmul (v.tab.5.3);

Tabela 5.3: Evoluția erorilor relative pentru modelele cu elemente nodale de ordin 1, diferite fineți de discretizare, pentru $V_a = 15V$. Eroarea limită impusă a fost de 0,03%. La numărul de iterații se mai adaugă și cea inițială (pentru deplasare inițială u = 0), iterație pentru care nu se poate determina eroarea relativă față de 0.



Figura 5.19: Evoluția erorilor relative pentru modelele cu elemente nodale de ordin 1, diferite fineți de discretizare pentru $V_a = 15V$ (v.tab.5.3).

kit	Referință	$Ref \times 0,7$	$Ref \times 3$	$Ref \times 2$
	$[\mu m]$	$[\mu m]$	$[\mu m]$	$[\mu m]$
0	0,227	0,085	$0,\!687$	0,686
1	0,471	$0,\!173$	1,121	1,122
2	0,715	0,266	1,422	1,423
3		0,362	$1,\!683$	1,684
4		0,461	1,931	1,932
5		0,562	2,182	2,185
6		0,667		

Tabela 5.4: Evoluția deplasărilor membranei mobile pentru modelele cu elemente nodale de ordin 1, diferite fineți de discretizare, pentru $V_a = 15V$. Iterația 0 corespunde determinării forței electrostatice pentru domeniul Ω_E nedeformat, imediat după aplicarea tensiunii de comandă. Eroarea limită impusă a fost de 0,03%.



Figura 5.20: Evoluția deplasărilor membranei mobile pentru modelele cu elemente nodale de ordin 1, diferite fineți de discretizare, pentru $V_a = 15V$ (v.tab.5.4).

Capitolul 5. Aspecte computationale

kit	Referință	$Ref \times 0,7$	$Ref \times 3$	$Ref \times 2$
	[mN]	[mN]	[mN]	[mN]
0	9,73	9,73	9,74	9,74
1	10,94	$10,\!16$	$14,\!18$	$14,\!23$
2	12,53	$10,\!65$	18,87	18,88
3		11,19	23,81	23,85
4		11,79	29,79	29,95
5		12,40	38,03	38,29
6		$13,\!21$		

Tabela 5.5: Evoluția forțelor electrostatice pentru modelele cu elemente nodale de ordin 1, diferite fineți de discretizare, pentru $V_a = 15V$. Iterația 0 corespunde determinării forței electrostatice pentru domeniul Ω_E nedeformat, imediat după aplicarea tensiunii de comandă. Eroarea limită impusă a fost de 0,03%.



Figura 5.21: Evoluția forțelor electrostatice pentru modelele cu elemente nodale de ordin 1, diferite fineți de discretizare, pentru $V_a = 15V$. (v.tab.5.5).



Figura 5.22: Distribuția potențialului electric în domeniul Ω_E , pentru rețeaua Ref×2 și $V_a = 15V$, penultima iterație ($u = 1,932 \mu m$).

Simulările au fost rulate pe un Laptop Dell Inspiron, dotat cu un procesor i7-8550U @1,80Ghz și 16Gb RAM, sub sistemul de operare Windows10©.

Din tabelele de date și graficele asociate, putem trage următoarele concluzii:

- 1. referitor la erori, viteză de convergență și timp, *putem despinde două grupări*: rețaua de referință și Ref.×0.7 au comportări apropiate ca precizie; rularea pentru rețeaua de referință s-a încheiat numai după 3 iterații, timpul de rezolvare fiind de 2 ori mai mic față de rețeaua Ref.×0.7(v.tab.5.2). A doua grupare este cea a rețelelor mai fine pentru care rezultatele sunt foarte apropiate; rețeaua Ref.×2 este totuși mai avantajoasă datorită numărului mai mic de DOF, respectiv a timpului de rezolvare care a fost cu 35% mai mic față de rețeaua Ref.×3.
- 2. din punct de vedere al apropierii valorilor finale ale deplasărilor, pot fi păstrate aceleași grupări ca și în cazul erorilor; soluția determinată cu ajutorul celei mai rare rețele este de cu 6,6% mai mică decît în cazul rețelei de referință. De remarcat însă, diferența mare dintre rezultatele celor două grupări: deplasarea determinată pe baza rețelelor de discretizare mai fine este de aproximativ de trei ori mai mare decât cea determinată cu ajutorul rețelelor mai rare; acest fapt poate fi explicat odată cu analiza evoluției forțelor electrostatice (v. pct. următor);

3. pe baza analizei forțelor (tab.5.5) se pot observa următoarele: după "iterația 0" forța electrostatică determinată cu ajutorul tuturor modelelor este foarte apropiată (între 0 și 0,1% diferență față de referință); răspunsul problemei mecanice la aceeași excitație însă, este sensibil diferit de la un caz la altul, fenomenul depinzând de densitatea rețelelor de discretizare: de remarcat, că răspunsurile modelelor mecanice cu rețele de discretizare mai fină, practic se confundă - și asta datorită faptului că între cazurile Ref.×2 și Ref.×3 nu există diferențe de finețe între rețelele problemei mecanice, ci numai între cele ale rețelelor de discretizare ale problemei electrice. Putem trage concluzia de aici că formularea slabă (5.64), tratată cu elemente nodale are o senzitivitate mare la finețea de reprezentare numerică, fapt nu tocmai avantajos; un studiu mai aprofundat al acestei teme depășește tematica prezentei lucrări, dar idei de abordare și de compensare ale acestor instabilități pot fi găsite în [74], [112], [86].

Din cele de mai sus, ținând cont de similitudini, am reținut doar două modele:

- 1. modelul cu cea mai rară discretizare a domeniului Ω_M (Ref×0,7), datorită apropierii rezultatelor cu cele de la modelul de referință, precum și datorită unui plus de stabilitate numerică - necesară evaluărilor pentru tensiuni mai apropiate de tensiunea V_{PI} , care, conform evaluărilor 1D ar trebui să depășească 20V;
- 2. modelul $\text{Ref} \times 2$ datorită similitudinii rezultatelor sale cu cele de la modelul $\text{Ref} \times 4$, dar cu avantajul de a avea mai puține ecuații de rezolvat.

Pentru aceste două modele, am efectuat simulări pentru $V_a = 22V$ și o precizie limită admisă tot de 0,03%. De data aceasta am reținut numai evoluția erorii relative, a vitezei de convergență și a evoluției deplasării membranei elastice. Rezultatele pot vizualizate în graficele de mai jos.


Figura 5.23: Evoluția erorilor relative pentru modelele cu elemente nodale de ordin 1 selectate din prima evaluare; $V_a = 22V$ și eroare limită impusă de 0,03%.

Capitolul 5. Aspecte computationale



Figura 5.24: Evoluția deplasărilor membranei mobile pentru modelele cu elemente nodale de ordin 1 selectate din prima evaluare; $V_a = 22V$ și eroare limită impusă de 0,03%.

Din graficele de mai sus, putem trage următoarele concluzii:

- 1. Simularea pentru modelul cu rețea de discretizare mai rară nu reușește să atingă eroarea impusă, după care devine divergent. Procesul a fost oprit pentru o deplasare $u = 3,029\mu m$, deoarece la următoarea iterație rezultatul depășea spațiul inițial $g_0 = 3,5\mu m$.
- 2. Modelul cu rețea mai fină este convergent după doar două iterații, iar rezultatul $u = 2,57\mu m$ este de presupus a fi în zona instabilă a caracteristicii $V_a(u)$, depășind mai mult de două ori estimarea inițială de $1/3g_0$, care ar fi corespuns limitei zonei de stabilitate.

5.6.5 Influența modului de determinare a forței electrostatice

Acum este momentul de a aduce în discuție cele două metode de a evalua forța electrostatică pe care câmpul electric din domeniul Ω_E o exercită asupra suprafeței lamelei elastice a șuntului capacitiv:

- evaluarea tensiunilor Maxwelliene, rezultate din rezolvarea problemei electrice la iterația k;
- prin raportarea variației energiei electrostatice la variația deplasării u:

$$\frac{W_{e_k} - W_{e_{k-1}}}{\delta u_{k-1}},\tag{5.78}$$

unde δu_{k-1} reprezintă deplasarea membranei elastice determinată din problema mecanică în iterația recedentă (v. algoritmul 1).

Cele două metode au fost discutate în varianta 1D, de unde a rezultat perfecta lor compatibilitate. Atunci când evaluările se efectuează pe domenii aproximate prin discretizări cu elemente finite, deși metodele au, teoretic, același efect, din punct de vedere numeric apar diferențe:

- prima metodă presupune evaluarea punctuală a tensorului tensiunilor Maxwelliene (2.32), reprezentând chiar tensorul eforturilor unitare $\overline{\overline{\sigma}}$ din problema de elasticitate (v.(2.63));
- a doua metodă presupune evaluarea energiilor în seturi de elemente finite care își schimbă forma de la o iterație la alta, chiar dacă ele sunt dispuse în aceeași zona a domeniului electric;

Din acest motiv, am rulat setul de simulări efectuate cu cele două modele pentru $V_a = 22V$, dar evaluând forța electrostatică prin a doua metodă, cu relația (5.78). In acest caz, întucât la pasul inițial (k = 0) nu este disponibilă delpasarea δu_{-1} forța electrostatică este determinată cu ajutorul tensunilor maxwelliene, iar în la iterațiile următoare ea este calculată prin metoda energetică.

Rezultatele sunt prezentate în graficele de mai jos:





Figura 5.25: Evoluția erorilor relative pentru modelele cu elemente nodale de ordin 1 selectate din prima evaluare. Forța electrostatică este determinată prin metoda energetică. $V_a = 22V$ și eroare limită impusă de 0,03%.



Figura 5.26: Evoluția deplasărilor membranei mobile pentru modelele cu elemente nodale de ordin 1 selectate din prima evaluare. Forța electrostatică este determinată prin metoda energetică. $V_a = 22V$ și eroare limită impusă de 0,03%.

Din graficele de mai sus, putem trage următoarele concluzii:

- 1. De data aceasta, spre deosebire de cazul anterior (v.fig.5.23), simularea pentru modelul cu rețea de discretizare mai rară este convergent, iar rezultatul este unul rezonabil: $u = 0,61\mu m$.
- 2. Modelul cu rețea mai fină este nu reușește să atingă precizia dorită, dar, spre deosebire de cazul precedent evoluția erorii pentru acest model este asemenea celei pentru modelul convergent. Eroarea minimă atinsă pe parcursul iterațiilor este de 0,043%, iar rezultatul corespunzător este $u = 2,44\mu m$ fiind mai redusă cu 5% față de rezultatul anterior. Rezultatul poate fi însă presupus în zona instabilă a caracteristicii $V_a(u)$, conform estimărilor calitativ numerice efectuate inițial pentru modelul 1D. Cu toate acestea, se poate concluziona că folosirea metodei energetice pentru evaluarea forțelor duce la o mai bună stabilitate a procesului iterativ către o eroare limită posibilă.

5.6.6 Caracteristica $V_a(u)$

In continuare, este util a trasa graficul dependenței tensiunii de acționare în funcție de deplasarea lamelei elastice rezultate din determinările numerice cu elemente finite, și de a o compara cu graficul determinat analitic pentru problema 1D. Pentru trasarea caracteristicii, au fost efectuate simulări pentru cele 2 modelele 2D selectate în cap.5.6.4 pentru un interval $V_a \in [5, 31]V$. Pasul h_v , de creștere a tensiunii de acționare a fost scăzut pe intervalul mai apropiat de tensiunea V_{PI} determinată teoretic:

$$h_{v} = \begin{cases} 1, 0V & \text{pentru } V_{a} \in [5, 10], \\ 0, 5V & \text{pentru } V_{a} \in (10, 20), \\ 0, 25V & \text{pentru } V_{a} \in [20, 31]. \end{cases}$$
(5.79)

Deoarece modelul discretizat pentru problema mecanică a rezultat a fi foarte sensibil la varierea fineței rețelei de discretizare, am preferat, pentru Ω_M , păstrarea rețelei rare din cazul Ref×0,7. Tot din acest motiv, punerea în evidență a influenței elementelor nodale de ordin II pentru Ω_M nu este relevantă atâta timp cât răspunsul numeric al modelului nu este controlabil. Ca urmare, am folosit elemente de ordin II, doar pentru problema electrică. Rețeaua de discretizare cu o bună uniformitate și frontiera $\partial \Omega_E$ cu doar câteva puncte de singularitate, nu implică obținerea unor diferențe spectaculoase, la utilizarea elementelor finite de ordin II față de cele de ordin I.

Folosind notațiile din tab.5.2, caracteristicile rețelelor utilizate cu elemente finite de ordin II pe Ω_E sunt:

- Ω_E : NT = 4877, 2841, 10558;
- Ω_M : NT = 286, NV = 288, DOF = 1148.

Rețeaua geometrică pentru cazul $\text{Ref} \times 2$ este aceeași, dar numărul de grade de libertate s-a mărit, ținând cont că pentru elementele finite H^1 conforme gradele de libertate sunt plasate atât în vertexuri, dar și în punctele mediane de pe laturile triunghiurilor.



Figura 5.27: Curba deplasării u a armăturii mobile funcție de tensiunea de acționare V_a .

Simulările au fost efectuate pentru cele două modalități de calcul al forței electrostatice: cu tensiui maxwelliene, respectiv energetic. Se poate observa din fig.5.27, că aplicarea metodei energetice duce la un parcurs mai neted al caracteristicii, care este și cea mai apropiată de cea teoretică pentru modelul 1D. Influența elementelor finite de ordin II se face simțită pentru cazul aplicării metodei tensiunilor maxwelliene, în sensul unei ușoare "neteziri" a caracteristicii.

5.6.7 Problema 3D

Modelul 3D al șuntului capacitiv descris în fig.5.9 și tab.5.1 include și adâncimea, în speță, lățimea lamelei elastice, precum și lungimea conductoarelor laterale și central. Considerând modelul ca pe o cutie paralelipipedică, putem completa restul de spațiu al cutiei cu aer (adică un material cu permitivitate relativă unitară), astfel încât simulările să poată pune în evidență și efectele de capăt ale intensității câmpului electric, între muchiile lamelei elastice și conductorul central.

Tehnica prin care **FreeFEM++** poate genera un model 3D este aceea de a copia în mai multe planuri paralele situate la diferite cote z un model 2D construit inițial în planul xOy. Mai mult, nu este obligatoriu ca aceste plane paralele să se suprapună exact (prisma obținută nu trebuie să fie neapărat dreaptă). Fiecare vertex din planul de bază are câte un corespondent în planurile superioare, iar curba care leagă aceste vertexuri omoloage, este descrisă de o ecuație parametrică în funcție de z. Practic, un model 3D în **FreeFEM++** este descris de o familie de suprafețe (în engleză "layer") definită în coordonare (x, y) reunită cu o familie de curbe definite în z.

Modelul 3D al microșuntului capacitiv poate fi generat prin multiplicarea pe axa Oz a modelului 2D generat în capitolul anterior; straturile în care lamela conductoare nu mai există, au proprietățile de material ale aerului.

In fig.5.28 - fig.5.30 sunt prezentate câteva vederi ale modelului 3D pentru domeniul electric Ω_E .

Modelul 3D al microșuntului capacitiv din figurile de mai sus a fost generat prin translatarea modelului 2D în varianta Ref×0,7 în ideea de a generat cât mai puține ecuații. Rețelelor de discretizare li s-au asociat grade de libertate de ordin I pentru problema mecanică, respectiv de ordin II pentru problema electrică. Folosind notațiile din tab.5.2 (NT reprezintă numărul de tetraedre), caracteristicile rețelelor utilizate cu elemente finite de ordin II pe Ω_E sunt:

• Ω_E : NT = 787.746, NV = 153.195, DOF = 1.136.844;

• Ω_M : NT = 6.864, NV = 2.592, DOF = 23.328.

Am efectuat o simulare similară cu cea din cap.5.6.4 pentru modelul 2D, și anume pentru $V_a = 15V$ și o eroare relativă limită impusă de 0,03%. Rezultatele, comparativ cu modelul 2D sunt în tabelul de mai jos:



Figura 5.28: Vedere de perspectivă a modelului 3D pentru domeniul Ω_E . Banda din mijloc este fața inferioară a lamelei elastice, flancată de zone de aer care acoperă conductorul central și dielectricul t_d . Acestea din urmă, fiind de aproximativ 10 ori mai înguste decât spațiul dintre conductoare pot fi văzute fie intr-o vedere de detaliu, fie într-o vedere în care nu sunt randate suprafețele ce închid volumul domeniului. Liniile paralele cu lamela elastică identifică suprafețele dintre layere-le folosite pentru a construi modelul 3D din cel 2D.

	3D		2D		
kit	err	u	err	u	
	[%]	$[\mu m]$	[%]	$[\mu m]$	
0	-	0,119	-	0,085	
1	0,061	0,246	0,0424	0,173	
2	0,048	0,379	0,0404	0,266	
3	0,036	0,517	0,0358	0,362	
4	0,035	0,659	0,0341	0,461	
5	0,023	0,805	0,0304	0,562	
6			0,0286	0,667	
t[s]		540		0,450	

Tabela 5.6: Comparativ al erorilor relative și a deplasărilor pentru modelele 2D și 3D. Finețea de discretizare a fost conform rețelei $\text{Ref} \times 0.7$, $V_a = 15V$, și eroarea limită impusă 0,03%.



Figura 5.29: Vedere de perspectivă a modelului 3D pentru domeniul Ω_E dar fără fețe opace. Vederea este la o scară mult prea mică pentru a face vizibile mai în detaliu rețeua de discretizare. Pot fi identificate acum ambele zone ale conductoarelor centrale - mobil și fix, precum și interfețele dintre layere-le ce construiesc modelul 3D.



Figura 5.31: Diagramă comparativă a erorilor relative pentru simulările cu modelele cu elemente nodale de ordin I 2D și 3D. Finețea de discretizare a fost conform rețelei Ref×0,7, $V_a = 15V$, și eroarea limită impusă 0,03%. 226



Figura 5.30: Detaliu în vedere de perspectivă a modelului 3D pentru o porțiune a domeniului Ω_E dinspre baza sa. Pot fi identificate suprafața superioară a conductorului central fix, precum și dielectricul depus peste acesta.



Figura 5.32: Diagramă comparativă a deplasării u pentru simulările cu modelele cu elemente nodale de ordin I 2D și 3D. Finețea de discretizare a fost conform rețelei Ref×0,7, $V_a = 15V$ și eroarea limită impusă 0,03%.



Figura 5.33: Vedere de ansamblu a unei secțiuni yOz prin dispozitiv. Linia cotată reprezintă lățimea lamelei elastice, iar pe șirurile de elemente marcate cu săgeți se poate identifica umflarea liniilor da câmp electric între muchia lamelei și planul conductorului central.

Din tab.5.6 și fig.5.31, 5.32 rezultă că simularea pentru modelul 3D a ajuns la precizia solicitată cu o iterație mai puțin decât în cazul 2D. Timpul de calcul a fost însă de 1200 de ori mai mare, datorită faptului că modelul 3D a presupus rezolvarea la fiecare iterație un număr de 330 de ori mai mare de ecuații. Diferența relativă dintre rezultatele celor două simulări este 17% față de simularea 2D cu rețea de discretizare Ref×0,7. Față de simularea cu modelul de referință 2D (v. tab.5.4) diferența este de doar 1,5% față de acesta din urmă. Rezultatul este coerent cu faptul că simularea 3D surprinde și efectul de capăt al liniilor de câmp electric manifestat între cele două armături pe toată lățimea contactului fix, în cosecințăenergia de acționare determinată în 3D este mai mare decât cea determinată în 2D. In continuare sunt prezentate câteva imagini cu rezultatele simulării 3D. Posibilitățile intefeței grafice ale **FreeFEM**++ nu permit vizualizarea corectă a vectorilor liniilor de cîmp electric \vec{E} , respectiv a vectorilor câmpului vectorial deplasare \vec{u} . Prin urmare sunt afișate variațiile modulelor pe componente și anume E_y pentru câmpul electric, respectiv separat u_x și u_y pentru câmpul deplasare.



Figura 5.34: Detaliu al fig.5.34 cu imaginea și valorile codificate colorat pentru modulul componentei E_y între muchia lamelei și planul conductorului central.



Figura 5.36: Distribuția componentei u_y a vectorului deplasare pe domeniul lamelei elastice Ω_M . Deplasarea este maximă în zona de mijloc a lamelei. Valorile, indicate prin codul culorilor sunt cele corespunzătoare simulării pentru $V_a = 15V$ (v.5.6).





Figura 5.35: Distribuția potențialului electric pe o secțiune "în trepte" în planul yOx prin domeniul Ω_E . Se poate observa distribuția potențialului pe spațiul dintre electrodul central și lamela elastică, pentru mai multe cote z.



Figura 5.37: Distribuția componentei u_x a vectorului deplasare pe domeniul lamelei elastice Ω_M . Componenta după x a deplasării își schimbă semnul la trecerea prin axa de simetrie transversală, după axa Ox lamela fiind solicitată la întindere.

In final, am trasat dependența $V_a(u)$ corespunzătoare modelului 3D utilizat și în determinările de mai sus. Pasul h_v , de creștere a tensiunii de acționare a fost constant, de 0,5V. Pe același grafic sunt reprezentate atât caracteristica ce corespunde modelului 3D, cât și cea trasată în fig.5.27 corespunzătoare modelului 2D cu care a fost generat modelul 3D curent. Ambele folosesc elemente finite nodale de ordin I, iar metoda de calcul a forței electrostatice este cea a evaluării tensiunilor maxwelliene pe suprafața inferioară a electrodului elastic.



Figura 5.38: Curba deplasării u a armăturii mobile funcție de tensiunea de acționare V_a . Comparație între caracteristica trasată prin simulări cu modelul 2D, cu rețea de discretizare Ref×0,7 și cea trasată prin simulări cu modelul 3D, obținut prin multiplicarea pe mai multe nivele a aceleiași rețele 2D - Ref×0,7.

Comparația grafică din fig.5.38 validează modelul 3D în raport cu cel 2D.

Caracteristica 3D este mai aproape de realitate întrucât surprinde mai realist distribuția câmpului electric, în comparație cu modelul2D. Cu toate acestea, simularea 3D a durat 153 minute, în comparație cu cea 2D care a durat numai 148sec., deci de aproximativ 60 de ori mai mult! Deși modelul dispozitivului studiat, datorită simetriilor proprii, ar fi putut fi redus la 1/2 în cazul 2D, respectiv la 1/4 în cazul 3D am preferat folosirea întregii sale geometrii tocmai pentru a putea efectua comparații cu un volum mai mare de calcule. La ora actuală sisteme de ordinul 10^6 ecuații sunt comune, punându-se problema tratării unor modele ce genereaza ecuații cu câteva ordine mai mari.

5.7 Procesarea paralelă

Simularea microsuntului capacitiv din cap.5.6 a fost efectuată atât pentru modelele 2D cât și pentru cel 3D utilizând un algoritm serial (v. pseudocodul 1). Principalul modul consumator de timp este cel de rezolvare a sistemului de ecuatii liniare. El este apelat la fiecare iterație din setul de iterații neliniare, parcurse pentru fiecare dintre tensiunile de acționare V_a . De aici și timpii considerabili de rezolvare menționați în subcapitolul precedent. Metoda de rezolvare implicită utilizată de FreeFEM++ este una directă, selectată din biblioteca matematică UMFPACK, metodă care are avantajul de a fi robustă, dar mare consumatoare de resurse de memorie, fapt constatat în principal la modelul 3D. O clasă de metode alternativă de tratare numerică ar fi cea a metodelor iterative, care, pe lângă avantajul unui consum mult mai mic de memorie, este un candidat bun pentru implementări pe procesoare paralele [61]. Dezavantajul major al metodelor de rezolvare iterative, mai ales pentru probleme de mari dimensiuni, constă în lipsa de robustețe. O metodă hibridă, care împacă în mod rezonabil consumul de resurse de calcul cu dezideratul de stabilitate numerică este metoda descompunerii domeniilor, metodă ce este foarte bine implementabilă și pe procesoare paralele [61], [157], [144], [70].

Ideea centrală este aceea de a împărți problema definită pe un domeniu dat, în mai multe probleme ce se pot rezolva, separat, pe partiții ale domeniului inițial. Condiția principală este de a se asigura convergența către aceeași limită pentru soluțiile de pe zonele comune ale subdomeniilor în cauză. Aceste zone pot fi porțiuni comune, în cazul în care interioarele subdomeniilor au zone de intersecție nevidă (partiționare cu suprapuneri), sau frontiera comună dintre ele, atunci când subdomeniile sunt adiacente (partiționare disjunctă).

Clasa de aplicații dedicată rezolvării ecuațiilor cu derivate parțiale (PDE) poate căpăta diverse aspecte, în funcție de modul de abordare ([144],[157]) și anume:

- descompunerea structurilor de date utilizate astfel încât procesul de rezolvare să poată fi "paralelizat", adică distribuit spre execuție mai multor procesoare (v. subcap.1.4);
- descompunerea domeniului fizic al problemei în regiuni disjuncte, sau parțial suprapuse ce pot fi tratate matematic cu ecuații diferite; în acest caz se acordă importanță asigurării condițiilor de continuitate la trecerea prin frontierele de separație pentru componentele normale şi/sau tangențiale ale mărimilor vectoriale implicate;
- descompunerea unui sistem de ecuații liniare de mari dimensiuni, respectiv a vectorului soluție, în mai multe blocuri mai mici ce pot fi tratate separat; sistemul de ecuații inițial este rezultatul discretizării ecuațiilor PDE pe întreg domeniul de calcul; acest caz este de fapt o descompunere a metodei de rezolvare a sistemului de ecuații în "submetode" mai mici ce pot fi tratate în paralel; în acest sens, se poate menționa folosirea soluțiilor pe domeniile descompuse ca precondiționare pentru accelerarea convergenței unor metode de tip Krylov.

De remarcat este faptul că, în cazul aplicațiilor științifice profesionale, toate cele trei aspecte de mai sus pot fi întâlnite simultan în același pachet software dedicat. In consecință, în contextul actual al dezvoltării metodelor de procesare paralelă (v. cap.1), metodele de descompunere a domeniului sunt atractive din următoarele motive principale:

- implementare naturală și deci eficientă pe sisteme de procesare paralelă;
- oferă metode de tratare mai simplă a domeniilor cu geometrie complicată;
- algoritmii corespunzători proprietăți de convergență foarte bune tinzând spre complexitate $\mathcal{O}(N)$ pentru o problemă cu N grade de libertate ([144]).

In cazul de față, ținând cont că este domeniul care solicită cel mai mare efort de calcul, am descompus problema de electrostatică 2D în 4 subdomenii care se suprapun pe câte o porțiune îngustă, aleasă convenabil pentru a permite descrierea ușoară a geometriilor asociate (v. fig.5.39). Alegerea corespunde numărului de procesoare reale disponibile pe calculatorul pe care am efectuat simulările (v.5.6.3). Procesul de rezolvare este unul iterativ. Forma tare a problemei presupune rezolvarea, la fiecare iterație (k), a ecuației eliptice (v. cap. 3.3) de ordin doi,



Figura 5.39: Subdomeniul Ω_i . Frontierele suprapuse cu subdomeniile învecinate Ω_j și Ω_p sunt Γ_{ij} , respectiv Γ_{ip} . Porțiuni din frontiera întregului domeniu al PE sunt Γ_{Eb_i} pentru lama elastică, respectiv Γ_{Ed_i} pentru bara centrală.



Figura 5.40: Domeniul Ω_E împărțit în patru subdomenii care se suprapun pe câte o fâșie îngustă în jurul fiecărei frontiere comune.

pe fiecare subdomeniu Ω_i :

$$-\nabla \cdot \left(\varepsilon \nabla v^{(k)}\right) = 0 p e \Omega_i, \tag{5.80}$$

$$v^{(k)} = v_j^{(k-1)} \text{ pe } \Gamma_{ij}, \text{ pentru } j = 1, \cdots, na_i$$
 (5.81)

$$\boldsymbol{\psi}^{(k)} = \boldsymbol{V}_a \text{ pe } \boldsymbol{\Gamma}_{\boldsymbol{E}_{d_i}}, \tag{5.82}$$

$$v^{(k)} = 0 \text{ pe } \Gamma_{E_{b_i}},$$
 (5.83)

$$\frac{\partial v}{\partial n} = 0 \ pe \Gamma_{E_{a_i}},$$

unde na_i este numărul de subdomenii vecine cu Ω_i . Formele slabe sunt similare conform celor discutate în subcap.5.4.2.

In fig. 5.40 pot fi observate limitele celor patru subdomenii în care a fost împărțit domeniul problemei electrice.

In fig. 5.41 - 5.43 se pot observa evoluțiile liniilor echipotențiale la frontiera dintre subdomeniile Ω_0 și Ω_1 .



Figura 5.41: Simulare prin metoda descompunerii domeniului. Problema de electrostatică 2D, a microșuntului este rezovată în paralel pe patru procesoare, fiecare ocupându-se de câte un subdomeniu (v.fig.5.40). Inițial, pe frontierele comune sunt impuse condiții Dirichlet omogene. Potențialul de acționare a lamei elastice este $V_a = 15V$. Se pot observa liniile de potențial constant din vecinătatea frontierei comune dintre subdomeniile Ω_0 și Ω_1 , corespunzătoare primei iterații. Eroarea în sens euclidian dintre valorile din stânga și dreapta frontierei comune este de ordinul 10^{-2}



Figura 5.42: Simulare prin metoda descompunerii domeniului pentru problema de electrostatică 2D a microșuntului capacitiv. Sunt figurate liniile de potențial constant din vecinătatea frontierei comune dintre subdomeniile Ω_0 și Ω_1 , corespunzătoare iterației a șasea. Se poate observa evoluția acestora față de prima iterație. Eroarea în sens euclidian dintre valorile din stânga și dreapta frontierei comune este de ordinul 10^{-3}

Capitolul 5. Aspecte computationale



Figura 5.43: Simulare prin metoda descompunerii domeniului pentru problema de electrostatică 2D a microșuntului capacitiv. Sunt figurate liniile de potențial constant din vecinătatea frontierei comune dintre subdomeniile Ω_0 și Ω_1 , corespunzătoare ultimei iterații. Eroarea în sens euclidian dintre valorile din stânga și dreapta frontierei comune este de ordinul 10^{-4} .

Din punctul de vedere al efortului de calcul, s-a dovedit a fi relevantă comparația evoluțiilor rezolvării problemei de elctrostatică din cadrul unei singure iterații neliniare a algoritmului (1). Am rezumat datele relevante în tab.5.7. Pentru paralelizare am folosit modelul MPI - prezentat în subcap.1.6.3. FreeFEM++ are un modul (ff-mpirun) și un set de funcții dedicate paralelismului conform bibliotecii asociate modelului amintit. De data aceasta testele au fost efectuate pe același Laptop Dell Inspiron, dotat cu un procesor i7-8550U @1,80Ghz și 16Gb RAM, dar sub sistemul de operare LINUX, varianta UBUNTU V18.10 (https://wiki.ubuntu.com/CosmicCuttlefish). Procesarea paralelă a fost gestionată de Mpi V4.0.

Pe baza tab.5.7 și tab.5.7 se pot face următoarele observații:

- inițial, finețea de discretizare a unui domeniu este stabilită în FreeFEM++
 prin impunerea numărului de noduri pe fiecare frontieră a acestuia; rețeua
 de discretizare este generată apoi automat; consecința este că rețelele de
 discretizare asociate subdomeniilor rezultate din descompunere sunt sensibil
 mai fine decât rețeaua de discretizare care acoperă zona corespondentă din
 modelul întreg (v. tab.5.7);
- cu toate acestea, timpul de calcul consumat de implementarea paralelă este mai mic decât varianta serială, acest timp fiind de fapt impus de cel mai lent dintre procesele paralele, la care se adaugă timpul consumat de secvența ce

$\text{Procesare} \rightarrow$	Serial			Paralel			
Ordin FF	DOF	t	$t_{\rm spec}$	DOF	t	$t_{\rm spec}$	S
		$[\mathbf{s}]$	$[\mu s/DOF]$		$[\mathbf{s}]$	$[\mu s/DOF]$	
P1	4381	0,19	43	7073	0,13	18	2,39
P2	16492	0,86	52	26912	0,62	23	2,26

Tabela 5.7: Timpii de rulare pentru o iterație a problemei electrice. Comparativ între algoritmul serial și cel paralel (descompunerea domeniilor), pentru elemente finite de ordin 1 (P1), respectiv ordin 2 (P2). Numărul gradelor de libertate se referă la suma gradelor de libertate utilizate de toate procesele implicate. t_{spec} reprezintă timpul specific pe grad de libertate - determinat cu relația t/DOF.

	P0	P1	P2	P3	Total
P1	2811	719	720	2823	7073
P2	10847	2631	2635	10799	26912

Tabela 5.8: Numărul de grade de libertate pe fiecare din cele patru subdomenii atribuite celor patru procesoare paralele.

rulează serial;

- pentru a putea efectua o comparație reală am determinat *timpul specific*, notat cu t_{spec} și determinat ca raportul dintre timpul total consumat de proces și numărul de grade de libertate (DOF) ce stabilesc dimensiunea problemei atribuită procesorului respectiv.
- se poate constata că creșterea de viteză **S** dereminată ca $t_{spec_{serial}}/t_{spec_{paralel}}$ rămâne practic invariantă la creșterea de aproximativ patru ori a dimensiunii problemei.
- dacă se folosește expresia (1.4) dată de Legea lui Amdahl pentru S = 2, 3 și p = 4 procesoare, rezultă aproximativ ponderea porțiunii seriale a algoritmului asociat metodei de descompunere a domeniilor, și anume $\alpha = 25\%$.

5.8 Concluzii referitoare la aspecte computaționale

In prezentul capitol au fost parcurse practic, pentru un microșunt capacitiv, etapele procesului de modelare multifizică prezentate în cap.2.1:

- modelare conceptuală, soldată cu constucția unui domeniu geometric simplu, dar relevant;
- modelarea analitic-aproximativă printr-o problemă 1D care pemite încadrarea corectă fizic și matematic a comportamentului dispozitivului studiat;
- modelare matematică pe parcursul căreia au fost exprimate *formulările tari* prin scrierea ecuațiilor tipice legilor ce guvernează câmpul electrostatic, respectiv mecanica statică; de aici au rezultat mărimile ce trebuie determinate pentru caracterizarea anumitor aspecte funcționale ale microșuntului, precum și spațiile vectoriale în care acestea vor fi căutate; urmează apoi tratarea în sens distribuțional a problemei ce duce la generarea *formulării slabe* a acesteia, constituită din ecuații cu drivate parțiale împreună cu spațiile Hilbert de care aparțin soluțiile;
- modelare numerică utilizând metoda elementelor finite; prin precizarea spațiilor funcționale pentru soluții se pot alege elementele finite, conforme cu spațiile precizate, potrivite pentru discretizarea geometrică și numerică a formulării slabe.
- validarea modelului. Nu am intenționat în acest capitol compararea rezultatelor simulărilor cu rezultate practice sau teoretice de referință: am preferat evidențierea câtorva din problemele ce pot apărea pe parcursul simulării numerice, și unele din cauzele care le pot produce. Mi s-a părut importantă evidențierea multitudinii de aspecte (fără a avea pretenția că au fost epuizate toate) ce trebuiesc luate în considerare în procesul de modelare multifizică: ecuațiile formulării tari sunt fidele fenomenelor fizice studiate, reflectând continuitatea acestora. În momentul migrării în lumea numerică, ce nu le poate trata, deocamdată, decât discontinuu, prin discretizare, aceleași ecuații sunt afectate de erori numerice, care, dacă nu sunt controlate corect duc la rezultate imprevizibile sau incoerente. Aici stăpânirea matematicii, în speță a analizei funcționale, este de o extremă importanță pentru înțelegerea și tratarea corectă a erorilor ce, în mod inevitabil fac parte din universul numeric.

Tot în acest capitol, sunt evidențiate legăturile cu principiile elementului finit expuse în cap. 4: fără o bună înțelegere a acestora, modelarea numerică prin această metodă se reduce la un interesant exercițiu de utilizare a unor pachete software dedicate. Este motivul pentru care am folosit **FreeFEM**++, software care nu poate fi abordat nicicum fără o înțelegere prealabilă a principiilor de bază ale analizei funcționale, precum și ale elementului finit.

• In urma rezolvării problemei 3D asociate șuntului capacitiv, a devenit evident faptul că, chiar și pentru un model de o complexitate redusă în contextul tehnic actual, algoritmul serial este un consumator important de spațiu și timp în universul digital; una din variantele ce are mare potențial de a reduce timpul de calcul, este procesarea paralelă, lucru dovedit pe rulările efectuate, ce-i drept cu numai patru procesoare, pentru problema 2D: aici trebuie avute în vedere atât matematica ce face posibilă paralelizarea algoritmilor seriali, precum și modelele software de implementare a acestora.

Capitolul 5. Aspecte computationale

Capitolul 6

Concluzii

6.1 Concluzii generale

Primul capitol conturează contextul tehnic actual al lumii digitale.

Lumea digitală a devenit, pe departe, cea mai importantă resursă de informație solicitată în orice domeniu: de la cea mai simplă rețetă de bucătărie, până la complexe rețete tehnico-științifice.

De aici tendința, aparent de neoprit, pentru posibilitatea de a prelucra și a stoca un volum cât mai mare de date, într-un timp cât mai redus, respectiv într-un spațiu cât mai redus. Comunitatea științifică are un dublu rol: pe de o parte cea de consumatoare masivă de resurse de calcul, iar pe de altă parte cea de creatoare și dezvoltatoare ale acelor resurse de calcul. Din punct de vedere hardware, o primă etapă de dezvoltare a fost aceea în care cresterea frecventei de lucru per procesor a fost limitată de cantitatea mare de căldură degajată de acest regim de lucru. Acum ne aflăm în plină dezvoltare a etapei următoare și anume cea a folosirii a cât mai multe procesoare în paralel. În primul capitol (subcap. 1.2 - 1.3) sunt evidentiate principalele jaloane care au marcat evolutia arhitecturilor hardware ale procesoarelor, arhitecturi care, în ultimele lor variante, conțin ele însele unități logice ce pot rula seturi de instrucțiuni în paralel. Mai mult, procesorul ce conduce unitatea centrală a unui calculator poate lucra, pentru rezolvarea aceleiași probleme, în paralel cu procesorul grafic al aceluiași sistem. Dezvoltarea rapidă a arhitecturilor hardware, a atras după sine, deschiderea către aplicatii software noi, materializate prin algoritmi dedicați, preocupați mai puțin de a face economie de memorie, ci mai mult de cresterea si contolul acuratetii calculelor numerice, precum și de reducerea timpului de calcul. Modelul clasic al programării structurate, utilizat pentru implementarea algoritmilor seriali a fost înlocuit cu modele ale programării

paralele, prezentate în subcap. 1.4. Este subliniat aici faptul că modalitățile de paralelizare ale unui algoritm nu se pot înscrie unor așa zise rețete, ci, mai degrabă, trebuie să respecte anumite principii, având ca scop esențial îndeplinirea unor criterii generale de eficiență ale programului. Tot în acest capitol sunt prezentate două metodologii de paralelizare:

- una datorată din 1995 lui Foster([69]) și descriptibilă prin secvența: "Partition -> Communication -> Agglomeration -> Mapping" (prescurtat PCAM);
- cealaltă datorată lui Mattson [109] ce propune în 2004 o sistematizare a procesului de programare paralelă, grupând multitudinile de tehnici tipice acestui proces (v. fig.1.16) în grupe de *modele (patterns)* definite la nivel abstract.

O concluzie importantă a subcap. 1.4 este faptul că unul din cele mai potrivite modele de paralelizare ale unui algoritm complex, este un model hibrid de programare paralelă, reprezentativ pentru majoritatea cazurilor ce apar în practică, model ce poartă denumirea de "Single Program Multiple Data" (SPMD) - v. fig. 1.18.

O altă concluzie ar fi faptul că algoritmii nu pot fi paralelizați în totalitate, în fiecare dintre aceștia existând segmente seriale. De aici apar limitări ale programării paralele evidențiate pe scurt, în subcap.1.5, prin cuantificările efectuate de Amdahl, respectiv de Gustavson.

Pentru ca tabloul să fie complet, era necesară și prezentarea instrumentelor software utilizate pentru implementarea practică a principiilor descrise mai sus, prezentare realizată în subcap. 1.6. Una din concluziile importante al subcapitolului ar fi că trecerea de la implementarea algoritmilor secvențiali, la implementări paralele, nu se poate face (cel puțin la ora actuală) cu ajutorul unor "compilatoare automate", dedicate acestei conversii. Putem desprinde însă două clase mari de instrumente software în domeniul aplicațiilor paralele, după modul de gestionare a datelor: procesare a datelor stocate în zone de memorie partajată (bibliotecile Pthreads și OpenMP), respectiv procesare a datelor stocate în zone de memorie distribuite, conectate prin intermediul unui sistem de mesaje (modelul MPI).

Concluzia principală a primului capitol privește problemele științifice și inginerești actuale, de mare complexitate, necesitând, pentru a fi rezolvate, o putere de calcul pe măsură. În consecință, s-a dezvoltat o nouă disciplină numită "Calcule Științifice de Inaltă Performanță" (High Performance Scientific Computing, prescurtat - HPSC). Caracteristica esențială a HPSC este interdisciplinaritatea: abordarea ei presupune cunoștințe de fizică, inginerie, matematică, metode numerice, precum și de tehnologia informației

Este motivul pentru care capitolul al doilea este dedicat aspectelor legate de domeniile fizicii implicate în activitatea de cercetare desfășurată pe parcursul elaborării prezentei teze și anume: teoria câmpului electromagnetic și teoria elasticității în medii liniare. Pentru fiecare din aceste domenii, sunt prezentate pe scurt mărimile fizice caracteristice, proprietăți de material, mărimi fizice de măsură, sistemul legilor fundamentale ce guvernează domeniul fizic respectiv, împreună cu legile de conservare caracteristice. Concluzia esențială a capitolului se referă la modul în care aceste mărimi caracterizează domeniile fizice (local, global), cu implicații directe în scrierea sistemelor de ecuații caracteristice legilor fundamentale, de material, respectiv de conservare: membrul stâng al acestor ecuații este o reflectare a proprietăților de material din domeniul fizic considerat, iar membrul drept conține mărimile ce descriu sursele de câmp, fie interne, fie externe, caz în care sunt reprezentate prin condiții de frontieră.

Capitolul al treilea, este dedicat, în mod natural, aspectelor matematice implicate în modelarea problemelor multifizice. In acest tip de probleme coexistă ecuații, care chiar dacă sunt din aceeași categorie (de ex. exprimă legi fundamentale, sau de conservare) ele sunt scrise în funcție de mărimi fizice diferite, pot fi aplicate pe spații geometrice de spețe diferite, și sunt măsurabile cu unități de măsură diferite. Sunt cauze pentru care un astfel de spațiu, compus din mai multe domenii fizce, este dificil de tratat în această formă, cu un aparat matematic unitar din categoria analizei matematice clasice.

Soluția pe care matematica o oferă pentru analiza fenomenelor multifizice (și nu numai) este analiza funcțională, prin conceptul ei fundamental de *spațiu abstract*. Apartenența unui element la acest spațiu presupune respectarea unor anumite axiome matematice, dar nu impune nicio constrângere referitoare la semnificația fizică elementului, respectiv mulțimii de elemente în cauză. Gradul de abstractizare al acestor spații, ține cont totuși de o proprietate ce caracterizează toate mărimile din lumea fizică macroscopică ce ne înconjoară și anume *continuitatea*. Noțiunea de continuitate este însă mult mai generală decât cea specificată de analiza matematică. In analiza funcțională continuitatea este asimilată termenilor de "continuitate aproape peste tot", sau "continuitate în sens generalizat", termeni descriși pe baza definirii noțiunii de convergență în sensul dat de teoria distribuțiilor (v. relațiile (3.2) - (3.3)). De aici rezultă mai apoi noțiunea de derivare în sens generalizat (distribuțional), respectiv noțiunile de gradient, rotor, divergență generalizate. Formulările matematice asociate spațiilor fizice reale sunt caracterizate de ecuatii ce constituie forma tare a acestor formulări. Spațiile funcționale abstracte pot fi generate, pentru un anumit domeniu, prin seturi de funcții de bază ce satisfac condiții nule pe limitele domeniului respectiv (funcții de test). Trecerea din spațiul fizic real în spațiul funcțional abstract se realizează prin proiecția ecuațiilor caracteristice formei tari pe setul de funcții de test, obținându-se astfel formularea slabă a problemei multifizice de rezolvat - de forma (3.222). Forma slabă a problemelor multifizice, presupune existența în membrul stâng a unei funcționale biliniare - $\mathcal{A}(u, v)$ - respectiv în membrul drept a unei funcționale liniare - $\mathcal{F}(v)$.

Funcționala corespunzătoare formulării slabe este întodeauna derivabilă în sens generalizat, deci continuă. Pe baza formulării slabe se demonstrează existența și unicitatea soluțiilor (v.subcap.3.2.9). Cum ecuațiile de rezolvat sunt diferențiale, soluțiile lor aparțin unui spațiu funcțional Sobolev (v.subcap.3.2.2).

Concluzia esențială a capitolului subliniază importanța identificării cadrului funcțional al problemei, adică a spațiilor corespunzătoare funcționalei biliniare $\mathcal{A}(u, v)$ și de funcționala liniara $\mathcal{F}(v)$. Un ghid remarcabil pentru această identificare, este secvența exactă de Rham (3.128), secvență ce reflectă izomorfismul existent între spațiile implicate în acest complex, precum și metoda de construcție a unei clase de funcții din alta.

Rezolvarea ecuațiilor formei slabe poate fi efectuată în majoritatea covârșitoare a cazurilor numai apelând la tehnici numerice ceea ce presupune trecerea din lumea continuă în universul digital, discontinuu. Este tema capitolului patru al prezentei lucrări. Trecerea susmenționată nu se poate face dacât pe baza aproximării spațiului geometric al problemei multifizice cu un set de celule de forme geometrice simple, cel mai frecvent triunghiuri sau tetraedre. Pe acest domeniu aproximativ, de dimensiune finită, se definește un spațiu funcțional izomorf cu cel al funcțiilor de test, numit spațiul funcțiilor de formă.

Soluția se aproximează în fiecare celulă cu un polinom de grad p, cel mai simplu fiind cazul în care p = 1. Valorile soluției numerice se determină în N noduri ale rețelei de discretizare. Aici prin *nod* se înțelege o noțiune abstractă prin care se identifică domeniul spațial (vertex, latură, față, sau volum) asociat funcției de formă, domeniu căruia i se atribuie și funcția *variabilă nodală*. Variabilele nodale alcătuiesc gradele de libertate ale problemei, și determină în mod univoc soluția numerică, soluție ce păstrează proprietatea de continuitate pe întreg domeniul de calcul. Spațiul gradelor de libertate, dual al spațiului funcțiilor de formă, este un element esențial al metodei. Soluția numerică, reprezentată prin valorile gradelor de libertate, se obține prin rezolvarea unui sistem de N ecuații algebrice liniare, sistem generat din forma slabă discretă $\mathcal{A}(u_h, v_h) = \mathcal{F}(v_h)$. Alegerea corectă a cadrului funcțional, implică determinarea unei soluții numerice care este grad-, rot-, sau div-conformă. Ca urmare, și elementele finite utilizate, după caz, vor fi conforme cu operatorii diferențiali menționați. Aceste elemente sunt strâns legate de *forma diferențială* (de ordin 1, 2 sau 3) caracteristică mărimii locale prin care se descrie problema fizică. Așa cum spațiile Sobolev grad-, rot-, sau div-conforme, sunt conectate între ele prin secvența exactă de Rham, tot așa și elementele finite conforme corespunzător, respectă aceleași secvență - discretă de data aceasta (v.secvența (4.158)). Concluzia esențială este că forma diferențială, prin imaginea secvenței de Rham, reprezintă legătura esențială între aspectele fizice, matematice și numerice ale problemelor multifizice.

6.2 Lucrări publicate.

In capitolul 5 au fost parcurse practic, pentru un microșunt capacitiv, etapele procesului de modelare multifizică prezentate în cap.2.1. Aplicația este una care a constituit subiectele de cercetare ale proiectelor "ToMeMs" PN-II-PT-PCCA-2011-3 (http://mems.lmn.pub.ro/), respectiv din cadrul programului "Excellence Research Grant Program", UPB – GEX 2017. Ctr. Nr. 05/25.09.2017, desfășurate în cadrul *Laboratorului de Modelare Numerică* al Facultății de Inginerie electrică, al Universității Politehnica București. Tematica prezentei lucrări a generat, în contextul proiectelor amintite, mai multe lucrări prezentate în cadrul unor conferințe internaționale de prestigiu:

- "An Object Oriented Data Structure Designed for Multiphysics Simulations on Parallel Computers"; autori: Mihai Popescu, Aurel-Sorin Lup, Gabriela Ciuprina, Daniel Ioan; lucrare prezentată la conferința ATEE-2015, București [129]. Lucrarea propune o structură de date orientată pe obiecte ce poate fi utilizată pentru implementarea algoritmilor specifici simulării multifizice. Structurile de date au fost implementate în C++ și testate parțial pe simularea unui microcomutator MEMS. Lucrarea a reprezentat un început de drum pentru generarea unei biblioteci de clase C++ dedicate modelării multifizice;
- "Using Object Oriented Data Structures for Optimizing MEMS Devices on Parallel Computers"; autori: Mihai Popescu, Aurel-Sorin Lup, Ruxandra Bărbulescu, Gabriela Ciuprina și Daniel Ioan; lucrare prezentată la The XVII International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering, Valencia, Spania, 2015 [127]. Lucrarea prezintă structura de date orientată pe obiecte propusă inițial în [129], dar mult

îmbunătățită, în sensul că permite, împărțirea unui domeiu fizic în subdomenii și tratarea acestora pe procesoare paralele, fără a impune o comunicație interprocese intensivă, deci consumatoare de timp. Sunt prezentate rezultate ale rulărilor pe cluster-ul Laboratorului de Modelare Numerică (v.subcap.1.7);

- "Parametric Multiphysics 3D Modelling of a Bridge Type MEMS Capacitive Switch"; autori: Aurel-Sorin Lup, Gabriela Ciuprina, Mihai Popescu şi Daniel Ioan; lucrare prezentată la The XVII International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering, Valencia, Spania, 2015 [104]. Lucrarea prezintă simulări multifizice 3D ale unui microșunt capacitiv, realizate cu pachetul software Comsol Multiphysics V4.4. Simulările au fost efectuate atât în regim static, precum şi dinamic pentru diverse tensiuni de acționare, dar şi pentru diferite lungimi ale membranei elastice (v.cap.5);
- "HPC in Multiphysics Analysis of RF-mems Capacitive Switches"; autori: Sorin Lup, Gabriela Ciuprina, Mihai Popescu și Daniel Ioan; lucrare prezentată la SIELA 2018 - The XXth International Symposium on Electrical Apparatus and Technologies, Burgas, Bulgaria [105]. Lucrarea prezintă o comparație între simulările neliniare de regim static efectuate pe de o parte cu pachetul software Comsol Multiphysics și pe de altă parte cu **FreeFEM**++ în scopul de a determina tensiunea de minimă acționare a unui microcomutator capacitiv. Iterațiile Comsol au fost efectuate serial, pe când cele cu **FreeFEM**++ au fost efectuate în paralel pe 8 procesoare;
- "A Parallel Algorithm for Multipysics Analysys of RF-MEMS"; autori: Mihai Popescu, Sorin Lup, Gabriela Ciuprina, Daniel Ioan și Ruxandra Bărbulescu, SCEE 2018 – The 12th International Conference on Scientific Computing in Electrical Engineering, 23 – 27 September 2018, Sicilia, Italia [128]. Lucrarea reia tema prezentată în [105], dar cu o abordare teoretică efectuată cu ajutorul formulărilor slabe. Implementarea cu FreeFEM++ rezultă în acest mod natural, rularea pe procesoare paralele fiind mai eficientă. Comparațiile au fost efectuate cu un model de referință Comsol Multiphysics.

6.3 Contribuții originale

1. Contribuții la elaborarea a trei lucrări prezentate la conferințe cotate ISI, dintre care una ca prim autor:

- "An Object Oriented Data Structure Designed for Multiphysics Simulations on Parallel Computers"; autori: Mihai Popescu, Aurel-Sorin Lup, Gabriela Ciuprina, Daniel Ioan; lucrare prezentată la conferința ATEE-2015, București [129];
- "Parametric Multiphysics 3D Modelling of a Bridge Type MEMS Capacitive Switch"; autori: Aurel-Sorin Lup, Gabriela Ciuprina, Mihai Popescu și Daniel Ioan; lucrare prezentată la The XVII International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering, Valencia, Spania, 2015 [104].
- "Coupled multiphysics-RF reduced models for MEMS"; autori: Gabriela Ciuprina, Daniel Ioan, Aurel-Sorin Lup, Mihai Popescu, Ruxandra Barbulescu, Alexandra Stefanescu; lucrare prezentată la IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES) 2016- Dehli [50].
- 2. Prezentarea sintetică, într-un singur document (cap.1), a celor mai importante instrumente hardware și software împreună cu modelele de programare, care sunt folosite în mod curent pentru programarea de înaltă performanță (High Performance Computing);
- 3. Sintetizarea, într-o ordine logică, a aparatului matematic, foarte complex, care stă la baza acelei laturi a analizei funcționale folosită în tratarea formulărilor slabe ale problemelor multifizice. Sarcina nu a fost tocmai simplă, întrucât literatura matematică în domeniu este foarte bogată, avându-și începuturile în prima parte a sec. XIX. Mai mult, aproape fiecare matematician ce a adus contribuții importante în domeniu, a introdus notații, respectiv demonstrații proprii, care au îngreunat lectura, înțelegerea și mai ales conectarea cu lucrările altor matematicieni, precum și cu realitatea fizică;
- 4. Expunerea într-o ordine coerentă a întreg aparatului matematic ce duce la construcția elementului finit, conform caracterizării sale generale efectuată în primul subcapitol (subcap.4.1). Apoi, folosind acest aparat, sunt specificate modalitățile de construcție ale elementelor finite de ordin I, grad-, rot-, și divconforme, în corespondență cu secvența de Rham discretă (4.158), discutată în ultima parte (subcap.4.1.8) a capitolului dedicat aspectelor numerice.

Cu toate că proiectele materializate prin lucrările publicate și expuse mai sus au avut, prin colaboratorii săi și aspecte fizic practice, lucrarea de față nu și-a propus compararea rezultatelor simulărilor cu rezultate practice sau teoretice de referință, comparații care au fost dealtfel subiect al lucrărilor prezentate în conferințele mentionate. În lucrare am preferat evidențierea câtorva din problemele ce pot apărea pe parcursul simulării numerice, și unele din cauzele care le pot produce. Este importantă evidentierea multitudinii de aspecte (fără a avea pretenția că au fost epuizate toate) ce trebuie luate în considerare, pe parcursul procesului de modelare multifizică: ecuațiile formulării tari sunt fidele fenomenelor fizice studiate, reflectând continuitatea acestora. Trecerea în universul funcțional al formulărilor slabe este necesară ca metodă ce reprezintă un pas important spre formularea numerică. Acestă trecere relaxează condițiile de existență ale proprietății de continuitate, permitând analiza cu un aparat unitar a unei largi varietăți de domenii fizice. In momentul migrării în universul numeric, ce nu poate trata ecuațiile decât pe portiuni, prin discretizare, calculele sunt afectate de erori numerice, care, dacă nu sunt controlate corect, duc la rezultate imprevizibile sau incoerente. De aici si importanta respectării modului de constructie al elementelor finite (v. secvența De Rham), deoarece conformitatea cu formele diferențiale ce descriu ecuațiile reprezintă garanția păstrării proprietăților de continuitate esențiale inclusiv în universul digital. Aici merită subliniat ceea ce am afirmat mai sus, și anume că formele diferențiale, prin imaginea secvenței de Rham, reprezintă legătura esențială între aspectele fizice, matematice și numerice ale problemelor multifizice.

5. Consider că acest parcurs, prezentat sintetic, dar complet și coerent, este contribuția originală principală adusă de prezenta lucrare. Urmând exemplul din cap.5, pot fi abordate și alte probleme multifizice, urmărind toate aspectele fizice, matematice și computaționale descrise în ordinea firească a aplicabilității lor, pe parcursul prezentei teze. Totodată această lucrare intenționează a atrage atenția asupra importanței însușirii, de către orice specialist, doritor a lucra în domeniul modelării problemelor multifizice, a principiilor analizei funcționale. Stăpânirea matematicii, în speță a analizei funcționale, este de o extremă importanță pentru înțelegerea și tratarea corectă a problemelor multifizice. In încheiere, merită accentuat faptul că fără o bună înțelegere a matematicii menționate, modelarea numerică multifizică se reduce la un interesant exercițiu de utilizare a unor pachete software dedicate, dar fără o reală și concretă utilitate practică.

6.4 Perspective

Cu toate că lucrul la prezenta teză a fost marcat de multe discontinuități și frământări, a existat un moment de la care ideile expuse au început să capete coerență și claritate. Unul dintre cele mai benefice experiențe pentru doctorand și implicit și pentru lucrare, a fost procesul de studiu și selecție al mai multor lucrări valoroase de fizică, matematică pură, matematică aplicată și tehnici de programare paralelă. Toate acestea, în încercarea de a contura o metodă de lucru utilă oricărui specialist aflat la început de drum în universul modelării multifizice. In mod cert calea propusă nu este unică, dar la fel de cert este că și alte metode de lucru ar trebui să poposească cel puțin în fiecare din următoarele universuri:

- 1. mărimi și legi fizice ale tuturor fenomenelor implicate în modelul studiat;
- 2. analiză funcțională;
- 3. modelare numerică;
- 4. High Performance Computing (HPC).

O bună parte din lucrarea de față, este dedicată primelor trei aspecte enumerate mai sus. Deși teza conține informații de natură teoretică (cap. 1) și practică (cap. 5) referitoare la cel de-al patrulea aspect, ramâne în perspectivă completarea acestora cu cercetări și lucrări dedicate următoarelor tematici:

- dezvoltarea unei metode de lucru, bogat documentată cu exemple și studii de caz, ce presupune descompunerea automată a domeniului multifizc studiat în mai multe subdomenii geometrice, în funcție de numărul de nuclee de procesare ("cores") disponibile. Fiecărui subdomeniu i se atașează atât câte o rețea de discretizare, precum și câte o implementare a unei formulări variaționale. În acest mod, întreaga problemă poate fi tratată numeric cu ajutorul mai multor nuclee de procesare, fiecare responsabil de câte un subdomeniu;
- pentru rezolvarea sistemului de ecuații global asamblat cu ajutorul procesoarelor menționate pot fi utilizate:
 - solvere paralele ce utilizează metode directe (de ex. MUMPS [34], PETSc [33]);
 - solvere paralele bazate pe metode semiiterative cum ar fi gradienți conjugați (GC), sau reziduu minim generalizat (GMRES), utilizând ca precondiționare descompunerea domeniilor menționată mai sus.

Capitolul 6. Concluzii

- provocarea la ora actuală este că tehnologia hardware avansează mult mai rapid decât aplicațiile soft dedicate: un software scris acum, nu va rula neapărat mai eficient pe o nouă generație hardware care bate la ușă. Iată motivul pentru care, varii metode de modelare multifizică ar trebui implementate cu ajutorul a cel puțin două sau trei pachete software disponibile în universul HPC, dintre care cel puțin, unul comercial.
- în lucrarea de față am folosit FreeFEM++ care s-a îmbogățit în ultimii doi ani cu o bibliotecă de funcții dedicată metodelor menționate mai sus (https: //doc.freefem.org/documentation/ffddm/);
- un pachet software similar, aflat în plină dezvoltare este și "ONELAB" (https://onelab.info/), care are la bază generatorul de rețea de discretizare Gmsh (https://gmsh.info/) [71] având avantajul unei interfețe grafice proprii mult mai performante decât FreeFEM++ ; beneficiind și de un solver propriu, ONELAB poate fi interfațat cu ușurință cu diverse biblioteci matematice utile în modelarea multifizică prin tehnici de HPC;
- metode de analiză asemanătoare pot fi testate și cu pachete software comerciale, cum ar fi COMSOL Multiphysics ©(https://www.comsol.com); robustețea solverului, posibilitatățile de pre și posprocesare, precum și suportul documentaristic de care se bucură un astfel de software, sunt umbrite de lipsa de flexibilitate impusă de lipsa de acces la codurile sursă ale programului.

Am enumerat mai sus câteva din perspectivele ce se deschid spre continuarea dezvoltării tematicii propuse de prezenta teză. O parte din ele vor constitui subiectul unor cercetări și publicații viitoare ale autorului, cu speranța că ele vor atrage și specialiști mai tineri pe această cale, nu tocmai ușoară, dar cu un deosebit potențial științific, al modelării multifizice.

A1. Cod FreeFem++ problema 2D.

```
******
* model 2D cu notatii identice cu cele din textul lucrarii;
* algoritmul iterativ descris de Pseudocodul 1 din lucrare
* scalari: nici una
* versiunea folosita pentru teza
* scris de: Mihai Popescu
* Ver: 3.1/20.09.2021
// ====== parametri geometrici generali ============
real lb = 280e-6, // lungime membrana
                 // grosime membrana
     hb = 0.4e-6,
                  // latime membrana
     wb = 120e - 6,
     lel = 240e-6,
                  // lungime conductor central
     wel = 120e-6, // latime conductor central
     hel = 0.4e-6, // inaltime conductor central
     td = 0.1e-6, // grosime dielectric peste c.central
     hcpw = 4e-6, // inaltime conductor lateral
lcpw = 240e-6, // lungime conductor lateral
wcpw = 120e-6, // latime conductor lateral
                   // gap initial
     g0 = 3.5e-6;
// ======= excitatii ==============
real VaMax = 31, // tensiune electr. fix - mobil MAX
     VaMin = 5,
                     // tensiune electr. fix - mobil MIN
     Vref = 0;
                     //potential de referinta
// caracteristici de material in domeniul electric
real EPS0 = 1e-9/(36*pi),
    Er = 7.;
// caracteristici de material membrana mobila
real E = 70e9; // Young Al
real niu = 0.35; // coef Poisson Al
real rho = 2700; // kg/mc densitate Al
// parametrii Lame
real mu = E/(2*(1+niu));
real lambda = E*niu/((1+niu)*(1-2*niu));
```

```
// ====== variabile comune celor doua probleme ============
real errdu = -1; // err. relativa a deplasarii max. la iteratia curenta
bool FORCEOK = false; // semnalizare forta de actionare ok
bool TOOSMALL = false;
                     // semnalizare forta de actionare prea mica
bool TOOLARGE = false; // semnalizare forta de actionare prea mare
                    // numar maxim de iteratii neliniare
// numar foarte mic, folosit pt aproximari
int NITMAX = 100;
real dlt = 0.01e-6:
// frontiere domeniul electric (OmegaE) -> etichete - [80 - 89]
int GammaEd = 80.
   GammaEb = 81,
   GammaCPW1 = 82,
   GammaCPW2 = 83:
// frontiere fixe zona aer intre dielectric, electrod mobil si pereti laterali
// front. superioara este a contactului mobil; se va deforma la fiecare iteratie
border GEcpw1(t=0, hcpw) {x=-lb/2; y=-t; label=GammaCPW1; }; // peretele stang
border GEs1(t=0, (lb-wel)/2-td) {x=-lb/2+t; y=-hcpw;} // baza substrat stanga
border db1(t=0, td) {x=-wel/2-td+t; y=-hcpw;};
                                                     // baza dielectric td st.
border GEd1(t=0, hel) {x=-wel/2; y=-hcpw+t; label=GammaEd;}; // st. electrod fix
border GEdt(t=0, wel) {x=-wel/2+t; y=-hcpw+hel; label=GammaEd;}// top electrod fix
border GEdr(t=0, hel) {x=wel/2; y=-hcpw+hel-t; label=GammaEd;} // dr. electrod fix
                                                    // baza dielectric td dr.
border db2(t=0, td) {x=wel/2+t; y=-hcpw;};
border GEs2(t=0, (lb-wel)/2-td) {x=wel/2+td+t; y=-hcpw;} // baza substrat dreapta
border GEcpw2(t=-hcpw, 0) {x=lb/2; y=t; label=GammaCPW2;};// perete dreapta
border GEb(t=-lb/2, lb/2) {x=-t; y=0; label=GammaEb;};
                                                     // contact mobil
   nedeformat
border dr(t=0, hel+td) {x=wel/2+td; y=-hcpw+t;};
                                                     // dreapta dielectric
border dt(t=0, wel+2*td) {x=wel/2+td-t; y=-hcpw+hel+td;}; // top dielectric
border dl(t=0, hel+td) {x=-wel/2-td; y=-hcpw+hel+td-t;}; // stanga dielectric
// frontiere domeniu mecanic (omegaM) -> etichete - [90 - 99]
int GammaMb = 90,
   GammaMl1 = 91,
   GammaMl2 = 92;
// frontiere initiale electrod mobil; mesh-ul se va deforma la fiecare iteratie
border GMl1(t=0, hb) { x=-lb/2; y=hb-t; label=GammaMl1;}; // stanga el. mobil
border GMb(t=0, lb) { x=-lb/2+t; y=0 ;label=GammaMb;}; // baza el. mobil
border GM12(t=0, hb) { x=lb/2; y=t ;label=GammaM12;};
                                                      // dreapta el.mobil
border GMt(t=0, lb) \{ x=lb/2-t; y=hb; \};
                                                       // top el.mobil
// ====== constante utilizate pentru discretizare
                       // nr. intervale pe GammaCPWx
int ncpw = 4,
   nb=ncpw*ceil(lb/hcpw),
                          // nr. interv. pe Gamma_b in raport cu GammaCPWx
   multb = 2, // multiplicator.nr. intervale pe GammaMb/GammaEb
   nes = ncpw*ceil((lb-wel)/2./hcpw), //n interv pe GammaEs in raport cu hcpw
                     // nr. intervale pe bazele dielectricului
   ndb = 4,
                     // nr. intervale pe inaltimile cond. central
   nhel = 6.
   nwel = nb*ceil(wel/lb); // nr. interv. pe fata cond central
                     // nr. interv pe grosime lama elastica
   int nl1 = 4:
verbosity = 0;
```
```
// contor interatii neliniare
// contor iteratii Va
int itk = 0,
     itv = 1;
real VaNew = 0,
                      // tensiune Va la pasul curent
       stepV = 1;
                      // pas de incrementare tensiune Va
real gap0 = hcpw-hel-td,
    margErrPi = 0.03, // marja de eroare initial [%]
                    // gap-uri generate in iteratiile nelin.
     gapnou = 0,
     gapvechi = gap0;
real dltUynou = 0, // deplasare calc. la itk
     dltUyvechi = 0; // deplasare calc. la itk-1
                    // masura timp
real t0, deltaT;
ofstream carVa("VPI_OmEM_P1P1_W.txt"); // fisier date iesire
for ( VaNew = VaMin; VaNew <= VaMax; VaNew = VaNew+stepV ) {</pre>
    // === initializari pentru fiecare valoare Va =========
   itk = 0:
   gapvechi = gap0;
   t0 = clock();
   real WeOld = 0;
                      // energie electrica itk-1
   TOOLARGE = false;
                      // semnalizare forta de actionare prea mare
   TOOSMALL = false; // semnalizare forta de actionare prea mica
   // pas incrementare Va functie de interval
   if ( (VaNew > 10) && (VaNew < 20) ) stepV = 0.5;</pre>
   if (VaNew >= 20) stepV = 0.25;
    cout << endl << "Iteratia " << itv << " pentru Va = " << VaNew << "V";</pre>
    // prb. mecanica ---- revenire la lamela nedeformata -----
   mesh Thm = buildmesh(GMl1(nl1) + GMb(ceil(nb/2)) + GMl2(nl1) + GMt(ceil(nb/1)))
    //Thm = splitmesh(Thm, 2); // rafinare de 4 ori a retelei
   //Thm = splitmesh(Thm, 1+1*(abs(x)>=1.1*wel)); // rafinare in jurul colturilor
    fespace Vmh(Thm, P1); // spatiu functional pb. mecanica, elemente H1 de ord.1
                          // [ux, uy]->deplasari nec.; [wx, wy]->func.de test
   Vmh ux, uy, wx, wy;
    Vmh uycrt = 0,
                          // deplasarea cumulata pt. iteratia curenta pe y si x
       uxcrt = 0;
    Vmh sigma11, sigma22, sigma12;
while ( itk < NITMAX ) {</pre>
       // mesh prb. electric cu deformatie contact mobil
        border GEbDef(t=-lb/2, lb/2) {
              \mathbf{x} = -\mathbf{t};
              y=uycrt(t,0)*(t>=-lb/2+dlt)*(t<=lb/2-dlt);</pre>
              label=GammaEb;
       }; // contact mobil deformat
       // mesh PE deformat la fiecare iteratie
       mesh The = buildmesh( GEcpw1(ncpw) + GEs1(nes) + db1(ndb)
                     + GEdl(nhel) + GEdt(nwel) + GEdr(nhel)
                     + db2(ndb) + GEs2(nes) + GEcpw2(ncpw) + GEbDef(multb*nb)
                     + dr(nhel) + dt(nwel)+ dl(nhel) );
```

```
fespace Veh(The, P1); // spatiu functional PE, elemente H1, de ord. 1
fespace Ph(The, P0);
                        // spatiu funct. PE, elemente L2, ord. 1
                        // uh->potentiale necunoscute, fi->func.de test
Veh uh, fi;
                       // tensiuni maxwelliene din problema electrica
Veh T11, T22, T12;
Veh Ex, Ey;
                        // comp. vector intensitate camp electric
//The = splitmesh( The, 2 ); // rafinare de 4 ori
int rd = The(-wel/2.-td/2., -hcpw+td).region; // pct. interior dielectric
Ph EPS = EPS0*(1 + (Er-1)*(region==rd)); // spatiu constante material
// problema de electrostatica
macro grad(u) [dx(u), dy(u)] //EOM
solve Electro(uh, fi) = int2d(The) ( EPS*grad(uh)'*grad(fi) ) //'
                          + on(GammaEd, uh=VaNew)
                          + on(GammaEb, uh=Vref)
                          + on(GammaCPW1, uh=Vref)
                          + on(GammaCPW2, uh=Vref);
// calcul camp electrostatic
Ex = -dx(uh);
Ey = -dy(uh);
//calcul tensiuni maxwelliene
T11 = EPSO*(Ex*Ex - 0.5*(Ex*Ex + Ey*Ey));
T22 = EPSO*(Ey*Ey - 0.5*(Ex*Ex + Ey*Ey));
T12 = EPS0 * Ex * Ey;
// energia electrostatica
real We = 0.5*int2d(The) ( EPS*grad(uh) '*grad(uh) ); //'
// problema elasticitate, sigma_xy este furnizat de problema electrica
sigma11 = T11;
sigma22 = T22;
sigma12 = T12;
real FeW = 0; // forta determinata din variatia de energie
if ( itk > 0 ) FeW = (We - WeOld)/dltUyvechi;
WeOld = We;
// lungime lama deformata din iteratia precedenta
real lbcrt = int1d(Thm, GammaMb)(1.);
// problema mecanica de elasticitate cu excitatie din tens. maxwelliene
macro div(p, r) ( dx(p) + dy(r) ) //
problem ElasticTM([ux, uy], [wx, wy]) =
int2d(Thm)(
             lambda*div(ux,uy)*div(wx,wy) +
             2.*mu*(dx(ux)*dx(wx) + dy(uy)*dy(wy) +
             0.5*(dx(uy)+dy(ux))*(dx(wy)+dy(wx)))
            )
         - int1d(Thm, GammaMb)( ( sigma11*N.x*wx + sigma22*N.y*wy +
                                   sigma12*(N.x*wy + N.y*wx) ) )
         + on (GammaMl1, ux=0, uy=0)
         + on (GammaMl2, ux=0, uy=0);
// problema mecanica de elasticitate cu excitatie din calcul energetic
problem ElasticFW([ux, uy], [wx, wy]) =
int2d(Thm)(
             lambda*div(ux,uy)*div(wx,wy) +
             2.*mu*(dx(ux)*dx(wx) + dy(uy)*dy(wy) +
```

```
0.5*(dx(uy)+dy(ux))*(dx(wy)+dy(wx)))
           )
        - int1d(Thm, GammaMb)( FeW*wy/lbcrt )
        + on(GammaMl1, ux=0, uy=0)
        + on (GammaMl2, ux=0, uy=0);
// pt. calcul energetic prima it. se face cu tensiuni maxwelliene % \mathcal{T}_{\mathrm{r}}
// daca se folosesc numai tens. maxwelliene atunci
// tb. pusa conditia ( itk \geq 0 )
if ( itk == 0 ) ElasticTM;
else ElasticFW:
// energia mecanica de deformatie
real Wm = 0.5*int2d(Thm)(2*mu*(dx(ux)*dx(ux) + dy(uy)*dy(uy) +
          (dx(uy)+dy(ux))*(dx(uy)+dy(ux))/2)+lambda*div(ux,uy)*div(ux,uy))
              ;
                                    // deformatie curenta
dltUynou = (uy[].min);
gapnou = gapvechi - abs(dltUynou); // gap curent
if ( itk > 0 )
   errdu = (dltUynou - dltUyvechi)/dltUyvechi; // eroare re. deformatie
else
    errdu = -1.;
// afisare rezultate la iteratia neliniara curenta
cout << endl << "DOF = " << 2*ux.n << "\tWe = " << 1e9*We << "nJ; Wm = "</pre>
                 << 1e9*Wm << "nJ; " << endl;
cout << endl << "Itk: " << itk << ";\tDltUy_nou = " << dltUynou</pre>
<< "\tgapnou = " << gapnou*1e6 << "\terr_delta_u = " << errdu*100
<< "%\t raport_g = " << gapnou/gap0 << "; u = " << (g0-gapnou)*1e6 << endl
if (gapnou < 0) {
                      // daca am depasit gap-ul initial, atunci iesire
   if ( itk == 0 ) { // daca se intampla la prima iter., Va e prea mare
        TOOLARGE = true;
        TOOSMALL = false;
    }
    else {
        TOOLARGE = false;
       TOOSMALL = true; // daca nu e prima iteratie, treci la Va urmator
    7
   break;
}
if (abs(errdu) < margErrPi) { // daca am atins err admisibila, iesire</pre>
   TOOSMALL = true;
   TOOLARGE = false;
   plot(Thm, The, wait=1); // pauza de verificare mesh-uri
    break:
7
uycrt = uycrt + uy;
                            // noua forma a lamei deformabile
uxcrt = uxcrt + ux;
dltUyvechi = dltUynou;
                            // memorez deplasare curenta
gapvechi = gapnou;
                            // memorez gap curent
Thm = movemesh(Thm, [x+ux, y+uy]); // deformez mesh lama elastica
itk = itk + 1;
                            // iteratia urmatoare
```

11

}

```
}
   if ( itk >= NITMAX ) { // nu am ajuns la precizie in NITMAX
       TOOLARGE = false;
       TOOSMALL = true;
   }
   // afisare rezultate iteratii neliniare
   cout << "TOOLARGE = " << TOOLARGE << "; TOOSMALL = " << TOOSMALL << endl;</pre>
   cout << "niter = " << itk << "; gap pentru Va = " << VaNew << "V este " <<</pre>
       gapvechi
        << " adica " << gapvechi/gap0 << "din gap ini,; u = " << g0 - gapvechi
        << endl << "======\n";
   deltaT = clock() - t0;
                                 // masor timp pe setul de ietratii neliniare
   // scriere date in fisier
   carVa << itv << "; " << itk << "; " << VaNew << "; " << gapnou*1e6 << "; "</pre>
       << (g0 - gapnou)*1e6 << "; " << deltaT << endl;
   itv++;
   // afisare diverse rezultate
   //plot(Thm, [ux, uy], value=1, wait=1);
   //plot(uh, fill=1, wait=1);
   //plot(Thm,[Ex, Ey], value=1, wait=1);
carVa.flush;
```

A2. Cod FreeFem++ problema 3D.

```
*******
* model 3D cu notatii identice cu cele din textul lucrarii;
* algoritmul iterativ descris de Pseudocodul 1 din lucrare
* scalari: nici una
* versiunea folosita pentru teza
* scris de: Mihai Popescu
* Ver: 2.0/22.09.2021
load "msh3" // modul discretizare 3D
load "medit"
            // modul vizualizare 3D
wb = 120e-6, // latime membrana
    lel = 240e-6, // lungime conductor central
    wel = 120e-6, // latime conductor central
    hel = 0.4e-6, // inaltime conductor central
    td = 0.1e-6, // grosime dielectric peste c.central
hcpw = 4e-6, // inaltime conductor lateral
lcpw = 240e-6, // lungime conductor lateral
    wcpw = 120e-6, // latime conductor lateral
    g0 = 3.5e-6;
                // gap initial
// ====== excitatii ==============
real VaMax = 31, // tensiune electr. fix - mobil MAX
                   // tensiune electr. fix - mobil MIN
    VaMin = 5,
    Vref = 0;
                  //potential de referinta
// caracteristici de material in domeniul electric
real EPS0 = 1e-9/(26*pi),
    Er = 7.;
// caracteristici de material membrana mobil
real E = 70e9; // Young Al
real niu = 0.35; // coef Poisson Al
real rho = 2700; // kg/mc densitate Al
// parametrii Lame
```

```
real mu = E/(2*(1+niu));
real lambda = E*niu/((1+niu)*(1-2*niu));
// ====== variabile comune celor doua probleme
real errdu = -1; // err. relativa a deplasarii max. la iteratia curenta
// ====== constante diverse =============================
bool FORCEOK = false; // semnalizare forta de actionare ok
bool TOOSMALL = false;
                        // semnalizare forta de actionare prea mica
                       // semnalizare forta de actionare prea mare
// numare maxim de iteratii neliniare
bool TOOLARGE = false;
int NITMAX = 100;
                      // numar foarte mic, folosit pt aproximari
real dlt = 0.01e-6;
// frontiere domeniul electric 2D (OmegaE) -> etichete - [80 - 89]
int GammaEd = 80,
   GammaEb = 81,
   GammaCPW1 = 82,
   GammaCPW2 = 83,
   GammaEs = 84:
// fr. fixe zona aer intre dielectric, electrod mobil si pereti laterali izolatori
// fr. superioara este cea a contactului mobil; se va deforma la fiecare iteratie
border GEcpw1(t=0, hcpw) {x=-lb/2; y=-t; label=GammaCPW1; }; // peretele stang
border GEs1(t=0, (lb-wel)/2-td)
                   {x=-lb/2+t; y=-hcpw; label=GammaEs; }; // baza substrat stanga
border db1(t=0, td) {x=-wel/2-td+t; y=-hcpw;};
                                                         // baza dielectric td st
border GEdl(t=0, hel) {x=-wel/2; y=-hcpw+t; label=GammaEd;}; // stanga electrod fix
border GEdt(t=0, wel) {x=-wel/2+t; y=-hcpw+hel; label=GammaEd;};// top electrod fix
border GEdr(t=0, hel) {x=wel/2; y=-hcpw+hel-t; label=GammaEd;}; // dr. electrod fix
border db2(t=0, td) {x=wel/2+t; y=-hcpw;};
                                                         // baza dielectric td dr
border GEs2(t=0, (lb-wel)/2-td)
                    {x=wel/2+td+t; y=-hcpw; label=GammaEs; }; // baza substrat dr.
border GEcpw2(t=-hcpw, 0) {x=lb/2; y=t; label=GammaCPW2;}; // perete dreapta
                                                      // contact mobil nedef.
// dreapta dielectric
border GEb(t=-lb/2, lb/2) {x=-t; y=0; label=GammaEb;};
border dr(t=0, hel+td) {x=wel/2+td; y=-hcpw+t;};
border dt(t=0, wel+2*td) {x=wel/2+td-t; y=-hcpw+hel+td;}; // top dielectric
border dl(t=0, hel+td) {x=-wel/2-td; y=-hcpw+hel+td-t;};
                                                        // stanga dielectric
// frontiere domeniu mecanic 2D (omegaM) -> etichete - [90 - 99]
int GammaMb = 90,
   GammaMl1 = 91,
   GammaMl2 = 92;
// frontiere initiale electrod mobil;
// reteaua de discretizare se va deforma la fiecare iteratie
border GMl1(t=0, hb) { x=-lb/2; y=hb-t; label=GammaMl1;}; // stanga el. mobil
border GMb(t=0, lb) { x=-lb/2+t; y=0 ;label=GammaMb;}; // baza el. mobil
                                                          // dreapta el.mobil
border GM12(t=0, hb) { x=lb/2; y=t ;label=GammaM12;};
border GMt(t=0, lb) { x=lb/2-t; y=hb;};
                                                          // top el.mobil
// ====== constante utilizate pentru discretizare
                            // nr. intervale pe GammaCPWx
int ncpw = 4,
   nb=ncpw*ceil(lb/hcpw),
                            // nr. interv. pe Gamma_b in raport cu inaltimea hcpw
   multb = 2,
                      // multiplicator. nr. noduri pe fr. (GammaMb/GammaEb)
    nes = ncpw*ceil((lb-wel)/2./hcpw), //n interv pe GammaEs in raport cu hcpw
```

```
// nr. interv. pe bazele dielectricului
   ndb = 4.
   nhel = 6,
                      // nr. interv. pe inaltimile cond. central
   nwel = nb*ceil(wel/lb); // nr.interv. pe fata cond.central
                           // nr. interv pe grosime lama elastica
   int nl1 = 4;
verbosity = 0;
int itk = 0,
                    // contor interatii neliniare
   itv = 1;
                    // contor iteratii Va
real VaNew = 0,
                       // tensiune Va la pasul curent
                   // pas de incrementare tensiune Va
    stepV = 0.5;
real gap0 = hcpw-hel-td,
    margErrPi = 0.03, // marja de eroare initial [%]
    gapnou = 0,
                    // gap-uri generate in iteratiile nelin.
    gapvechi = gap0,
        uCrt = 0;
                        // deplasarea curenta
                    // deplasare calc. la itk
real dltUynou = 0,
    dltUyvechi = 0; // deplasare calc. la itk-1
                    // masura timp
real t0, deltaT;
ofstream carVa("caract_VaT.txt");
for ( VaNew = VaMin; VaNew <= VaMax; VaNew = VaNew+stepV ) {</pre>
   // === initializari pentru fiecare valoare Va ==========
   itk = 0;
   gapvechi = gap0;
   t0 = clock();
   real WeOld = 0;
                      // energie electrica itk-1
   TOOLARGE = false;
                      // semnalizare forta de actionare prea mare
                     // semnalizare forta de actionare prea mica
   TOOSMALL = false;
   cout << endl << "Iteratia " << itv << " pentru Va = " << VaNew << "V";</pre>
   // prb. mecanica 2D ---- revenire la lamela nedeformata -----
   mesh Thm = buildmesh( GMl1(nl1) + GMb(ceil(nb/2)) +
                        GMl2(nl1) + GMt(ceil(nb/1)));
   //Thm = splitmesh(Thm, 2); // rafinare de 4 ori a retelei
   //Thm = splitmesh(Thm, 1+1*(abs(x)>=1.1*wel)); // rafinare in jurul colturilor
   // mesh 3D - problema mecanica
   mesh3 Thm3 = buildlayers( Thm, 8, zbound=[-wb/2., wb/2.] );
    fespace Vmh(Thm3, [P1, P1, P1]); // spatiu functional PM, elemente H1, ord.1
   Vmh [ux, uy, uz], [wx, wy, wz]; // [ux, uy, uz]->deplasari nec.;
                                   // [wx, wy, wz]->functii de test
    Vmh [uxcrt, uycrt, uzcrt] = [0, 0, 0]; // deplasarea cumulata
while ( itk < NITMAX ) {</pre>
// mesh prb. electric cu deformatie contact mobil
       t0 = clock();
       border GEbDef(t=-lb/2, lb/2) {
             \mathbf{x} = -\mathbf{t}:
              y=uycrt(t,0,0)*(t>=-lb/2+dlt)*(t<=lb/2-dlt);</pre>
              label=GammaEb:
       }; // contact mobil deformat
```

```
real ymin = uycrt[].min;
   cout << endl << ">>>> " << ymin << " <<< "; // poz.curenta a lamei</pre>
   // mesh PE deformat la fiecare iteratie
   mesh The = buildmesh( GEcpw1(ncpw) + GEs1(nes) + db1(ndb)
                 + GEdl(nhel) + GEdt(nwel) + GEdr(nhel)
                 + db2(ndb) + GEs2(nes) + GEcpw2(ncpw) + GEbDef(multb*nb)
                 + dr(nhel) + dt(nwel)+ dl(nhel) );
// The = splitmesh(The, 1+
11
           1*(sqrt(square(x+wel/2+td)+square(y+hcpw-hel-td))<=hcpw)+
11
            1*(sqrt(square(x-wel/2-td)+square(y+hcpw-hel-td))<=hcpw));</pre>
   int[int] potreftoair = [GammaEb, GammaEs];
   int cpwlayers = 8;
   // compunere layer-e mesh
   // conductor central - mijloc z \ln [-wb/4., wb/4.] - 6 layers
   mesh3 The3cec = buildlayers(The, 6, zbound=[-wb/4., wb/4.]);
   // conductor central - spate z \in [-wb/2., -wb/4.] - 6 layers
   mesh3 The3ceb = buildlayers(The, 6, zbound=[-wb/2., -wb/4.]);
   // conductor central - fatza z \in [wb/4., wb/2.] - 6 layers
   mesh3 The3cef = buildlayers(The, 6, zbound=[wb/4, wb/2]);
   // conductor central - 18 layers
   mesh3 The3ce = The3ceb + The3cec + The3cef;
                                                   // mesh central
   // mesh aer spate z \in [-wb, -3*wb/4] - 2 layers
   mesh3 The3bk1 = buildlayers(The, 2, zbound=[-wb, -3*wb/4]);
   // mesh aer inainte de cond.central z i [-3*wb/4, -wb/2] - 6 layers
   mesh3 The3bk2 = buildlayers(The, 6, zbound=[-3*wb/4, -wb/2]);
   mesh3 The3bk = The3bk1 + The3bk2;
                                                  // mesh spate
   The3bk = change(The3bk, label=potreftoair);
                                                  // propagare atribute
   // mesh aer dupa cond.central z \in [wb/2, 3*wb/4] - 6 layers
   mesh3 The3fr1 = buildlayers(The, 6, zbound=[wb/2, 3*wb/4]);
   // mesh aer fatza z \in [3*wb/4, wb] - 6 layers
   mesh3 The3fr2 = buildlayers(The, 2, zbound=[3*wb/4, wb]);
   mesh3 The3fr = The3fr1 + The3fr2;
                                                   // mesh fatza
   The3fr = change(The3fr, label=potreftoair);
                                                   // propagare atribute
   mesh3 The3 = The3bk + The3ce + The3fr;
                                                  // mesh complet PE
   int diel = The(-(wel+td)/2, -hcpw+td).region; // id regiune dielectric
   int aer = The(-wel-2*td, -hcpw+td).region;
                                                   // id regiune aer;
   fespace Veh(The3, P2); // spatiu functional PE, elemente H1, ord. 2
   fespace Ph(The3, PO); // spatii funct. PE, elem. L2
                          // uh->func.necunoscute, fi->func.de test
   Veh uh, fi;
   Veh T11, T22, T33,
       T12,T13, T23;
                          // tensiuni maxwelliene din problema electrica
   Veh Ex, Ey, Ez;
                          // comp. vector int.camp electric
   Ph EPS = EPSO*(1 + (Er-1)*(region==diel)); // atrib. prop. de material
   // afisare caracteristici mesh + DOF
   cout << endl << "[DOF_E, DOF_M] = " << uh[].n << ", " << 3*ux[].n
        << "; NTe = " << The3.nt << "; NVe = " << The3.nv
        << "; NTm = " << Thm3.nt << "; NVm = " << Thm3.nv << endl;
```

```
// problema de electrostatica
   macro grad(u) [dx(u), dy(u), dz(u)] //EOM
   solve Electro(uh, fi, solver=CG, eps=1e-4) =
      int3d(The3) ( EPS*grad(uh)'*grad(fi) ) //'
                           + on(GammaEd, uh=VaNew)
                           + on(GammaEb, uh=Vref)
                           + on(GammaCPW1, uh=Vref)
                           + on(GammaCPW2, uh=Vref);
   // calcul camp electrostatic
   Ex = -dx(uh);
   Ey = -dy(uh);
   Ez = -dz(uh);
   // calcul tensiuni maxwelliene
   T11 = EPS*(Ex*Ex - 0.5*(Ex*Ex + Ey*Ey + Ez*Ez));
   T22 = EPS*(Ey*Ey - 0.5*(Ex*Ex + Ey*Ey + Ez*Ez));
   T33 = EPS*(Ez*Ez - 0.5*(Ex*Ex + Ey*Ey + Ez*Ez));
  T12 = EPS*Ex*Ey; T13 = EPS*Ex*Ez; T23 = EPS*Ey*Ez;
   real deltatE = clock() - t0; // timp solver electric
   t0 = clock();
// Problema de elasticitate
   real sqrt2 = sqrt(2.);
   macro epsilon(u1, u2, u3) [
      dx(u1), dy(u2), dz(u3),
      (dz(u2) + dy(u3))/sqrt2,
      (dz(u1) + dx(u3))/sqrt2,
       (dy(u1) + dx(u2))/sqrt2 ] //
   macro div(u1, u2, u3) (dx(u1) + dy(u2) + dz(u3)) //
   solve Elastic ([ux, uy, uz], [wx, wy, wz])
       = int3d(Thm3)(
              lambda*div(ux, uy, uz)*div(wx, wy, wz) +
              2.*mu*( epsilon(ux, uy, uz)'*epsilon(wx, wy, wz) )
         ) //>
       - int2d(Thm3, GammaMb)(
               T11*N.x*wx + T22*N.y*wy + T33*N.z*wz +
               T12*(N.x*wy + N.y*wx) +
               T23*(N.y*wz + N.z*wy) +
               T13*(N.z*wx + N.x*wz)
          )
       + on (GammaMl1, ux=0, uy=0, uz=0)
       + on(GammaM12, ux=0, uy=0, uz=0);
                                       // deformatie curenta
   dltUynou = (uy[].min);
   gapnou = gapvechi - abs(dltUynou); // gap curent
   real pg = gapnou/gap0;
                                       // (gap curent)/gap0
   real deltatM = clock() - t0;
                                      // timp solver mecanic
   if ( itk > 1 )
       errdu = (dltUynou - dltUyvechi)/dltUyvechi;
   else
       errdu = -1.;
   // energii pe subdomenii - zona umflari camp electric, zona centrala
   real Wt3d = int3d(The3)(EPS*(Ex*Ex + Ey*Ey + Ez*Ez)/2.);
   real Wc3d = int3d(The3ce)(EPS*(Ex*Ex + Ey*Ey + Ez*Ez)/2.);
```

```
real Wc3ext = int3d(The3fr)(EPS*(Ex*Ex + Ey*Ey + Ez*Ez)/2.) +
                  int3d(The3bk)(EPS*(Ex*Ex + Ey*Ey + Ez*Ez)/2.);
    real Wm3 = int3d(Thm3) (
                    lambda*div(ux, uy, uz)*div(ux, uy, uz)
                    + 2.*mu*( epsilon(ux, uy, uz)'*epsilon(ux, uy, uz) ) //'
                ):
    real Wm3nJ = 1e12*Wm3; // energia in pJ
    real We3nJ = Wt3d*1e12; // energia in pJ
    // afisare rezultate la iteratia neliniara curenta
    cout << endl << "Itk: " << itk << "; DltUy_nou = " << dltUynou</pre>
         << "; gapnou = " << 1e6*gapnou << "; errdu = " << errdu*100
         << "%; rap_g = " << gapnou/gap0 << "; u = " << (gap0-gapnou)*1e6
         << endl;
        gapnou < 0) { // daca am depasit gap-ul initial, atunci iesire
if ( itk == 0 ) { // daca se intampla la prima iteratie, Va e prea
   if (gapnou < 0) {</pre>
            mare
            TOOLARGE = true;
            TOOSMALL = false;
        }
        else {
            TOOLARGE = false;
            TOOSMALL = true;
        }
        break;
    }
    if (abs(errdu) < margErrPi) { // daca am atins err admisibila, iesire</pre>
        TOOSMALL = true;
        TOOLARGE = false;
        [uxcrt, uycrt, uzcrt] = [uxcrt, uycrt, uzcrt] + [ux, uy, uz];
        uCrt = abs(uycrt[].min);
        // scriere rezultate
        carVa << itk << "; " << VaNew << "; " << uCrt*1e6 << "; " << pg</pre>
        << "; " << abs(errdu) << "; " << deltatE+deltatM << "; " << We3nJ
        << "; " << Wm3nJ <<"; " << uh[].n+3*uy[].n << endl;
        break;
    7
    [uxcrt, uycrt, uzcrt] = [uxcrt, uycrt, uzcrt] +
                            [ux, uy, uz]; // noua forma a lamei elastice
    uCrt = abs(uycrt[].min);
                                           // memorez deplasare curenta
    carVa << itk << "; " << VaNew << "; " << uCrt*1e6 << "; " << pg
          << "; " << abs(errdu) << "; " << deltatE+deltatM << ";
          << We3nJ << "; " << Wm3nJ <<"; " << uh[].n+3*uy[].n << endl;
    dltUyvechi = dltUynou;
    gapvechi = gapnou;
    Thm3 = movemesh(Thm3, [x+ux, y+uy, z]);
    itk = itk + 1;
7
if ( itk > NITMAX ) {
    TOOLARGE = false;
    TOOSMALL = true;
7
cout << "TOOLARGE = " << TOOLARGE << "; TOOSMALL = " << TOOSMALL << endl;</pre>
cout << "niter = " << itk << "; gap pentru Va = " << VaNew << "V este "</pre>
    << gapvechi << " adica " << gapvechi/gap0 << "din gap initial" <<
```

Anexa 3. Cod paralel

A3. Cod FreeFem++ DDM - paralel.

```
*******
* model 2D cu notatii identice cu cele din textul lucrarii;
* algoritmul iterativ descris de Pseudocodul 1 din lucrare
* rulat pentru o singura iteratie neliniara - test DDM
* problema de electrostatica ruleaza pe 4 procesoare in paralel
* scalari: nici una
* versiunea folosita pentru teza
* scris de: Mihai Popescu
* Ver: 1.1/20.10.2021
// ====== initializere comunitate procesoare in paralel
mpiComm comm(mpiCommWorld, 0, 0);
if (mpisize != 4)
      cout << endl << "Numar incorect de procesoare!" <<</pre>
                   " Rulati numai cu 4 procese!" << endl;
real lb = 280e-6, // lungime membrana
    hb = 0.4e-6,
                // grosime membrana
    wb = 120e - 6,
               // latime membrana
    lel = 240e-6, // lungime conductor central
    wel = 120e-6, // latime conductor central
    hel = 0.4e-6, // inaltime conductor central
    td = 0.1e - 6,
                 // grosime dielectric peste c.central
    hcpw = 4e-6,
                 // inaltime conductor lateral
    lcpw = 240e-6, // lungime conductor lateral
    wcpw = 120e-6, // latime conductor lateral
                 // gap initial
    g0 = 3.5e - 6;
// ====== excitatii ==========
real VaMax = 15, // tensiune electr. fix - mobil MAX
    VaMin = 15,
                   // tensiune electr. fix - mobil MIN
    Vref = 0;
                  //potential de referinta
// caracteristic de material in domeniul electric
real EPS0 = 1e-9/(36*pi),
   Er = 7.;
```

```
// caracteristici de material membrana mobila
real E = 70e9; // Young Al
real niu = 0.35; // coef Poisson Al
real rho = 2700; // kg/mc densitate Al
// parametrii Lame
real mu = E/(2*(1+niu));
real lambda = E*niu/((1+niu)*(1-2*niu));
real errdu = -1; // err. relativa a deplasarii max. la iteratia curenta
real errpe = 1e-3; // err.euclidiana intre solutiile a 2 subd.adiacente
bool FORCEOK = false; // semnalizare forta de actionare ok
bool TOOSMALL = false; // semnalizare forta de actionare prea mica
bool TOOLARGE = false; // semnalizare forta de actionare prea mare
                       // numar maxim de iteratii neliniare
int NITMAX = 100;
                      // numar foarte mic, folosit pt aproximari
real dlt = 0.01e-6;
// ====== constante utilizate pentru discretizare
                            // n intervale pe GammaCPWx
int ncpw = 4,
   nb=ncpw*ceil(lb/hcpw),
                            // n. interv. pe Gamma_b
   multb = 2, // multiplicator. nr. noduri pe fr. comuna PE - PB
   nes = ncpw*ceil((lb-wel)/2./hcpw), //n interv pe GammaEs
   ndb = 4.
                      // n interv. pe bazele dielectricului
   nhel = 6,
                      // n interv. pe inaltimile cond. central
   nwel = nb*ceil(wel/lb); // n interv. pe fatza cond central
   int nl1 = 4; // n interv pe grosime lama elastica
// ======= initializare ===============================
verbosity = 0;
int itk = 0,
                  // contor interatii neliniare
    itv = 1;
                  // contor iteratii Va
                  // tensiune Va la pasul curent
real VaNew = 0,
    stepV = 1;
                  // pas de incrementare tensiune Va
real gap0 = hcpw-hel-td,
    margErrPi = 0.03, // marja de eroare initial [%]
    gapnou = 0,
                    // gap-uri generate in iteratiile nelin.
    gapvechi = gap0;
real dltUynou = 0, // deplasare calc. la itk
    dltUyvechi = 0; // deplasare calc. la itk-1
real t0, deltaT;
                    // masura timp
string fname = "VPI_P1P1_proc" + mpirank + ".txt";
ofstream carVa(fname); // fisier date iesire
real[int] xrd(mpisize), // coordonate puncte din dielectric
         yrd(mpisize); // pt.setare materiale pt.fiecare retea The
xrd[0] = -wel/2.-td/2.; // coordonate punct din dielectric in The[0]
yrd[0] = -hcpw+td;
xrd[1] = -wel/4.;
                    // coordonate punct din dielectric in The[1]
yrd[1] = -hcpw + hel + td/2.;
xrd[1] = +wel/4.;
                   // coordonate punct din dielectric in The[2]
yrd[1] = -hcpw + hel + td/2.;
xrd[0] = +wel/2. + td/2.; // coordonate punct din dielectric in The[3]
yrd[0] = -hcpw + td;
```

```
// frontiere domeniul electric (OmegaE) -> etichete - [80 - 89]
// frontiera GammaEb este cea a contactului mobil; se deform. la fiecare iter.
int GammaEd = 80, // eticheta cond.fr. potential bara centrala
    GammaEb = 81, // eticheta cond.fr. potential lama elastica
    GammaCPW1 = GammaEb, // eticheta cond.fr. pereti lat.;
    GammaCPW2 = GammaEb,
    Gamma01 = 84, // eticheta cond.fr. intre PeO si Pe1.;
    Gamma12 = 85, // eticheta cond.fr. intre Pe1 si Pe2.;
Gamma23 = 86; // eticheta cond.fr. intre Pe2 si Pe3.;
// fiecare subdom. are cate o fr.de vecinatate in stanga si una in dreapta
int[int] GammaLeft = [GammaCPW1, Gamma01, Gamma12, Gamma23],
         GammaRight = [Gamma01, Gamma12, Gamma23, GammaCPW2];
for ( VaNew = VaMin; VaNew <= VaMax; VaNew = VaNew+stepV ) {</pre>
    // === initializari pentru fiecare valoare Va =========
    itk = 0;
    gapvechi = gap0;
    t0 = clock();
    real WeOld = 0;
                        // energie electrica itk-1
    TOOLARGE = false; // semnalizare forta de actionare prea mare
    TOOSMALL = false; // semnalizare forta de actionare prea mica
    // pas incrementare Va functie de interval
    if ( (VaNew > 10) && (VaNew < 20) ) stepV = 0.5;
    if (VaNew >= 20) stepV = 0.25;
     cout << endl << "Iteratia " << itv << " pentru Va = " << VaNew << "V";</pre>
11
// frontiere domeniu mecanic (omegaM) -> etichete - [90 - 99]
int GammaMb0 = 90,
    GammaMb1 = 91,
    GammaMb2 = 92,
    GammaMb3 = 94,
   GammaMll = 95.
   GammaM12 = 96,
    GammaMb = 97;
    // frontiere init.electrod mobil; reteaua de discr. se def.la fiecare iter.
    border GMl1(t=0, hb) { x=-lb/2; y=hb-t; label=GammaMl1;}; // stanga el. mobil
    border GMb(t=0, lb) { x=-lb/2+t; y=0 ; label=GammaMb;}; // baza el. mobil
border GMl2(t=0, hb) { x=lb/2; y=t ; label=GammaMl2;}; // dreapta el.mobil
    border GM12(t=0, hb) { x=lb/2; y=t ;label=GammaM12;};
                                                              // top el.mobil
    border GMt(t=0, lb) { x=lb/2-t; y=hb;};
    // prb. mecanica ---- revenire la lamela nedeformata -----
    mesh Thm = buildmesh(GMl1(nl1) + GMb(ceil(nb/2)) + GMl2(nl1) + GMt(ceil(nb/1)))
       ;
    fespace Vmh(Thm, P1); // spatiu functional pb. mecanica, elemente H1 de ord.1
                           // deplasarea cumulata pt. iteratia curenta pe y si x
    Vmh uvcrt = 0;
    Vmh ux, uy, wx, wy; // [ux, uy]->deplasari nec.; [wx, wy]->func.de test
    Vmh sigma11=0, sigma22=0, sigma12=0;
while ( itk < NITMAX ) { // studiem o singua iteratie
11
```

```
// ======= MESH ELECTRO =============
t0 = clock();
   mesh[int] The(mpisize);
                              // retele de discretizare locale
   if (mpirank == 0) { // ----- Rank0 ----- Rank0 -----
    // frontiere procesor 0
                                    0],
    real[int, int] CO = [ [-1b/2,
                          [-1b/2,
                                     -hcpw],
                                     -hcpw],
                          \left[-wel/2\right]
                          [-wel/2]
                                     uycrt(-wel/2, 0)],
                                     -hcpw+hel],
                          [-wel/2]
                          [-wel/2]
                                     -hcpw+hel+td],
                          [-wel/2-td, -hcpw+hel+td],
                          [-wel/2-td, -hcpw] ];
    // peretele stang
   border GEcpw1(t=0, CO(0,1)-CO(1,1)) {x=CO(0, 0); y=CO(0,1)-t;
                                       label=GammaCPW1;}
    // baza substrat st.
   border GEs1(t=0, CO(7,0)-CO(1,0)) {x=CO(1,0)+t; y=CO(1,1);}
    // baza dielectric st.
   border db1(t=0, CO(2,0)-CO(7,0))
                                    \{x=CO(7,0)+t; y=CO(7,1);\}
   // stanga electrod fix
   border GEdl(t=0, CO(4,1)-CO(2,1)) {x=CO(2,0); y=CO(2,1)+t;
                                       label=GammaEd;}
   // strat dielectric
   border dl01(t=0, CO(5,1)-CO(4,1)) {x=CO(4,0); y=CO(4,1)+t;
                                       label=Gamma01;}
   border GE01(t=0, C0(3,1)-C0(5,1)) { //aer pana la pozitia lamei elastice
            x = CO(5,0);
            y = CO(5, 1) + t;
            label = Gamma01;
   }
   border GEb0(t=C0(3,0), C0(0,0)) { // portiunea 0 din lama elastica
            x = t;
            y = uycrt(t, 0) * (t \ge CO(0, 0) + dlt);
            label = GammaEb;
   }
    // strat diel.oriz.
   border dh0(t=0, CO(5,0)-CO(6,0)) {x=CO(5,0)-t; y=CO(6,1);}
   // fatza diel.stanga
   border dl(t=0, C0(6,1)-C0(7,1)) {x=C0(6,0); y=C0(6,1)-t;}
    // mesh PEO deformat la fiecare iteratie
   The[0] = buildmesh( GEcpw1(ncpw) + GEs1(nes) + db1(ndb) + GEd1(nhel) +
                       dl01(ndb) + GE01(ncpw) + GEb0(multb*nb/4) +
                       dh0(ndb) + dl(nhel) );
} // ---- end Rank0 -----
 else if (mpirank == 1) { // ------ rank 1 ------
   // frontiere procesor 1
   real[int, int] C1 = [ [-wel/2-td, uycrt(-wel/2-td, 0)],
                         [-wel/2-td, -hcpw+hel],
                                   -hcpw+hel],
                         [td,
                         [td,
                                    uycrt(td, 0)],
                         [-wel/2,
                                    -hcpw+hel],
                                    -hcpw+hel+td],
                         [td,
```

```
[-wel/2-td, -hcpw+hel+td] ];
    border GE10(t=0, C1(0,1)-C1(6,1)) { // fr. stanga x=-wel/2-td
           x = C1(0,0);
           y = C1(0, 1) - t;
           label = Gamma01;
    7
    // strat dielectric
    border dl10(t=0, C1(6,1)-C1(1,1)) {x=C1(1,0); y=C1(6,1)-t; label=Gamma01;}
    // strat dielectric oriz.
    border dh1(t=0, C1(4,0)-C1(1,0)) {x=C1(1,0)+t; y=C1(4,1); }
    // top electrod fix
    border GEdt1(t=0, C1(2,0)-C1(4,0)) {x=C1(4,0)+t; y=C1(4,1); label=GammaEd;}
    //fr. dr. dielectric x=td
    border dc12(t=0, C1(5,1)-C1(2,1)) {x=C1(2,0); y=C1(2,1)+t; label=Gamma12;}
    // fr. dreapta aer x=td
    border GE12(t=0, C1(3,1)-C1(5,1)) {x=C1(5,0); y=C1(5,1)+t; label=Gamma12;}
border GEb1(t=C1(3,0), C1(0,0)) { // portiunea 1 din lama elastica
           x = t;
           y = uycrt(t, 0);
           label = GammaEb;
    }
    border dt1(t=0, C1(5,0)-C1(6,0)) {x=C1(5,0)-t; y=C1(5,1);} // top diel.
    // mesh PE1 deformat la fiecare iteratie
    The[1] = buildmesh( GE10(ncpw) + dl10(ndb) + dh1(ndb) + GEdt1(nwel) +
                         dc12(ndb) + GE12(ncpw) +
                                  GEb1(multb*nb/4) + dt1(nwel) );
} // endif miprank == 1
else if (mpirank == 2) { // ----- rank 2 -----
    // frontiere procesor 2
    real[int, int] C2 = [-td],
                                        0],
                                      -hcpw+hel],
                           \begin{bmatrix} -td \end{bmatrix}
                           [wel/2+td, -hcpw+hel],
                            [wel/2+td, 0],
                                        -hcpw+hel+td],
                            ſ−td,
                                        -hcpw+hel],
                            [wel/2,
                            [wel/2+td, -hcpw+hel+td] ];
    // fr. stanga x=-td
    border GE21(t=0, C2(0,1)-C2(4,1)) {x=C2(0,0); y=C2(0,1)-t; label=Gamma12;}
    // strat dielectric x=-td
    border dc21(t=0, C2(4,1)-C2(1,1)) {x=C2(4,0); y=C2(4,1)-t; label=Gamma12;}
    // top electrod fix
    border GEdt2(t=0, C2(5,0)-C2(1,0)) {x=C2(1,0)+t; y=C2(1,1); label=GammaEd;}
    border db21(t=0, C2(2,0)-C2(5,0)) {x=C2(5,0)+t; y=C2(5,1);}
border dl23(t=0, C2(6,1)-C2(2,1)) {x=C2(2,0); y=C2(2,1)+t; label=Gamma23;}
    // fr.dr.procesor 2
    border GE23(t=0, C2(3,1)-C2(6,1)) {x=C2(6,0); y=C2(6,1)+t; label=Gamma23;}
    border GEb2(t=C2(3,0), C2(0,0)) { // portiunea 2 din lama elastica
           x = t;
           y = uycrt(t, 0);
            label = GammaEb;
    7
    border dt2(t=0, C2(6,0)-C2(4,0)) {x=C2(6,0)-t; y=C2(6,1);} // top
        dielectric
    // mesh PE2 deformat la fiecare iteratie
    The[2] = buildmesh( GE21(ncpw) + dc21(ndb) + GEdt2(nwel) +
```

Anexa 3. Cod paralel

```
db21(ndb) + dl23(ndb) + GE23(ncpw) +
                           GEb2(multb*nb/4) + dt2(nwel) );
    } // endif miprank == 2
    else if (mpirank == 3) { // ----- rank 3 -----
    // frontiere procesor 3
    real[int, int] C3 = [[wel/2,
                                          uycrt(wel/2,0)],
                             [wel/2,
                                          -hcpw],
                             [1b/2,
                                          -hcpw],
                             [1b/2,
                                          0],
                             [wel/2,
                                          -hcpw+hel+td],
                             \left[ \frac{wel}{2} \right]
                                          -hcpw+hel],
                             [wel/2+td, -hcpw],
                             [wel/2+td, -hcpw+hel+td] ];
    // fr. stanga proc.3
    border GE32(t=0, C3(0,1)-C3(4,1)) {x=C3(0,0); y=C3(0,1)-t; label=Gamma23;}
    border dl32(t=0, C3(4,1)-C3(5,1)) {x=C3(4,0); y=C3(4,1)-t; label=Gamma23;}
    // dr. electrod fix
    border GEdr(t=0, C3(5,1)-C3(1,1)) {x=C3(5,0); y=C3(5,1)-t; label=GammaEd;}
    // baza dielectric td dr.
    border db2(t=0, C3(6,0)-C3(1,0)) {x=C3(1,0)+t; y=C3(1,1);}
    // baza substrat dreapta
    border GEs2(t=0, C3(2,0)-C3(6,0)) {x=C3(6,0)+t; y=C3(6,1);}
    // perete dreapta
    border GEcpw2(t=0, C3(3,1)-C3(2,1))
                                        {x=C3(2,0); y=C3(2,1)+t; label=GammaCPW2;}
    border GEb3(t=C3(3,0), C3(0,0)) { // portiunea 3 din lama elastica
              x = t;
              y=uycrt(t,0)*(t<(C3(3,0)-dlt));</pre>
              label=GammaEb;
    }
    border dr(t=0, C3(7,1)-C3(6,1)) {x=C3(7,0); y=C3(6,1)+t; // dreapta diel.
    border dh3(t=0, C3(7,0)-C3(4,0)) {x=C3(7,0)-t; y=C3(7,1);} // top diel.
    // mesh PE3 deformat la fiecare iteratie
    The[3] = buildmesh( GE32(ncpw) + dl32(ndb) + GEdr(nhel) +
                           db2(ndb) + GEs2(nes) + GEcpw2(ncpw) +
                           GEb3(multb*nb/4) + dr(nhel) + dh3(ndb) );
} // endif miprank == 3
mpiBarrier(comm);
// migrare retea discretizare catre celelalte procese
broadcast(processor(0), The[0]); // Th[0] catre celelalte procese
broadcast(processor(1), The[1]); // Th[1] catre celelalte procese
broadcast(processor(2), The[2]); // Th[2] catre celelalte procese
broadcast(processor(2), fme[2]), // fm[2] catter celefalte processe
broadcast(processor(3), The[3]); // Th[3] catter celefalte processe
cout << endl << "Processor " << mpirank << " mesh ready! " << endl;</pre>
    // ====== FESPACE ==================
mpiBarrier(comm);
func Pk = P1; // ordin elemente finite
int inf = (mpirank - 1)*(mpirank > 0);
int sup = (mpirank + 1)*(mpirank < mpisize-1) +</pre>
            mpirank * (mpirank == mpisize-1);
fespace Veh(The[mpirank], Pk); // spatiu functional PE, elemente H1, de ord.1
fespace Ph(The[mpirank], P0); // spatii funct. PE, elem. L2 stocare prop.mat.
```

```
Veh uh=0, fi;
                             // uh->potentiale necunoscute, fi->func.de test
    Veh T11, T22, T12; // tensiuni maxwelliene din problema electrica
    Veh Ex, Ey;
                             // comp. vector intensitate camp electric
    fespace VehLeft(The[inf], Pk);
    fespace VehRight(The[sup], Pk);
    VehLeft uRxLeft = 0; // spatiu receptor de la subdom. vecin din stanga
VehRight uRxRight = 0; // spatiu receptor de la subdom. vecin din dreapta
    // ===== problema ==========
    int rd = The[mpirank](xrd[mpirank], yrd[mpirank]).region; //pct.dielectric
    Ph EPS = EPSO*(1 + (Er-1)*(region==rd));
                                                   // atrib. constante de material
    // problema de electrostatica
    macro grad(u) [dx(u), dy(u)] //EOM
    problem
               pe(uh, fi, init=ik, solver=Cholesky)
               = int2d(The[mpirank]) ( EPS*grad(uh)'*grad(fi) )//'
                        + on(GammaEd, uh=VaNew)
                         + on(GammaEb, uh=Vref)
                         + on(GammaLeft[mpirank], uh=uRxLeft)
                         + on(GammaRight[mpirank], uh=uRxRight);
                 // ========== LOOP ================
     if (mpirank == 0) cout << endl << " start loop " << endl;</pre>
         for (int i = 0; i < 15 ; i++ ) {</pre>
          real[int] errlocal(mpisize);
          real errt;
          mpiRequest[int] rq(4*mpisize); // status Rx/Tx
          pe;
                                            // rezolva problema
          cout << endl << "problem PE" << mpirank << " solved! " << endl;</pre>
          mpiBarrier(comm);
          if (mpirank > 0) // update frontiere
             Irecv(processor(inf, comm, rq[mpirank]), uRxLeft[]);
          if (mpirank < mpisize)</pre>
             Irecv(processor(sup, comm, rq[mpisize+mpirank]), uRxRight[]);
          if (mpirank > 0)
             Isend(processor(inf, comm, rq[2*mpisize+mpirank]), uh[]);
          if (mpirank < mpisize)</pre>
             Isend(processor(sup, comm, rq[3*mpisize+mpirank]), uh[]);
          for (int j = 0; j < 4*mpisize; ++j) int k = mpiWaitAny(rq);</pre>
              cout << endl << "Transfer between proc. " << mpirank <<</pre>
                                   " and proc. " << inf << " ready! " << endl;
11
          if (mpirank < mpisize-1) { // - calc.err pe fr.de vecinatate</pre>
              errlocal[mpirank] = int1d(The[mpirank], GammaRight[mpirank])
                                   (square(uh - uRxRight));
          }
          broadcast(processor(1), errlocal[1]);
          broadcast(processor(2), errlocal[2]);
          if (mpirank == 0) {
              deltaT = clock() - t0;
              errt = sqrt(errlocal[0] + errlocal[1] + errlocal[2]);
              cout << " Errt la iter. " << i << " = " << errt <<
                      "; t = " << deltaT << endl;
          }
```

11

Anexa 3. Cod paralel

```
broadcast(processor(0), errt);
if (errt <= errpe) break;
}
cout << endl << "Proc. " << mpirank << "; DOF = " << uh.n <<
        "; timp = " << clock()-t0 << endl;
// calcul camp electrostatic
Ex = -dx(uh);
Ey = -dy(uh);
//calcul tensiuni maxwelliene
T11 = EPS0*(Ex*Ex - 0.5*(Ex*Ex + Ey*Ey));
T22 = EPS0*(Ey*Ey - 0.5*(Ex*Ex + Ey*Ey));
T12 = EPS0*Ex*Ey;
// energia electrostatica
real We = 0.5*int2d(The[mpirank]) ( EPS*grad(uh)'*grad(uh) );//'
carVa << "proc. " << mpirank << "We = " << We << endl
        << "sigma11 = " << sigma11[];
}
carVa.flush;
```

Bibliografie

- [1] https://en.wikipedia.org/wiki/InfiniBand.
- [2] ***. Cisco Catalyst 2960-X and 2960-XR Series Switches Data Sheet. CISCO.
- [3] ***. Moore law. http://en.wikipedia.org/wiki/Moore%27s_law. accesat 06-2014.
- [4] ***. Nvidia launches the world's first graphics processing unit: Geforce 256. Technical report, NVIDIA Corp., 1999.
- [5] ***. Nvidia's next generation cuda compute architecture: Fermi. Technical report, NVIDIA Corporation, 2009.
- [6] ***. Cache (computing). http://en.wikipedia.org/wiki/Cache_ %28computing%29, 2013.
- [7] ***. Computer cluster. http://en.wikipedia.org/wiki/Cluster_ %28computing%29., 2013.
- [8] ***. Eniac. http://en.wikipedia.org/wiki/ENIAC, 2013.
- [9] ***. Harvard architecture. http://en.wikipedia.org/wiki/Harvard_ architecture, 2013.
- [10] ***. Harvard mark i. http://en.wikipedia.org/wiki/Harvard_Mark_I, 2013.
- [11] ***. Harvard mark ii. http://en.wikipedia.org/wiki/Harvard_Mark_II, 2013.
- [12] ***. Non uniform memory access. http://en.wikipedia.org/wiki/Non-Uniform_Memory_Access, 2013.

- [13] ***. Non-uniform memory access. http://en.wikipedia.org/wiki/Non-Uniform_Memory_Access., 2013.
- [14] ***. OpenMP Application Program Interface. OpenMP Architecture, July 2013. Version 4.0.
- [15] ***. Parallel computing. http://en.wikipedia.org/wiki/Parallel_ computing, 2013.
- [16] ***. Pipeline (computing). http://en.wikipedia.org/wiki/Pipelining, 2013.
- [17] ***. Semaphore (programming). http://en.wikipedia.org/wiki/ Semaphore_%28programming%29, 2013.
- [18] ***. Simd. http://en.wikipedia.org/wiki/SIMD, 2013.
- [19] ***. Top 500 supercomputer sites. http://www.top500.org/statistics/ overtime/, 2013.
- [20] ***. Uniform memory access. http://en.wikipedia.org/wiki/Uniform_ memory_access, 2013.
- [21] ***. Von neumann architecture. http://en.wikipedia.org/wiki/Von_ Neumann_architecture, 2013.
- [22] ***. Amd opteron. http://en.wikipedia.org/wiki/AMD_Opteron, 2014.
- [23] ***. Data types. http://en.wikipedia.org/wiki/Data_type, 2014.
- [24] ***. Frequency scaling. http://en.wikipedia.org/wiki/Frequency_ scaling, 2014.
- [25] ***. Instruction set architecture. http://en.wikipedia.org/wiki/ Instruction_set_architecture, 2014.
- [26] ***. Intel pentium 4. http://en.wikipedia.org/wiki/Intel_Pentium_4, 2014.
- [27] ***. Memory hierarchy. http://en.wikipedia.org/wiki/memory_ hierarchy, 2014.
- [28] ***. Cuda programming guide v7.5. http://docs.nvidia.com/cuda/pdf/ CUDA_C_Programming_Guide.pdf, 2015.

- [29] ***. Opencl. http://en.wikipedia.org/wiki/OpenCL, 2015.
- [30] ***. The guide to the program posix certified by ieee and the open group. Technical report, The IEEE Open Group, 2017.
- [31] ***. Medical mems. https://www.innovationservices.philips.com/lookingexpertise/mems-micro-devices/mems-micro-devices-applications/medicalmems/, February 2017.
- [32] ***. Mems inertial switch for military applications, 2017.
- [33] ***. Petsc/tao users manual. https://petsc.org/release/docs/manual/, 2020.
- [34] ***. Multifrontal Massively Parallel Solver (MUMPS 5.4.1), 2021.
- [35] R.A. Adams and J.J.F. Fournier. *Sobolev Spaces*. Elsevier, 2003.
- [36] Neculai Andrei. Modele matematice în mecanică. Research Institute for Informatics, Center for Advanced Modeling and Optimization, 2009.
- [37] Donu Arapura. Introduction to differential forms, jun 2016.
- [38] Stefan Banach. *Theorie des Operations Lineaires*. Z Subwencji Funduszu Kultury Narodowej, 1932.
- [39] A. Behzadan and M. Holst. Sobolev-slobodeckij spaces on compact manifolds, revisited. Technical report, Departament of Mathematics, University of California San Diego, 2018.
- [40] B. Blaise and L. Livermore. Posix threads programming. Technical report, Lawrence Livermore National Lab. HPC Tutorials, 2021.
- [41] Alain Bossavit. Electromagnetisme, en vue de la modelisation. Springer Verlag Paris, 1993.
- [42] Alain Bossavit. Computational Electromagnetism. Elsevier, Academic Press, 1997.
- [43] H. Brezis. Functional Analysis, Sobolev Spaces and Partial Differential Equations. Springer, 2010.
- [44] M. Brinskiy and M. Lubin. An introduction to mpi-3 shared memory programming. Technical report, Intel Corporation, 2018.

- [45] Louis L. Bucciarelli. Solid mechanics. MIT Civil and Environmental Engineering Course 1.050., 2004.
- [46] Theo Buhler and Dietmar Salamon. FUNCTIONAL ANALYSIS. ETH Zurich, 2017.
- [47] J Capitan Francisco. Barycentric coordinates. International Journal of Computer Discovered Mathematics, 0(0):pp.32–48, September 2015.
- [48] Jean Cea. Annales de l'institut fourier. In Approximation variationnelle des problèmes aux limites, volume 14, pages 345–444. Imprimerie Durand, 28 -Luisant, 1964.
- [49] P. G. Ciarlet. The Finite Element Method for Elliptic Problems. North-Holland Publishing Company, Amsterdam, 2nd edition, 1979.
- [50] G. Ciuprina, D. Ioan, M. Popescu, Ş. Alexandra, and L. Sorin. Coupled multiphysics-rf reduced models for mems. In *IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems* (ICPEICES) - Dehli, 2016.
- [51] G Ciuprina, D. Ioan, M. Popescu, I. Munteanu, and R. Popa. Metode numerice în ingineria electrică. MATRIX ROM, 1995.
- [52] Frank H. Clarke. Generalized gradients and applications. Transactions of the American Mathematical Society, 205, 1975.
- [53] Keith Conrad. Bilinear Forms. Technical report, University of Connecticut, 2017.
- [54] CRAY RESEARCH, INC. CRAY-1 Computer System. Hardware Reference Manual, 1977.
- [55] Bernard Dacorogna. INTRODUCTION TO THE CALCULUS OF VARIATIONS. Imperial College Press, 2004.
- [56] Daniel. Bazele Teoretice ale Ingineriei Electrice. Editura 2000, 1989.
- [57] Robert Resnick David Halliday and Jearl Walker. Fundamentals of physics. John Wiley & Sons, Inc., 9th edition, 2008.
- [58] A.S. Demidov. Generalized functions in mathematical physics Main ideas and concepts. Nova Science Publishers, 2000.

- [59] L. Demkowicz, M. Wozniak, et al. Fast parallel integration for three dimensional discontinuous petrov galerkin method. *Procedia Computer Science*, (101):8–17, 2016. YSC 2016. 5th International Young Scientist Conference on Computational Science.
- [60] Leszek Demkowicz. Computing With Hp-adaptive Finite Elements. Institute for Computational Engineering and Sciences The University of Texas at Austin, 2005.
- [61] Victorita Dolean, Pierre Jolivet, and Frédéric Nataf. An Introduction to Domain Decomposition Methods: algorithms theory and parallel implementation. HAL archives-ouvertes.fr, May 2016.
- [62] R. Duncan. A survey of parallel computer architectures. Computer, 23(2):5– 16, 1990.
- [63] V. Eijkhout. Introduction to high performance scientific computing. http:// pages.tacc.utexas.edu/~eijkhout/Articles/EijkhoutIntroToHPC.pdf, 2014.
- [64] et alii Esfandyari Jay. Mems sensors for advanced mobile applications. an overview. Technical report, STMicroelectronics, 2011.
- [65] J. V. Fernandez, W.H.A Schilders, and L. Miguel Silveira. Order reduction techniques for coupled multi-domain electromagnetic based models. Technical report, CASA, 2008.
- [66] Richard Feynman. Fizica modernă, volume 1. Editura Tehnică, 1969.
- [67] Mocanu Florentina. *Mecanica Mediilor Deformabile*. Universitatea Tehnică "Gheorghe Asachi" Iasi, 2000. Note de curs.
- [68] M. J. Flynn. Some computer organizations and their effectiveness. *IEEE Transactions on Computers*, C-21(9):948–960, 1972.
- [69] I. Foster. Designing and Building Parallel Programs: Concept and Tools for Parallel Software Engineering. Addison-Wesley, 1995.
- [70] M. Gander, F. Nataf, and L. Halpern. Optimized Schwartz Methods. 12th International Conference on Domain Decomposition Methods, 2001.

- [71] C. Geuzaine and J.F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2019.
- [72] V. Girault and P.A. Raviart. *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms.* Springer Berlin Heidelberg, 2012.
- [73] C. Grossmann, H. Roos, and M Stynes. Numerical Treatment of Partial Differential Equations. Springer-Verlag Berlin Heidelberg, 3rd edition edition, 2007.
- [74] S.D.A. Hannot. Modeling Strategies for Electro-Mechanical Microsystems with Uncertainty Quantification. Technische Universiteit Delft, 2010.
- [75] F. Hecht. New development in freefem++. Journal of Numerical Mathematics, 20(3-4):251–256, 2012.
- [76] F. Hecht. FreeFEM Documentation. Release 4.0, April 2019.
- [77] David Patterson John Hennessy. Computer Organization and Design The Hardware and Software Interface. ELSEVIER, 5 edition, 2014.
- [78] R. Hiptmair. Finite elements in computational electromagnetism. Acta Numerica, 11:237–339, 2002.
- [79] J. Necas I. Hlavácek. Mathematical Theory of Elastic and Elasto-Plastic Bodies, volume 3. ELSEVIER, 1st edition, 1981.
- [80] Ronald H.W. Hoppe. Finite element methods. University of Huston -Webpage, 2016.
- [81] Jacopo Iannacci. RF-MEMS Technology for High-Performance Passives. 2053-2563. IOP Publishing, 2017.
- [82] D. Ioan and G Ciuprina. Reduced Order Models of On-Chip Passive Components and Interconnects, Workbench and Test Structures, chapter III.8, pages 447–467. Springer Berlin Heidelberg, 2008.
- [83] D. Ioan, G. Ciuprina, and W.H.A. Schilders. Model order reduction, applications, chapter "Complexity reduction of electromagnetic systems", pages 145–200. De Gruyter, 2021.

- [84] Daniel Ioan. Metoda Elementului Finit pentru Modelarea Electromagnetică. Universitatea Politehnica Bucuresti, 2012.
- [85] Daniel Ioan. Modeling multiphysics systems and their complexity reduction. Ph.D Courses - Wupertal Universitat, 2017.
- [86] Marsden J.E. and Hughes Thomas J.R. *Mathematical Foundations of Elasticity*. Dover Publications Inc., 1983.
- [87] D. Kanter. Inside barcelona: Amd's next generation. https://www.realworldtech.com/barcelona/, 2007.
- [88] D. Kanter. Inside nehalem: Intel's future processor and system. https://www.realworldtech.com/nehalem/, 2008.
- [89] Y Katherine. Introduction to parallel programming for multicore. http: //www.cs.berkeley.edu/~yelick/cs194f07, August 2007.
- [90] Y Katherine. Memory hierarchies and optimizations. http://www.cs. berkeley.edu/~yelick/cs194f07, August 2007.
- [91] Y Katherine. Parallel cluster hardware. http://www.cs.berkeley.edu/ ~yelick/cs194f07, August 2007.
- [92] Y Katherine. Parallel programming. creating and using threads. http://www. cs.berkeley.edu/~yelick/cs194f07, August 2007.
- [93] Y Katherine. Shared memory hardware: Case study in matrix multiplication. http://www.cs.berkeley.edu/~yelick/cs194f07, August 2007.
- [94] Robert C. Kirby, A. Logg, and A. Terrel. Common and unusual finite elements. http://www.logg.org/anders/pub/papers/KirbyLoggEtAl2012a.pdf, 2015.
- [95] et alii Korvink, J. G. Inkjet-based Micromanufacturing. Wiley-VCH Verlag Gmbh & Co., 2012.
- [96] A.M. Krall. A review of orthogonal polynomials satifying boundary value problems. In Othogonal Polynomials and their Applications, volume 1329. Springer-Verlag, 1986.
- [97] Erwin Kreyszig. Introductory Functional Analysis with Applications. Wiley & Sons, 1989.

- [98] Cornelius Lanczos. The Variational Principles of Mechanic. University of Toronto Press, 1970.
- [99] Peter D. Lax. Functional Analysis. John Wiley & Sons, Inc., 2002.
- [100] G. Leoni. A First Course in Sobolev Spaces, volume v. 105. American Mathematical Society, 2009.
- [101] Christian Lessig. A primer on differential forms. Computing + Mathematical Sciences California Institute of Technology, 2012.
- [102] R.M. Lin and W.J. Wang. Structural dynamics of microsystems current state of research and future directions. In *Mechanical systems and signal processing*, volume 20, pages 1015–1043, 2006.
- [103] S. Lup. Multiphysics Modelling of Radio Frequency Micro-Electro-Mechanical-Systems. PhD thesis, POLITEHNICA University Bucharest, 2016.
- [104] S. Lup, G. Ciuprina, M. Popescu, and D. Ioan. Parametric multiphysics 3d modelling of a bridge type mems capacitive switch. In *ISEF*, 2015.
- [105] S. Lup, M. Popescu, G. Ciuprina, and D. Ioan. Hpc in multiphysics analysis of rf-mems capacitive switches. In XXth International Symposium on Electrical Apparatus and Technologies., 2018.
- [106] N. et alii MacDonald. Writing message passingparallel programs with mpi. a two day course on mpi usage. Technical report, Edinburgh Parallel Computing Centre, 2013.
- [107] Lawrence E. Malvern. Introduction to the Mechanics of Continuous Medium. Prentice-Hall, Inc, 1969.
- [108] Tarek P.A. Mathew. Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations. Springer, 2008.
- [109] B. Mattson, B. Sanders, and B. Massingill. Patterns for Parallel Programming. Addison-Wesley Professional, 2004.
- [110] W. McLean. Strongly Elliptic Systems and Boundary Integral Equations. Cambridge University Press, 2000.
- [111] D.F.Hendry M.D. Goodfrey. The Computer as von Neumann Planned It. IEEE Transaction on Magnetics, 15(1), 1993.

- [112] R. M. C. Mestrom. Multiphysics Modelling and Experimental Validation of Microelectromechanical Resonator Dynamics. PhD thesis, Technische Universiteit Eindhoven, 2009.
- [113] C. I. Mocanu. Teoria câmpului electromagnetic. Editura Didactică şi Pedagogică, Bucureşti, 1981.
- [114] Peter Monk. Finite Element Methods for Maxwell's Equations. Clarendon Press. Oxford, 2003.
- [115] G. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 19, 1965.
- [116] I. Munteanu, G. Ciuprina, and F. Tomescu. *Modelarea numerică a câmpului* electromagnetic prin programe Scilab. Printech - București, România, 2000.
- [117] Jean-Claude Nédélec. Mixed finite element in 3d in h(div) and h(curl). Equadiff 6, Proceedings of the International Conference on Differential Equations and Their Applications held in Brno, Czechoslovakia, Aug. 26 -30, 1985:321–325, August 1985.
- [118] J. Necas. Direct Methods in the Theory of Elliptic Equations. Springer Monographs in Mathematics. Springer-Verlag Berlin Heidelberg, 2012.
- [119] J. C. Nédélec. Mixed finite elements in \mathbb{R}^3 . In Numerical Mathematics. Springer-Verlag, 1980.
- [120] A. Nentchev. Numerical Analysis and Simulation in Microelectronics by Vector Finite Elements. PhD thesis, Techniche Universität Wien, January 2006.
- [121] Patrick Newberry. Differential forms. Mathematics Department of the University of Colorado Boulder, sep 2016.
- [122] Jarkko Niiranen. Finite Element Methods for 2d and 3d Elasticity. Aalto University, Helsinki, Engineering Computation and Simulation, 31.10.2016-20.12.2016, Course 9.
- [123] Mícheál O'Searcoid. Metric Spaces. Springer Science & Business Media, 2006.
- [124] P. S. Pacheco. An Introduction to Parallel Programming. Morgan Kaufmann Publishers, 2011.

- [125] J. Peddie. Is it time to rename the gpu. Technical report, IEEE Computer Society - The 75th Aniversary., 2021.
- [126] Iordan Petrescu. Introducere în metoda elementelor finite, cu aplicații în mecanica construcțiilor. Note de curs, 2012.
- [127] M. Popescu, S. Lup, and R. Bărbulescu. Using object oriented data structures for optimizing mems devices on parallel computers. In *ISEF*, 2015.
- [128] M. Popescu, S. Lup, G. Ciuprina, R. Bărbulescu, and D. Ioan. A parallel algorithm for multipysics analysys of rf-mems. In SCEE 2018 – The 12th International Conference on Scientific Computing in Electrical Engineering, 2018.
- [129] M. Popescu, S. Lup, G. Ciuprina, and D. Ioan. An object oriented data structure designed for multiphysics simulations on parallel computers. In *ATEE*, 2015.
- [130] M. Preda, P. Cristea, and F. Spinei. Bazele Electrotehnicii, volume I şi II. Editura Didactică și Pedagogică, București, 1980.
- [131] J.Y. Qian, G.P. Li, and F. De Flaviis. A parametric model of mems capacitive switch operating at microwave frequencies. In *IEEE MTT-S International Microwave Symposium digest*, 2000.
- [132] A. Quarteroni and A. Valli. Numerical Approximation of Partial Differential Equations. Springer-Verlag Berlin Heidelberg, second edition, 2008.
- [133] G. Rebeiz. Rf mems switches: Status of the technolgy. In *The 12th Conference* on Solid State Sensors, Actuators and Microsystems, 2003.
- [134] G. M. Rebeiz. RF MEMS: Theory, Design and Technology. John Wiley & Sons, 2003.
- [135] M.G. Rebeiz and J.B. Muldavin. High-isolation cpw mems shuntswitches—part 1: Modeling. IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES, 48(6):1045–1052, June 2000.
- [136] M. Rognes, Kirby R., et al. Efficient assembly of h(div) and h(curl) conformingfinite elements. SIAM Journal on Scientific Computing, 31(6):pp. 4130–4151, 2009.

- [137] W. Rudin. Functional Analysis. McGraw-Hill, Inc, 2-nd edition, 1991.
- [138] D Rusu. Analiză funcțională. Ed. Performantica, 2005.
- [139] M. Rychlik. An introduction to orthogonal polynomials. Technical report, University of Arizona, 2009.
- [140] W.H.A. Schilders, Henk A. van der Vorst, and J Rommes, editors. Model order reduction: theory, research aspects and applications, volume 13. Springer-Verlag Berlin Heidelberg, 1 edition, 2008.
- [141] J. A. Scott. Some examples of the use of areal coordinates in triangle geometry. *The Mathematical Gazette*, 83(498):472–477, 1999.
- [142] Tom Simonite. Moore's Law Is Dead. Now What? https://www. technologyreview.com/s/601441/moores-law-is-dead-now-what/, May 2016.
- [143] Marián Slodička. Partial Differential Equations. Lecture Notes for Master Courses - Department of Mathematical Analysis Ghent University, 2012.
- [144] Barry Smith. Domain Decomposition. Parallel Multilevel Methods for Elliptic Partial Differential Equations. Cambridge University Press, 1996.
- [145] Edwin H. Spanier. Algebraic Topology, Cap. 4.5. McGraw-Hill Book Company, 1995.
- [146] A. Spector and D. Gifford. The space shuttle primary computer system. Communications of the ACM, 1984.
- [147] Mircea Stan. Mecanica mediilor continue. Matrixrom, 2001.
- [148] A. Timotin, V. Hortopan, A. Ifrim, and M. Preda. Lecții de Bazele Electrotehnicii. Editura Didactică și Pedagogică, 1970.
- [149] F. M. G. Tomescu and A. Tomescu. Bazele electrotehnicii. Câmp electromagnetic. Matrix Rom, 2000.
- [150] E. Tonti. The reason for analogies between physical theories. In C. Brebbia, editor, *Applied Mathematical Modelling*, volume 1, pages 37–50. IPC Science and Technology Press Ltd, UK, Dept. of Civil Engineering, University of Southampton, UK, 1976.

- [151] J. Ubranic. Introduction to openmp. Technical report, Pittsburgh Supercomputing Center, 2021.
- [152] Mark van Kraaij. Strain and deformation, a global overview, may 2006.
- [153] Piero Villagio. Mathematical Models for Elastic Structures. Cambridge University Press, 1997.
- [154] John von Neumann. First draft of a report on the edvac. Technical report, Moore School of Electrical Engineering University of Pennsylvania, 1945.
- [155] S. Weiss. Course on parallel computing. Hunter College of the City University of New York.
- [156] H. Whitney. Geometric Integration Theory. Princeton University Press, 1957.
- [157] O. Windlund and A. Toselli. Domain Decomposition Methods Algorithms and Theory, volume 34 of Springer Series in Computational Mathematics. Springer Verlag, 2005.
- [158] Hugh Young and Roger Freedman. University Physics with Modern Physics. Addison-Wesley, the 13th edition, 2012.
- [159] Sabine Zaglmayr. High Order Finite Element Methods for Electromagnetic Field Computation. PhD thesis, JOHANNES KEPLER UNIVERSITÄT LINZ, 2006.
- [160] Bai Zhaojun, M. Patrick, Dewilde Roland, and W. Freund. Reduced-order modeling. *Handbook of Numerical Analysis*, 13, 2005.