

THESE DE DOCTORAT
DE L'ETABLISSEMENT UNIVERSITE BOURGOGNE FRANCHE-COMTE
PREPAREE A L'UNIVERSITE DE FRANCHE-COMTE

Ecole doctorale n°37
Sciences Pour l'Ingenieur et Microtechniques
Doctorat de Mécanique

Par

Nguyen-Nhat-Binh TRINH

Modélisation asymptotique d'un réseau de micro-miroirs et design de logiciel pour
les modèles multi-échelles dérivés automatiquement

Asymptotic modeling of a micro-mirror array and software design for automatically
derived multiscale models

Thèse présentée et soutenue à Besançon, le 07 octobre 2021

Composition du Jury :

M. Horatiu CIRSTEAN	Professeur à l'Université de Lorraine	Rapporteur
M. Antonio GAUDIELLO	Professeur à l'Université de Naples Federico II, Italie	Rapporteur
Mme. Renata BUNOIU-SCHILTZ	Maîtresse de conférences à l'Université de Lorraine	Examinatrice
M. Michel LENCZNER	Professeur à l'Université Technologie de Belfort- Montbéliard (UTBM)	Directeur de thèse
M. Frédéric ZAMKOTSIAN	Directeur de recherche CNRS au Laboratoire d'Astrophysique de Marseille (LAM)	Codirecteur de thèse

ACKNOWLEDGEMENTS

Accomplishing this PhD dissertation has been an inspiring experience for me which I will never forget. I would like to acknowledge all the people who supported and encouraged me during the thesis preparation time.

First, I am sincerely grateful to my supervisor Prof. Michel LENZNER and my co-supervisor, Dr Frédéric ZAMKOTSIAN, for all the precious support and invaluable research advice during my PhD study, especially for their time to read a manuscript carefully and their helpful suggestion of its improvement in time of writing this thesis. I would also like to thank Dr Nicolas RATIER as the third co-supervisor for his valuable discussion, worthy help. All of them are not only my teachers but also my friends, and the time under their supervision marked the maturity of my scientific research.

Second, I would like to express my great appreciation to all the members of the jury, Prof. Horatiu CIRSTEAN, Prof. Antonio GAUDIELLO and Dr Renata BUNOIU-SCHILTZ, for not only their precious time in reading the manuscript, their valuable scientific comments for thesis improvement, but also their exciting questions and worthwhile discussions during my thesis defence.

Third, I would like to offer my special thanks to all my colleagues in the FEMTO-ST institute, especially in the Time-Frequency Department, Dr Duy-Duc Nguyen, a Ph.D Student Santerelli Falzon TETSING TALLA for the scientific discussion and the enthusiastic help in the difficulties of the life in France.

Besides, I would like to extend my sincere thanks to all my Vietnamese friends and the Vietnamese community in Besançon, where I did the PhD thesis far away from my home country for the appropriate time together. All of them are my second family in France.

Moreover, I would like to express special thanks to all my close friends for their support and encouragement. Although they may be a long way from me, I know that they are always with me in tough times, and I feel that I am probably lucky to have such friends like them.

Furthermore, I would like to express sincere thanks to the Buddhist monks as my spiritual teachers and my inner Buddhist group for their encouragement and assistance. They are one of the tremendous spiritual backbones that helped me through the difficult moments in completing the thesis. With a genuine heart, I would like to thank Bodhisattva Avalokiteshvara, who gave me this chance, a wonderful experience.

Finally, last but not least, I would like to express my infinite gratitude to my big family in Vietnam, my parents (Ba Chau va Me Sang), elder sister (Chi Hai Oanh) and younger brother (Ut Binh), niece as a little angel (Nhat Anh), and two beloved deceased grandmothers. They are my great encouragement, accompanying me and always supporting me in life. Without them, the success of this thesis would not have been possible.

CONTENTS

1	Introduction	1
1.1	Micro-Mirror Array	1
1.1.1	A Short Overview of Micro-Mirror Array Technologies	1
1.1.2	MIRA	4
1.2	Homogenization Methods	6
1.2.1	Example of the Periodic Homogenization	6
1.2.2	Possible Application of Asymptotic Methods to MMAs	9
1.2.3	Multiscale Modeling of MIRA	10
1.3	MEMSALab Software	11
1.3.1	MEMSALab's Objective	11
1.3.2	MEMSALab's Flowchart	11
1.3.3	Previous Result	12
1.3.4	The Extension-Combination Method	13
1.4	Thesis Objectives	16
1.4.1	Thesis's Contribution	16
1.4.2	Thesis's Organization	20
2	Asymptotic Models for Electro-Mechanical Problems in the Two-Dimensional Array MIRA	23
2.1	Introduction	23
2.2	Mathematical Equations	25
2.3	Asymptotic Analysis Notations and some Lemmas	29
2.4	Global Scalings	31
2.5	Two-Scale Transform Operators and the Proof Algorithm	32
2.5.1	Internal Two-Scale Transform Operator	33

2.5.2	Boundary Two-Scale Transform Operator	37
2.5.3	The Reference Algorithm for Model Proofs	39
2.6	Electrostatic Models	40
2.6.1	Periodic Model	41
2.6.2	Lateral Boundary Layer Model	43
2.6.2.1	Geometry Notations	44
2.6.2.2	Boundary Layer Two-Scale Transform Operator	46
2.6.2.3	Derivation of a Lateral Boundary Model	48
2.6.3	Exterior Edge Model	52
2.6.3.1	Geometry Notations	52
2.6.3.2	Exterior Edge Boundary Layer Two-Scale Operator	54
2.6.3.3	Derivation of an Exterior Edge Model	55
2.6.4	Interface Model	58
2.6.4.1	Geometry Notations	58
2.6.4.2	Interface Boundary Layer Two-Scale Transform Operator	59
2.6.4.3	Derivation of an Interface Model	61
2.6.5	Internal Edge Model	65
2.6.5.1	Geometry Notations	66
2.6.5.2	Internal Edge Boundary Layer Two-Scale Operator	68
2.6.5.3	Derivation of an Internal Edge Model	70
2.7	Linear Elasticity Models	75
2.7.1	Periodic Model	77
2.7.2	Lateral Boundary Model	80
2.7.3	Exterior Edge Model	83
2.7.4	Interface Model	85
2.7.5	Internal Edge Model	88
3	Asymptotic Model Based Simulation of MIRA	93
3.1	Introduction	93
3.2	Description of MIRA Design	94

3.3	Asymptotic Model Implementations	100
3.4	Pull-in Analysis	106
3.4.1	Description of the Pull-in Phenomenon	106
3.4.2	Pull-In Voltage Computation	107
3.4.3	Optimization of the Pull-in Voltage	109
3.5	Effect of a neighbour imposed voltage	111
4	Data Structure and Rewriting Strategies for Multiscale Models Construction	115
4.1	Introduction	115
4.2	Nodes of Abstract Syntax Tree for Multiscale Model Representation	117
4.2.1	index	121
4.2.2	ident	122
4.2.3	paramReg	124
4.2.4	region	126
4.2.5	variable	130
4.2.6	function	131
4.2.7	operator	133
4.2.8	equality	134
4.2.9	equation	135
4.2.10	bc and ic	136
4.2.11	pde	138
4.2.12	model	138
4.2.13	subModel	139
4.2.14	subScale	142
4.3	Strategies Nodes	143
4.3.1	goto	143
4.3.2	insert	144
4.3.3	remove	145
4.3.4	replace	145
4.3.5	rule	145

4.3.6	rename	146
4.3.7	store	146
4.3.8	leftChoice	146
4.3.9	compose	147
4.4	Construction of a Family of Multiscale Electrostatic Models of MIRA	147
4.4.1	Domain Definitions and PDE Settings for the Physical Problem	148
4.4.2	Periodic Models	152
4.4.2.1	Data Structure for a Periodic Model	152
4.4.2.2	Strategy for a Periodic Model	155
4.4.3	Boundary Layer Models	156
4.4.3.1	Data Structure for a Boundary Layer Model	156
4.4.3.2	Strategy for a Boundary Layer Model	159
4.4.4	Edge Boundary Layer Models	159
4.4.4.1	Data Structure for an Edge Boundary Layer Model	159
4.4.4.2	Strategy for an Edge Boundary Layer Model	162
4.4.5	Interface Boundary Layer Models	162
4.4.5.1	Data Structure for an Interface Boundary Layer Model	162
4.4.5.2	Strategy for an Interface Boundary Layer Model	166
4.4.6	Interface Edge Boundary Layer Models	167
4.4.6.1	Data Structure for an Interface Edge Boundary Layer Model	167
4.4.6.2	Strategy for an Interface Edge Boundary Layer Model	170
5	A Rewriting and Description Logic Concurrent System	173
5.1	Introduction	173
5.2	Description Logic	175
5.2.1	Semantics	175
5.2.2	Knowledge Base	175
5.2.3	Different Description Logics	176
5.2.4	Inferences	176

5.2.5	Reasoning Tasks	177
5.3	A Short Overview of the MEMSALab Rewriting System	177
5.3.1	Node, Pattern and Tree	178
5.3.2	Strategy	178
5.3.3	Extended Grammars of Expressions and Strategies	180
5.4	Representation of Rewriting Notions in Description Logic	181
5.4.1	Trees Represented by Individuals	182
5.4.2	Patterns Represented by Concepts and Roles	183
5.4.3	Strategies Represented by Concepts and Roles	185
5.4.4	Summary of the Conversion Between Rewriting and Description Logic	186
5.5	The Concurrent Rewriting and Description Logic System	186
5.5.1	The Components of a RDL System	187
5.5.2	Strategy 0 of Ontology Construction	189
5.5.3	Commands Q and QData of Ontology Interrogation	189
5.5.4	Application of a Strategy Defined in an Ontology	190
5.6	RDL Representation of a Family of Models of MMAs	190
5.6.1	Roles Between Submodels, Model Properties and Strategies	190
5.6.2	Strategies in RDL	192
5.6.2.1	Concept, Data and Role Formulas	192
5.6.2.2	Strategies of the Property Table	197
6	Conclusions and Perspectives	201
I	Appendix	223
6.1	The First Eletrostatic Interface Model	225
6.1.1	Component	225
6.1.2	Global Definition	225
6.1.3	Geometry	225
6.1.4	Material	226
6.1.5	Physics (internal and external periodic models)	226

6.1.5.1	External Periodic Model	226
6.1.5.2	Internal periodic model	227
6.1.6	Study (compute two periodic solutions)	228
6.1.7	Physics set up for the first interface model	228
6.1.7.1	First part of the interface corrector	228
6.1.7.2	Second part of the interface corrector	229
6.1.8	Study (compute interface solutions)	230
6.1.9	Results (assembly of solutions)	230

INTRODUCTION

1.1/ MICRO-MIRROR ARRAY

Let us start with a quick overview of the techniques used in micro-mirror arrays before presenting the one that is the subject of this thesis.

1.1.1/ A SHORT OVERVIEW OF MICRO-MIRROR ARRAY TECHNOLOGIES

Micro-Mirror Arrays, abbreviated as MMAs, are devices related to Micro-Optical-Electromechanical Systems (MOEMS) family with mirrors in their components. The size of the mirror is very small, millimetre-sized, micro-sized, or smaller, with the principal goal being steering or monitoring the light amplitude, the light phase, or both. According to the statistics in 2018 of authors in [87], there are about 277 MMA designs from 49 companies and 23 academic research groups. They are widely used in various fields such as optics, telecommunications, astronomy, biology, etc.

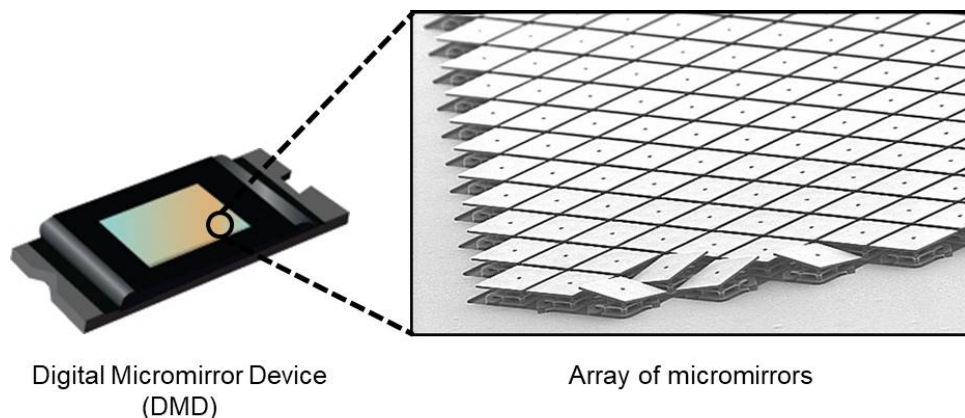


Figure 1.1: The digital micro-mirror device (DMD) was invented by solid state physicist and Texas Instruments Fellow Emeritus Dr. Larry Hornbeck in 1987. This device is the heart of projection technology which uses millions of mirrors to generate a visual screen. The figure is taken from [11].

We now take a brief glimpse at the classification of MMAs, the comprehensive reviews can be found in [59, 81, 87]. MMAs can be categorized according to the type of their actuators, including into four groups: *electrostatic*, *electrothermal*, *piezoelectric*, and *magnetic*.

The fundamental principle of the electrostatic actuator's operation is that the mechanical displacement is caused by the electrostatic forces generated by two conductors when voltages are applied. Based on the configuration and the management of the movement of actuators on activation, they are classified in three groups, see Figure 1.2. The first group is with surface actuators, whose conductors are two parallel planes: one flexible plate and one fixed plate Figure 1.2a. The other groups are with comb drivers, as the conductor configuration is similar to the comb shape. The direction of motion of these actuators can be either same as or perpendicular to the length of comb figures. We denote the former as the in-plane comb drivers, as in Figure 1.2b, and the latter as the out-plane comb drivers, as in Figure 1.2c.

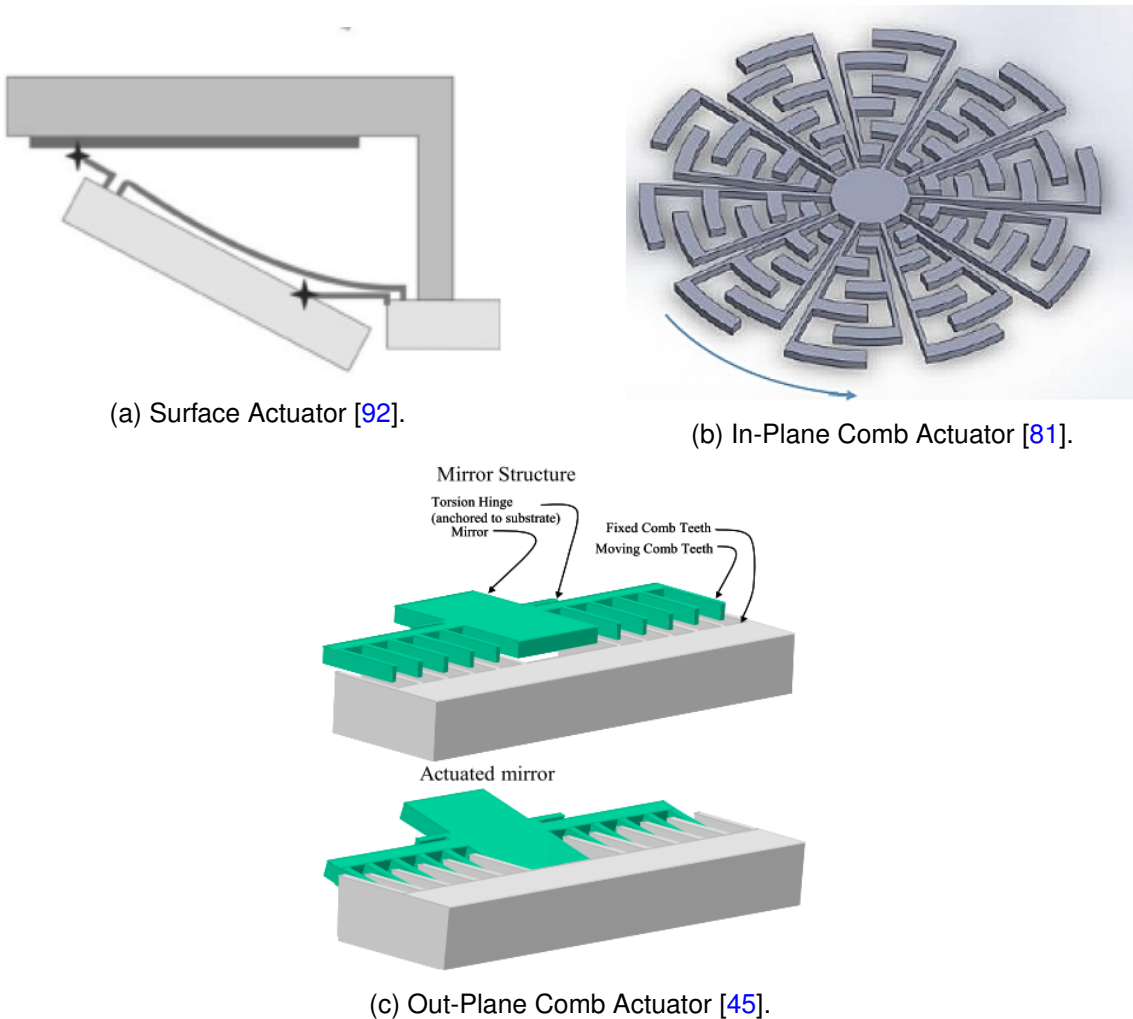
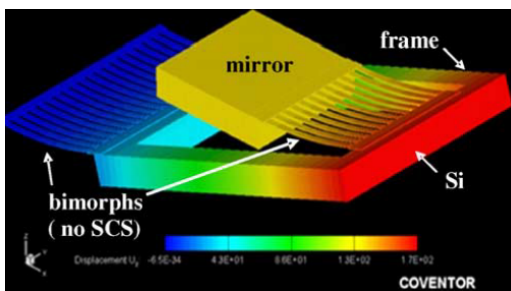
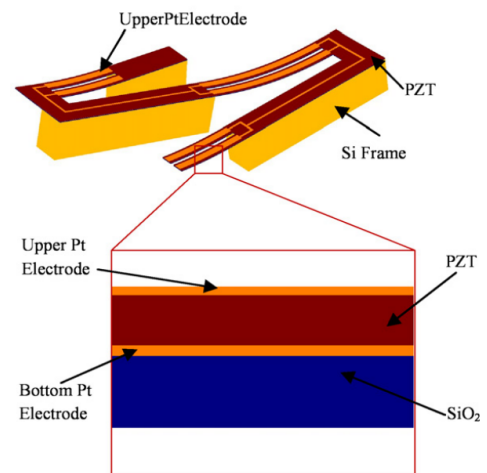


Figure 1.2: The electrostatic actuator group with three subgroups: (a) the surface actuators, (b) and (c) the in-plane and out-plane comb actuators.

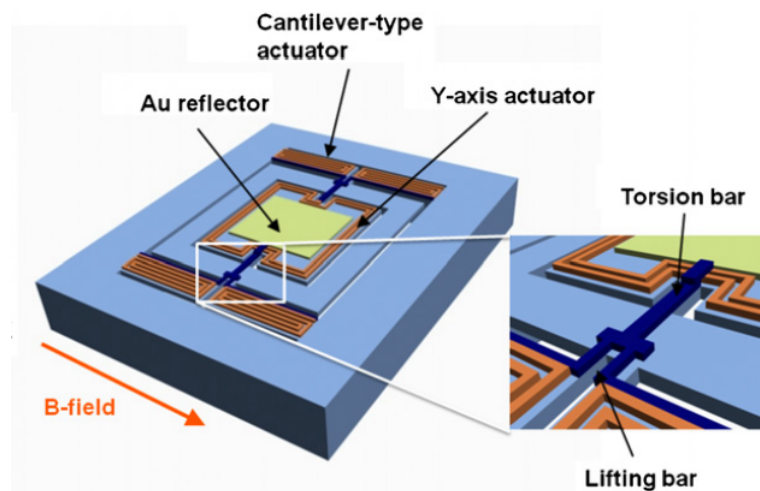
Electrothermal actuation is through assembly of components with large differences of thermal expansion coefficients. As a result, when a current is applied to them, they emit heat, leading to the generation of mechanical movements, see Figure 1.3a. Piezoelectric actuation is with piezoelectric materials which can be deformed upon applying a voltage. It is typical to combine several layers of materials to achieve the desired sizable mechanical displacement, see an example in Figure 1.3b. Magnetic actuation uses the Lorentz force generated by coils when an electric current is applied, see an example in Figure 1.3c.



(a) An electrothermal actuator [60].



(b) A piezoelectric actuator [97].



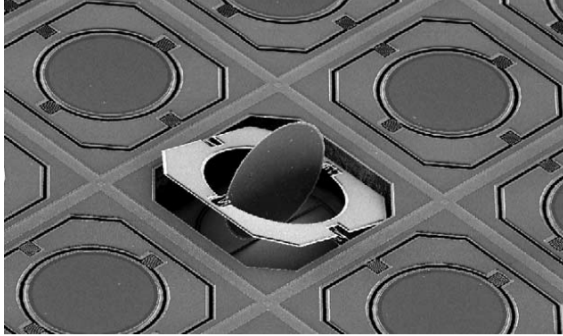
(c) A magnetic actuator [34].

Figure 1.3: Examples of the three other groups of actuators.

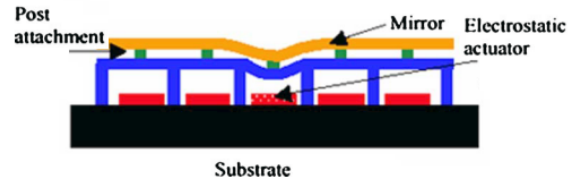
Another aspect of the classification is based on the kind of mirror surface. Two groups are distinguished, the discrete and the continuous one. In the former, the mirrors are disconnected from that of the adjacent cells, so their movements are independent. In the

latter, the mirrors in each cell are continuously linked to each other. In other words, there is only one mirror in the structure of the devices in this group, see Figure 1.4.

The number of mirrored elements in the array depends on the function of the device, can vary from one cell to thousands and can be placed in a one or two dimensional array. These arrays can be operated following the command algorithms: the direct addressing, line addressing, and the line-column addressing scheme, see more in [28, 30, 31].



(a) A discrete surface mirror [92].



(b) A continuous surface mirror [46].

Figure 1.4: Examples of discrete and continuous mirrors.

We next introduce MIRA as the MMA that the thesis considers.

1.1.2/ MIRA

For many years the Laboratoire d'Astrophysique de Marseille (LAM) has developed first in a collaboration with EPFL (Switzerland) and recently together with CSEM (Switzerland) an array of electrostatically actuated tilting mono-crystalline silicon micro-mirrors called MIRA, see its top view in Figure 1.5. It has been designed with stringent requirements such as a mirror size of $200 \times 100 \mu\text{m}^2$, a tilt angle of more than 20° , a filling factor of more than 80%, a contrast ratio of more than 1000, a wavelength bandwidth from visible to IR, an actuation voltage lower than 100V and an operating temperature ranging from room temperature to less than 100K. For more detail see [95].

This project aims to produce a new generation of MMAs devices applied in Multi-Object Spectroscopy (MOS), whose goal is to study the spectral characterization of faint astronomical objects like stars or galaxies. The MIRA is placed on the telescope's focal plane playing a role as a reflective field selector, which allows choosing the astronomical objects, see Figure 1.6.

The MIRA array is a member of the electrostatic actuation group, more precisely belongs in the surface actuation group. When an electrical potential is applied to conductors, an electrostatic force is generated and makes the mirror move towards the electrode placed under the mirror. The tilt angle is achieved by the support of the stopper beam located

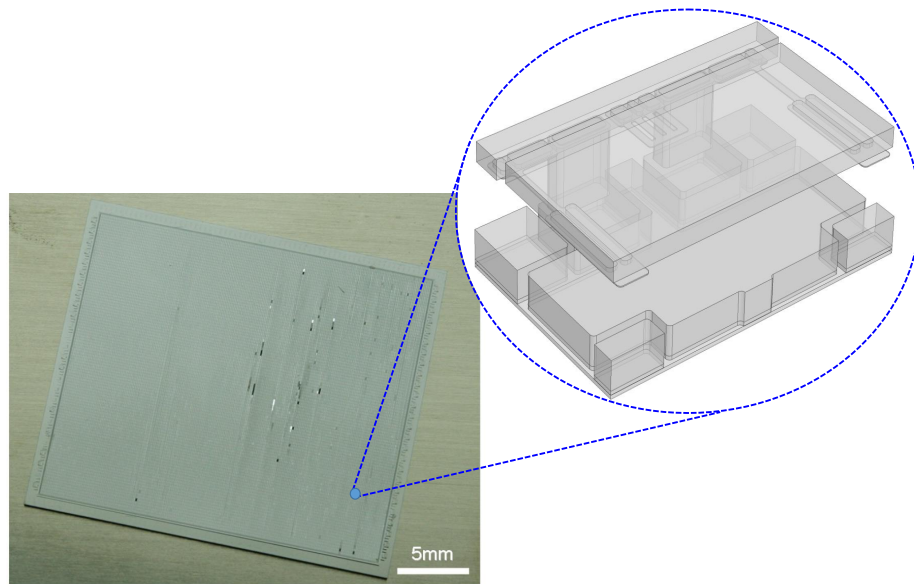


Figure 1.5: Top view of the MIRA array with 100×200 cells. The zoom represents a single cell.

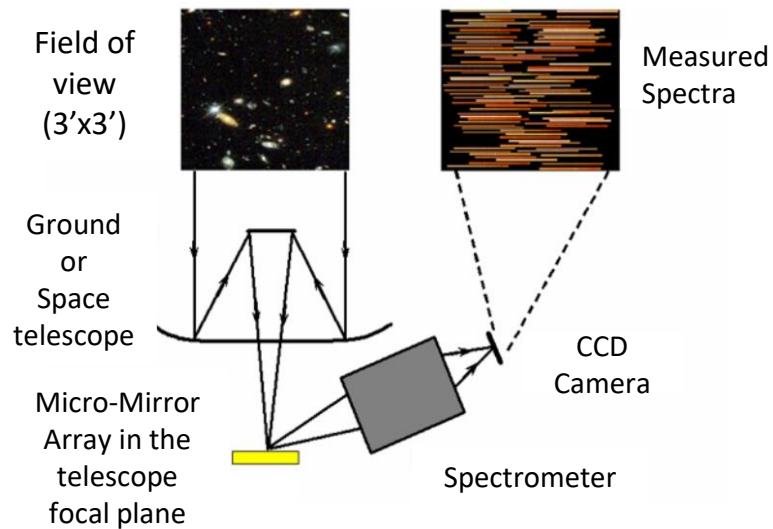


Figure 1.6: Application of the MIRA array in the the Multi-Object Spectroscopy. The figure taken from [93].

under the frame. When the voltage is switched off, the restoring force caused by two flexible beams installed under the mirror pulls it back to its resting position.

1.2/ HOMOGENIZATION METHODS

The asymptotic methods are the second main theme of the thesis. More precisely, the models studied belong to the class of periodic homogenization. Thus, we start with the most emblematic example of periodic homogenization, which allows us to situate the contributions of this work. Finally, we take advantage of the presentation of micromirror array technologies to suggest a number of possible applications of asymptotic methods to this field.

1.2.1/ EXAMPLE OF THE PERIODIC HOMOGENIZATION

Let us begin by considering the well-known example introduced in [5, 39, 42, 26] as the static heat conduction in a composite material located in a domain Ω illustrated in Figure 1.7 with the homogeneous Dirichlet boundary condition,

$$\begin{cases} -\operatorname{div}(\gamma \nabla u) & = f \text{ in } \Omega \\ u & = 0 \text{ on } \partial\Omega. \end{cases} \quad (1.2.1)$$

Here u represents the temperature in Ω , f is a given heat source, and γ a thermal conductivity. For simplicity, we consider the material to be isotropic then γ is scalar and assume that it takes distinct constant values $\gamma = \gamma_1$ in Ω_1 and $\gamma = \gamma_2$ in Ω_2 .

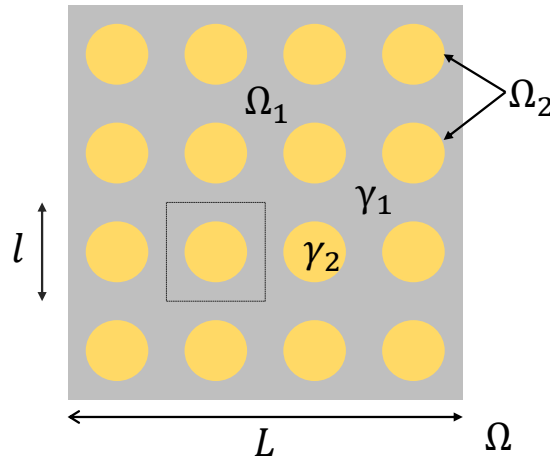


Figure 1.7: Illustration of a composite material. It occupies the domain $\Omega =]0, L[^2$. The structure in Ω has two components: a matrix in Ω_1 with conductivity γ_1 and inclusions in Ω_2 with conductivity γ_2 . We assume that the inclusions have a size in the same order than l and is periodically distributed with period l in each direction.

It can be observed that when the number of inclusions increases, their size becomes significantly small compared to the whole domain size. The classical parameter ε of the asymptotic methods is originated from here with $\varepsilon = l/L$. Periodic homogenization con-

sists in searching for the homogeneous material whose behavior would approach that of the composite material. In the case of thermics, it is necessary to identify a conductivity matrix of a homogeneous material that leads to a good approximation of the global temperature field while, in a first approach, ignoring the effects of local oscillations due to heterogeneities.

More generally, let us consider a heterogeneous anisotropic composite material in Ω with periodic inclusions with period ε as depicted in Figure 1.8a. The parameter ε is supposed to be very small compared to the whole domain size. The conductivity is thus a conductivity tensor γ^ε which has the form $\gamma^\varepsilon(x) = \gamma(x/\varepsilon)$ where $\gamma(y) = \{\gamma_{ij}(y)\}_{i,j \in \{1,2\}}$ is defined in the unit cell Y shown in Figure 1.8b. It is assumed that each entry $\gamma_{ij} \in L^\infty(\mathbb{R}^2)$, that the tensor is symmetrical i.e. $\gamma_{ij} = \gamma_{ji}$ for all $i, j \in \{1, 2\}$, and that γ is Y -periodic. In addition, γ satisfies the ellipticity condition,

$$\alpha|\xi|^2 \leq \sum_{i,j \in \{1,2\}} \gamma_{ij} \xi_i \xi_j \leq \beta|\xi|^2, \text{ for all } \xi = (\xi_1, \xi_2) \in \mathbb{R}^2,$$

for some $0 < \alpha \leq \beta$.

Taking into account of the dependence of the temperature on ε , equation 1.2.1 is rewritten with the rapidly oscillating coefficient γ^ε as

$$\begin{cases} -\operatorname{div}(\gamma^\varepsilon \nabla u^\varepsilon) & = f \text{ in } \Omega \\ u^\varepsilon & = 0 \text{ on } \partial\Omega. \end{cases} \quad (1.2.2)$$

For all $f \in L^2(\Omega)$, equation 1.2.2 has a unique solution u^ε in the Sobolev space $H_0^1(\Omega)$

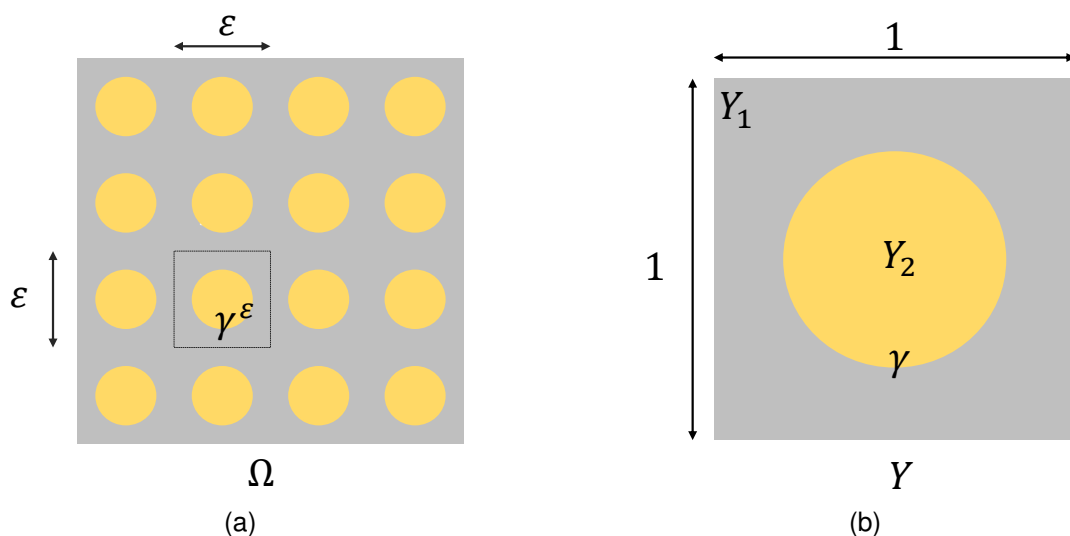


Figure 1.8: Description of (a) the composite material domain Ω and of (b) the unit cell Y . Y is the union of a matrix Y_1 and an inclusion Y_2 with conductivity tensor $\gamma(y)$. Obviously, Ω can be covered by cells of the form $\varepsilon Y = \varepsilon Y_1 \cup \varepsilon Y_2$.

satisfying the a priori estimate

$$\|u^\varepsilon\|_{H_0^1(\Omega)} \leq C \|f\|_{L^2(\Omega)}. \quad (1.2.3)$$

The homogenization method consists in observing that the solution u^ε depends on two scales: a macroscopic scale and a microscopic one. The former is defined by a slow variable x representing the position of a point in Ω . For the latter we use the variable $y := x/\varepsilon$ called the *fast variable* identifying the position of a point in Y . Evidently, the variable y locates whether the position is in Y_1 or in Y_2 , then it identifies the value of the thermal conductivity. Roughly speaking, the variable x is attached to the global behavior of the temperature, while the variable y concerns local effects i.e. microscopic oscillations.

From the point of view of mathematical theory, the passage to the limit when ε goes to 0 leads to many questions as to whether the solution u^ε converges to some limit, what kind of convergence it is, whether this limit is a good approximation to u^ε , etc. The goal of homogenization theory is to answer such questions.

Under the estimate 1.2.3, an homogenization theorem proves that u^ε converges to a limit u^0 in $H^1(\Omega)$ when ε tends to 0, u^0 being a function of x independent of y , i.e. $u^0(x, y) = u^0(x)$, and satisfies the so-called homogenized problem,

$$\begin{cases} -\operatorname{div}_x(\gamma^0 \nabla_x u^0) & = f \text{ in } \Omega \\ u^0 & = 0 \text{ on } \partial\Omega, \end{cases} \quad (1.2.4)$$

with the homogenized conductivity tensor γ^0 defined by

$$\gamma_{ij}^0 = \int_Y \gamma_{ij} dy - \sum_{k=1}^2 \int_Y \gamma_{ik} \frac{\partial \chi_j}{\partial y_k} dy, \text{ for all } i, j \in \{1, 2\}.$$

The functions χ_j are solutions to the cell problems

$$\begin{cases} -\operatorname{div}_y(\gamma(y) \nabla_y \chi_j) & = -\operatorname{div}_y(\gamma(y) e_j) \text{ in } Y \\ \chi_j & \text{is } Y\text{-periodic} \\ \int_Y \chi_j dy & = 0 \end{cases}, \quad (1.2.5)$$

$\{e_j\}_{j=1,2}$ being the canonical basis in \mathbb{R}^2 .

In addition, a classical result yields the following estimate,

$$\|u^\varepsilon - u^0\|_{H^1(\Omega)} \leq C \varepsilon^{\frac{1}{2}},$$

so u^0 is a reasonable approximation to u^ε when ε is small enough.

Consequently, instead of calculating u^ε by solving equation 1.2.2, we can approximate it

by u^0 through solving equation 1.2.4 and equation 1.2.5. From a numerical point of view, this is very interesting since solving the two last equations is not too much complex and far less costly than solving equation 1.2.2. Indeed, the homogenized equation is free from ε and then coarser meshes can be used regardless the size of ε .

Besides, the coefficients γ^ε of equation 1.2.2 play a role as a main characteristic of the composite material. In practice, it is so useful that the global behavior of this composite material can be described approximately by the behavior of the homogeneous one represented by the coefficients γ^0 of equation 1.2.4.

It is worth noting that there are many homogenization methods to derive the homogenized equation 1.2.4, as the oscillating test function method [88] and [80], the two-scale convergence initially introduced by Nguetseng [75] and then developed by Allaire [4], the adaption of two-scale convergence method by Casado-Díaz et al. [32, 33], the two-scale convergence method or unfolding method of Lenczner [63, 64], Cioranescu, Damlamian, and Griso [36, 37] to cite only three. Among the reference books on these subjects, we mention only a few, [25, 42, 38, 5, 26, 79, 27, 39, 17, 67, 89].

In this thesis, we utilize Lenczner's two-scale convergence method to derive the asymptotic models for the electromechanical problem in the two-dimensional MIRA array. We remark that the oscillations of this model are not much due to the coefficients of the partial differential equations but in the geometry of the array. Other applications of asymptotic methods to MMAs are introduced in the following.

1.2.2/ POSSIBLE APPLICATION OF ASYMPTOTIC METHODS TO MMAS

The classification of micromirror arrays in Section 1.1.1 reveals a set of asymptotic modeling possibilities for mechanical problems coupled with electrical or thermal or electromagnetic effects.

Piezoelectric actuators are multi-layered thin plates and as such can be modeled following the asymptotic techniques for thin plates as in [29, 54, 91, 66]. Electrothermal actuation uses thin multilayer plates or beams as piezoelectric actuators. Some actuators in this category can be structured as periodic fingers so that they can be modeled using periodic homogenization in addition. Combdribe actuators can be modeled by periodic homogenization methods for boundaries with large oscillations. This problem has been addressed in [50] taking into account the electrical potential only. Useful techniques are those of homogenization of oscillating boundary structures which have been extensively studied. A good entry point into the literature is the book of Meln'yk [71] and the articles of Gaudiello [49] which cite many references. When the combdribe performs a circular motion, the periodicity to be taken into account is in the radial direction.

Modeling of actuation by magnetic coils can be done by periodic homogenization taking

into account the quasi-periodicity in the radial direction if the number of coils is large enough.

Some actuators are complex and contain other thin structures, like the beams that support the MIRA mirrors. It is then possible to model them with asymptotic methods dedicated to thin structures and the boundary layer correctors used at their junction to their support.

The type of distribution of the micromirrors, discrete or continuous, introduces another distinction in the homogenization methods that are used in addition to those used at the cell scale. For MMAs with a continuous distribution of micromirrors, if there is no anchoring of each cell to the substrate, the usual theory of periodic homogenization applies with macroscopic components representing the mean fields and microscopic components for local corrections, see the bibliography of the previous section. Conversely, if each cell is anchored, either for a continuous distribution or for a discrete distribution of mirrors, the macroscopic component is null and only the microscopic periodic component remains, which then represents the first order effects. This is the case for MIRA. Then boundary layer correctors play a role at first order and must be taken into account, see the bibliography on homogenization with boundary layer effects in Chapter 2.

1.2.3/ MULTISCALE MODELING OF MIRA

The first contribution to the asymptotic models for the MIRA array is achieved in Duy-Duc Nguyen's thesis [77]. In his work, the asymptotic models are derived using the two-scale transform method for the electrostatic problem in the vacuum of one-dimensional array MIRA. The equation is governed as the Poisson system with Dirichlet conditions on the internal boundaries and Neumann conditions on the lateral boundaries.

As we mentioned above, the oscillation of this model does not much come from coefficients of PDE in the geometry. In this case, the coefficients take a constant value 1 since the domain is the vacuum. In addition, the array is a discrete array whose mirrors are independent of each other, and it is assumed that there are two different zones with two distinct constant applied voltages. Thus, as discussed in the previous section, the macroscopic solution will disappear, and the only contribution from the microscopic periodic one is left in the first-order effects to the approximation of potential.

Moreover, as there are two different imposed voltage zones, it produces two periodic models with solutions being periodic with different periods. This leads to the discontinuity of the approximated potential at the interface of two zones. Further, the periodic solutions do not satisfy the boundary conditions at the lateral boundary. Therefore, the correctors at the interface and the lateral boundary are proposed to deal with these problems, then the interface model and the boundary models are investigated. We note that their proofs for these models are compatible with the reference proof in MEMSALab as the goal of

software development.

In our thesis, we will further contribute by considering the coupled electromechanical problem in the two-dimensional array MIRA including all the boundary effects that will be discussed more in the following.

1.3/ MEMSALAB SOFTWARE

In order to properly locate the contribution of this thesis to the MEMSALab software, we will introduce an overview of MEMSALab with results already obtained in the development of its package before the start of this thesis in the following.

1.3.1/ MEMSALAB'S OBJECTIVE

In a general sense, a multiscale phenomenon is characterized by the presence of one or more small or large parameters, which constitutes the starting point of the construction of reduced models by asymptotic or singular perturbation methods . These methods are applied to algebraic equations, ordinary differential equations or partial differential equations. Small or large parameters can come from temporal phenomena (e.g large ratios between frequencies) or spatial phenomena (periodic structures with a large number of cells, large variations of coefficients, large ratio between lengths, etc). As we have discussed above, the result of an asymptotic analysis is a system of equations requiring much less computation time than the nominal model. Their mathematical foundations are solid and the resulting models offer a good compromise between precision and computation time. Their major disadvantage is that the construction of models is done on a case by case basis and requires a mathematical expertise, limiting their integration in general simulation software.

This integration is the objective of the MEMSALab project which promises the construction of software ensuring the generation of asymptotic models based on an original principle of reuse, inspired by the way humans operate. It combines two-scale approximation methods established in [65] from mathematics and rewriting methods from theoretical computer science.

1.3.2/ MEMSALAB'S FLOWCHART

A global view of the expected MEMSALab operation is depicted in the flowchart represented in Figure 1.9. A multi-scale model derivation starts with an input Partial Differential Equation (PDE) extracted from a PDE solver (FEM in the figure) and being expressed in the User Language. Together with the features to be taken into account for the asymptotic

analysis, they are transmitted to the core that generates the asymptotic model. Then, the latter is sent to the PDE solver for simulation.

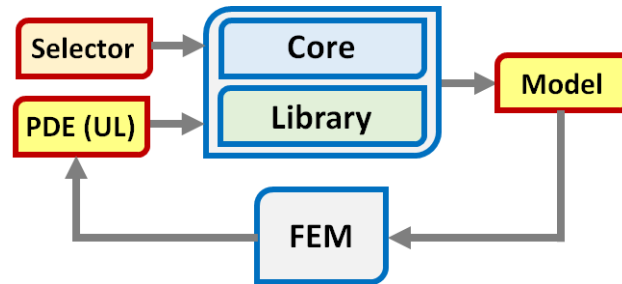


Figure 1.9: Expected flowchart of MEMSALab starting from an input PDE and producing an output sent in a simulation tool for PDEs as a finite element method software package (FEM). The input can also be generated by a user thanks to the User Language (UL). The *Selector* is an interface that selects the multiscale characteristics to be taken into account for building the multi-scale model in the *Core*.

1.3.3/ PREVIOUS RESULT

We recall the results achieved in the development of MEMSALab before the beginning of this thesis. A language of rewriting strategies for performing steps in mathematical proofs was introduced in [18]. A grammar of hierarchical description of the objects involved in partial differential equations has been defined in [94]. It includes domains together with their boundary, variables on domains or boundaries, functions of these variables, operators of these functions and finally partial differential equations with their boundary conditions. Beside, it is possible to write mathematical, arithmetic and logical expressions in this grammar as well as to make symbolic computations on them. Note that the hierarchical description of these objects is consistent with the needs of the formulation and derivation of multiscale models. Rewriting strategies applied to well-structured objects ensure that the performed proof operations are mathematically correct. The concept of extension-combination, recalled below, was announced in [94]. A specific grammar was introduced in [22] and [76], but its definitive form, the complete formulas for calculating the combinations and the proof of their correction are completely detailed in a submitted paper [19] whose preliminary version is available in [20].

A Domain Specific Language (DSL), written over the OCaml Language [78], including all the above grammars and thus allowing the writing of partial differential equations, of strategies of model derivation, of extensions and combinations was introduced in [22] and [76] and since, was significantly developed. It allowed the writing of first proofs of asymptotic models as partially reported in [21] using strategies, extensions and combinations. This approach is named the extension-combination method which is shortly summarized in the next section.

1.3.4/ THE EXTENSION-COMBINATION METHOD

We first briefly recall from the earlier works [94, 22] the general theoretical framework in which the software operates. A multiscale model derivation is characterized by the features taken into account in the asymptotic analysis. The mathematical objects are grounded into the three following levels.

1. *An input PDE or system of PDEs* is defined on a given geometry involving multiple scales. A reference PDE, PDE_{Ref} , as for instance a second order boundary value problem $-(a_e u')' = f$ posed in an open bounded interval Ω of \mathbb{R} , with mixed boundary conditions, the Dirichlet condition $u = h$ on Γ^D and the Neumann one $a_e u' = g$ on Γ^N , where Γ^D and Γ^N are two complementary parts of the boundary of Ω . The declaration of such a problem in the user language is given in Figure 1.10.
2. *A proof* consists of strategies and implements an asymptotic model derivation. There is a unique and particular initial proof called the *reference* proof $\text{Proof}_{\text{Ref}}$ illustrated in Figure 1.11 which deals with PDE_{Ref} . Indeed, a proof is applied to an input PDE to produce a multiscale model.
3. *An extension* is made with of other kinds of strategies. Precisely, an extension is an object that takes into account special features of a model. Each extension can be applied to the reference PDE and to the reference proof yielding the extended or the enriched proof. Furthermore, two extensions can be automatically *combined* or merged to produce another extension that includes the features of the input extensions.

In general, proofs for complex models are incrementally constructed by applying extensions to already existing proofs starting from the reference model, and such extensions can be automatically combined to produce more complex extensions.

```

Index
i_BC : "i_BC" ["D","N"] ["Given"]
i_BC_D : "i_BC" ["D"] ["Given"]
i_BC_N : "i_BC" ["D"] ["Given"]
Region
gamma : "gamma" [i_BC] [] [] --
gamma_D : "gamma" [i_BC_D] [] [] --
gamma_N : "gamma" [i_BC_N] [] [] --
Variable
xg : "xg" [] gamma
xg_D : "xg" [] gamma_D
xg_N : "xg" [] gamma_N
Function
normal : "normal" [] [xg] [] ["Given"]
Region
omega : "omega" [] [] [] gamma normal

Variable
x : "x" [] omega
Function
a : "a" [] [x] [] ["Given"]
f : "f" [] [x] [] ["Given"]
h : "h" [] [xg_D] [] ["Given"]
g : "g" [] [xg_N] [] ["Given"]
u : "u" [] [x] [] ["Unknown"]
Expression
equation : - ∂ ( a • ∂ u / ∂ x ) / ∂ x = f
PDE
pde : "pde" [] equation [u, h, xg_D
      , a • ∂ u / ∂ x, g, xg_N] [] [0]
pde : "pde_set" [] 0 [] [] [pde]

```

Figure 1.10: Script defining the differential equation of PDE_{Ref} in the earlier version of the User Language. Functions used in boundary conditions are also defined. The boundary conditions are omitted since their writing is similar to that of a PDE. The functions a , f and u are depending on the variable x which is defined on the domain ω . The latter has boundary γ and for outward unit normal n . The functions h and g are functions of xg_D or xg_N two variables defined on two parts γ_D and γ_N of the boundary.

```

proof'ref : "proof'ref"
; forPde'ref{["set"]} ("pde'ref", "indexInit")
; forPde{["weakToVeryWeakForm", "inPde"]}
; forPde{["setTestFunction", "inPde"]}
; forOpB{["toOpTS", "ref"]}
; forOpTS{["adjoint", "inIntegral", "ref"]}
; forPde{["passToTheLimit", "inPde"]}
; forPde{["veryWeakToStrongForm", "inPde"]}

```

Figure 1.11: The script in the User Language of the reference proof $Proof_{Ref}$ that is applied to PDE_{Ref} . It is based on the proof algorithm that underlies the construction of the models developed in the thesis. The proof is made as a composition of strategies expressed by semi-columns.

For the construction of an asymptotic model taking into account several characteristics of the PDE and/or of the proof, the extension-combination method begins with the construction of extension operators associated with several characteristics, cf Ext_1 and Ext_2 in Figure 1.12. When applied to the reference pair $(PDE_{Ref}, Proof_{Ref})$ they lead to new pairs $(PDE_1, Proof_1)$ and $(PDE_2, Proof_2)$. It is assumed that the extensions are *correct* in the sense that for each pair $(Proof_n, PDE_n)$ the application of $Proof_n$ to PDE_n provides an asymptotic model $Model_n$ taking correctly into account the n^{th} characteristics.

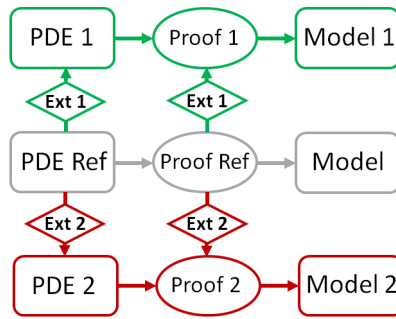


Figure 1.12: Schematic view of the scheme of asymptotic model generation for the pair $(PDE_{Ref}, Proof_{Ref})$ and their extensions $(PDE_1, Proof_1)$ and $(PDE_2, Proof_2)$ by Ext_1 and Ext_2 .



Figure 1.13: Schematic view of the combination of Ext_1 , Ext_2 and the complementary extension Ext'_{12} built to generate a correct extension Ext_{12} .

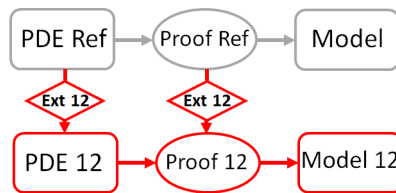


Figure 1.14: Schematic view of the pair $(Proof_{12}, PDE_{12})$ built by the combination Ext_{12} and the asymptotic model $Model_{12}$ obtained by application of $Proof_{12}$ to PDE_{12} .

By construction, the combination of several extensions, e.g. $Ext_1 + Ext_2$, is another extension covering all the characteristics of the involved extensions but which is not necessarily *correct* in the above sense. To reach correctness, it is generally, but not always, necessary to combine $Ext_1 + Ext_2$ with a complementary extension, e.g. Ext'_{12} in Figure 1.13. Figure 1.14 shows the final step of the method, i.e. the pair $(Proof_{12}, PDE_{12})$, built by application of the extension Ext_{12} , generating the expected model $Model_{12}$.

Models based on strategies and extensions have been built but the process of combination has not been fully implemented and therefore used. Indeed, we noticed that the size of the extensions resulting from combinations grows very quickly which prevents a systematic use. However, the good news is that they contain many redundant and inaccessible parts. This raises the question of minimizing or reducing the extensions, which requires additional theoretical work that could not yet be done.

1.4/ THESIS OBJECTIVES

The objectives of the thesis were to contribute to fundamental aspects of the development of the MEMSALab software. For this, it was chosen to start with an initiation to the development of models obtained by asymptotic methods. This was done in the continuity of Duc Duy Nguyen's thesis [77], i.e. on the modeling of MMAs. The models obtained were to serve as a basis for contributions to MEMSALab. Initially, it was planned to contribute to its library of models based on the extension-combination method. However, as explained in the following, it appeared during the developments that the design of the software suffered from characteristics that could hamper the scaling up to complex problems in the short term. It was then decided to direct the contributions rather towards new principles in the hope that they will open to better long-term perspectives.

Another aspect of MEMSALab that has not yet been systematically addressed is that of its interfacing with software for simulation of boundary value problems. At this level, there are essentially two directions of study. On the one hand, there is the one concerning the representation of multi-scale geometric structures which should be compatible with those used in simulation software such as the Constructive Solid Geometry (CSG), to mention only the simplest one. On the other hand, there is the determination of algorithms for the implementation of multiscale models which often combine in a complex way several sub-models. These algorithms are to be distinguished from those for the resolution of partial differential equations, they are higher level algorithms that build the interactions between the sub-models. It is on this point that the thesis contributes through model implementation in a commercial simulation software, because the implementation in a simulation software of the obtained asymptotic models provides at the same time the algorithm of their implementation. In addition to the interests specific to the development of MEMSALab, the simulations were also done with the concern to provide answers to questions that arose during the design of MIRA. This led to the simulation of the pull-in phenomenon and to the search for parameters that would guarantee a minimum voltage. Further work has been carried out but has not been completed, so it is not reported in this thesis.

1.4.1/ THESIS'S CONTRIBUTION

We start by constructing asymptotic models of the electromechanical coupling problem in MIRA. The technique of derivation uses the two-scale transform (or unfolding) method. The aim is to obtain models and their proofs compatible with the extension-combination method based on a reference proof, also called reference algorithm. Much effort has been put into designing proofs in the form of computational steps that can be realized by symbolic computation and expressed by extension of the reference proof. Some mathe-

matical justifications have then been sacrificed. For example, the existence of an asymptotic development of the two-scale transformation of the solutions is admitted because this yields significant reduction in proof length without losing in generality.

As mentioned above, one of the contributions of this thesis concerns the asymptotic models for MIRA. The array is two-dimensional with two zones where the imposed voltages are different. The equations are those of electromechanics coupling the electrostatic one to the linear elasticity system. The two-scale transformation method is used to treat this problem in the same way as in the thesis work of Duy-Duc Nguyen [77]. Nevertheless, the latter had only treated the asymptotic analysis of the electrostatics problem in a one-dimensional array. As in his work, the existence of two regions with different imposed voltages leads to two periodic models: the internal periodic model and the external model whose solutions are both periodic but with distinct periodic cell. As a result, the approximation constructed using these two periodic solutions is discontinuous at the interface of the two regions and does not satisfy the boundary conditions of the physical problem on the lateral border.

To correct these defects, boundary layer models at the interface and at the lateral boundary are constructed. They are constructed separately for each piece of boundary or interface, which leads to further defects. Precisely, since the solution in the vicinity of the outer boundary edges is an assembly of the outer periodic solution and the two boundary layer contributions from the adjacent lateral boundaries, it results a loss of continuity at the edges. The problem is similar to the edges of the interface between the two zones. To correct these defects, two other correctors are introduced. They are solutions of the so-called outer edge and inner edge models. These models constitute contributions in the field of asymptotic models beyond the particular case of arrays of microsystems such as MIRA. In summary, five types of asymptotic models have been developed: the periodic model, the interface model and the lateral boundary model, and the two edge models. A particular aspect of this work is that all the proofs of the models follow the frame of a reference proof in order to be implementable using the extension-combination method.

In the course of this work, we realized that there is no particular difficulty in applying the two-scale transformation method to the coupled electromechanical system. On the other hand, the difficulties lie in the construction of the boundary layer corrector models. To begin with, it is necessary to choose a suitable two-scale transformation and to understand how the solutions of other models must be taken into account. The reader cannot help but notice that the cumbersome notations make this work unnecessarily difficult to understand. We believe that even if better choices could have been made, the complexity is inherent to the problem of building complex asymptotic models for coupled boundary value problems. This particularity must be properly taken into account in the symbolic computation system.

The asymptotic electrostatics models have been implemented in COMSOL. The approximation of the electric potential is done at each point by summing a periodic solution and, for points close to an boundary or an interface, boundary layer correctors. An example of a complete approximation is provided in the vicinity of an outer edge. It results from the assembly of the periodic exterior solution and the contribution of three correctors: two lateral boundary layer correctors and the one of the edge. Moreover, the pull-in analysis is performed, including the pull-in voltage calculation and its optimization. The pull-in voltage calculation is conducted in one cell by solving an invert problems for the coupled electromechanical problem in COMSOL. Solving this problem is time-consuming as it is a nonlinear one and the mesh generation is also complicated. The reasonably complex algorithm has been developed to treat this problem. A surface response model of the pull-in voltage with respect to the two most significant parameters serves as a reduced order. It has been deployed to avoid long computation time. It is the base of an optimization process done with SIMBAD, a homemade software.

After having been introduced to model building techniques using the asymptotic two-scale transformation method, the aim was to bring contributions to MEMSALab. At this stage we started from the gained experience. The developments of MEMSALab that preceded the thesis led a number of conclusions regarding the design choices that have been done. It turns out that a tree-like data structure is suitable for representing boundary value problems and for applying rewriting strategies, navigation rewriting strategies are efficient even for very large trees, term matching is the tool of choice for hypothesis testing, and rewriting strategies are adequate to perform term transformations necessary for the application of elementary or more complex mathematical properties. As already said, these concepts have been tested by the implementation of a proof algorithm used as a reference in the extension-combination method as well as of some of its extensions. The above conclusions relate to the positive aspects of the design choices that were made. They have therefore been retained. In the following, we focus on the limitations and the means we had to remedy them.

The previous data structure was implemented in such a way that its modification required a change in the language grammar as well as changes in several places in the core. Specifically, the kinds of data (*index*, *region*, *variable*, *function*, *operator*, *pde* etc) were defined in the core. The models were described by objects of type *pde* including other types of objects. As the work evolved it became clear that this data structure was not rich enough to represent a wide variety of multi-scale models. The main limitations were that most of the multiscale features of the physical domains were almost not represented, the referencing of the geometric domains was not complete enough to represent a tree structure of domains at different scales or different submodels, and most importantly, the asymptotic models related to a physical model were built separately from each other, in the sense that a global structure was missing.

For these reasons, we developed a new form of construction for the data structure and the strategy based on forms of abstract syntax trees (AST) called Nodes. As a result, the new data structure is now formulated in a uniform and flexible way with more concepts taken into account, e.g. each data structure field is characterized by its identifier, the domain characteristics are outlined entirely. In addition, the global recursive structure is created with a "subScale" node defining the multiscale characteristics of the models in a vertical architecture and with a "subModel" node detailing the asymptotic model types, e.g., periodic model, boundary layer model, in a horizontal architecture. Generally speaking, the new data structure now possesses more capability to describe a family of asymptotic models as models of a thin structure, models of PDEs with strongly heterogeneous coefficients, models with oscillating boundaries, models with multiple scales, various boundary layer problems, a combination of these asymptotic characteristics. Further, another notable achievement is that the strategy is now independent of the data structure. This improvement step has a significant contribution to the algorithmic construction. The strategy now can be manufactured generally with an extensive range of functions.

The next step was the validation of the data structure using the asymptotic models of MIRA. Given their complexity, it appeared that an intermediate step before the implementation of proofs by the extension-combination method would be essential. It was decided to replace the proofs by a so-called direct method which consists in obtaining the asymptotic models by direct transformation of physical models, and by taking into account the asymptotic characteristics stored in the data structure. In other words, for each model the direct method is equivalent to applying a strategy corresponding to the statement of the corresponding theorem. Precisely, the strategy verifies the hypotheses and transforms the physical problem into the asymptotic model that is written at the dedicated place of the data structure. Thus, for an instance of each of the five asymptotic models of the electrostatics problem, the data structure content has been precisely detailed and the strategy for applying the corresponding theorem has been expressed as a meta-algorithm based on elementary strategies.

As mentioned above, rewriting strategies are very efficient on complex data structures stored in tree form. They are well adapted to hypothesis testing and to the application of mathematical rules even if they are complex. More specifically, this approach is well suited to represent partial differential equations and to apply mathematical properties to them. On the other hand, we find that they are not appropriate to manage a system of hypotheses and conclusions for a "large" corpus of theorems. For example, if we consider a number of theorems whose hypotheses and conclusions do not differ much from each other, we would need a classification system that would allow us to select the right conclusions corresponding to a set of hypotheses. After a lot of research and various attempts, especially with different logic systems, we came to the conclusion that the Description Logic (DL) used for ontology representation could fulfill this function. Indeed,

description logic uses sets of individuals and allows to define operations on sets and relations between sets or individuals. In itself, this is sufficient to define classifications. Of course DL also allows other operations such as the use of elementary data like strings or numbers.

Once this solution was identified, it remained to know how to interface the two worlds, that of rewriting on trees and that of ontologies. Moreover, it soon became clear that it makes no sense to represent equational expressions in an ontology, which is obviously not appropriate for this purpose, and that conversely, it would be interesting to store relational information between objects in the ontology that cannot be expressed by a tree structure. In short, it seemed more natural to have two separate representations, one of the data tree for boundary problems, their assumptions and asymptotic models and the other of the ontology. Since we want to represent theorem applications, we need to be able to represent the rewriting strategies in addition to the data. Thus, the work that was done was to build a correspondence between a rewriting system made of terms and rewriting strategies and an ontology in order to be able to build the necessary interactions between the two systems. For example, it is necessary that a strategy can enrich the ontology or query it, while the ontology must be able to store information about the success of a strategy applied to a term.

1.4.2/ THESIS'S ORGANIZATION

In addition to **Chapter 1** for the introduction, the thesis is comprised of the following chapters.

Chapter 2: Asymptotic Models for Electromechanical Problems in the Two-Dimensional Array MIRA. We utilize the two-scale transform method to derive asymptotic models of the two-dimensional periodic arrays MIRA. The equations are governed by the electromechanical system of equations. Five kinds of models are built for both the electrical and mechanical effects. The main models have periodic solutions. Then, corrector models are derived for boundary layers at the boundaries and at the interfaces between zones of different actuation. They are essential to take into account the boundary conditions and the transmission conditions. Finally, correctors are established at the edges of the boundaries and of the interfaces. They are necessary to fix incompatibilities between boundary layer correctors of adjacent faces, and between boundary layer correctors of interfaces. A noteworthy point with respect to MEMSALab is that the proof of each model has been designed to be implementable following the extension-combination method. More precisely, the proofs made for the electrostatic model are all variations of the same reference proof so that they are prepared to be expressed as extensions of

the latter. Those made for the elasticity model follow the same pattern so that they are also prepared to be obtained as extensions or even by combining the extensions of the electrostatic models with the extension from the electrostatic problem to the elasticity system.

Chapter 3: Asymptotic Model Based Simulation of MIRA. We recall the design of a cell of MIRA i.e. all the parameters and materials of its components. Numerical simulation results of the five electrostatic models derived in **Chapter 2** are reported. Then, the full approximation of the solution at an outer edge involving the solution of four models is presented. It involves a periodic solution, the solutions of the two boundary layer problems at the adjacent boundaries and the solution of the boundary layer model at the edge. The implementations was done in COMSOL Multiphysics. Moreover, a pull-in analysis was carried out in a single cell. It involves simulations of the coupling of the electrical and mechanical problems. Given the complexity and mesh sensitivity of the computations, a relatively complex algorithm has been deployed. Moreover, an optimization process of the pull-in voltage in a single cell has been carried out and is presented. Since the simulation time of the electromechanical problem is very long, the optimization was done using a reduced model.

Chapter 4: Data Structure and Rewriting Strategies for Multiscale Model Construction. First, the AST based data structure is defined and illustrated with simple examples. Then it is validated with the data of the electrostatic problem posed in the MIRA geometry and with the characteristics of the five asymptotic models of **Chapter 2**. In each case, a meta-algorithm of strategies building the model is provided.

Chapter 5: A Rewriting and Description Logic Concurrent System. First, we recall the elementary principles that found the description logic and the grammar of the MEMSALab rewriting system. Then, a representation of a fragment of the rewriting system is expressed in description logic. This leads to the formulation of the concurrent RDL system based on the two methods. Finally, the meta-algorithms of Chapter 4 are expressed as a single algorithm of the RDL system.

Chapter 6: Conclusions and Perspectives.

ASYMPTOTIC MODELS FOR ELECTRO-MECHANICAL PROBLEMS IN THE TWO-DIMENSIONAL ARRAY MIRA

2.1/ INTRODUCTION

The direct simulation of physical phenomena in a micromirror array is very costly in computing time due to a large number of degrees of freedom as its enormous size and the existence of various scales in its complicated structure. One approach to overcome this difficulty is to use an approximate model with accepted error obtained by deploying asymptotic methods for periodic problems, [23, 89, 40].

In this chapter, we derive two-scale models of the device MIRA designed by [31]. First models were investigated in Duc Duy Nguyen's thesis [77]. In his work, multi-scale models of a one-dimensional array have been constructed for the electrostatic problem by utilizing the unfolding method [63, 38, 36, 37, 10, 32] also called two-scale convergence since it generalizes the two scale convergence introduced in [75] and developed in [4]. The modeling was including boundary layer models at the lateral boundaries and at interfaces between zones of different actuations. In our work, we continue to contribute to the modeling by considering two-dimensional arrays of the same micromirrors governed by the equations of electrostatics coupled with those of linear elasticity.

We assume that the array is divided into two zones where the actuation is uniform. The electrostatic potential and the mechanical displacement of the asymptotic model are periodic in each of these zones. As a result, the electrostatic field and its normal derivative as well as the field of the mechanical displacements and the normal stress vectors are discontinuous at those interfaces. In addition, they do not satisfy the boundary conditions at the lateral boundaries of the array. To get rid of defects, boundary layer correctors are introduced at the interfaces and at the lateral boundaries. Besides, these corrections are

formulated separately on each face of the interfaces and of the lateral boundaries, leading to the discontinuity of the sum of their contribution at the face junctions, namely at the edges. Then, boundary layer correctors are also introduced at the edges.

Boundary layer problems in the homogenization problems have been much investigated, e.g. see in [24, 6, 82, 53, 56, 85, 52, 51, 8, 74]. In this work, our contribution is to derive interface, edge and internal edge corrector models which have not been studied yet. In total, we derive five kinds of models with the following features: periodic solution, lateral boundary layer, interface boundary layer, internal edge boundary layer, and exterior edge boundary layer, see in Figure 2.1. For each kind, we provide only one model instance, the other ones being obtained without difficulty.

As stated in the thesis introduction, this work is carried out with the perspective of developing symbolic computation algorithms for model building. Thus, particular attention is paid to the algorithmic structure of the model proofs and here we have endeavored to write them all following the framework of a single algorithm. Variations from this reference algorithm can be expressed by the extension-combination method. Here, we do not expose this aspect but it has been the subject of our work [21] achieved for simpler models but for the same algorithm. Notice that a complete theory of extension-combination is available in [20] while an extended version has been submitted for a journal publication. It can be observed that in the above mentioned algorithm, most of the operations are done on a very weak formulation instead on a weak formulation as it is usual. This leads to shortened proof lengths due to the absence of need of weak convergences of derivatives. Another characteristic of our choice in designing symbolic computation algorithm is to adopt a compromise between assumptions and algebraic computations. Thus our attention is more on calculations that can be algebraized than on fine mathematical analysis. In our algorithm, we assume priori estimates on the physical solutions. Thus in the following model derivations, we adopt the same assumptions which apply to the solution as well as to the boundary layer correctors. In addition, the boundary layer correctors and their gradients are assumed to converge exponentially to zero at infinity. This might be proven as e.g. [6, 89]. Another characteristic of this work, which shows the interest of having models automatically derived, is the choice to deal with a real problem whose complexity exceeds by far the one usually treated in academic works. While the complexity of the MIRA cells is not so high, nevertheless its handling in the framework of asymptotic methods quickly leads to having to manage extremely heavy notations, which is quickly prohibitive for a manual treatment. In this sense, this work provides a very interesting (indeed precious) family of models to guide the development of a rather general symbolic computation tool. Still in the perspective of developing systematic proofs, despite the fact that the imposed electric voltage is assumed to be piecewise constant in the MIRA, it is treated as if it could be continuously varying inside some zones and discontinuous at their interface. This leads to additional boundary layer effects that could find applications for

other devices.

As the model proofs all follow the same pattern, it would be unnecessarily long to write them all in detail. It has been chosen to detail them for the first models, then to reduce them for the next ones. On the other hand, in the mechanical models, the electrostatic force is assumed to converge to some limits which form has not yet been investigated, however it is in the scope of our algorithm to establish it. In this regard, we mention that the approach of [50] to pass to the limit in the electrostatic force in a model of comb actuator seems to be a possible solution.

Finally, we mention that the asymptotic models derived in this chapter are used to illustrate the data structure reported in the second part of the thesis as well as our new approach of model representation combining rewriting techniques with description logic based ontologies.

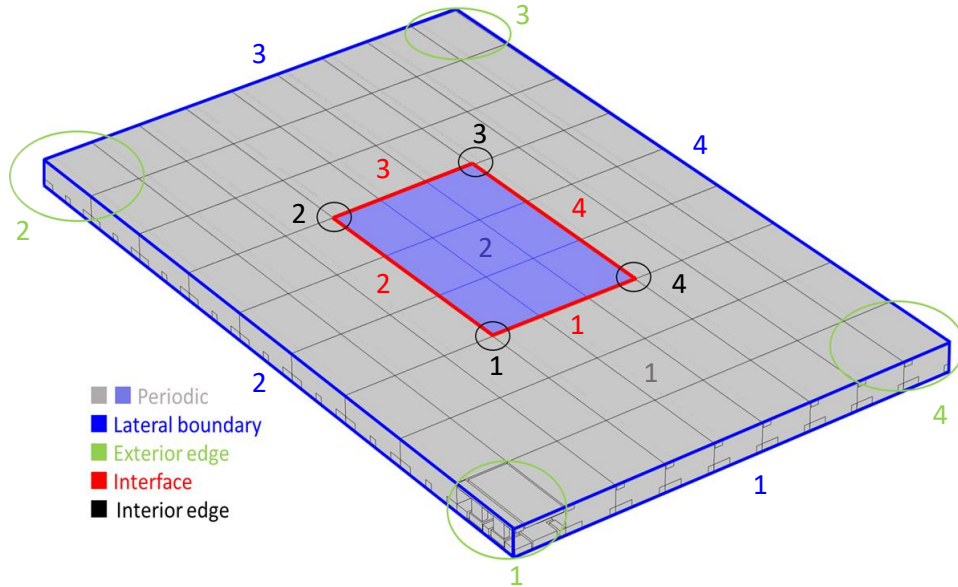


Figure 2.1: Zones where the asymptotic models are taken into account. The corresponding color numbers indicate the models'index. Assume that the array is actuated by two different imposed voltages in the internal and external zones, then it leads to two periodic models named the internal and the external periodic models.

2.2/ MATHEMATICAL EQUATIONS

We begin by describing the geometry of the MIRA array. It occupies the region Ω decomposed into Ω^{mec} and Ω^{vac} where the mechanical part and the vacuum surrounding it are located. Its width, length and thickness are respectively L_1, L_2 and L_3 , see Figure 2.2. It includes $n_1 \times n_2$ cells Ω_c of sizes l_1, l_2 , and l_3 .

Thus $\Omega = \cup_c \Omega_c$, where c is a multi-index belonging to $\mathcal{I}_{mul} = \{c = (c_1, c_2), c_1 \in 1, \dots, n_1$

and $c_2 \in \{1, \dots, n_2\}$. Each cell Ω_c , includes the mechanical part Ω_c^{mec} and the vacuum Ω_c^{vac} , see Figure 2.3. The mechanical structure consists of two parts, the mirror $\Omega_{mir,c}^{mec}$ and the electrode $\Omega_{ele,c}^{mec}$ so that $\Omega_c^{mec} = \Omega_{mir,c}^{mec} \cup \Omega_{ele,c}^{mec}$. We also use the decomposition of the domains Ω^{mec} and Ω^{vac} of the array as the unions $\Omega^{mec} = \cup_c \Omega_c^{mec}$ and $\Omega^{vac} = \cup_c \Omega_c^{vac}$, and the same for the domains consisting of all mirrors and electrodes $\Omega_{mir}^{mec} = \cup_c \Omega_{mir,c}^{mec}$ and $\Omega_{ele}^{mec} = \cup_c \Omega_{ele,c}^{mec}$.

The boundary of Ω^{mec} is defined as a union $\Gamma_0^{mec} \cup \Gamma_1^{mec}$, where Γ_0^{mec} is the boundary of Ω_{ele}^{mec} and Γ_1^{mec} is the boundary of Ω_{mir}^{mec} . Moreover, $\Gamma_0^{mec} = \cup_c \Gamma_{0,c}^{mec}$ and $\Gamma_1^{mec} = \cup_c \Gamma_{1,c}^{mec}$ where $\Gamma_{0,c}^{mec}$ and $\Gamma_{1,c}^{mec}$ denote respectively the boundary of the electrode $\Omega_{ele,c}^{mec}$ and of the mirror $\Omega_{mir,c}^{mec}$ of the mechanical body in a cell Ω_c^{mec} . The boundary of Ω^{vac} is denoted by $\partial\Omega^{vac}$ and is constructed as the union of the internal boundary Γ_{int}^{vac} and the external boundary Γ_{ext}^{vac} , where Γ_{int}^{vac} is defined by $\Gamma_0^{mec} \cup \Gamma_1^{mec}$ and Γ_{ext}^{vac} is the union of the lateral boundary Γ_{lat}^{vac} and of the top boundary Γ_{top}^{vac} of the vacuum part, $\Gamma_{ext}^{vac} = \Gamma_{lat}^{vac} \cup \Gamma_{top}^{vac}$.

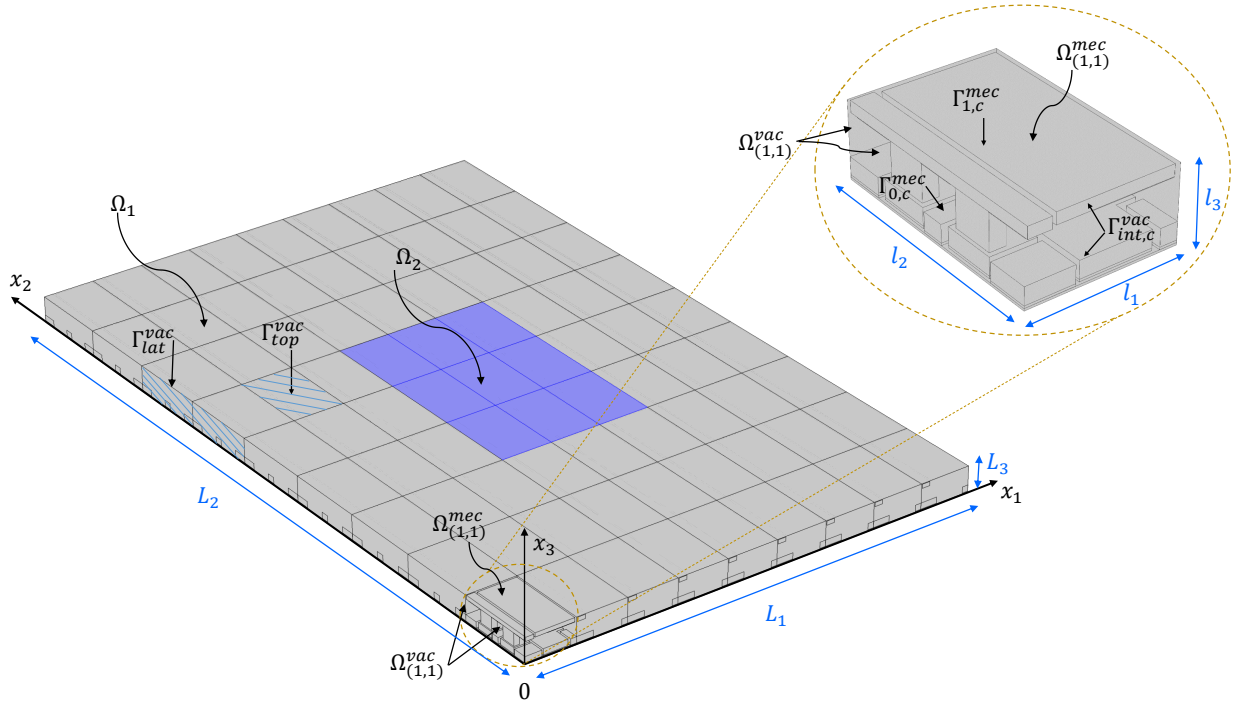


Figure 2.2: Representation of two zones the external zone Ω_1 and the internal zone Ω_2 with different actuation voltage in the MIRA array. The zoom illustrates one cell $\Omega_{(1,1)}$ of the array with the mechanical structure in $\Omega_{(1,1)}^{mec}$ surrounded by the vacuum in $\Omega_{(1,1)}^{vac}$.

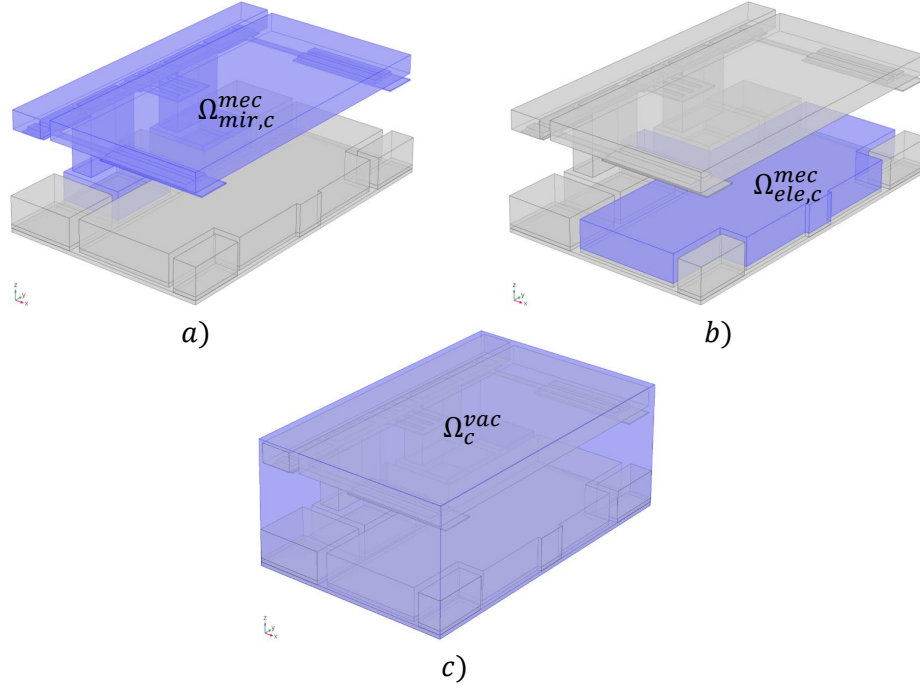


Figure 2.3: Illustration of the components of the cell Ω_c of the MIRA array. The mechanical part Ω_c^{mec} is made with two components, (a) the mirror part $\Omega_{mir,c}^{mec}$ and (b) the electrode part $\Omega_{ele,c}^{mec}$. Figure (c) represents the vacuum part Ω_c^{vac} .

For the sake of simplicity but without losing generality, we consider that Ω is split into two zones Ω_1 and Ω_2 in which the imposed voltages noted as V_1 and V_2 are different. Hereafter, we add the subscripts 1, 2 in geometrical notations to represent to which zones they belong, for example, Ω_1^{vac} and Ω_2^{vac} is a vacuum part of Ω_1 and Ω_2 , $\Gamma_{1,int}^{vac}$ and $\Gamma_{2,int}^{vac}$ is the internal boundary of Ω_1^{vac} and Ω_2^{vac} , and note that all previous geometrical notations without the subscripts 1, 2 now are understood as a union of two elements related to zones Ω_1 and Ω_2 , e.g. $\Gamma_{int}^{vac} = \Gamma_{1,int}^{vac} \cup \Gamma_{2,int}^{vac}$.

The field of electric potential ϕ in the vacuum is governed by the equation of electrostatics, see [55],

$$\begin{cases} -\Delta\phi & = 0 & \text{in } \Omega^{vac} \\ \phi & = V & \text{on } \Gamma_{int}^{vac} \\ \nabla\phi \cdot \mathbf{n} & = 0 & \text{on } \Gamma_{ext}^{vac} \end{cases}, \quad (2.2.1)$$

where V is the imposed voltage taking two distinct constant values V_1 in Ω_1 and V_2 and Ω_2 , and \mathbf{n} is the outward unit normal vector. The continuity of the potential and the electrostatic field at the interface Γ_{interf}^{vac} of Ω_1^{vac} and Ω_2^{vac} are given as

$$\phi|_{\Omega_1^{vac}} = \phi|_{\Omega_2^{vac}} \text{ and } \nabla\phi|_{\Omega_1^{vac}} \cdot \mathbf{n}^1 = -\nabla\phi|_{\Omega_2^{vac}} \cdot \mathbf{n}^2,$$

where \mathbf{n}^1 and \mathbf{n}^2 are the outward unit normal vectors of Ω_1^{vac} and Ω_2^{vac} on Γ_{interf}^{vac} , $\mathbf{n}^1 = -\mathbf{n}^2$.

Let us introduce a Hilbert space $H_{\Gamma_{int}^{vac},0}^1(\Omega^{vac}) \doteq \{v \in H^1(\Omega^{vac}), v = 0 \text{ in } \Gamma_{int}^{vac}\}$ endowed with the norm

$$\|v\|_{H_{\Gamma_{int}^{vac},0}^1(\Omega^{vac})} = \|\nabla v\|_{L^2(\Omega^{vac})},$$

for all $v \in H_{\Gamma_{int}^{vac},0}^1(\Omega^{vac})$.

Then a variational problem of (2.2.1) is to find $\phi \in H_{\Gamma_{int}^{vac},V}^1(\Omega^{vac}) \doteq \{\phi \in H^1(\Omega^{vac}), \phi = V \text{ in } \Gamma_{int}^{vac}\}$ such that

$$\int_{\Omega^{vac}} \nabla \phi \nabla v \, dx = 0,$$

for all $v \in H_{\Gamma_{int}^{vac},0}^1(\Omega^{vac})$.

The mechanical behavior is described by a displacement $\mathbf{u} = (u_1, u_2, u_3)$. The mechanical deformations are caused by the electrostatic force generated by the difference of imposed voltages between the mirror and the electrode. They are solutions to the following linearized elasticity system for homogeneous isotropic material, see [70]. In this part, the Einstein summation convention is employed to reduce formulas.

$$\begin{cases} -\partial_j(\lambda e_{pp}(\mathbf{u})\delta_{ij} + 2\mu e_{ij}(\mathbf{u})) & = f_i \text{ in } \Omega^{mec} \\ \mathbf{u} & = \mathbf{0} \text{ on } \Gamma_0^{mec} \\ (\lambda e_{pp}(\mathbf{u})\delta_{ij} + 2\mu e_{ij}(\mathbf{u}))n_j & = g_i \text{ in } \Gamma_1^{mec} \end{cases}, \quad (2.2.2)$$

where λ and μ are the Lamé constants, $e(\mathbf{u}) = (e_{ij}(\mathbf{u}))_{ij}$ is a linearized strain tensor defined by

$$e_{ij}(\mathbf{u}) = \frac{1}{2}(\partial_i u_j + \partial_j u_i),$$

$\mathbf{f} = (f_1, f_2, f_3)$ is the external body force density and $\mathbf{g} = (g_1, g_2, g_3)$ is the electrical force per unit area applied on the surface of conductors oriented along the outward normal vector \mathbf{n} of Γ_{int}^{vac} . It is given by, see [62],

$$\mathbf{g} = \frac{1}{2}\epsilon_0|\mathbf{E}|^2\mathbf{n}, \quad (2.2.3)$$

where $\mathbf{E} = -\nabla\phi$ is the electrical field and ϵ_0 is the vacuum permittivity. The continuity of the displacement \mathbf{u} and the normal stress tensor at the interface Γ_{interf}^{mec} between Ω_1^{mec} and Ω_2^{mec} are given as

$$\mathbf{u}|_{\Omega_1^{mec}} = \mathbf{u}|_{\Omega_2^{mec}} \text{ and } (\lambda e_{pp}(\mathbf{u})\delta_{ij} + 2\mu e_{ij}(\mathbf{u}))|_{\Omega_1^{mec}}n_j^1 = -(\lambda e_{pp}(\mathbf{u})\delta_{ij} + 2\mu e_{ij}(\mathbf{u}))|_{\Omega_2^{mec}}n_j^2,$$

where \mathbf{n}^1 and \mathbf{n}^2 are the outward normal vectors of Ω_1^{mec} and Ω_2^{mec} at Γ_{interf}^{mec} , $\mathbf{n}^1 = -\mathbf{n}^2$.

We now introduce a linearized stress tensor $\sigma(\mathbf{u})$, thank to the Hooke's law in the linearized elasticity,

$$\sigma(\mathbf{u}) = \lambda(\nabla \cdot \mathbf{u})I + 2\mu e(\mathbf{u}) \text{ or } \sigma_{ij}(\mathbf{u}) = \lambda e_{pp}(\mathbf{u})\delta_{ij} + 2\mu e_{ij}(\mathbf{u}),$$

where I is the 3×3 identity matrix. Hence (2.2.2) takes the shorter form

$$\begin{cases} -\operatorname{div}\sigma(\mathbf{u}) &= \mathbf{f} \text{ in } \Omega^{mec} \\ \mathbf{u} &= \mathbf{0} \text{ on } \Gamma_0^{mec} \\ \sigma(\mathbf{u})\mathbf{n} &= \mathbf{g} \text{ in } \Gamma_1^{mec}. \end{cases} \quad (2.2.4)$$

We introduce the Hilbert space $\mathcal{H}_{\Gamma_0^{mec},0}^1(\Omega^{mec}) \doteq \{\mathbf{v} = (v_1, v_2, v_3), v_i \in H^1(\Omega^{mec}), v_i = 0 \text{ on } \Gamma_0^{mec}\}$ with the norm

$$\|\mathbf{v}\|_{\mathcal{H}_{\Gamma_0^{mec},0}^1(\Omega^{mec})} = \left(\sum_{i=1}^3 \|\nabla v_i\|_{L^2(\Omega^{mec})}^2 \right)^{1/2},$$

and assume that $\mathbf{f} \in \mathbf{L}^2(\Omega^{mec}) \doteq [L^2(\Omega^{mec})]^3$ and $\mathbf{g} \in \mathbf{L}^2(\Gamma_1^{mec}) \doteq [L^2(\Gamma_1^{mec})]^3$.

Multiplying both sides of (2.2.2) by a vector test function \mathbf{v} of $\mathcal{H}_{\Gamma_0^{mec},0}^1(\Omega^{mec})$ and integrating over Ω^{mec} , by the partial integration formula, we obtain

$$\int_{\Omega^{mec}} (\lambda e_{pp}(\mathbf{u})\delta_{ij} + 2\mu e_{ij}(\mathbf{u}))\partial_j v_i \, dx = \int_{\Omega^{mec}} f_i v_i \, dx + \int_{\Gamma_1^{mec}} g_i v_i \, ds(x).$$

It is easily seen that $\delta_{ij}\partial_j v_i = \partial_j v_j = e_{qq}(\mathbf{v})$ and $\partial_j v_i = e_{ij}(\mathbf{v}) + \widetilde{e}_{ij}(\mathbf{v})$, where \widetilde{e}_{ij} is defined by $\widetilde{e}_{ij}(\mathbf{v}) = (\partial_i v_j - \partial_j v_i)/2$. Since $e_{ij}(\mathbf{u})$ is symmetric and $\widetilde{e}_{ij}(\mathbf{v})$ is anti-symmetric then we get $e_{ij}(\mathbf{u})\partial_j v_i = e_{ij}(\mathbf{u})e_{ij}(\mathbf{v})$. The weak form of (2.2.2) is to find $\mathbf{u} \in \mathcal{H}_{\Gamma_0^{mec},0}^1(\Omega^{mec})$ such that

$$\int_{\Omega^{mec}} (\lambda e_{pp}(\mathbf{u})e_{qq}(\mathbf{v}) + 2\mu e_{ij}(\mathbf{u})e_{ij}(\mathbf{v}))dx = \int_{\Omega^{mec}} f_i v_i dx + \int_{\Gamma_1^{mec}} g_i v_i ds(x), \quad (2.2.5)$$

for all $\mathbf{v} \in \mathcal{H}_{\Gamma_0^{mec},0}^1(\Omega^{mec})$.

2.3/ ASYMPTOTIC ANALYSIS NOTATIONS AND SOME LEMMAS

In this part, we introduce Landau Symbols, the big O and the small o , for more details see in [48] and [47].

Definition 1:

Let $\varphi(x, \varepsilon)$ and $\psi(x, \varepsilon)$ be real functions of the variable x varying in a domain $D \subset \mathbb{R}^3$ and of the parameter $0 < \varepsilon \leq \varepsilon_0$, ε_0 a given positive number and assume that $\psi(x, \cdot) \neq 0$ for all $x \in D$. We say that:

1. The big O: $\varphi(x, \varepsilon) = O(\psi(x, \varepsilon))$ in D if only if there exists a constant C such that for all $x \in D$, $|\varphi(x, \varepsilon)| \leq C|\psi(x, \varepsilon)|$ for ε small enough .
2. The big O of 1: $\varphi(x, \varepsilon) = O(1)$ in D if only if there exists a constant C such that for all $x \in D$, $|\varphi(x, \varepsilon)| \leq C$ for ε small enough. This is a special case of the first item.
3. The small o: $\varphi(x, \varepsilon) = o(\psi(x, \varepsilon))$ in D if only if $|\varphi(x, \varepsilon)|/|\psi(x, \varepsilon)| \rightarrow 0$ as $\varepsilon \rightarrow 0$ for all $x \in D$ such that $\psi(x, \varepsilon) \neq 0$.
4. The small o of 1: $\varphi(x, \varepsilon) = o(1)$ in D if only if $|\varphi(x, \varepsilon)| \rightarrow 0$ as $\varepsilon \rightarrow 0$ for all $x \in D$ such that $\psi(x, \varepsilon) \neq 0$. This is a special case of the third item.

Example: $\sin(x) = O(1)$, $\varepsilon^3 = o(\varepsilon^2)$.

Definition 2:

Let Y a subset of \mathbb{R}^3 defined as $Y =]-h_i/2, h_i/2[$ with outward normal vector \mathbf{n} , where h_i is a positive number, $i \in I = \{1, 2, 3\}$ and γ be a subboundary of ∂Y such that γ is a union of pairs of parallel planes that are also subboundaries of ∂Y . Let φ be a function defined a.e on \mathbb{R}^3 and $\{e_1, e_2, e_3\}$ be a canonical basis vector of \mathbb{R}^3 . We say that:

1. φ is Y - periodic in a direction e_i , for $i \in I$ if only if $\varphi(x + ke_i h_i) = \varphi(x)$ a.e on \mathbb{R}^3 for all $k \in \mathbb{Z}$.
2. φ is γ - periodic if only if for all $x \in \gamma$ there exists $x' \in \gamma$ such that $\varphi(x) = \varphi(x')$, $x' = x + h_i e_i$ for some $i \in I$.
3. φ is γ - antiperiodic if only if for all $x \in \gamma$ there exists $x' \in \gamma$ such that $\varphi(x) = -\varphi(x')$, $x' = x + h_i e_i$ for some $i \in I$.

With this definition, we have some following lemmas.

Lemma 1:

If φ is in $L^2(Y)$ such that

$$\int_Y \varphi v \, dx = 0,$$

for any $v \in C_0^\infty(Y)$ then $\varphi = 0$ a.e in Y .

Lemma 2:

If for any $v \in C_0^\infty(Y)$ such that

$$\int_\gamma \varphi \nabla v \cdot \mathbf{n} \, ds(x) = 0$$

then $\varphi = 0$ a.e on γ .

Lemma 3:

If for any $v \in C^\infty(Y)$, Y -periodic in the periodic direction such that

$$\int_\gamma \varphi \nabla v \cdot \mathbf{n} \, ds(x) = 0, \text{ resp } \int_\gamma v \nabla \varphi \cdot \mathbf{n} \, ds(x) = 0.$$

then φ is γ -periodic, resp. $\nabla \varphi \cdot \mathbf{n}$ is γ -antiperiodic.

2.4/ GLOBAL SCALINGS

Let us introduce the dimensionless small parameter $\varepsilon = \min\{l_i/L_j\}$ over $i, j \in \{1, 2, 3\}$ which is the characteristic parameter of our asymptotic methods. We study the behavior of solutions to the above models when ε tends to 0 for fixed L_i . First of all, we deploy a superscript ε for all geometrical notations, normal vectors, variables, functions, and etc, for example $\Omega^\varepsilon, \Gamma_{int}^{\varepsilon, vac}, n^\varepsilon, x^\varepsilon, \phi^\varepsilon$ instead of $\Omega, \Gamma_{int}^{vac}, n, x, \phi$, resp.

Due to the array size is very small, we introduce a space scale L , a magnitude of the array size, to make a scaling of the array and its cells by putting $\widehat{x}^\varepsilon = x^\varepsilon/L$ that yields a scaling of Ω^ε into $\widehat{\Omega}^\varepsilon$ and Ω_c^ε into $\widehat{\Omega}_c^\varepsilon$ with sizes $\widehat{L}_i = L_i/L$ and $\widehat{l}_i = l_i/L$ for $i = 1, 2, 3$, resp. We now add a hat $\widehat{\cdot}$ into all geometrical notations to represent new scaled domains and their boundaries, i.e $\widehat{\Omega}^{\varepsilon, vac}, \widehat{\Gamma}_{int}^{\varepsilon, vac}$ are scaled regions of $\Omega^{\varepsilon, vac}, \Gamma_{int}^{\varepsilon, vac}$ and add the variables as subscripts to operators such as Laplace Δ , Divergence div to indicate these operators with respect to the variables. For instance, $\Delta_{\widehat{x}^\varepsilon}, div_{\widehat{x}^\varepsilon}$ and Δ_{x^1}, div_{x^1} are Laplace, Divergence operators of the variables \widehat{x}^ε , resp x^1 .

In particular, we note that the parameter $\varepsilon = \min\{l_i/L_j\} = \min\{\widehat{l}_i/\widehat{L}_j\}$ is remained unchanged. As a simplifying approach, we assume that $\widehat{l}_i = \varepsilon$ for all $i = 1, 2, 3$ then the

volume of a scaled cell and the scaled array are $|\Omega_c^\varepsilon| = \prod_i \widehat{l}_i = \varepsilon^3$ and $|\Omega^\varepsilon| = \prod_i \widehat{L}_i = \varepsilon$. From now on, we say ε tends to 0 in the sense that the number of cells in the array $N_c = n_1 \times n_2 \rightarrow +\infty$ with $\varepsilon = 1/N_c$.

We now deal with the scaling for the electrostatic potential and the mechanical displacement. In the electrostatic part, the space scale L is reused, we set $\widehat{V}^\varepsilon = V^\varepsilon/L$ and $\widehat{\phi}^\varepsilon = \phi^\varepsilon/L$ and the electrostatic force (2.2.3) is left unchanged $\widehat{\mathbf{g}}^\varepsilon = \mathbf{g}^\varepsilon$, with $\widehat{\mathbf{g}}^\varepsilon = -\frac{1}{2}\epsilon_0|\widehat{\mathbf{E}}^\varepsilon|^2\widehat{\mathbf{n}}^\varepsilon$, where $\widehat{\mathbf{E}}^\varepsilon$ is the electrical field of the scaled potential $\widehat{\phi}^\varepsilon$. Plugging these new scaled fields into the equation (2.2.1), we obtain the following equations for the scaled potential $\widehat{\phi}^\varepsilon$,

$$\begin{cases} -\Delta_{x^\varepsilon} \widehat{\phi}^\varepsilon & = 0 & \text{in } \widehat{\Omega}^{\varepsilon,vac} \\ \widehat{\phi}^\varepsilon & = \widehat{V}^\varepsilon & \text{on } \widehat{\Gamma}_{int}^{\varepsilon,vac} \\ \nabla_{x^\varepsilon} \widehat{\phi}^\varepsilon \cdot \widehat{\mathbf{n}}^\varepsilon & = 0 & \text{on } \widehat{\Gamma}_{ext}^{\varepsilon,vac}. \end{cases} \quad (2.4.1)$$

In the linear elasticity part, we propose the scaled displacement $\widehat{\mathbf{u}}^\varepsilon = \mathbf{u}^\varepsilon/\varepsilon L$ and the scaled external force $\widehat{\mathbf{f}}^\varepsilon = \varepsilon \mathbf{f}^\varepsilon/L$, then them into (2.2.4), we get the scaled linear elasticity systems,

$$\begin{cases} -\text{div}_{x^\varepsilon} \widehat{\sigma}_{x^\varepsilon}(\widehat{\mathbf{u}}^\varepsilon) & = \varepsilon^{-2} \widehat{\mathbf{f}}^\varepsilon & \text{in } \widehat{\Omega}^{\varepsilon,mec} \\ \widehat{\mathbf{u}}^\varepsilon & = \mathbf{0} & \text{on } \widehat{\Gamma}_0^{\varepsilon,mec} \\ \widehat{\sigma}_{x^\varepsilon}(\widehat{\mathbf{u}}^\varepsilon) \widehat{\mathbf{n}}^\varepsilon & = \varepsilon^{-1} \widehat{\mathbf{g}}^\varepsilon & \text{in } \widehat{\Gamma}_1^{\varepsilon,mec}. \end{cases} \quad (2.4.2)$$

Remark 1:

For simplicity of notation, we hereafter remove the hat $\widehat{\cdot}$ from all the notations, for instance, $\Omega^{\varepsilon,mec}$, ϕ^ε replaces $\widehat{\Omega}^{\varepsilon,mec}$, resp. $\widehat{\phi}^\varepsilon$, and we employ the notation Γ referring to a subboundary of a domain with name sharing with the domain name, for example, $\Gamma^{\varepsilon,vac}$ is a subboundary of $\Omega^{\varepsilon,vac}$.

2.5/ TWO-SCALE TRANSFORM OPERATORS AND THE PROOF ALGORITHM

In this section, we recall the two-scale transform operator or unfolding operator in a domain introduced in [63, 38, 36, 37, 10, 32], and define the boundary two-scale transform operator equivalent to the boundary unfolding operator found in [41, 65]. These operators are used to build the periodic solution model. For each of the other four models, specific two-scale transformations will be introduced in the related sections. The last part of this section is devoted to the formulation of the algorithm that underlies all the model constructions in this chapter.

2.5.1/ INTERNAL TWO-SCALE TRANSFORM OPERATOR

Let us begin by introducing $\Omega^\sharp \subset \mathbb{R}^2$ such that $\Omega^\varepsilon = \Omega^\sharp \times]0, \varepsilon[$ with a partition $\{\Omega_c^\sharp\}_c$ where $\Omega_c^\sharp = [(c_1 - 1)\varepsilon, c_1\varepsilon[\times [(c_2 - 1)\varepsilon, c_2\varepsilon[$, $c = (c_1, c_2) \in \mathcal{I}_{mul}$, and $x^{\sharp,c}$ is the center of the cell Ω_c^\sharp defined as $x^{\sharp,c} = (c_1\varepsilon - \varepsilon/2, c_2\varepsilon - \varepsilon/2)$. It follows that $\Omega_c^\varepsilon = \Omega_c^\sharp \times]0, \varepsilon[$ and that $x^{\varepsilon,c} = (x^{\sharp,c}, \varepsilon/2)$ where $x^{\varepsilon,c}$ is the center of the cell Ω_c^ε .

We now represent the reference cell also called the unit periodicity cell Ω^1 residing at the position $] -1/2, 1/2[^3$, see Figure 2.4. Its boundaries of the vacuum and mechanical parts are denoted by $\partial\Omega^{1,vac} = \Gamma_{int}^{1,vac} \cup \Gamma_{per}^{1,vac} \cup \Gamma_{top}^{1,vac}$ and $\partial\Omega^{1,mec} = \Gamma_0^{1,vac} \cup \Gamma_1^{1,vac} \cup \Gamma_{per}^{1,vac}$. Obviously, if $x^\varepsilon \in \Omega_c^\varepsilon$, $c \in \mathcal{I}_{mul}$ then $(x^\varepsilon - x^{\varepsilon,c})/\varepsilon \in \Omega^1$, and $\Omega^\varepsilon = \cup_{c \in \mathcal{I}_{mul}} (z_c + (1/2, 1/2, 1/2) + \Omega^1)$ where $z_c = (z_{c_1}, z_{c_2}, 0)$, $z_{c_1}, z_{c_2} \in \mathbb{Z}$.

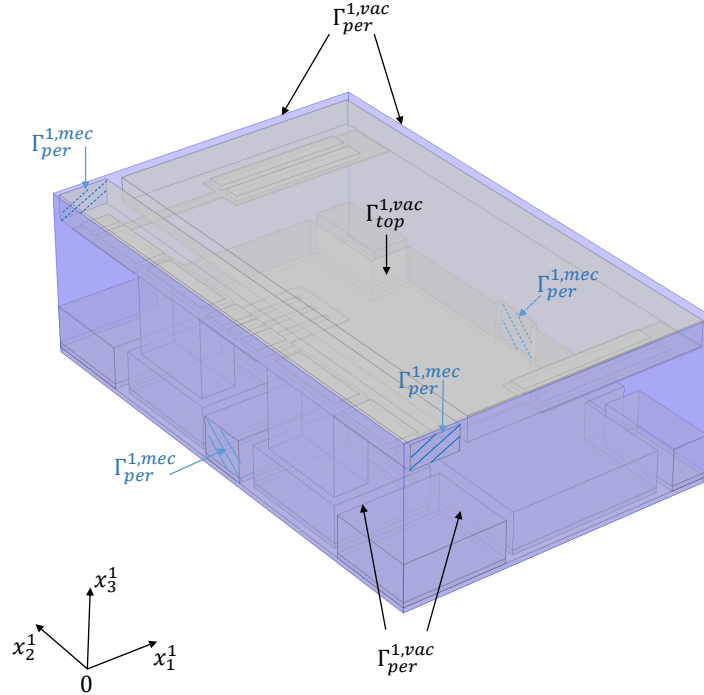


Figure 2.4: The reference cell Ω^1 constructed by the mechanical body $\Omega^{1,mec}$ surrounded by the vacuum part $\Omega^{1,vac}$ occupies the block $] -1/2, 1/2[^3$

Definition 3:

For all φ in $L^2(\Omega^\varepsilon)$, the two-scale transform operator T^ε is defined as

$$T^\varepsilon(\varphi)(x^\sharp, x^1) = \sum_c \chi_{\Omega_c^\sharp}(x^\sharp) \varphi(x^{\varepsilon,c} + \varepsilon x^1),$$

for all $x^\sharp \in \Omega^\sharp$ and $x^1 \in \Omega^1$, where χ is the characteristic function.

Proposition 1:

The two-scale transform operator has the following properties:

1. T^ε is a linear and continuous operator from $L^2(\Omega^\varepsilon)$ to $L^2(\Omega^\# \times \Omega^1)$.
2. For every $\varphi, \psi \in L^2(\Omega^\varepsilon)$, $T^\varepsilon(\varphi\psi) = T^\varepsilon(\varphi)T^\varepsilon(\psi)$.
3. For every $\varphi \in L^1(\Omega^\varepsilon)$, we have

$$\int_{\Omega^\varepsilon} \varphi dx^\varepsilon = \varepsilon \int_{\Omega^\# \times \Omega^1} T^\varepsilon(\varphi) dx^\# dx^1.$$

4. For every $\varphi \in L^2(\Omega^\varepsilon)$, we obtain $\|\varphi\|_{L^2(\Omega^\varepsilon)} = \sqrt{\varepsilon} \|T^\varepsilon(\varphi)\|_{L^2(\Omega^\# \times \Omega^1)}$.
5. For all $\varphi \in H^1(\Omega^\varepsilon)$, $\nabla_{x^1} T^\varepsilon(\varphi) = \varepsilon T^\varepsilon(\nabla_{x^\varepsilon} \varphi)$

Proof. By Definition 3, we obtain (1) and (2) and by changing a variable $x^\varepsilon = x^{\varepsilon,c} + \varepsilon x^1$ yields

$$\begin{aligned} \int_{\Omega^\# \times \Omega^1} T^\varepsilon(\varphi) dx^\# dx^1 &= \sum_c \int_{\Omega_c^\# \times \Omega^1} \varphi(x^{\varepsilon,c} + \varepsilon x^1) dx^\# dx^1 \\ &= \frac{1}{\varepsilon^3} \sum_c |\Omega_c^\#| \int_{\Omega_c^\varepsilon} \varphi(x^\varepsilon) dx^\varepsilon \\ &= \frac{1}{\varepsilon} \int_{\Omega^\varepsilon} \varphi dx^\varepsilon. \end{aligned}$$

From that, (3) and (4) are straightforward.

We now prove (5), for a fixed ε , let φ be in $W^{1,2}(\Omega^\varepsilon)$. By density, there exists $\{\varphi_n\}$ in $C^1(\Omega^\# \times \Omega^1) \cap W^{1,2}(\Omega^\varepsilon)$ such that φ_n converges to φ in $W^{1,2}(\Omega^\varepsilon)$ then we obtain $\nabla_{x^1} T^\varepsilon(\varphi_n) = \varepsilon T^\varepsilon(\nabla_{x^\varepsilon} \varphi_n)$. By (1), it follows that $T^\varepsilon(\varphi_n) \rightarrow T^\varepsilon(\varphi)$ strongly in $L^2(\Omega^\# \times \Omega^1)$ and $T^\varepsilon(\nabla_{x^\varepsilon} \varphi_n) \rightarrow T^\varepsilon(\nabla_{x^\varepsilon} \varphi)$ strongly in $L^2(\Omega^\# \times \Omega^1)$. It leads to $\nabla_{x^1} T^\varepsilon(\varphi) = \varepsilon T^\varepsilon(\nabla_{x^\varepsilon} \varphi)$. \square

Remark 2:

The new norm $\|\cdot\| = \varepsilon^{-1/2} \|\cdot\|$ is designed to avoid the factor $\varepsilon^{1/2}$ from the height of a thin domain.

We next recall the adjoint operator of T^ε and its properties.

Definition 4:

For every $\varphi \in L^2(\Omega^\varepsilon)$, define the adjoint operator $T^{\varepsilon*}$ of T^ε by

$$\frac{1}{\varepsilon} \int_{\Omega^\varepsilon} \varphi T^{\varepsilon*}(\psi) dx^\varepsilon = \int_{\Omega^\# \times \Omega^1} T^\varepsilon(\varphi) \psi dx^\# dx^1,$$

for all $\psi \in L^2(\Omega^\# \times \Omega^1)$.

From this definition, we obtain the explicit formula of $T^{\varepsilon*}$,

$$T^{\varepsilon*}(\psi)(x^\varepsilon) = \frac{1}{\varepsilon^2} \sum_c \int_{\Omega_c^\#} \psi \left(x^\#, \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) dx^\# \chi_{\Omega_c^\varepsilon}(x^\varepsilon). \quad (2.5.1)$$

It is evident that $T^{\varepsilon*}(\psi)$ is not regular. Then we will define the smoothing operator to approximate it.

Let us assume that $\psi \in C^1(\Omega^\# \times \Omega^1)$ and the second derivative of ψ with respect to $x_i^\#$ exists for all $i = 1, 2$, then we apply the Taylor formula for ψ at a point $((x_1^\varepsilon, x_2^\varepsilon), (x^\varepsilon - x^{\varepsilon,c})/\varepsilon)$ with respect to $x^\#$:

$$\begin{aligned} \psi(x^\#, (x^\varepsilon - x^{\varepsilon,c})/\varepsilon) &= \psi((x_1^\varepsilon, x_2^\varepsilon), (x^\varepsilon - x^{\varepsilon,c})/\varepsilon) + (x_1^\# - x_1^\varepsilon) \partial_{x_1^\#} \psi((x_1^\varepsilon, x_2^\varepsilon), (x^\varepsilon - x^{\varepsilon,c})/\varepsilon) \\ &\quad + (x_2^\# - x_2^\varepsilon) \partial_{x_2^\#} \psi((x_1^\varepsilon, x_2^\varepsilon), (x^\varepsilon - x^{\varepsilon,c})/\varepsilon) + \varepsilon O(\varepsilon). \end{aligned}$$

The approximation of the integral is

$$\begin{aligned} \int_{\Omega_c^\#} \psi \left(x^\#, \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) dx^\# \chi_{\Omega_c^\varepsilon}(x^\varepsilon) &= \int_{\Omega_c^\#} \psi \left((x_1^\varepsilon, x_2^\varepsilon), \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) dx^\# \chi_{\Omega_c^\varepsilon}(x^\varepsilon) \\ &\quad + \int_{\Omega_c^\#} (x_1^\# - x_1^\varepsilon) \partial_{x_1^\#} \psi \left((x_1^\varepsilon, x_2^\varepsilon), \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) dx^\# \chi_{\Omega_c^\varepsilon}(x^\varepsilon) \\ &\quad + \int_{\Omega_c^\#} (x_2^\# - x_2^\varepsilon) \partial_{x_2^\#} \psi \left((x_1^\varepsilon, x_2^\varepsilon), \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) dx^\# \chi_{\Omega_c^\varepsilon}(x^\varepsilon) \\ &\quad + \int_{\Omega_c^\#} \varepsilon O(\varepsilon) dx^\# \chi_{\Omega_c^\varepsilon}(x^\varepsilon) \\ &= \psi \left((x_1^\varepsilon, x_2^\varepsilon), \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) |\Omega_c^\#| \chi_{\Omega_c^\varepsilon}(x^\varepsilon) + O(\varepsilon) |\Omega_c^\#| \chi_{\Omega_c^\varepsilon}(x^\varepsilon). \end{aligned}$$

Plugging it into (2.5.1), we obtain

$$\begin{aligned} T^{\varepsilon*}(\psi)(x^\varepsilon) &= \frac{1}{\varepsilon^2} \sum_c \psi \left((x_1^\varepsilon, x_2^\varepsilon), \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) |\Omega_c^\#| \chi_{\Omega_c^\varepsilon}(x^\varepsilon) + \frac{1}{\varepsilon^2} \sum_c O(\varepsilon) |\Omega_c^\#| \chi_{\Omega_c^\varepsilon}(x^\varepsilon) \\ &= \sum_c \psi \left((x_1^\varepsilon, x_2^\varepsilon), \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) \chi_{\Omega_c^\varepsilon}(x^\varepsilon) + O(\varepsilon). \end{aligned}$$

It leads to define a smoothing operator B^ε in order to approximate $T^{\varepsilon*}$.

Definition 5:

For all $\psi \in C^1(\Omega^\# \times \Omega^1)$, the operator B^ε is defined as follows

$$B^\varepsilon(\psi)(x^\varepsilon) = \psi\left(P(x^\varepsilon), \frac{x^\varepsilon}{\varepsilon} - \frac{1}{2}\right),$$

where $P(x^\varepsilon) = (x_1^\varepsilon, x_2^\varepsilon)$.

Proposition 2:

For all ψ in $C^1(\Omega^\# \times \Omega^1)$ and Ω^1 -periodic in the directions x_1^1 and x_2^1 , we have

$$T^{\varepsilon*}(\psi)(x^\varepsilon) = B^\varepsilon(\psi)(x^\varepsilon) + O(\varepsilon).$$

Proposition 3:

Let φ^ε be a sequence in $L^2(\Omega^\varepsilon)$ that satisfies

$$\|\|\varphi^\varepsilon\|\|_{L^2(\Omega^\varepsilon)} \leq C \quad \text{and} \quad \varepsilon \|\|\nabla_{x^\varepsilon}\varphi^\varepsilon\|\|_{L^2(\Omega^\varepsilon)} \leq C.$$

Then, there exist a subsequence (still denoted ε) $T^\varepsilon(\varphi^\varepsilon)$ and a function φ^0 in $L^2(\Omega^\#; H^1(\Omega^1))$, Ω^1 -periodic in directions x_1^1, x_2^1 , such that

1. i.: $T^\varepsilon(\varphi^\varepsilon) \rightharpoonup \varphi^0$ weakly in $L^2(\Omega^\# \times \Omega^1)$,
2. ii.: $\varepsilon T^\varepsilon(\nabla_{x^\varepsilon}\varphi^\varepsilon) \rightharpoonup \nabla_{x^1}\varphi^0$ weakly in $L^2(\Omega^\# \times \Omega^1)$.

Remark 3:

In this chapter, the value of a constant C will be changed place by place.

Let us briefly recall the definition of two-scale convergence for a thin domain, see in [69], arising from the classical two-scale convergence which was originally introduced in [75, 4], see also [96].

Definition 6:

A bounded sequence φ^ε in $L^2(\Omega^\varepsilon, \|\|\cdot\|\|)$ is said to be two-scale convergent to a limit φ^0 belonging to $L^2(\Omega^\# \times \Omega^1)$ if, for any smooth function ψ defined in $\Omega^\# \times \Omega^1$ and Ω^1 -periodic in the directions x_i^1 , $i = 1, 2$, we have

$$\frac{1}{\varepsilon} \int_{\Omega^\varepsilon} \varphi^\varepsilon(x^\varepsilon) \psi\left((x_1^\varepsilon, x_2^\varepsilon), \frac{x^\varepsilon}{\varepsilon} - \frac{1}{2}\right) dx^\varepsilon \rightarrow \int_{\Omega^\# \times \Omega^1} \varphi^0(x^\#, x^1) \psi(x^\#, x^1) dx^\# dx^1.$$

The following result proves that the weak convergence of the two-scale transform operator is equivalent to the original two-scale convergence.

Proposition 4:

Let φ^ε be a bounded sequence in $L^2(\Omega^\varepsilon, \|\cdot\|)$, then the following properties are equivalent

1. $T^\varepsilon(\varphi^\varepsilon)$ converges weakly to φ^0 in $L^2(\Omega^\sharp \times \Omega^1)$,
2. φ^ε two-scale converges to φ^0 .

Proof. By the definition of B^ε , we can write

$$\frac{1}{\varepsilon} \int_{\Omega^\varepsilon} \varphi^\varepsilon(x^\varepsilon) \psi \left((x_1^\varepsilon, x_2^\varepsilon), \frac{x^\varepsilon}{\varepsilon} - \frac{1}{2} \right) dx^\varepsilon = \frac{1}{\varepsilon} \int_{\Omega^\varepsilon} \varphi^\varepsilon(x^\varepsilon) B^\varepsilon(\psi)(x^\varepsilon) dx^\varepsilon.$$

By Proposition 2 and definition of the adjoint operator, it follows that

$$\begin{aligned} \frac{1}{\varepsilon} \int_{\Omega^\varepsilon} \varphi^\varepsilon(x^\varepsilon) B^\varepsilon(\psi)(x^\varepsilon) dx^\varepsilon &= \frac{1}{\varepsilon} \int_{\Omega^\varepsilon} \varphi^\varepsilon(x^\varepsilon) T^{\varepsilon*}(\psi)(x^\varepsilon) dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Omega^\varepsilon} \varphi^\varepsilon(x^\varepsilon) O(\varepsilon) dx^\varepsilon \\ &= \int_{\Omega^\sharp} \int_{\Omega^1} T^\varepsilon(\varphi^\varepsilon)(x^\sharp, x^1) \psi(x^\sharp, x^1) dx^\sharp dx^1 + O(\varepsilon). \end{aligned}$$

And we obtain the result. □

2.5.2/ BOUNDARY TWO-SCALE TRANSFORM OPERATOR

We introduce the two-scale transform operator T_b^ε defined on the boundary of Ω^ε equivalent to the boundary unfolding operator found in [41].

Definition 7:

Let Γ^ε be a periodic internal subboundary of $\partial\Omega^\varepsilon$, Γ^1 be a subboundary of $\partial\Omega^1$, for all φ in $L^2(\Gamma^\varepsilon)$, we define the boundary two-scale transform operator T_b^ε by

$$T_b^\varepsilon(\varphi)(x^\sharp, x^1) = \sum_c \chi_{\Omega_c^\sharp}(x^\sharp) \varphi(x^{\varepsilon,c} + \varepsilon x^1),$$

for a.e. $x^\sharp \in \Omega^\sharp$, $x^1 \in \Gamma^1$.

Proposition 5:

The boundary two-scale transform operator has the following properties:

1. For every $\varphi \in L^1(\Gamma^\varepsilon)$, we have

$$\int_{\Gamma^\varepsilon} \varphi \, ds(x^\varepsilon) = \int_{\Omega^\# \times \Gamma^1} T_b^\varepsilon(\varphi) \, dx^\# \, ds(x^1).$$

2. For every $\varphi \in L^1(\Gamma^\varepsilon)$, we obtain $\|\varphi\|_{L^1(\Gamma^\varepsilon)} = \|T_b^\varepsilon(\varphi)\|_{L^1(\Omega^\# \times \Gamma^1)}$.
3. For all $\varphi \in H^1(\Gamma^\varepsilon)$, $\nabla_{x^1} T_b^\varepsilon(\varphi) = \varepsilon T_b^\varepsilon(\nabla_{x^\varepsilon} \varphi)$.

Definition 8:

For every $\varphi \in L^2(\Gamma^\varepsilon)$, we define the adjoint operator $T_b^{\varepsilon*}$ of T_b^ε by

$$\int_{\Gamma^\varepsilon} \varphi T_b^{\varepsilon*}(\psi) \, ds(x^\varepsilon) = \int_{\Omega^\# \times \Gamma^1} T_b^\varepsilon(\varphi) \psi \, dx^\# \, ds(x^1),$$

for all $\psi \in L^2(\Omega^\# \times \Gamma^1)$.

Hence the explicit formula of $T_b^{\varepsilon*}$ is written

$$T_b^{\varepsilon*}(\psi)(x^\varepsilon) = \frac{1}{\varepsilon} \sum_c \int_{\Omega_c^\#} \psi \left(x^\#, \frac{x^\varepsilon - x^{\varepsilon,c}}{\varepsilon} \right) dx^\# \chi_{\Omega_c^\varepsilon}(x^\varepsilon).$$

Definition 9:

For all v in $C^1(\Omega^\# \times \Gamma^1)$, the operator B_b^ε is defined by

$$B_b^\varepsilon(\psi)(x^\varepsilon) = \psi \left(P(x^\varepsilon), \frac{x^\varepsilon}{\varepsilon} - \frac{1}{2} \right),$$

where $P(x^\varepsilon) = (x_1^\varepsilon, x_2^\varepsilon)$.

Proposition 6:

For every ψ in $C^1(\Omega^\# \times \Gamma^1)$ and Γ^1 -periodic in the directions x_1^1 and x_2^1 , then

$$T_b^{\varepsilon*}(\psi)(x^\varepsilon) = B_b^\varepsilon(\psi)(x^\varepsilon) + O(\varepsilon),$$

Proposition 7:

Let φ^ε be a sequence in $L^2(\Gamma^\varepsilon)$ and assume that

$$\|\varphi^\varepsilon\|_{L^2(\Gamma^\varepsilon)} \leq C \quad \text{and} \quad \varepsilon \|\nabla_{x^\varepsilon} \varphi^\varepsilon\|_{L^2(\Gamma^\varepsilon)} \leq C.$$

Then, there exist a subsequence (still denoted ε) $T^\varepsilon(\varphi^\varepsilon)$ and a function φ^0 in $L^2(\Omega^\#; H^1(\Gamma^1))$, Γ^1 -periodic in the directions x_1^1 and x_2^1 such that

1. $T^\varepsilon(\varphi^\varepsilon) \rightharpoonup \varphi^0$ weakly in $L^2(\Omega^\# \times \Gamma^1)$,
2. $\varepsilon T^\varepsilon(\nabla_{x^\varepsilon} \varphi^\varepsilon) \rightharpoonup \nabla_{x^1} \varphi^0$ weakly in $L^2(\Omega^\# \times \Gamma^1)$.

Let us briefly recall the two-scale convergence on the surface introduced in [7], [83].

Definition 10:

A bounded sequence φ^ε in $L^2(\Gamma^\varepsilon)$ is said to two-scale convergence to a limit φ^0 belonging to $L^2(\Omega^\# \times \Gamma^1)$ if for any smooth function ψ defined in $\Omega^\# \times \Gamma^1$ and Γ^1 -periodic, we have

$$\int_{\Gamma^\varepsilon} \varphi^\varepsilon(x^\varepsilon) \psi\left(x_1^\varepsilon, x_2^\varepsilon, \frac{x^\varepsilon}{\varepsilon} - \frac{1}{2}\right) ds(x^\varepsilon) \rightarrow \int_{\Omega^\#} \int_{\Gamma^1} \varphi^0(x^\#, x^1) \psi(x^\#, x^1) dx^\# ds(x^1).$$

We also have the equivalence between the original two-scale convergence on the surface and weak convergence of the two-scale transform operator by the following proposition.

Proposition 8:

Let φ^ε be a bounded sequence in $L^2(\Gamma^\varepsilon, \|\cdot\|)$, then the following properties are equivalent

1. $\{T_b^\varepsilon(\varphi^\varepsilon)\}$ converges weakly to φ^0 in $L^2(\Omega^\# \times \Gamma^1)$,
2. φ^ε two-scale converges to φ^0 .

2.5.3/ THE REFERENCE ALGORITHM FOR MODEL PROOFS

This section is devoted to the symbolic computation algorithm that served as a reference for the construction of the models reported in [21] and based on the extension-combination method. It is this same algorithm that drives the construction of the five models of this chapter. The operations described therein are high level, the implementation details not being explained because they strongly depend on the way partial differential equations are represented in a symbolic computing environment, see our approach in [94] and in the following chapters of the thesis.

The starting point of the algorithm is a boundary value problem either in strong form or in weak form. It uses the definition of a two-scale transformation T^ε and its associated operators $T^{\varepsilon*}$ and B^ε . These operators and their properties may depend on the model. In this chapter, to each model corresponds a different transformation T^ε . In step (iii) we use the derivation rule of $B^\varepsilon(w)$ which has not been recalled in the previous subsections but will be detailed during each model derivation. In the same way, the approximation of $T^{\varepsilon*}$ by B^ε is established for each model derivation. In the implementation reported in [21] the properties of the operators T^ε , $T^{\varepsilon*}$, and B^ε are used without being reconstructed in order to avoid time-consuming calculations.

- i) Define a two-scale transform (i.e. unfolding) operator T^ε , its adjoint $T^{\varepsilon*}$, and a smooth approximation B^ε of $T^{\varepsilon*}$.
- ii) Derive the very weak form of the boundary value problem with a test function v .
- iii) Replace v by $\varepsilon^k B^\varepsilon(w)$ for some $k \in \mathbb{Z} \setminus \{0\}$, and apply the rule of the derivative of $B^\varepsilon(w)$.
- iv) Replace B^ε by its approximation in terms of $T^{\varepsilon*}$.
- v) Apply the adjoint rule to replace the instances of $T^{\varepsilon*}$ by instances of T^ε that are now applied to an expression Ψ^ε .
- vi) Assuming that $T^\varepsilon(\Psi^\varepsilon)$ is bounded when ε vanishes, an extracted subsequence weakly converges to a limit Ψ^0 .
- vii) Convert the very weak form satisfied by Ψ^0 into its strong form.

2.6/ ELECTROSTATIC MODELS

We now recall the electrostatic equation 2.4.1 in the vacuum domain of MIRA,

$$\begin{cases} -\Delta_{x^\varepsilon} \phi^\varepsilon & = 0 & \text{in } \Omega^{\varepsilon, vac} \\ \phi^\varepsilon & = V^\varepsilon & \text{on } \Gamma_{int}^{\varepsilon, vac} \\ \nabla_{x^\varepsilon} \phi^\varepsilon \cdot \mathbf{n}^\varepsilon & = 0 & \text{on } \Gamma_{ext}^{\varepsilon, vac} . \end{cases} \quad (2.6.1)$$

By the same method in [40], it follows that there exists ϕ^ε in $H_{\Gamma_{int}^{\varepsilon, vac}, V^\varepsilon}^1(\Omega^{\varepsilon, vac}) = \{v^\varepsilon \in H^1(\Omega^{\varepsilon, vac}), v^\varepsilon = V^\varepsilon \text{ on } \Gamma_{int}^{\varepsilon, vac}\}$ such that

$$\int_{\Omega^{\varepsilon, vac}} \nabla_{x^\varepsilon} \phi^\varepsilon \nabla_{x^\varepsilon} v^\varepsilon \, dx^\varepsilon = 0, \quad (2.6.2)$$

for all v^ε in $H_{\Gamma_{int}^{\varepsilon,vac},0}^1(\Omega^{\varepsilon,vac})$.

Applying Green's formula, we have a very weak formulation of Problem 2.6.1,

$$\int_{\Omega^{\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} v^\varepsilon \, dx^\varepsilon = \int_{\Gamma_{int}^{\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} v^\varepsilon \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon) + \int_{\Gamma_{ext}^{\varepsilon,vac}} \varphi^\varepsilon \nabla_{x^\varepsilon} v^\varepsilon \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon), \quad (2.6.3)$$

for all v^ε in $H_{\Gamma_{int}^{\varepsilon,vac},0}^2(\Omega^{\varepsilon,vac}) = \{v^\varepsilon \in H^2(\Omega^{\varepsilon,vac}), v^\varepsilon = 0 \text{ on } \Gamma_{int}^{\varepsilon,vac}\}$.

Assumption 1:

Assume that $\|\phi^\varepsilon\|_{L^2(\Omega^{\varepsilon,vac})}$ and $\varepsilon \|\nabla_{x^\varepsilon} \varphi^\varepsilon\|_{L^2(\Omega^{\varepsilon,vac})}$ are bounded uniformly with respect to ε .

2.6.1/ PERIODIC MODEL

Proposition 9:

Let ϕ^ε be a solution of (2.6.2) and assume that ϕ^ε satisfies Assumption 1. Then, there exists $\phi^0 \in L^2(\Omega^\sharp, H^1(\Omega^{1,vac}))$ $\Omega^{1,vac}$ -periodic in the directions x_1^1, x_2^1 such that $T^\varepsilon \phi^\varepsilon \rightharpoonup \phi^0$ weakly in $L^2(\Omega^\sharp \times \Omega^{1,vac})$. Moreover for a.e $x^\sharp \in \Omega^\sharp$, ϕ^0 is a solution to

$$\begin{cases} -\Delta_{x^1} \phi^0 = 0 & \text{in } \Omega^{1,vac} \\ \phi^0 = V^0 & \text{on } \Gamma_{int}^{1,vac} \\ \nabla_{x^1} \phi^0 \cdot \mathbf{n}^1 = 0 & \text{on } \Gamma_{top}^{1,vac} \\ \nabla_{x^1} \phi^0 \cdot \mathbf{n}^1 & \text{is } \Gamma_{per}^{1,vac}\text{-antiperiodic} \\ \phi^0 & \text{is } \Gamma_{per}^{1,vac}\text{-periodic,} \end{cases}$$

where V^0 is a weak limit of $T^\varepsilon(V^\varepsilon)$ in $L^2(\Omega^\sharp \times \Gamma_{int}^{1,vac})$.

Remark 4:

For the sake of simplicity of notation, we use the same notation for the two-scale transform operator T^ε , the adjoint operator $T^{\varepsilon*}$ and an approximate operator B^ε of $T^{\varepsilon*}$ defined in a domain and those defined on a boundary.

Proof. Thanks to Proposition 3 and Assumption 1, we obtain the existence and the periodicity of ϕ^0 . The proof is completed by showing that ϕ^0 satisfies the above equations.

Let us take w in $C^\infty(\Omega^\sharp \times \overline{\Omega^{1,vac}})$ such that $w = 0$ on $\Gamma_{int}^{1,vac}$ and $\nabla_{x^1} w \cdot \mathbf{n}^1 = 0$ on $\Gamma_{top}^{1,vac} \cup \Gamma_{per}^{1,vac}$.

Obviously, $B^\varepsilon w = 0$ on $\Gamma_{int}^{\varepsilon, vac}$ then we replace v^ε in (2.6.3) by $\varepsilon B^\varepsilon w$,

$$\int_{\Omega^{\varepsilon, vac}} \phi^\varepsilon \Delta_{x^\varepsilon} B^\varepsilon w \, dx^\varepsilon = \int_{\Gamma_{int}^{\varepsilon, vac}} V^\varepsilon \nabla_{x^\varepsilon} B^\varepsilon w \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon) + \int_{\Gamma_{ext}^{\varepsilon, vac}} \varphi^\varepsilon \nabla_{x^\varepsilon} B^\varepsilon w \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon). \quad (2.6.4)$$

It is clear that

$$\begin{aligned} \frac{\partial B^\varepsilon w}{\partial x_i^\varepsilon} &= B^\varepsilon \left(\chi_{I^\#}(i) \frac{\partial w}{\partial x_i^\#} + \frac{1}{\varepsilon} \frac{\partial w}{\partial x_i^1} \right), \\ \frac{\partial}{\partial x_i^\varepsilon} \frac{\partial B^\varepsilon w}{\partial x_i^\varepsilon} &= B^\varepsilon \left(\chi_{I^\#}(i) \frac{\partial}{\partial x_i^\#} \frac{\partial w}{\partial x_i^\#} + \chi_{I^\#}(i) \frac{2}{\varepsilon} \frac{\partial}{\partial x_i^\#} \frac{\partial w}{\partial x_i^1} + \frac{1}{\varepsilon^2} \frac{\partial}{\partial x_i^1} \frac{\partial w}{\partial x_i^1} \right), \end{aligned}$$

for all $i \in I = \{1, 2, 3\}$, $I^\# = \{1, 2\}$.

By a straightforward calculation, the left-hand side (*l.h.s*) of (2.6.4) becomes

$$\begin{aligned} l.h.s &= \varepsilon \int_{\Omega^{\varepsilon, vac}} \phi^\varepsilon B^\varepsilon \left(\sum_{i=1}^2 \frac{\partial}{\partial x_i^\#} \frac{\partial w}{\partial x_i^\#} + \frac{2}{\varepsilon} \sum_{i=1}^2 \frac{\partial}{\partial x_i^\#} \frac{\partial w}{\partial x_i^1} + \frac{1}{\varepsilon^2} \Delta_{x^1} w \right) dx^\varepsilon \\ &= \frac{1}{\varepsilon} \int_{\Omega^{\varepsilon, vac}} \phi^\varepsilon B^\varepsilon (\Delta_{x^1} w) dx^\varepsilon + O(\varepsilon), \end{aligned} \quad (2.6.5)$$

where

$$O(\varepsilon) = \varepsilon \int_{\Omega^{\varepsilon, vac}} \phi^\varepsilon B^\varepsilon \left(\sum_{i=1}^2 \frac{\partial}{\partial x_i^\#} \frac{\partial w}{\partial x_i^\#} \right) dx^\varepsilon + 2 \int_{\Omega^{\varepsilon, vac}} \phi^\varepsilon B^\varepsilon \left(\sum_{i=1}^2 \frac{\partial}{\partial x_i^\#} \frac{\partial w}{\partial x_i^1} \right) dx^\varepsilon.$$

Similarly, the right-hand side (*r.h.s*) of (2.6.4) becomes

$$\begin{aligned} r.h.s &= \varepsilon \int_{\Gamma_{int}^{\varepsilon, vac}} V^\varepsilon \left[\sum_{i=1}^2 B^\varepsilon \left(\frac{\partial w}{\partial x_i^\#} \right) n_i^\varepsilon + \frac{1}{\varepsilon} B^\varepsilon (\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon \right] ds(x^\varepsilon) \\ &+ \varepsilon \int_{\Gamma_{ext}^{\varepsilon, vac}} \varphi^\varepsilon \left[\sum_{i=1}^2 B^\varepsilon \left(\frac{\partial w}{\partial x_i^\#} \right) n_i^\varepsilon + \frac{1}{\varepsilon} B^\varepsilon (\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon \right] ds(x^\varepsilon). \end{aligned}$$

It is clear from $\nabla_{x^1} w \cdot \mathbf{n}^1 = 0$ on $\Gamma_{top}^{1, vac} \cup \Gamma_{per}^{1, vac}$ that $B^\varepsilon (\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon = 0$ on $\Gamma_{ext}^{\varepsilon, vac} = \Gamma_{top}^{\varepsilon, vac} \cup \Gamma_{lat}^{\varepsilon, vac}$, then

$$r.h.s = \int_{\Gamma_{int}^{\varepsilon, vac}} V^\varepsilon B^\varepsilon (\nabla_{x^1} w \cdot \mathbf{n}^1) \, ds(x^\varepsilon) + O(\varepsilon), \quad (2.6.6)$$

where

$$O(\varepsilon) = \varepsilon \sum_{i=1}^2 \int_{\partial \Omega^{\varepsilon, vac}} \varphi^\varepsilon B^\varepsilon \left(\frac{\partial w}{\partial x_i^\#} \right) n_i^\varepsilon \, ds(x^\varepsilon).$$

Combining with (2.6.5) and (2.6.6), we can assert that

$$\frac{1}{\varepsilon} \int_{\Omega^{\varepsilon, vac}} \phi^\varepsilon B^\varepsilon (\Delta_{x^1} w) \, dx^\varepsilon = \int_{\Gamma_{int}^{\varepsilon, vac}} V^\varepsilon B^\varepsilon (\nabla_{x^1} w \cdot \mathbf{n}^1) \, ds(x^\varepsilon) + O(\varepsilon).$$

Approximating B^ε by $T^{\varepsilon*}$ from Proposition 2 and Proposition 6 gives

$$\frac{1}{\varepsilon} \int_{\Omega^{\varepsilon,vac}} \phi^\varepsilon T^{\varepsilon*} (\Delta_{x^1} w) dx^\varepsilon = \int_{\Gamma_{int}^{\varepsilon,vac}} V^\varepsilon T^{\varepsilon*} (\nabla_{x^1} w \cdot \mathbf{n}^1) ds(x^\varepsilon) + O(\varepsilon).$$

The definition of $T^{\varepsilon*}$ yields

$$\int_{\Omega^\# \times \Omega^{1,vac}} T^\varepsilon(\phi^\varepsilon) \Delta_{x^1} w dx^\# dx^1 = \int_{\Omega^\# \times \Gamma_{int}^{1,vac}} T^\varepsilon(V^\varepsilon) \nabla_{x^1} w \cdot \mathbf{n}^1 dx^\# ds(x^1) + O(\varepsilon).$$

Passing ε to 0 with Proposition 3 and Proposition 7, we get

$$\int_{\Omega^\# \times \Omega^{1,vac}} \phi^0 \Delta_{x^1} w dx^\# dx^1 = \int_{\Omega^\# \times \Gamma_{int}^{1,vac}} V^0 \nabla_{x^1} w \cdot \mathbf{n}^1 dx^\# ds(x^1).$$

Applying Green's formula twice, combining with conditions satisfied by w and decomposing $\partial\Omega^{1,vac} = \Gamma_{int}^{1,vac} \cup \Gamma_{per}^{1,vac} \cup \Gamma_{top}^{1,vac}$, we obtain

$$\begin{aligned} & \int_{\Omega^\# \times \Omega^{1,vac}} \Delta_{x^1} \phi^0 w dx^\# dx^1 - \int_{\Omega^\# \times (\Gamma_{per}^{1,vac} \cup \Gamma_{top}^{1,vac})} \nabla_{x^1} \phi^0 \cdot \mathbf{n}^1 w dx^\# ds(x^1) \\ & + \int_{\Omega^\# \times \Gamma_{int}^{1,vac}} \phi^0 \nabla_{x^1} w \cdot \mathbf{n}^1 dx^\# ds(x^1) = \int_{\Omega^\# \times \Gamma_{int}^{1,vac}} V^0 \nabla_{x^1} w \cdot \mathbf{n}^1 dx^\# ds(x^1). \end{aligned}$$

Now, we choose w such that $w = 0$ on $\Gamma_{per}^{1,vac} \cup \Gamma_{top}^{1,vac}$ and $\nabla_{x^1} w \cdot \mathbf{n}^1 = 0$ on $\Gamma_{int}^{1,vac}$. By Lemma 1 we deduce that

$$\Delta_{x^1} \phi^0 = 0 \text{ in } \Omega^{1,vac}.$$

Next, we choose w such that $w = 0$ on $\Gamma_{per}^{1,vac} \cup \Gamma_{top}^{1,vac}$, Lemma 2 gives

$$\phi^0 = V^0 \text{ on } \Gamma_{int}^{1,vac}.$$

And then, we choose $w = 0$ on $\Gamma_{per}^{1,vac}$, it follows from Lemma 2 that

$$\nabla_{x^1} \phi^0 \cdot \mathbf{n}^1 = 0 \text{ on } \Gamma_{top}^{1,vac}.$$

Finally, we conclude from Lemma 3 that

$$\nabla_{x^1} \phi^0 \cdot \mathbf{n}^1 \text{ is } \Gamma_{per}^{1,vac} \text{ - antiperiodic.}$$

□

2.6.2/ LATERAL BOUNDARY LAYER MODEL

One observe that the solution ϕ^0 of the periodic model of the external zone, see Proposition 9, does not satisfy the boundary conditions due to its periodicity on the boundary. It

leads to propose a corrector term on the lateral boundary to eliminate this problem. We now introduce boundary layer terms $\phi_{bl}^\varepsilon = \phi^\varepsilon - B^\varepsilon(\phi^0)$ and $v_{bl}^\varepsilon = V^\varepsilon - B^\varepsilon(V^0)$, where B^ε is an approximate operator of the adjoint operator $T^{\varepsilon*}$ of the two-scale transform operator T^ε defined in the previous part, V^0 is a weak limit of $T^\varepsilon(V^\varepsilon)$ in $L^2(\Omega^\# \times \Gamma_{1,int}^{1,vac})$ given in Proposition 9. We now investigate the convergence of ϕ_{bl}^ε at the first lateral boundary. The convergence on the other boundaries can be derived in the same way.

2.6.2.1/ GEOMETRY NOTATIONS

Let $\Omega_{bl,1}^{\alpha\varepsilon}$ be a subdomain of Ω^ε defined as $\Omega_{bl,1}^{\alpha\varepsilon} = \cup_{c \in I_{bl,1}} \Omega_c^\varepsilon$ where $I_{bl,1} := \{c = (c_1, c_2) : c_1 \in \overline{1, n_1} \text{ and } c_2 \in \overline{1, \alpha}\}$, with $\alpha \in \mathbb{N}^*$ such that $\alpha\varepsilon < L_1^2$, where L_1^2 is a positive number, see Figure 2.5. The boundary of the vacuum zone $\Omega_{bl,1}^{\alpha\varepsilon}$ is denoted as $\partial\Omega_{bl,1}^{\alpha\varepsilon}$ assembled by the union of $\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}$ and $\Gamma_{bl,1,ext}^{\alpha\varepsilon,vac} = \Gamma_{bl,1,\alpha}^{\alpha\varepsilon,vac} \cup \Gamma_{bl,1,top}^{\alpha\varepsilon,vac} \cup \Gamma_{bl,1,lat}^{\alpha\varepsilon,vac}$, where $\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}$, $\Gamma_{bl,1,top}^{\alpha\varepsilon,vac}$, $\Gamma_{bl,1,lat}^{\alpha\varepsilon,vac}$ are respectively subboundaries of $\Gamma_{1,int}^{\varepsilon,vac}$, $\Gamma_{1,top}^{\varepsilon,vac}$, $\Gamma_{1,lat}^{\varepsilon,vac}$, while $\Gamma_{bl,1,\alpha}^{\alpha\varepsilon,vac}$ is a subboundary of the internal vacuum boundary between cells of the external zone Ω_1^ε . The boundary of the mechanical body $\Omega_{bl,1}^{\alpha\varepsilon,mec}$ is defined as $\partial\Omega_{bl,1}^{\alpha\varepsilon,mec}$ constructed by $\Gamma_{bl,1,0}^{\alpha\varepsilon,mec} \cup \Gamma_{bl,1,1}^{\alpha\varepsilon,mec} \cup \Gamma_{bl,1,\alpha}^{\alpha\varepsilon,mec}$, where $\Gamma_{bl,1,0}^{\alpha\varepsilon,mec}$ and $\Gamma_{bl,1,1}^{\alpha\varepsilon,mec}$ are respectively subboundaries of $\Gamma_{1,0}^{\varepsilon,mec}$ and $\Gamma_{1,1}^{\varepsilon,mec}$, $\Gamma_{bl,1,\alpha}^{\alpha\varepsilon,mec}$ is a subboundary of the internal mechanical subboundary between cells of Ω_1^ε .

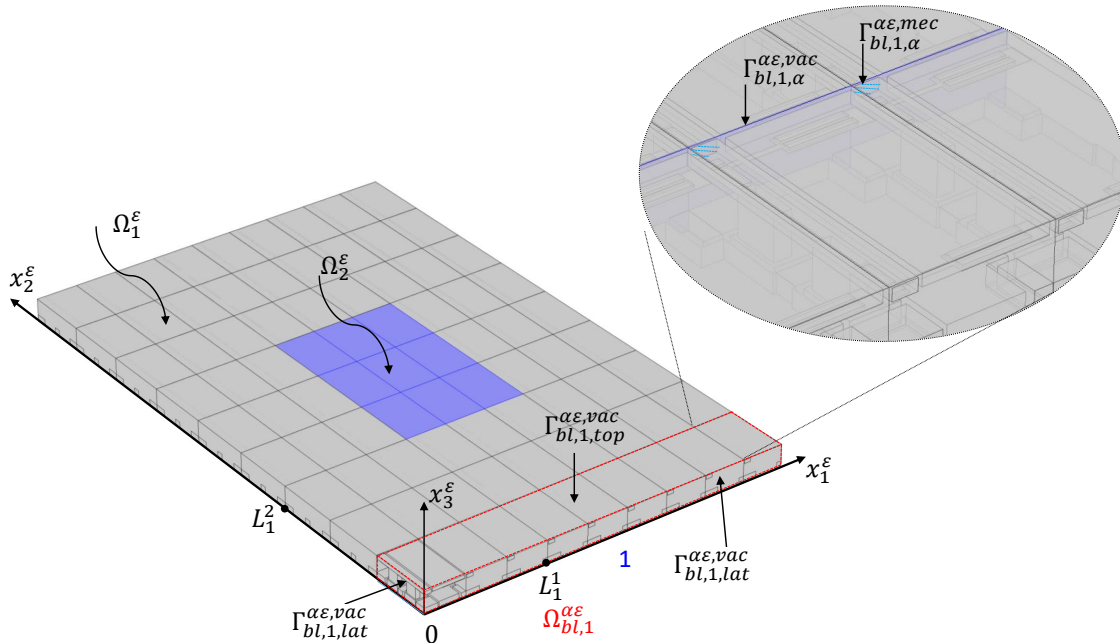


Figure 2.5: The physical domain $\Omega_{bl,1}^{\alpha\varepsilon}$ for the first lateral boundary model with two subdomains the mechanical body $\Omega_{bl,1}^{\alpha\varepsilon,mec}$ and the vacuum part $\Omega_{bl,1}^{\alpha\varepsilon,vac}$ with $\alpha = 1$. The zoom represents the internal subboundaries of the vacuum and the mechanical part between cells of the external zone.

We next denote $\Omega_{bl,1}^\# = [0, L_1[$, with a partition $\{\Omega_{bl,1c_1}^\#\}_{c_1}$, $\Omega_{bl,1c_1}^\# = [(c_1 - 1)\varepsilon, c_1\varepsilon[$, $c_1 = 1, \dots, n_1$ and denote $x^{\#,c_1} = c_1\varepsilon - \varepsilon/2$ as the center of $\Omega_{bl,1c_1}^\#$.

The finite microscopic domain $\Omega_{bl,1}^1$ is built by $\Omega_{bl,1}^1 = \cup_{\xi=0}^{\alpha-1} (\Omega^1 + (0, 1/2 + \xi, 1/2))$, see Figure 2.6. The boundary of the finite vacuum part is denoted as $\partial\Omega_{bl,1}^{1,vac} = \Gamma_{bl,1,int}^{1,vac} \cup \Gamma_{bl,1,per}^{1,vac} \cup \Gamma_{bl,1,top}^{1,vac} \cup \Gamma_{bl,1,0}^{1,vac} \cup \Gamma_{bl,1,\alpha}^{1,vac}$ and the boundary of the finite mechanical part is expressed $\partial\Omega_{bl,1}^{1,mec} = \Gamma_{bl,1,0}^{1,mec} \cup \Gamma_{bl,1,1}^{1,mec} \cup \Gamma_{bl,1,per}^{1,mec} \cup \Gamma_{bl,1,Or}^{1,mec} \cup \Gamma_{bl,1,\alpha}^{1,mec}$.

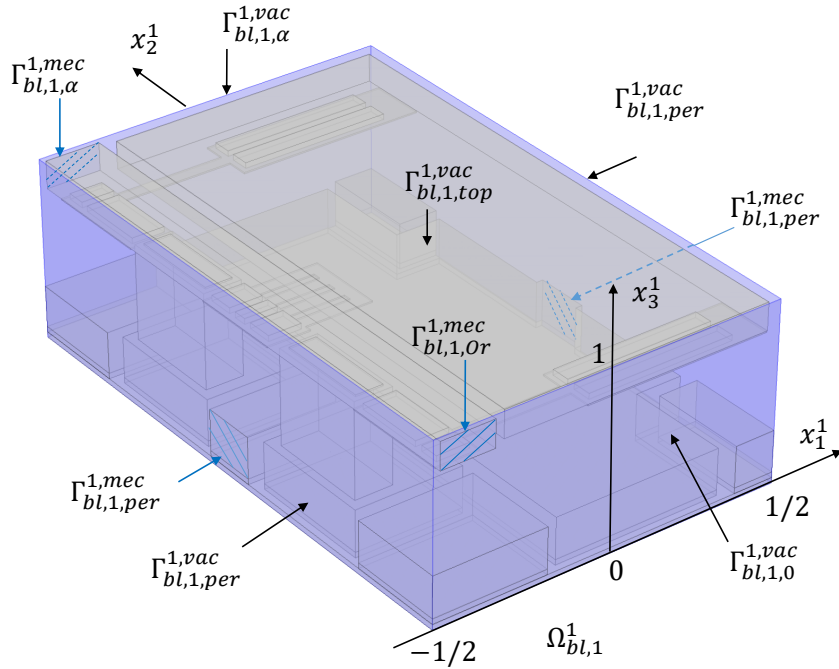


Figure 2.6: The microscopic domain $\Omega_{bl,1}^1$ with two subdomains $\Omega_{bl,1}^{1,mec}$ and $\Omega_{bl,1}^{1,vac}$ with $\alpha = 1$.

The infinite microscopic domain $\Omega_{bl,1}^\infty$ is defined as $\Omega_{bl,1}^\infty = \cup_{\xi=0}^{+\infty} (\Omega^1 + (0, 1/2 + \xi, 1/2))$ with the boundary of the infinite vacuum part $\partial\Omega_{bl,1}^{\infty,vac} = \Gamma_{bl,1,int}^{\infty,vac} \cup \Gamma_{bl,1,per}^{\infty,vac} \cup \Gamma_{bl,1,top}^{\infty,vac} \cup \Gamma_{bl,1,0}^{\infty,vac} \cup \Gamma_{bl,1,+\infty}^{\infty,vac}$ and the boundary of the infinite mechanical part $\partial\Omega_{bl,1}^{\infty,mec} = \Gamma_{bl,1,0}^{\infty,mec} \cup \Gamma_{bl,1,1}^{\infty,mec} \cup \Gamma_{bl,1,per}^{\infty,mec} \cup \Gamma_{bl,1,Or}^{\infty,mec} \cup \Gamma_{bl,1,+\infty}^{\infty,mec}$. These subboundaries are defined as the extension of the subboundaries of the finite domain $\Omega_{bl,1}^1$.

Remark 5:

We use the subscript $i = 1, 2, 3, 4$ for all geometrical notations and operators, the superscript i for all functions to indicate which lateral boundary models they belong to, corresponding to the index in Figure 2.1. For instance, $\Omega_{bl,1}^{\alpha\varepsilon}$ and $\Omega_{bl,2}^{\alpha\varepsilon}$ are the first and the second physical domains, $T_{bl,1}^\varepsilon$ and $T_{bl,2}^\varepsilon$ are the first and the second boundary layer two-scale transform operators, ϕ_{bl}^1 and ϕ_{bl}^2 are the solutions of the first and the second lateral boundary models.

When we say for each α , it means that for all α satisfied the condition 2.6.2.1, $\alpha \in \mathbb{Z}^+$ such that $\alpha\varepsilon < L_1^2$, L_1^2 is a positive given number.

Next, we introduce the two-scale transform for the first lateral model.

2.6.2.2/ BOUNDARY LAYER TWO-SCALE TRANSFORM OPERATOR**Definition 11:**

For every φ in $L^2(\Omega_{bl,1}^{\alpha\varepsilon})$ (resp. φ in $L^2(\Gamma_{bl,1}^{\alpha\varepsilon})$), we define a boundary layer two-scale transform operator $T_{bl,1}^\varepsilon$ by

$$T_{bl,1}^\varepsilon(\varphi)(x^\#, x^1) = \sum_{c_1} \chi_{\Omega_{bl,1c_1}^\#}(x^\#) \varphi(x^\#, c_1 + \varepsilon x_1^1, \varepsilon x_2^1, \varepsilon x_3^1),$$

for a.e. $x^\# \in \Omega_{bl,1}^\#$, $x^1 \in \Omega_{bl,1}^1$ (resp. $x^1 \in \Gamma_{bl,1}^1$).

Definition 12:

We define the adjoint operator $T_{bl,1}^{\varepsilon*}$ of $T_{bl,1}^\varepsilon$ by

$$\frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon}} \varphi T_{bl,1}^{\varepsilon*}(\psi) dx^\varepsilon = \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,\alpha}} T_{bl,1}^\varepsilon(\varphi) \psi dx^\# dx^1,$$

for all $\psi \in L^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^1)$, $\varphi \in L^2(\Omega_{bl,1}^{\alpha\varepsilon})$, respectively by

$$\frac{1}{\varepsilon} \int_{\Gamma_{bl,1}^{\alpha\varepsilon}} \varphi T_{bl,1}^{\varepsilon*}(\psi) ds(x^\varepsilon) = \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1}^{1,\alpha}} T_{bl,1}^\varepsilon(\varphi) \psi dx^\# ds(x^1),$$

for all $\psi \in L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1}^1)$, $\varphi \in L^2(\Gamma_{bl,1}^{\alpha\varepsilon})$.

From the definition, the explicit formula follows

$$T_{bl,1}^{\varepsilon*}(\psi)(x^\varepsilon) = \frac{1}{\varepsilon} \sum_{c_1} \int_{\Omega_{bl,1c_1}^\#} \psi \left(x^\#, \frac{x_1^\varepsilon}{\varepsilon} - \left(c_1 - \frac{1}{2} \right), \frac{x_2^\varepsilon}{\varepsilon}, \frac{x_3^\varepsilon}{\varepsilon} \right) dx^\# \chi_{\Omega_{bl,1c_1}^\#}(x_1^\varepsilon),$$

for all $\psi \in L^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^1)$, resp. for all $\psi \in L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1}^1)$.

Definition 13:

The operator $B_{bl,1}^\varepsilon$ is defined as:

$$B_{bl,1}^\varepsilon(\psi)(x^\varepsilon) = \psi\left(P(x^\varepsilon), \frac{x_1^\varepsilon}{\varepsilon} - \frac{1}{2}, \frac{x_2^\varepsilon}{\varepsilon}, \frac{x_3^\varepsilon}{\varepsilon}\right)$$

for all $\psi \in C^1(\Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^1})$, where $P(x^\varepsilon) = x_1^\varepsilon$.

Proposition 10:

For all ψ in $C^1(\Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^1})$, $\overline{\Omega_{bl,1}^1}$ - periodic in the direction x_1^1 , then

$$T_{bl,1}^{\varepsilon*}(\psi)(x^\varepsilon) = B_{bl,1}^\varepsilon(\psi)(x^\varepsilon) + O(\varepsilon).$$

Proposition 11:

For each α , for all $\psi \in C^1(\Omega^\#, L^2(\Omega^1)) \cap C^0(\Omega^\#, H^1(\Omega^1))$, Ω^1 - periodic in the direction x_1^1 , then

$$T_{bl,1}^\varepsilon(B^\varepsilon(\psi)) \rightarrow \tilde{\psi} \text{ in } L^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^1) \text{ and in } L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1}^1) \text{ when } \varepsilon \rightarrow 0,$$

where $\tilde{\psi}(x^\#, x^1) = \psi\left((x^\#, 0), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right)$.

Proof. By the definition of $T_{bl,1}^\varepsilon$ and B^ε , it follows that

$$\begin{aligned} T_{bl,1}^\varepsilon(B^\varepsilon(\psi))(x^\#, x^1) &= \sum_{c_1} \chi_{\Omega_{bl,1c_1}^\#}(x^\#) B^\varepsilon(\psi)(x^{\#,c_1} + \varepsilon x_1^1, \varepsilon x_2^1, \varepsilon x_3^1) \\ &= \sum_{c_1} \chi_{\Omega_{bl,1c_1}^\#}(x^\#) \psi\left((x^{\#,c_1} + \varepsilon x_1^1, \varepsilon x_2^1), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right). \end{aligned}$$

Applying the first order Taylor expansion,

$$\psi\left((x^{\#,c} + \varepsilon x_1^1, \varepsilon x_2^1), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right) = \psi\left((x^\#, \varepsilon x_2^1), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right) + O(\varepsilon).$$

Next, passing ε to 0, we have

$$T_{bl,1}^\varepsilon(B^\varepsilon(\psi)) \rightarrow \psi\left((x^\#, 0), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right),$$

in $L^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^1)$ as well as in $L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1}^1)$. \square

2.6.2.3/ DERIVATION OF A LATERAL BOUNDARY MODEL

We now recall the boundary layer terms $\phi_{bl}^\varepsilon = \phi^\varepsilon - B^\varepsilon \phi^0$, $v_{bl}^\varepsilon = V^\varepsilon - B^\varepsilon V^0$ and make the following assumptions where it is understood that by construction $\Omega_{bl,1}^{1,vac}$ depends on α .

Assumption 2:

We assume that:

1. For each α , there exist $\phi_{bl}^{1,\alpha}$ in $L^2(\Omega_{bl,1}^\#, H^1(\Omega_{bl,1}^{1,vac}))$, $\Omega_{bl,1}^{1,vac}$ -periodic in the direction x_1^1 , and v_{bl}^1 in $L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,vac})$ such that $T_{bl,1}^\varepsilon(\phi_{bl}^\varepsilon) \rightharpoonup \phi_{bl}^{1,\alpha}$ weakly in $L^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,vac})$ and $T_{bl,1}^\varepsilon(v_{bl}^\varepsilon) \rightharpoonup v_{bl}^{1,\alpha}$ weakly in $L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,vac})$ when $\varepsilon \rightarrow 0$.
2. There exist ϕ_{bl}^1 in $L^2(\Omega_{bl,1}^\#, H^1(\Omega_{bl,1}^{\infty,vac}))$, $\Omega_{bl,1}^{\infty,vac}$ -periodic in the direction x_1^1 , ϕ_{bl}^1 and its gradient exponentially decreasing to 0 when $x_2^1 \rightarrow +\infty$, and v_{bl}^1 in $L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{\infty,vac})$ such that $\phi_{bl}^{1,\alpha} \chi_{\Omega_{bl,1}^{1,vac}} \rightharpoonup \phi_{bl}^1$ weakly in $L^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^{\infty,vac})$ and $v_{bl}^{1,\alpha} \chi_{\Omega_{bl,1}^{1,vac}} \rightharpoonup v_{bl}^1$ weakly in $L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{\infty,vac})$ when $\alpha \rightarrow +\infty$.

For all ψ defined in $\Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^{1,vac}}$, we define the tilde function $\tilde{\psi}(x^\#, x^1)$ as $\tilde{\psi}(x^\#, x^1) = \psi((x^\#, 0), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2}))$ for all $(x^\#, x^1) \in \Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^{1,vac}}$.

Proposition 12:

For each α , if $\phi^0 \in C^1(\Omega^\#, L^2(\Omega^1)) \cap C^0(\Omega^\#, H^1(\Omega^1))$ and Assumption 2 holds then $T_{bl,1}^\varepsilon \phi^\varepsilon \rightharpoonup \phi_{bl}^{1,\alpha} + \tilde{\phi}^0$ weakly in $L^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,vac})$ and $T_{bl,1}^\varepsilon v^\varepsilon \rightharpoonup v_{bl}^{1,\alpha} + \tilde{V}^0$ weakly in $L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,vac})$.

Proof. The proof is by passing ε to 0 in $T_{bl,1}^\varepsilon \phi^\varepsilon = T_{bl,1}^\varepsilon(B^\varepsilon \phi^0) + T_{bl,1}^\varepsilon(\phi_{bl}^\varepsilon)$, $T_{bl,1}^\varepsilon v^\varepsilon = T_{bl,1}^\varepsilon(B^\varepsilon V^0) + T_{bl,1}^\varepsilon(v_{bl}^\varepsilon)$ and combining with Proposition 11 and Assumption 2. \square

Proposition 13:

The limit ϕ_{bl}^1 is a solution to

$$\begin{cases} -\Delta_{x^1} \phi_{bl}^1 = 0 & \text{in } \Omega_{bl,1}^{\infty,vac} \\ \phi_{bl}^1 = v_{bl}^1 & \text{on } \Gamma_{bl,1,int}^{\infty,vac} \\ \nabla_{x^1} \phi_{bl}^1 \cdot \mathbf{n}^1 = 0 & \text{on } \Gamma_{bl,1,top}^{\infty,vac} \\ \nabla_{x^1} \phi_{bl}^1 \cdot \mathbf{n}^1 & \text{is } \Gamma_{bl,1,per}^{\infty,vac} \text{ - antiperiodic} \\ \nabla_{x^1} \phi_{bl}^1 \cdot \mathbf{n}^1 = -\nabla \tilde{\phi}^0 \cdot \mathbf{n}^1 & \text{on } \Gamma_{bl,1,0}^{\infty,vac} \\ \phi_{bl}^1 & \text{is } \Gamma_{bl,1,per}^{\infty,vac} \text{ - periodic.} \end{cases}$$

Proof. The proof starts by finding the very weak form satisfied by the limit $\phi_{bl}^{1,\alpha}$ and then to pass to the limit on $\alpha \rightarrow \infty$ to find the very weak form satisfied by ϕ_{bl}^1 . The derivation of the corresponding strong form follows. Let us begin with α fixed and replace v^ε in (2.6.3) by a smooth function v_{bl}^ε in $\Omega_{bl,1}^{\alpha\varepsilon,vac}$ vanishing out of $\Omega_{bl,1}^{\alpha\varepsilon,vac}$ and $v_{bl}^\varepsilon = 0$ on $\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}$. This yields

$$\int_{\Omega_{bl,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} v_{bl}^\varepsilon dx^\varepsilon = \int_{\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} v_{bl}^\varepsilon \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon) + \int_{\Gamma_{bl,1,ext}^{\alpha\varepsilon,vac}} \phi^\varepsilon \nabla_{x^\varepsilon} v_{bl}^\varepsilon \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon).$$

Taking a function w in $C^\infty(\Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^{1,vac}})$, $\Omega_{bl,1}^{1,vac}$ - periodic in the direction x_1^1 satisfying $w = 0$ on $\Gamma_{bl,1,int}^{1,vac} \cup \Gamma_{bl,1,\alpha}^{1,vac}$ and $\nabla_{x^1} w \cdot \mathbf{n}^1 = 0$ on $\Gamma_{bl,1,per}^{1,vac} \cup \Gamma_{bl,1,top}^{1,vac} \cup \Gamma_{bl,1,0}^{1,vac} \cup \Gamma_{bl,1,\alpha}^{1,vac}$. It is obvious that $B_{bl,1}^\varepsilon(w) = 0$ on $\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}$, then replacing v_{bl}^ε by $B_{bl,1}^\varepsilon(w)$, we get

$$\int_{\Omega_{bl,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} B_{bl,1}^\varepsilon(w) dx^\varepsilon = \int_{\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} B_{bl,1}^\varepsilon(w) \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon) + \int_{\Gamma_{bl,1,ext}^{\alpha\varepsilon,vac}} \phi^\varepsilon \nabla_{x^\varepsilon} B_{bl,1}^\varepsilon(w) \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon). \quad (2.6.7)$$

A direct computation shows that

$$\begin{aligned} \frac{\partial B_{bl,1}^\varepsilon w}{\partial x_i^\varepsilon} &= B_{bl,1}^\varepsilon \left(\chi_{I^\#}(i) \frac{\partial w}{\partial x^\#} + \frac{1}{\varepsilon} \frac{\partial w}{\partial x_i^1} \right), \\ \frac{\partial}{\partial x_i^\varepsilon} \frac{\partial B_{bl,1}^\varepsilon w}{\partial x_i^\varepsilon} &= B_{bl,1}^\varepsilon \left(\chi_{I^\#}(i) \frac{\partial}{\partial x^\#} \frac{\partial w}{\partial x^\#} + \chi_{I^\#}(i) \frac{2}{\varepsilon} \frac{\partial}{\partial x^\#} \frac{\partial w}{\partial x_1^1} + \frac{1}{\varepsilon^2} \frac{\partial}{\partial x_i^1} \frac{\partial w}{\partial x_i^1} \right), \end{aligned}$$

for all $i \in I = \{1, 2, 3\}$ and with $I^\# = \{1\}$.

After a straightforward calculation, the *l.h.s* of (2.6.7) becomes

$$\begin{aligned} l.h.s &= \int_{\Omega_{bl,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B^\varepsilon \left(\frac{\partial}{\partial x^\#} \frac{\partial w}{\partial x^\#} + \frac{2}{\varepsilon} \frac{\partial}{\partial x^\#} \frac{\partial w}{\partial x_1^1} + \frac{1}{\varepsilon^2} \Delta_{x^1} w \right) dx^\varepsilon \\ &= \frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B^\varepsilon (\Delta_{x^1} w) dx^\varepsilon + O(\varepsilon), \end{aligned} \quad (2.6.8)$$

where

$$O(\varepsilon) = \int_{\Omega_{bl,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B^\varepsilon \left(\frac{\partial}{\partial x^\#} \frac{\partial w}{\partial x^\#} \right) dx^\varepsilon + \frac{2}{\varepsilon} \int_{\Omega_{bl,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B^\varepsilon \left(\frac{\partial}{\partial x^\#} \frac{\partial w}{\partial x_1^1} \right) dx^\varepsilon.$$

The *r.h.s* of (2.6.7) becomes

$$\begin{aligned} r.h.s &= \int_{\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \left[B^\varepsilon \left(\frac{\partial w}{\partial x^\#} \right) n_1^\varepsilon + \frac{1}{\varepsilon} B^\varepsilon (\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon \right] ds(x^\varepsilon) \\ &+ \int_{\Gamma_{bl,1,ext}^{\alpha\varepsilon,vac}} \phi^\varepsilon \left[B^\varepsilon \left(\frac{\partial w}{\partial x^\#} \right) n_i^\varepsilon + \frac{1}{\varepsilon} B^\varepsilon (\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon \right] ds(x^\varepsilon). \end{aligned}$$

Decomposing $\Gamma_{bl,1,ext}^{\alpha\varepsilon,vac}$ into $\Gamma_{bl,1,ext}^{\alpha\varepsilon,vac} = \Gamma_{bl,1,\alpha}^{\alpha\varepsilon,vac} \cup \Gamma_{bl,1,top}^{\alpha\varepsilon,vac} \cup \Gamma_{bl,1,lat}^{\alpha\varepsilon,vac}$ and combining with $\nabla_{x^1} w \cdot \mathbf{n}^1 = 0$

on $\Gamma_{bl,1,per}^{1,vac} \cup \Gamma_{bl,1,top}^{1,vac} \cup \Gamma_{bl,1,0}^{1,vac} \cup \Gamma_{bl,1,\alpha}^{1,vac}$ make it obvious that $B^\varepsilon (\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon = 0$ on $\Gamma_{bl,1,ext}^{\alpha\varepsilon,vac}$, then

$$r.h.s = \frac{1}{\varepsilon} \int_{\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon B^\varepsilon (\nabla_{x^1} w \cdot \mathbf{n}^1) ds(x^\varepsilon) + O(\varepsilon), \quad (2.6.9)$$

where

$$O(\varepsilon) = \int_{\partial\Omega^{\alpha\varepsilon,vac}} \varphi^\varepsilon B^\varepsilon \left(\frac{\partial w}{\partial x^\#} \right) n_1^\varepsilon ds(x^\varepsilon)$$

From (2.6.8) and (2.6.9), we have

$$\frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B_{bl,1}^\varepsilon (\Delta_{x^1} w) dx^\varepsilon = \frac{1}{\varepsilon} \int_{\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon B_{bl,1}^\varepsilon (\nabla_{x^1} w \cdot \mathbf{n}^1) ds(x^\varepsilon) + O(\varepsilon), \quad (2.6.10)$$

Replacing $B_{bl,1}^\varepsilon$ by $T_{bl,1}^{\varepsilon*}$ using Proposition 10, Equality (2.6.10) becomes

$$\frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon T_{bl,1}^{\varepsilon*} (\Delta_{x^1} w) dx^\varepsilon = \frac{1}{\varepsilon} \int_{\Gamma_{bl,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon T_{bl,1}^{\varepsilon*} (\nabla_{x^1} w \cdot \mathbf{n}^1) ds(x^\varepsilon) + O(\varepsilon),$$

By the definition of $T_{bl,1}^{\varepsilon*}$, we have

$$\int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,\alpha,vac}} T_{bl,1}^\varepsilon (\phi^\varepsilon) \Delta_{x^1} w dx^\# dx^1 = \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,\alpha,vac}} T_{bl,1}^\varepsilon (V^\varepsilon) \nabla_{x^1} w \cdot \mathbf{n}^1 dx^\# ds(x^1) + O(\varepsilon).$$

Passing ε to 0 and combining with Proposition 12 gives

$$\int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,\alpha,vac}} (\phi_{bl}^{1,\alpha} + \tilde{\phi}^0) \Delta_{x^1} w dx^\# dx^1 = \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,\alpha,vac}} (v_{bl}^{1,\alpha} + \tilde{V}^0) \nabla_{x^1} w \cdot \mathbf{n}^1 dx^\# ds(x^1), \quad (2.6.11)$$

for each α .

It follows that (2.6.11) still holds if w is taken on the form of $\tau_\alpha v$, where $(\tau_\alpha)_{\alpha \in [\alpha_0, +\infty]}$ is a family of smooth truncation functions with compact support in $\Omega_{bl,1}^\# \times \Omega_{bl,1}^{\infty,vac}$ such that $\tau_\alpha v \rightarrow v$ for all $v \in H^2(\Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^{\infty,vac}})$ and $v \in C^\infty(\Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^{\infty,vac}}) \cap H^2(\Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^{\infty,vac}})$ satisfies v is $\Omega_{bl,1}^{\infty,vac}$ -periodic in the direction x_1^1 , $v = 0$ on $\Gamma_{bl,1,int}^{\infty,vac}$, $\nabla_{x^1} v \cdot \mathbf{n}^1 = 0$ on $\Gamma_{bl,1,per}^{\infty,vac} \cup \Gamma_{bl,1,top}^{\infty,vac} \cup \Gamma_{bl,1,0}^{\infty,vac}$ as well as $|v|$, $|\nabla_{x^1} v|$, and $|\Delta_{x^1} v|$ exponentially decrease to 0 when $x_2^1 \rightarrow +\infty$. Thus,

$$\int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{\infty,vac}} (\phi_{bl}^{1,\alpha} + \tilde{\phi}^0) \chi_{\Omega_{bl,1}^{1,vac}} \Delta_{x^1} (\tau_\alpha v) dx^\# dx^1 = \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{\infty,vac}} (v_{bl}^{1,\alpha} + \tilde{V}^0) \chi_{\Omega_{bl,1}^{1,vac}} \nabla_{x^1} (\tau_\alpha v) \cdot \mathbf{n}^1 dx^\# ds(x^1).$$

Then, passing α to $+\infty$, by Assumption 2, we get

$$\int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{\infty,vac}} (\phi_{bl}^1 + \tilde{\phi}^0) \Delta_{x^1} v dx^\# dx^1 = \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{\infty,vac}} (v_{bl}^1 + \tilde{V}^0) \nabla_{x^1} v \cdot \mathbf{n}^1 dx^\# ds(x^1).$$

To carry out the interpretation of the very weak formulation, we then assume that v is vanishing out of a bounded domain. For simplicity of notation, we choose this domain

equal to $\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,vac}$. Then

$$\int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,vac}} (\phi_{bl}^1 + \widetilde{\phi}^0) \Delta_{x^1} v \, dx^\# dx^1 = \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,vac}} (v_{bl}^1 + \widetilde{V}^0) \nabla_{x^1} v \cdot \mathbf{n}^1 \, dx^\# ds(x^1),$$

for each α .

Applying Green's formula twice with decomposing $\partial\Omega_{bl,1}^{1,vac} = \Gamma_{bl,1,int}^{1,vac} \cup \Gamma_{bl,1,per}^{1,vac} \cup \Gamma_{bl,1,top}^{1,vac} \cup \Gamma_{bl,1,0}^{1,vac} \cup \Gamma_{bl,1,\alpha}^{1,vac}$ and combining with conditions satisfied by v , the results from Proposition 9 $\Delta_{x^1} \widetilde{\phi}^0 = 0$ in $\Omega_{bl,1}^{1,vac}$, $\phi_{bl}^1 = \widetilde{V}^0$ on $\Gamma_{bl,1,int}^{1,vac}$, $\nabla_{x^1} \widetilde{\phi}^0 \cdot \mathbf{n}^1 = 0$ on $\Gamma_{bl,1,top}^{1,vac}$, $\nabla_{x^1} \widetilde{\phi}^0 \cdot \mathbf{n}^1$ is $\Gamma_{bl,1,per}^{1,vac}$ -antiperiodic, the above equality becomes

$$\begin{aligned} & \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,vac}} \Delta_{x^1} \phi_{bl}^1 v \, dx^\# dx^1 + \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,vac}} \phi_{bl}^1 \nabla_{x^1} v \cdot \mathbf{n}^1 \, dx^\# ds(x^1) \\ & - \int_{\Omega_{bl,1}^\# \times (\Gamma_{bl,1,top}^{1,vac} \cup \Gamma_{bl,1,per}^{1,vac})} \nabla_{x^1} \phi_{bl}^1 \cdot \mathbf{n}^1 v \, dx^\# ds(x^1) + \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,0}^{1,vac}} \nabla_{x^1} (\phi_{bl}^1 + \widetilde{\phi}^0) \cdot \mathbf{n}^1 v \, dx^\# ds(x^1) \\ & = \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,vac}} v_{bl}^1 \nabla_{x^1} v \cdot \mathbf{n}^1 \, dx^\# ds(x^1). \end{aligned}$$

If $v = 0$ on $\Gamma_{bl,1,0}^{1,vac} \cup \Gamma_{bl,1,top}^{1,vac} \cup \Gamma_{bl,1,per}^{1,vac}$ and $\nabla_{x^1} v \cdot \mathbf{n}^1 = 0$ on $\Gamma_{bl,1,int}^{1,vac}$, then

$$\int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,vac}} \Delta_{x^1} (\phi_{bl}^1) v \, dx^\# dx^1 = 0.$$

We conclude that,

$$\Delta_{x^1} \phi_{bl}^1 = 0 \quad \text{in } \Omega_{bl,1}^{1,vac}.$$

Next, if $v = 0$ on $\Gamma_{bl,1,0}^{1,vac} \cup \Gamma_{bl,1,top}^{1,vac} \cup \Gamma_{bl,1,per}^{1,vac}$, then

$$\int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,vac}} (\phi_{bl}^1 - v_{bl}^1) \nabla_{x^1} v \cdot \mathbf{n}^1 \, dx^\# ds(x^1) = 0,$$

and we deduce

$$\phi_{bl}^1 = v_{bl}^1 \quad \text{on } \Gamma_{bl,1,int}^{1,vac}.$$

Now, if $v = 0$ on $\Gamma_{bl,1,0}^{1,vac} \cup \Gamma_{bl,1,per}^{1,vac}$, thus

$$\int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,top}^{1,vac}} \nabla_{x^1} \phi_{bl}^1 \cdot \mathbf{n}^1 v \, dx^\# ds(x^1) = 0,$$

and we obtain

$$\nabla_{x^1} \phi_{bl}^1 \cdot \mathbf{n}^1 = 0 \quad \text{on } \Gamma_{bl,1,top}^{1,vac}.$$

If $v = 0$ on $\Gamma_{bl,1,per}^{1,vac}$, hence

$$\int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,0}^{1,vac}} \nabla_{x^1} (\phi_{bl}^1 + \tilde{\phi}^0) \cdot \mathbf{n}^1 v \, dx^\# ds(x^1) = 0,$$

we assert that

$$\nabla_{x^1} \phi_{bl}^1 \cdot \mathbf{n}^1 = -\nabla_{x^1} \tilde{\phi}^0 \cdot \mathbf{n}^1 \text{ on } \Gamma_{bl,1,0}^{1,vac}.$$

Last, we get

$$\nabla_{x^1} \phi_{bl}^1 \cdot \mathbf{n}^1 \text{ is } \Gamma_{bl,1,per}^{1,vac} \text{ - antiperiodic.}$$

Since this equations are true for each α then the above equations hold in the infinite domain and the proof is complete. \square

2.6.3/ EXTERIOR EDGE MODEL

We assume that all lateral boundary models are already derived and identified by the index $i = 1, 2, 3, 4$ of the lateral boundaries, see Figure 2.1. Let us see the contributions of two lateral boundary models corresponding to the indices $i = 1$ and $i = 2$ at the first exterior edge. Obviously, the sum of contributions is not continuous at this edge, and then it leads to propose an edge corrector to overcome this problem. We introduce terms $\phi_{exe}^\varepsilon = \phi^\varepsilon - (B^\varepsilon \phi^0 + B_{bl,1}^\varepsilon \phi_{bl}^1 + B_{bl,2}^\varepsilon \phi_{bl}^2)$ and $v_{exe}^\varepsilon = V^\varepsilon - (B^\varepsilon V^0 + B_{bl,1}^\varepsilon v_{bl}^1 + B_{bl,2}^\varepsilon v_{bl}^2)$, where ϕ^0 is the solution of periodic model of the external zone, ϕ_{bl}^1 and ϕ_{bl}^2 are the solutions of the first and second lateral boundary near the first exterior edge, $B_{bl,1}^\varepsilon$ and $B_{bl,2}^\varepsilon$ are smooth approximate operators of the first and second adjoint boundary layer two-scale transform operator $T_{bl,1}^{\varepsilon*}$ and $T_{bl,2}^{\varepsilon*}$, v_{bl}^1 and v_{bl}^2 are weak limits of $v_{bl}^{1,\alpha}$ and $v_{bl}^{2,\alpha}$ which are the weak limits of $T_{bl,1}^\varepsilon(v_{bl}^\varepsilon)$ in $L^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,int}^{1,vac})$, resp. of $T_{bl,2}^\varepsilon(v_{bl}^\varepsilon)$ in $L^2(\Omega_{bl,2}^\# \times \Gamma_{bl,2,int}^{1,\alpha,vac})$.

2.6.3.1/ GEOMETRY NOTATIONS

Let $\Omega_{exe,1}^{\alpha\varepsilon}$ be a subdomain of Ω^ε assembled by $\Omega_{exe,1}^{\alpha\varepsilon} = \cup_{c \in I_{exe,1}} \Omega_c^\varepsilon$ where $I_{exe,1} := \{c = (c_1, c_2) : c_1, c_2 \in \overline{1, \alpha}\}$ with $\alpha\varepsilon < \min\{L_1^1, L_1^2\}$, see Figure 2.7. The boundary of the vacuum part $\partial\Omega_{exe,1}^{\alpha\varepsilon,vac}$ is a subboundary of $\partial\Omega_1^{\varepsilon,vac}$ denoted as $\partial\Omega_{exe,1}^{\alpha\varepsilon,vac} = \Gamma_{exe,1,int}^{\alpha\varepsilon,vac} \cup \Gamma_{exe,1,ext}^{\alpha\varepsilon,vac}$, with $\Gamma_{exe,1,ext}^{\alpha\varepsilon,vac} = \Gamma_{exe,1,\alpha}^{\alpha\varepsilon,vac} \cup \Gamma_{exe,1,top}^{\alpha\varepsilon,vac} \cup \Gamma_{exe,1,lat}^{\alpha\varepsilon,vac}$, where $\Gamma_{exe,1,int}^{\alpha\varepsilon,vac}$, $\Gamma_{exe,1,top}^{\alpha\varepsilon,vac}$, $\Gamma_{exe,1,lat}^{\alpha\varepsilon,vac}$ are respectively subboundaries of $\Gamma_{1,int}^{\varepsilon,vac}$, $\Gamma_{1,top}^{\varepsilon,vac}$, $\Gamma_{1,lat}^{\varepsilon,vac}$, and $\Gamma_{exe,1,\alpha}^{\alpha\varepsilon,vac}$ is a subboundary of the internal vacuum subboundary between cells of Ω_1^ε . The boundary of the mechanical body $\partial\Omega_{exe,1}^{\alpha\varepsilon,mec}$ is subboundary of $\partial\Omega_1^{\varepsilon,mec}$ defined as $\partial\Omega_{exe,1}^{\alpha\varepsilon,mec} = \Gamma_{exe,1,0}^{\alpha\varepsilon,mec} \cup \Gamma_{exe,1,1}^{\alpha\varepsilon,mec} \cup \Gamma_{exe,1,\alpha}^{\alpha\varepsilon,mec}$, where $\Gamma_{exe,1,0}^{\alpha\varepsilon,mec}$, $\Gamma_{exe,1,1}^{\alpha\varepsilon,mec}$ are subboundaries of $\Gamma_{1,0}^{\varepsilon,mec}$, resp. $\Gamma_{1,1}^{\varepsilon,mec}$, $\Gamma_{exe,1,\alpha}^{\alpha\varepsilon,mec}$ is a subboundary of the internal mechanical subboundary between cells of Ω_1^ε .

We introduce the finite microscopic domain $\Omega_{exe,1}^1$ defined by $\Omega_{exe,1}^1 = \cup_{\xi,\eta=0}^{\alpha-1} (\Omega^1 + (\xi + 1/2, \eta + 1/2, 1/2))$, see Figure 2.8, with the vacuum boundary $\partial\Omega_{exe,1}^{1,vac} = \Gamma_{exe,1,int}^{1,vac} \cup \Gamma_{exe,1,top}^{1,vac} \cup$

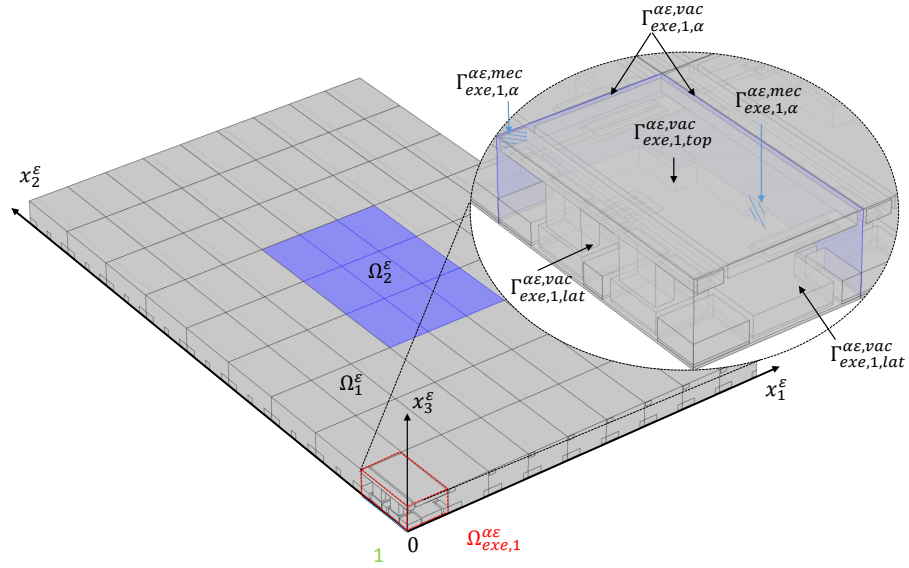


Figure 2.7: The first exterior edge physical domain $\Omega_{exe,1}^{\alpha\epsilon}$ including two subdomains $\Omega_{exe,1}^{\alpha\epsilon, vac}$ and $\Omega_{exe,1}^{\alpha\epsilon, mec}$ with $\alpha = 1$. The zoom illustrates their boundaries.

$$\Gamma_{exe,1,bl1}^{1, vac} \cup \Gamma_{exe,1,bl2}^{1, vac} \cup \Gamma_{exe,1,\alpha}^{1, vac} \text{ and the mechanical boundary } \partial\Omega_{exe,1}^{1, mec} = \Gamma_{exe,1,0}^{1, mec} \cup \Gamma_{exe,1,1}^{1, mec} \cup \Gamma_{exe,1,bl1}^{1, mec} \cup \Gamma_{exe,1,bl2}^{1, mec} \cup \Gamma_{exe,1,\alpha}^{1, mec}$$

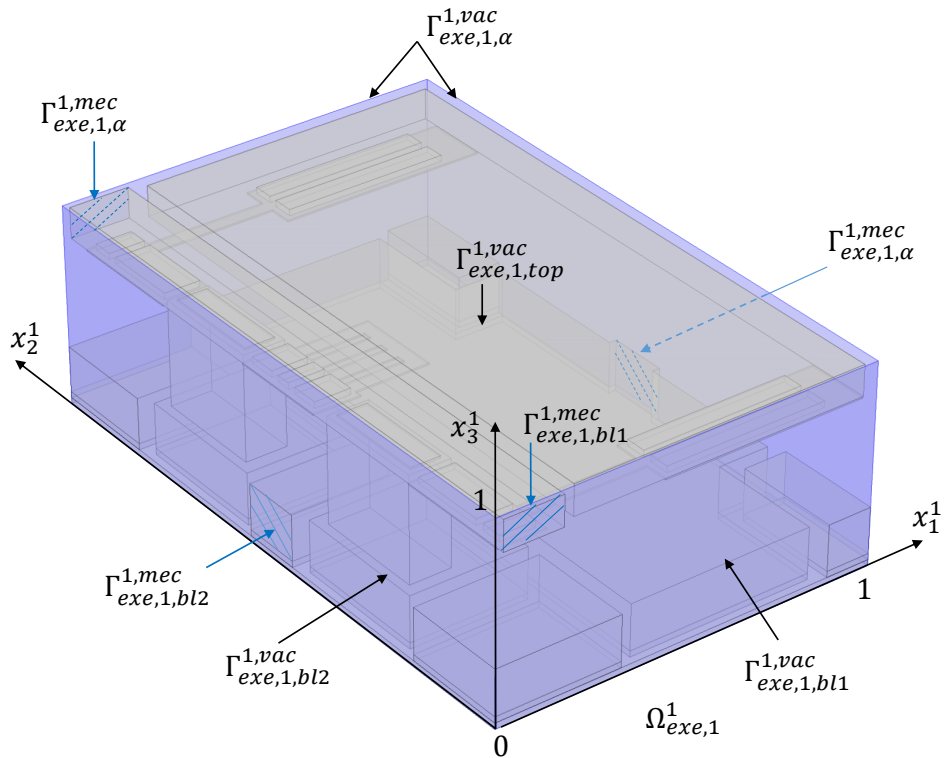


Figure 2.8: The first exterior edge physical domain $\Omega_{exe,1}^{\alpha\epsilon}$ with two subdomains $\Omega_{exe,1}^{\alpha\epsilon, vac}$ and $\Omega_{exe,1}^{\alpha\epsilon, mec}$.

The infinite microscopic domain $\Omega_{exe,1}^\infty$ and its related sets are defined as the limits of $\Omega_{exe,1}^1$ and related when α tends to infinity.

2.6.3.2/ EXTERIOR EDGE BOUNDARY LAYER TWO-SCALE OPERATOR

We now introduce a dilation operator $T_{exe,1}^\varepsilon$ for the first exterior edge model.

Definition 14:

For all φ in $L^2(\Omega_{exe,1}^{\alpha\varepsilon})$ (resp., φ in $L^2(\Gamma_{exe,1}^{\alpha\varepsilon})$), the operator $T_{exe,1}^\varepsilon$ is defined as

$$T_{exe,1}^\varepsilon(\varphi)(x^1) = \varphi(\varepsilon x^1),$$

for a.e. x^1 in $\Omega_{exe,1}^1$ (resp., x^1 in $\Gamma_{exe,1}^1$).

Definition 15:

We define the adjoint operator $T_{exe,1}^{\varepsilon*}$ of $T_{exe,1}^\varepsilon$ by

$$\frac{1}{\varepsilon^3} \int_{\Omega_{exe,1}^{\alpha\varepsilon}} \varphi T_{exe,1}^{\varepsilon*}(\psi) dx^\varepsilon = \int_{\Omega_{exe,1}^1} T_{exe,1}^\varepsilon(\varphi) \psi dx^1,$$

for all $\varphi \in L^2(\Omega_{exe,1}^{\alpha\varepsilon})$, $\psi \in L^2(\Omega_{exe,1}^1)$, respectively

$$\frac{1}{\varepsilon^2} \int_{\Gamma_{exe,1}^{\alpha\varepsilon}} \varphi T_{exe,1}^{\varepsilon*}(\psi) ds(x^\varepsilon) = \int_{\Gamma_{exe,1}^1} T_{exe,1}^\varepsilon(\varphi) \psi ds(x^1),$$

for all $\varphi \in L^2(\Gamma_{exe,1}^{\alpha\varepsilon})$, $\psi \in L^2(\Gamma_{exe,1}^1)$.

From the definitions, we obtain an explicit formula of $T_{exe,1}^{\varepsilon*}$ as

$$T_{exe,1}^{\varepsilon*}(\psi)(x^\varepsilon) = \psi\left(\frac{x^\varepsilon}{\varepsilon}\right),$$

for all $\psi \in L^2(\Omega_{exe,1}^1)$, resp. $\psi \in L^2(\Gamma_{exe,1}^1)$.

In this edge case, the operator $T_{exe,1}^{\varepsilon*}$ and its approximation $B_{exe,1}^\varepsilon$ are identical however both will be used to follow the algorithm of Section 2.5.3.

Proposition 14:

Let B^ε , $B_{bl,1}^\varepsilon$, $B_{bl,2}^\varepsilon$ be smooth approximate operators of the adjoints of T^ε , $T_{bl,1}^\varepsilon$, $T_{bl,2}^\varepsilon$ respectively, we have

1. For all $\psi \in C^1(\Omega^\sharp; L^2(\Omega^1)) \cap C^0(\Omega^\sharp; H^1(\Omega^1))$ and Ω^1 - periodic in the directions x_1^1, x_2^1 ,

$$T_{exe,1}^\varepsilon(B^\varepsilon\psi) \rightarrow \tilde{\psi} \text{ in } L^2(\Omega_{exe,1}^1) \text{ and in } L^2(\Gamma_{exe,1}^1),$$

where $\tilde{\psi}(x^1) = \psi(0, x^1 - 1/2)$.

2. For all $\psi \in C^1(\Omega_{bl,1}^\sharp; L^2(\Omega_{bl,1}^\infty)) \cap C^0(\Omega_{bl,1}^\sharp; H^1(\Omega_{bl,1}^\infty))$ and $\Omega_{bl,1}^\infty$ - periodic in the direction x_1^1 ,

$$T_{exe,1}^\varepsilon(B_{bl,1}^\varepsilon\psi) \rightarrow \tilde{\psi} \text{ in } L^2(\Omega_{exe,1}^1) \text{ and in } L^2(\Gamma_{exe,1}^{1,\infty}),$$

where $\tilde{\psi}(x^1) = \psi(0, (x_1^1 - 1/2, x_2^1, x_3^1))$.

3. For all $\psi \in C^1(\Omega_{bl,2}^\sharp; L^2(\Omega_{bl,2}^\infty)) \cap C^0(\Omega_{bl,2}^\sharp; H^1(\Omega_{bl,2}^\infty))$ and $\Omega_{bl,2}^\infty$ - periodic in the direction x_2^1 ,

$$T_{exe,1}^\varepsilon(B_{bl,2}^\varepsilon\psi) \rightarrow \tilde{\psi} \text{ in } L^2(\Omega_{exe,1}^1) \text{ and in } L^2(\Gamma_{exe,1}^1),$$

where $\tilde{\psi}(x^1) = \psi(0, (x_1^1, x_2^1 - 1/2, x_3^1))$.

2.6.3.3/ DERIVATION OF AN EXTERIOR EDGE MODEL

Let us recall that $\phi_{exe}^\varepsilon = \phi^\varepsilon - (B^\varepsilon\phi^0 + B_{bl,1}^\varepsilon\phi_{bl}^1 + B_{bl,2}^\varepsilon\phi_{bl}^2)$ and $v_{exe}^\varepsilon = V^\varepsilon - (B^\varepsilon V^0 + B_{bl,1}^\varepsilon v_{bl}^1 + B_{bl,2}^\varepsilon v_{bl}^2)$ and make the following assumptions.

Assumption 3:

we assume that

1. For each α , there exist $\phi_{exe}^{1,\alpha}$ in $H^1(\Omega_{exe,1}^{1,vac})$ and $v_{exe}^{1,\alpha}$ in $L^2(\Gamma_{exe,1,int}^{1,vac})$ such that $T_{exe,1}^\varepsilon(\phi_{exe}^\varepsilon) \rightarrow \phi_{exe}^{1,\alpha}$ weakly in $L^2(\Omega_{exe,1}^{1,vac})$ and $T_{exe,1}^\varepsilon(v_{exe}^\varepsilon) \rightarrow v_{exe}^{1,\alpha}$ weakly in $L^2(\Gamma_{exe,1,int}^{1,vac})$ when $\varepsilon \rightarrow 0$.
2. Assume that there exist ϕ_{exe}^1 in $H^1(\Omega_{exe,1}^{\infty,vac})$, ϕ_{exe}^1 and its gradient converge exponentially fast to zero when $x_1^1 + x_2^1 \rightarrow \infty$, and v_{exe}^1 in $L^2(\Gamma_{exe,1,int}^{\infty,vac})$ such that $\phi_{exe}^{1,\alpha}\chi_{\Omega_{exe,1}^{1,vac}} \rightarrow \phi_{exe}^1$ weakly in $L^2(\Omega_{exe,1}^{\infty,vac})$ and $v_{exe}^{1,\alpha}\chi_{\Omega_{exe,1}^{1,vac}} \rightarrow v_{exe}^1$ weakly in $L^2(\Gamma_{exe,1,int}^{\infty,vac})$ when $\alpha \rightarrow +\infty$.

The following proposition results from using Proposition 14.

Proposition 15:

For each α , if $\phi^0 \in C^1(\Omega^\#; L^2(\Omega^1)) \cap C^0(\Omega^\#; H^1(\Omega^1))$, $\phi_{bl}^1 \in C^1(\Omega_{bl,1}^\#; L^2(\Omega_{bl,1}^\infty)) \cap C^0(\Omega_{bl,1}^\#; H^1(\Omega_{bl,1}^\infty))$, $\phi_{bl}^2 \in C^1(\Omega_{bl,2}^\#; L^2(\Omega_{bl,2}^\infty)) \cap C^0(\Omega_{bl,2}^\#; H^1(\Omega_{bl,2}^\infty))$ and Assumption 3 holds then $T_{exe,1}^\varepsilon(\phi^\varepsilon) \rightharpoonup \phi_{exe}^{1,\alpha} + \widetilde{\phi}^0 + \widetilde{\phi}_{bl}^1 + \widetilde{\phi}_{bl}^2$ weakly in $L^2(\Omega_{exe,1}^{1,vac})$ and $T_{exe,1}^\varepsilon(V^\varepsilon) \rightharpoonup v_{exe}^{1,\alpha} + \widetilde{V}^0 + \widetilde{v}_{bl}^1 + \widetilde{v}_{bl}^2$ weakly in $L^2(\Gamma_{exe,1,int}^{1,vac})$ when $\varepsilon \rightarrow 0$, where $\widetilde{\varphi}^0(x^1) = \varphi^0(0, x^1 - 1/2)$, $\widetilde{\varphi}_{bl}^1(x^1) = \varphi_{bl}^1(0, (x_1^1 - 1/2, x_2^1, x_3^1))$, $\widetilde{\varphi}_{bl}^2(x^1) = \varphi_{bl}^2(0, (x_1^1, x_2^1 - 1/2, x_3^1))$.

Proposition 16:

The limit ϕ_{exe}^1 satisfies

$$\left\{ \begin{array}{ll} \Delta_{x^1} \phi_{exe}^1 = 0 & \text{in } \Omega_{exe,1}^{\infty,vac} \\ \phi_{exe}^1 = v_{exe}^1 & \text{on } \Gamma_{exe,1,int}^{\infty,vac} \\ \nabla_{x^1} \phi_{exe}^1 \cdot \mathbf{n}^1 = 0 & \text{on } \Gamma_{exe,1,top}^{\infty,vac} \\ \nabla_{x^1} \phi_{exe}^1 \cdot \mathbf{n}^1 = -\nabla_{x^1} \widetilde{\phi}_{bl}^2 \cdot \mathbf{n}^1 & \text{on } \Gamma_{exe,1,bl1}^{\infty,vac} \\ \nabla_{x^1} \phi_{exe}^1 \cdot \mathbf{n}^1 = -\nabla_{x^1} \widetilde{\phi}_{bl}^1 \cdot \mathbf{n}^1 & \text{on } \Gamma_{exe,1,bl2}^{\infty,vac} \end{array} \right.$$

Proof. The outline of the proof runs as the previous ones. Firstly, we take a fixed α and replace v^ε by a smooth function v_{exe}^ε in (2.6.3), where v_{exe}^ε is defined in $\Omega_{exe,1}^{\alpha\varepsilon,vac}$, $v_{exe}^\varepsilon = 0$ on $\Gamma_{exe,1,int}^{\alpha\varepsilon,vac}$ and vanishes out of $\Omega_{exe,1}^{\alpha\varepsilon}$, then

$$\int_{\Omega_{exe,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} v_{exe}^\varepsilon \, dx^\varepsilon = \int_{\Gamma_{exe,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} v_{exe}^\varepsilon \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon) + \int_{\Gamma_{exe,1,ext}^{\alpha\varepsilon,vac}} \varphi^\varepsilon \nabla_{x^\varepsilon} v_{exe}^\varepsilon \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon).$$

After that, we substitute v_{exe}^ε by $\varepsilon^{-1} B_{exe,1}^\varepsilon(w)$ where w is in $C^\infty(\overline{\Omega_{exe,1}^{1,vac}})$, $w = 0$ on $\Gamma_{exe,1,int}^{1,vac} \cup \Gamma_{exe,1,\alpha}^{1,vac}$ and $\nabla_{x^1} w \cdot \mathbf{n}^1 = 0$ on $\Gamma_{exe,1,top}^{1,vac} \cup \Gamma_{exe,1,\alpha}^{1,vac} \cup \Gamma_{exe,1,bl1}^{1,vac} \cup \Gamma_{exe,1,bl2}^{1,vac}$. Hence,

$$\frac{1}{\varepsilon} \int_{\Omega_{exe,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} B_{exe,1}^\varepsilon(w) \, dx^\varepsilon = \frac{1}{\varepsilon} \int_{\Gamma_{exe,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} B_{exe,1}^\varepsilon(w) \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon) + \frac{1}{\varepsilon} \int_{\Gamma_{exe,1,ext}^{\alpha\varepsilon,vac}} \varphi^\varepsilon \nabla_{x^\varepsilon} B_{exe,1}^\varepsilon(w) \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon).$$

We check at once that,

$$\frac{\partial B_{exe,1}^\varepsilon w}{\partial x_i^\varepsilon} = \frac{1}{\varepsilon} B_{exe,1}^\varepsilon \left(\frac{\partial w}{\partial x_i^1} \right) \text{ and } \frac{\partial}{\partial x_i^\varepsilon} \frac{\partial B_{exe,1}^\varepsilon w}{\partial x_i^\varepsilon} = \frac{1}{\varepsilon^2} B_{exe,1}^\varepsilon \left(\frac{\partial}{\partial x_i^1} \frac{\partial w}{\partial x_i^1} \right),$$

for all $i = 1, 2, 3$, and it obvious that $B_{exe,1}^\varepsilon(\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon = 0$ on $\Gamma_{exe,1,ext}^{\alpha\varepsilon,vac}$, then

$$\frac{1}{\varepsilon^3} \int_{\Omega_{exe,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B_{exe,1}^\varepsilon(\Delta_{x^1} w) \, dx^\varepsilon = \frac{1}{\varepsilon^2} \int_{\Gamma_{exe,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon B_{exe,1}^\varepsilon(\nabla_{x^1} w \cdot \mathbf{n}^1) \, ds(x^\varepsilon).$$

Replacing $B_{exe,1}^\varepsilon$ by $T_{exe,1}^{\varepsilon*}$ and combining with the definition of $T_{exe,1}^{\varepsilon*}$, we have

$$\int_{\Omega_{exe,1}^{1,vac}} T_{exe,1}^\varepsilon(\phi^\varepsilon) \Delta_{x^1} w \, dx^1 = \int_{\Gamma_{exe,1,int}^{1,vac}} T_{exe,1}^\varepsilon(V^\varepsilon) \nabla_{x^1} w \cdot \mathbf{n}^1 \, ds(x^1).$$

Passing ε to 0, by Proposition 15 gives

$$\int_{\Omega_{exe,1}^{1,vac}} \left(\phi_{exe}^{1,\alpha} + \tilde{\phi}^0 + \tilde{\phi}_{bl}^1 + \tilde{\phi}_{bl}^2 \right) \Delta_{x^1} w \, dx^1 = \int_{\Gamma_{exe,1,int}^{1,vac}} \left(v_{exe}^{1,\alpha} + \tilde{V}^0 + \tilde{v}_{bl}^1 + \tilde{v}_{bl}^2 \right) \nabla_{x^1} w \cdot \mathbf{n}^1 \, ds(x^1).$$

We now replace w by $\tau_\alpha v$, where τ_α is a smooth truncation function with compact support in $\Omega_{exe,1}^{1,vac}$ and $v \in C^\infty(\overline{\Omega_{exe,1}^{\infty,vac}}) \cap H^2(\overline{\Omega_{exe,1}^{\infty,vac}})$ satisfying $v = 0$ on $\Gamma_{exe,1,int}^{\infty,vac}$, $\nabla_{x^1} v \cdot \mathbf{n}^1 = 0$ on $\Gamma_{exe,1,top}^{1,vac} \cup \Gamma_{exe,1,bl1}^{1,vac} \cup \Gamma_{exe,1,bl2}^{1,vac}$, $|v|$, $|\nabla_{x^1} v|$ and $|\Delta_{x^1} v|$ converge exponentially fast to zero when $x_1^1 + x_2^1 \rightarrow \infty$, $\tau_\alpha v \rightarrow v$ in $H^2(\overline{\Omega_{exe,1}^{\infty,vac}})$ when $\alpha \rightarrow \infty$. We obtain

$$\int_{\Omega_{exe,1}^{\infty,vac}} \left(\phi_{exe}^{1,\alpha} + \tilde{\phi}^0 + \tilde{\phi}_{bl}^1 + \tilde{\phi}_{bl}^2 \right) \chi_{\Omega_{exe,1}^{1,vac}} \Delta_{x^1} (\tau_\alpha v) \, dx^1 = \int_{\Gamma_{exe,1,int}^{\infty,vac}} \left(v_{exe}^{1,\alpha} + \tilde{V}^0 + \tilde{v}_{bl}^1 + \tilde{v}_{bl}^2 \right) \chi_{\Omega_{exe,1}^{1,vac}} \nabla_{x^1} \tau_\alpha v \cdot \mathbf{n}^1 \, ds(x^1).$$

Passing α to $+\infty$, by Assumption 3, we get

$$\int_{\Omega_{exe,1}^{\infty,vac}} \left(\phi_{exe}^1 + \tilde{\phi}^0 + \tilde{\phi}_{bl}^1 + \tilde{\phi}_{bl}^2 \right) \Delta_{x^1} v \, dx^1 = \int_{\Gamma_{exe,1,int}^{\infty,vac}} \left(v_{exe}^1 + \tilde{V}^0 + \tilde{v}_{bl}^1 + \tilde{v}_{bl}^2 \right) \nabla_{x^1} v \cdot \mathbf{n}^1 \, ds(x^1).$$

Now, we choose v vanishing out of $\Omega_{exe,1}^{1,vac}$ for a given α ,

$$\int_{\Omega_{exe,1}^{1,vac}} \left(\phi_{exe}^1 + \tilde{\phi}^0 + \tilde{\phi}_{bl}^1 + \tilde{\phi}_{bl}^2 \right) \Delta_{x^1} v \, dx^1 = \int_{\Gamma_{exe,1,int}^{1,vac}} \left(v_{exe}^1 + \tilde{V}^0 + \tilde{v}_{bl}^1 + \tilde{v}_{bl}^2 \right) \nabla_{x^1} v \cdot \mathbf{n}^1 \, ds(x^1),$$

Applying Green's formula twice and decomposing $\partial \Omega_{exe,1}^{1,vac} = \Gamma_{exe,1,int}^{1,vac} \cup \Gamma_{exe,1,top}^{1,vac} \cup \Gamma_{exe,1,bl1}^{1,vac} \cup \Gamma_{exe,1,bl2}^{1,vac} \cup \Gamma_{exe,1,\alpha}^{1,vac}$, combining with conditions satisfied by v , the results from Proposition 9 and Proposition 13 $\Delta_{x^1} \tilde{\phi}^0 = \Delta_{x^1} \tilde{\phi}_{bl}^1 = \Delta_{x^1} \tilde{\phi}_{bl}^2 = 0$ in $\Omega_{exe,1}^{1,vac}$, $\tilde{\phi}^0 = \tilde{V}^0$, $\tilde{\phi}_{bl}^1 = \tilde{v}_{bl}^1$, $\tilde{\phi}_{bl}^2 = \tilde{v}_{bl}^2$ on $\Gamma_{exe,1,int}^{1,vac}$, $\nabla_{x^1} \tilde{\phi}^0 \cdot \mathbf{n}^1 = \nabla_{x^1} \tilde{\phi}_{bl}^1 \cdot \mathbf{n}^1 = \nabla_{x^1} \tilde{\phi}_{bl}^2 \cdot \mathbf{n}^1 = 0$ on $\Gamma_{exe,1,top}^{1,vac}$, $\nabla_{x^1} (\tilde{\phi}^0 + \tilde{\phi}_{bl}^1) \cdot \mathbf{n}^1 = 0$ on $\Gamma_{exe,1,bl1}^{1,vac}$, $\nabla_{x^1} (\tilde{\phi}^0 + \tilde{\phi}_{bl}^2) \cdot \mathbf{n}^1 = 0$ on $\Gamma_{exe,1,bl2}^{1,vac}$, we deduce that

$$\begin{aligned} & \int_{\Omega_{exe,1}^{1,vac}} \Delta_{x^1} (\phi_{exe}^1) v \, dx^1 - \int_{\Gamma_{exe,1,top}^{1,vac}} \nabla_{x^1} \phi_{exe}^1 \cdot \mathbf{n}^1 v \, ds(x^1) \\ & - \int_{\Gamma_{exe,1,bl1}^{1,vac}} \nabla_{x^1} (\phi_{exe}^1 + \tilde{\phi}_{bl}^2) \cdot \mathbf{n}^1 v \, ds(x^1) - \int_{\Gamma_{exe,1,bl2}^{1,vac}} \nabla_{x^1} (\phi_{exe}^1 + \tilde{\phi}_{bl}^1) \cdot \mathbf{n}^1 v \, ds(x^1) \\ & + \int_{\Gamma_{exe,1,int}^{1,vac}} \phi_{exe}^1 \nabla_{x^1} v \cdot \mathbf{n}^1 \, ds(x^1) = \int_{\Gamma_{exe,1,int}^{1,vac}} v_{exe}^1 \nabla_{x^1} v \cdot \mathbf{n}^1 \, ds(x^1). \end{aligned}$$

The rest of the proof runs as the previous proofs. \square

2.6.4/ INTERFACE MODEL

Let us consider the solution ϕ^0 in Proposition 9 again. It is easily seen that ϕ^0 is periodic with two distinct periodicity cells as different voltages are imposed in the two zones, then the electric potential approximated by ϕ^0 is discontinuous at the interface between two zones. It leads to propose an interface corrector to deal with this problem. We introduce the terms $\phi_{bl}^\varepsilon = \phi^\varepsilon - B^\varepsilon(\phi^0)$ and $v_{bl}^\varepsilon = V^\varepsilon - B^\varepsilon(V^0)$.

2.6.4.1/ GEOMETRY NOTATIONS

Let $\Omega_{in,1}^{\alpha\varepsilon}$ be a subdomain of Ω^ε defined as $\Omega_{in,1}^{\alpha\varepsilon} = \cup_{c \in I_{in,1}} \Omega_c^\varepsilon$, where $I_{in,1} := \{c = (c_1, c_2) : c_1 = \overline{i_1, j_1} \text{ and } c_2 \in \overline{i_2 - \alpha, i_2 + \alpha}, 2 \leq i, j \leq n_1\}$ and $\alpha \in \mathbb{Z}^+$, see Figure 2.9, and $\Omega_{in,1}^{\alpha\varepsilon}$ is assembled by two nonoverlapping subdomains $\Omega_{in,1}^{\alpha\varepsilon+}$ and $\Omega_{in,1}^{\alpha\varepsilon-}$, we write them $\Omega_{in,1}^{\alpha\varepsilon\pm}$ for short, which are respectively a subdomain of Ω_2^ε and Ω_1^ε . The interface between $\Omega_{in,1}^{\alpha\varepsilon+}$ and $\Omega_{in,1}^{\alpha\varepsilon-}$ is denoted as $\Gamma_{in,1,interf}^{\alpha\varepsilon}$ which is a subboundary of the $\Gamma_{interf}^\varepsilon$.

The boundary of the vacuum part $\Omega_{in,1}^{\alpha\varepsilon,vac}$ is built by a union $\partial\Omega_{in,1}^{\alpha\varepsilon,vac+} \cup \partial\Omega_{in,1}^{\alpha\varepsilon,vac-} \cup \Gamma_{in,1,interf}^{\alpha\varepsilon,vac}$, where $\partial\Omega_{in,1}^{\alpha\varepsilon,vac\pm}$ is a boundary of $\Omega_{in,1}^{\alpha\varepsilon,vac\pm}$ and $\Gamma_{in,1,interf}^{\alpha\varepsilon,vac}$ is the interface between them which is a subboundary of $\Gamma_{interf}^{\varepsilon,vac}$. More precisely, $\partial\Omega_{in,1}^{\alpha\varepsilon,vac+}$ and $\partial\Omega_{in,1}^{\alpha\varepsilon,vac-}$ are subboundaries of the internal vacuum part $\partial\Omega_2^{\varepsilon,vac}$ and the external vacuum part $\partial\Omega_1^{\varepsilon,vac}$ assembled by the unions $\Gamma_{in,1,int}^{\alpha\varepsilon,vac\pm} \cup \Gamma_{in,1,ext}^{\alpha\varepsilon,vac\pm} \cup \Gamma_{in,1,interf}^{\alpha\varepsilon,vac}$, with $\Gamma_{in,1,ext}^{\alpha\varepsilon,vac\pm} = \Gamma_{in,1,top}^{\alpha\varepsilon,vac\pm} \cup \Gamma_{in,1,lat}^{\alpha\varepsilon,vac\pm} \cup \Gamma_{in,1,\alpha}^{\alpha\varepsilon,vac\pm}$, where $\Gamma_{in,1,int}^{\alpha\varepsilon,vac+}$ and $\Gamma_{in,1,top}^{\alpha\varepsilon,vac+}$ is a subboundary of $\Gamma_{2,int}^{\varepsilon,vac}$ and $\Gamma_{2,top}^{\varepsilon,vac}$, $\Gamma_{in,1,lat}^{\alpha\varepsilon,vac+}$ and $\Gamma_{in,1,\alpha}^{\alpha\varepsilon,vac+}$ are subboundaries of $\Gamma_{interf}^{\varepsilon,vac}$ and the internal vacuum subboundary between cells of the internal zone Ω_2 , $\Gamma_{in,1,int}^{\alpha\varepsilon,vac-}$ and $\Gamma_{in,1,top}^{\alpha\varepsilon,vac-}$ are subboundaries of $\Gamma_{1,int}^{\varepsilon,vac}$ and $\Gamma_{1,top}^{\varepsilon,vac}$, $\Gamma_{in,1,lat}^{\alpha\varepsilon,vac-}$ and $\Gamma_{in,1,\alpha}^{\alpha\varepsilon,vac-}$ are subboundaries of the internal vacuum subboundary between cells of the external zone Ω_1^ε .

The boundary of the mechanical part $\Omega_{in,1}^{\alpha\varepsilon,mec}$ is determined by the union $\partial\Omega_{in,1}^{\alpha\varepsilon,mec+} \cup \partial\Omega_{in,1}^{\alpha\varepsilon,mec-} \cup \Gamma_{in,1,interf}^{\alpha\varepsilon,mec}$, where $\partial\Omega_{in,1}^{\alpha\varepsilon,mec\pm}$ are the boundaries of $\Omega_{in,1}^{\alpha\varepsilon,mec\pm}$ and $\Gamma_{in,1,interf}^{\alpha\varepsilon,mec}$ is the interface between them which is a subboundary of $\Gamma_{interf}^{\varepsilon,mec}$. Precisely, $\partial\Omega_{in,1}^{\alpha\varepsilon,mec+}$ and $\partial\Omega_{in,1}^{\alpha\varepsilon,mec-}$ are subboundaries of the internal mechanical part $\partial\Omega_2^{\varepsilon,mec}$ and the external mechanical part $\partial\Omega_1^{\varepsilon,mec}$ defined as the union $\Gamma_{in,1,1}^{\alpha\varepsilon,mec\pm} \cup \Gamma_{in,1,0}^{\alpha\varepsilon,mec\pm} \cup \Gamma_{in,1,interf}^{\alpha\varepsilon,mec}$, where $\Gamma_{in,1,1}^{\alpha\varepsilon,mec+}$ and $\Gamma_{in,1,0}^{\alpha\varepsilon,mec+}$ are subboundaries of $\Gamma_{2,1}^{\varepsilon,mec}$ and $\Gamma_{2,0}^{\varepsilon,mec}$, $\Gamma_{in,1,lat}^{\alpha\varepsilon,mec+}$ is a subboundary of $\Gamma_{interf}^{\varepsilon,mec}$ and the internal mechanical subboundary between cells of the internal zone Ω_2^ε , $\Gamma_{in,1,1}^{\alpha\varepsilon,mec-}$ and $\Gamma_{in,1,0}^{\alpha\varepsilon,mec-}$ is a subboundary of $\Gamma_{1,1}^{\varepsilon,mec}$ and $\Gamma_{1,0}^{\varepsilon,mec}$, $\Gamma_{in,1,lat}^{\alpha\varepsilon,mec-}$ is a subboundary of the internal mechanical subboundary between cells of the external zone Ω_1^ε .

We next denote $\Omega_{in,1}^\# = [L_1^1, L_2^1]$ with the partition $\{\Omega_{in,1,c_1}^\#\}_{c_1=\overline{i_1, j_1-1}}$, $\Omega_{in,1,c_1}^\# = [c_1\varepsilon, (c_1+1)\varepsilon]$, with i, j such that $L_1^1 = i_1\varepsilon$, $L_2^1 = j_1\varepsilon$, and denote $x^{\#,c_1}$ as a center of $\Omega_{in,1,c_1}^\#$, $x^{\#,c_1} = c_1\varepsilon + \varepsilon/2$.

We now represent the finite microscopic domain $\Omega_{in,1}^1$, see Figure 2.10, constructed by the union of two nonoverlapping subdomains $\Omega_{in,1}^{1+}$ and $\Omega_{in,1}^{1-}$, we write them $\Omega_{in,1}^{1\pm}$ for brevity, with the interface $\Gamma_{in,1,interf}^1$, where $\Omega_{in,1}^{1\pm}$ is determined by $\Omega_{in,1}^{1\pm} = \cup_{\eta=\overline{1,\alpha}} (\Omega^1 + (0, \pm(\eta-$

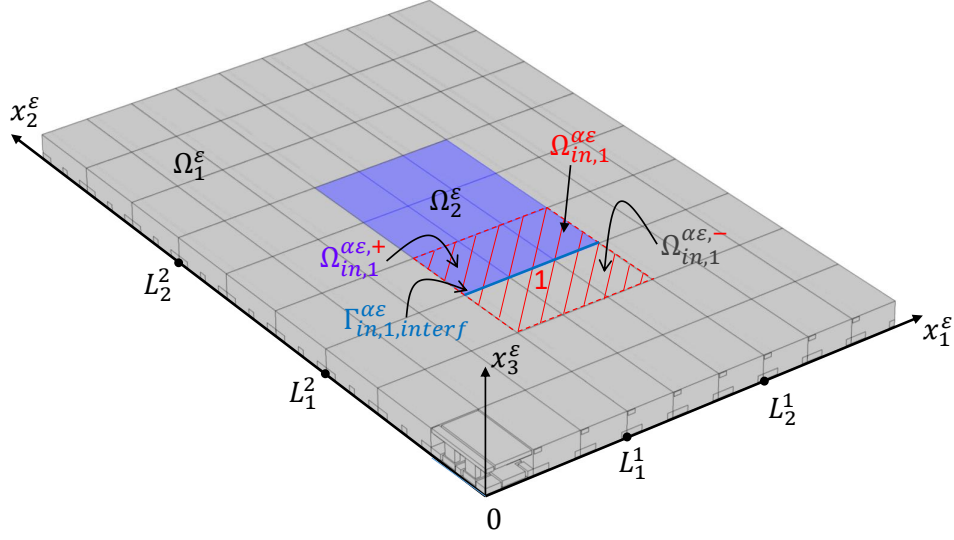


Figure 2.9: The first interface physical domain $\Omega_{in,1}^{\alpha\epsilon}$ with two nonoverlapping subdomains $\Omega_{in,1}^{\alpha\epsilon+}$ and $\Omega_{in,1}^{\alpha\epsilon-}$, each domain $\Omega_{in,1}^{\alpha\epsilon\pm}$ is assembled by two parts the vacuum part $\Omega_{in,1}^{\alpha\epsilon,vac\pm}$ and the mechanical part $\Omega_{in,1}^{\alpha\epsilon,mec\pm}$, with $\alpha = 1$.

$1/2), 1/2)$). The notation system built for the physical domain is transposed to the microscopic domain that we do not detail.

For all regular function v defined in $\Omega_{in,1}^1$, we denote v^+ and v^- as respectively the restriction of v in $\Omega_{in,1}^{1+}$ and $\Omega_{in,1}^{1-}$ and $[[v]] = v^+ - v^-$ as the jump of v at the interface $\Gamma_{in,1}^{1,interf}$.

The infinite microscopic domain $\Omega_{in,1}^\infty$ and its boundaries are defined as the limit of $\Omega_{in,1}^1$ and of its boundaries.

2.6.4.2/ INTERFACE BOUNDARY LAYER TWO-SCALE TRANSFORM OPERATOR

Let us introduce the interface boundary layer two-scale transform $T_{in,1}^\epsilon$.

Definition 16:

For all φ in $L^2(\Omega_{in,1}^{\alpha\epsilon})$ (resp. φ in $L^2(\Gamma_{in,1}^{\alpha\epsilon})$), $T_{in,1}^\epsilon$ is defined by

$$T_{in,1}^\epsilon(\varphi)(x^\#, x^1) = \sum_{c_1} \chi_{\Omega_{in,1}^{c_1}}(x^\#) \varphi(x^\#, c_1 + \epsilon x_1^1, L_1^2 + \epsilon x_2^1, \epsilon x_3^1),$$

for a.e. $x^\# \in \Omega_{in,1}^\#, x^1 \in \Omega_{in,1}^1$ (resp. $x^1 \in \Gamma_{in,1}^1$), $L_1^2 = i_2 \epsilon, i_2 \in \mathbb{Z}^+$.

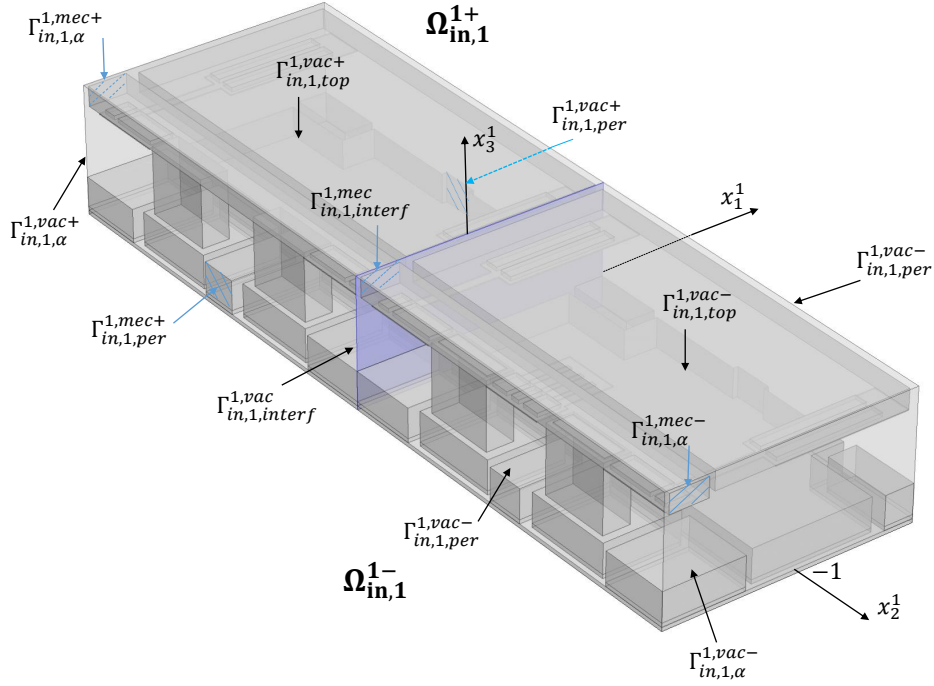


Figure 2.10: The first interface microscopic domain $\Omega_{in,1}^1$ with two nonoverlapping sub-domains $\Omega_{in,1}^{1,\pm}$, each of them also involves two parts, the vacuum part $\Omega_{in,1}^{1,vac\pm}$ and the mechanical part $\Omega_{in,1}^{1,mec\pm}$, in the case of $\alpha = 1$.

Definition 17:

We define the adjoint operator $T_{in,1}^{\varepsilon*}$ of $T_{in,1}^{\varepsilon}$ by

$$\frac{1}{\varepsilon^2} \int_{\Omega_{in,1}^{\alpha\varepsilon}} \varphi T_{in,1}^{\varepsilon*}(\psi) dx^\varepsilon = \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^1} T_{in,1}^{\varepsilon}(\varphi) \psi dx^\# dx^1,$$

for all $\psi \in L^2(\Omega_{in,1}^\# \times \Omega_{in,1}^1)$, $\varphi \in L^2(\Omega_{in,1}^{\alpha\varepsilon})$, and

$$\frac{1}{\varepsilon} \int_{\Gamma_{in,1}^{\alpha\varepsilon}} \varphi T_{in,1}^{\varepsilon*}(\psi) ds(x^\varepsilon) = \int_{\Omega_{in,1}^\# \times \Gamma_{in,1}^1} T_{in,1}^{\varepsilon}(\varphi) \psi dx^\# ds(x^1),$$

for all $\psi \in L^2(\Omega_{in,1}^\# \times \Gamma_{in,1}^1)$, $\varphi \in L^2(\Gamma_{in,1}^{\alpha\varepsilon})$.

From the definitions, we obtain an explicit formula of $T_{in,1}^{\varepsilon*}$ as

$$T_{in,1}^{\varepsilon*}(\psi)(x^\varepsilon) = \frac{1}{\varepsilon} \sum_{c_1} \int_{\Omega_{in,1,c_1}^\#} \psi \left(x^\#, \frac{x_1^\varepsilon - x_1^\# c_1}{\varepsilon}, \frac{x_2^\varepsilon - L_1^2}{\varepsilon}, \frac{x_3^\varepsilon}{\varepsilon} \right) dx^\# \chi_{\Omega_{in,1,c_1}^\#} (x_1^\varepsilon),$$

valid for all $\psi \in L^2(\Omega_{in,1}^\# \times \Omega_{in,1}^1)$ and $\psi \in L^2(\Omega_{in,1}^\# \times \Gamma_{in,1}^1)$.

Definition 18:

For every $\psi \in C^1(\Omega_{in,1}^\# \times \overline{\Omega_{in,1}^1})$, the operator $B_{in,1}^\varepsilon$ is defined by

$$B_{in,1}^\varepsilon(\psi)(x^\varepsilon) = \psi\left(P(x^\varepsilon), \frac{x_1^\varepsilon}{\varepsilon} - \frac{1}{2}, \frac{x_2^\varepsilon}{\varepsilon}, \frac{x_3^\varepsilon}{\varepsilon}\right),$$

where $P(x^\varepsilon) = x_1^\varepsilon$.

Proposition 17:

For every ψ in $C^1(\Omega_{in,1}^\# \times \overline{\Omega_{in,1}^1})$ and $\overline{\Omega_{in,1}^1}$ -periodic in the directions x_1^1 and x_2^1 , then

$$T_{in,1}^{\varepsilon*}(\psi)(x^\varepsilon) = B_{in,1}^\varepsilon(\psi)(x^\varepsilon) + O(\varepsilon).$$

Proposition 18:

For every $\psi \in C^1(\Omega^\#; L^2(\Omega^1)) \cap C^0(\Omega^\#; H^1(\Omega^1))$ and Ω^1 -periodic in the directions x_1^1, x_2^1 ,

$$T_{in,1}^\varepsilon(B^\varepsilon(\psi)) \rightarrow \tilde{\psi} \text{ in } L^2(\Omega_{in,1}^\# \times \Omega_{in,1}^1) \text{ and in } L^2(\Omega_{in,1}^\# \times \Gamma_{in,1}^1),$$

where $\tilde{\psi}(x^\#, x^1) = \psi\left(x^\#, L_1^2, (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right)$.

Proof. By the definitions of $T_{in,1}^\varepsilon$ and B^ε , we obtain

$$\begin{aligned} T_{in,1}^\varepsilon(B^\varepsilon(\psi))(x^\#, x^1) &= \sum_{c_1} \chi_{\Omega_{in,1,c_1}^\#}(x^\#) B^\varepsilon(\psi)(x^{\#,c_1} + \varepsilon x_1^1, L_1^2 + \varepsilon x_2^1, \varepsilon x_3^1) \\ &= \sum_{c_1} \chi_{\Omega_{in,1,c_1}^\#}(x^\#) \psi\left(x^{\#,c_1} + \varepsilon x_1^1, L_1^2 + \varepsilon x_2^1, (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right). \end{aligned}$$

By the first order Taylor expansion,

$$\psi\left(x^{\#,c_1} + \varepsilon x_1^1, L_1^2 + \varepsilon x_2^1, (x_1^1 + c_1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right) = \psi\left(x^\#, L_1^2 + \varepsilon x_2^1, (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})\right) + O(\varepsilon),$$

in $L^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{1,vac})$, $x^\#$ in $\Omega_{in,1,c_1}^\#$. Passing ε to 0, then

$$T_{in,1}^\varepsilon(B^\varepsilon(\psi)) \rightarrow \psi\left(x^\#, L_1^2, ((x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2}))\right) \text{ in } L^2(\Omega_{in,1}^\# \times \Omega_{in,1}^1) \text{ and in } L^2(\Omega_{in,1}^\# \times \Gamma_{in,1}^1).$$

□

2.6.4.3/ DERIVATION OF AN INTERFACE MODEL

Let us now recall the boundary layer terms $\phi_{bl}^\varepsilon = \phi^\varepsilon - B^\varepsilon(\phi^0)$, $v_{bl}^\varepsilon = V^\varepsilon - B^\varepsilon(V^0)$ and make the following assumptions.

Assumption 4:

We assume that

1. For each α , there exist $\phi_{in}^{1,\alpha} \in L^2(\Omega_{in,1}^\#, H^1(\Omega_{in,1}^{1,vac}))$, $\Omega_{in,1}^{1,vac}$ - periodic in the direction x_1^1 and $v_{in}^1 \in L^2(\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{1,vac})$ such that $T_{in,1}^\varepsilon(\phi_{bl}^\varepsilon) \rightharpoonup \phi_{in,1}^{1,\alpha}$ weakly in $L^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{1,vac})$ and $T_{in,1}^\varepsilon(v_{bl}^\varepsilon) \rightharpoonup v_{in}^{1,\alpha}$ weakly in $L^2(\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{1,vac})$.
2. There exist $\phi_{in}^1 \in L^2(\Omega_{in,1}^\#, H^1(\Omega_{in,1}^{\infty,vac}))$, $\Omega_{in,1}^{\infty,vac}$ - periodic in the direction x_1^1 , ϕ_{in}^1 and its gradient exponentially decrease to 0 when $|x_2^1| \rightarrow +\infty$, and $v_{in}^1 \in L^2(\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{\infty,vac})$ such that $\phi_{in}^{1,\alpha} \chi_{\Omega_{in,1}^{1,vac}} \rightharpoonup \phi_{in,1}^1$ weakly in $L^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{\infty,vac})$ and $v_{in}^{1,\alpha} \chi_{\Omega_{in,1}^{1,vac}} \rightharpoonup v_{in}^1$ weakly in $L^2(\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{\infty,vac})$.

The next proposition is a corollary of Proposition 18.

Proposition 19:

If $\phi^0 \in C^1(\Omega^\#; L^2(\Omega^1)) \cap C^0(\Omega^\#; H^1(\Omega^1))$ and Assumption 4 is fulfilled then $T_{in,1}^\varepsilon(\phi^\varepsilon) \rightharpoonup \phi_{in}^{1,\alpha} + \widetilde{\phi}^0$ weakly in $L^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{1,vac})$ and $T_{in,1}^\varepsilon(V^\varepsilon) \rightharpoonup v_{in}^{1,\alpha} + \widetilde{V}^0$ weakly in $L^2(\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{1,vac})$, where $\widetilde{\psi}(x^\#, x^1) = \psi((x^\#, L_1^2), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2}))$.

Proposition 20:

The limit ϕ_{in}^1 is a solution to

$$\left\{ \begin{array}{ll} -\Delta_{x^1} \phi_{in}^1 = 0 & \text{in } \Omega_{in,1}^{\infty,vac} \\ \phi_{in}^1 = v_{in}^1 & \text{on } \Gamma_{in,1,int}^{\infty,vac} \\ \nabla_{x^1} \phi_{in}^1 \cdot \mathbf{n}^1 = 0 & \text{on } \Gamma_{in,1,top}^{\infty,vac} \\ \nabla_{x^1} \phi_{in}^1 \cdot \mathbf{n}^1 & \text{is } \Gamma_{in,1,per}^{\infty,vac} \text{ - antiperiodic} \\ \left[\left[\nabla_{x^1} \phi_{in}^1 \right] \right] \cdot \mathbf{n}^1 = - \left[\left[\nabla_{x^1} \widetilde{\phi}^0 \right] \right] \cdot \mathbf{n}^1 & \text{on } \Gamma_{in,1,interf}^{\infty,vac} \\ \left[\left[\phi_{in}^1 \right] \right] = - \left[\left[\widetilde{\phi}^0 \right] \right] & \text{on } \Gamma_{in,1,interf}^{\infty,vac} \\ \phi_{in}^1 & \text{is } \Gamma_{in,1,per}^{\infty,vac} \text{ - periodic.} \end{array} \right.$$

Proof. Let us repeat some steps of the previous proof. We replace v^ε by a smooth function v_{in}^ε in (2.6.3), where v_{in}^ε is defined in $\Omega_{in,1}^{\alpha\varepsilon,vac}$, $v_{in}^\varepsilon = 0$ on $\Gamma_{in,1,int}^{\alpha\varepsilon,vac}$ and vanishes out of $\Omega_{in,1}^{\alpha\varepsilon,vac}$.

$$\int_{\Omega_{in,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} v_{ex}^\varepsilon dx^\varepsilon = \int_{\Gamma_{in,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} v_{ex}^\varepsilon \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon) + \int_{\Gamma_{in,1,ext}^{\alpha\varepsilon,vac}} \varphi^\varepsilon \nabla_{x^\varepsilon} v_{ex}^\varepsilon \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon).$$

Then, we substitute v_{in}^ε by $B_{in,1}^\varepsilon(w)$, where w is in $C^\infty(\Omega_{in,1}^\# \times \overline{\Omega_{in,1}^{1,vac}})$, $\Omega_{in,1}^{1,vac}$ - periodic in the

directions x_1^1, x_2^1 , $w = 0$ on $\Gamma_{in,1,int}^{1,vac\pm} \cup \Gamma_{in,1,\alpha}^{1,vac\pm}$ and $\nabla_{x^1} w \cdot \mathbf{n}^1 = 0$ on $\Gamma_{in,1,top}^{1,vac,\pm} \cup \Gamma_{in,1,per}^{1,vac\pm} \cup \Gamma_{in,1,\alpha}^{1,vac\pm}$, we get

$$\int_{\Omega_{in,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} B_{in,1}^\varepsilon(w) dx^\varepsilon = \int_{\Gamma_{in,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} B_{in,1}^\varepsilon(w) \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon) + \int_{\Gamma_{in,1,ext}^{\alpha\varepsilon,vac}} \phi^\varepsilon \nabla_{x^\varepsilon} B_{in,1}^\varepsilon(w) \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon).$$

Clearly,

$$\begin{aligned} \frac{\partial B_{in,1}^\varepsilon w}{\partial x_i^\varepsilon} &= B_{in,1}^\varepsilon \left(\chi_{I^\sharp}(i) \frac{\partial w}{\partial x^\sharp} + \frac{1}{\varepsilon} \frac{\partial w}{\partial x_i^1} \right), \\ \frac{\partial}{\partial x_i^\varepsilon} \frac{\partial B_{in,1}^\varepsilon w}{\partial x_i^\varepsilon} &= B_{in,1}^\varepsilon \left(\chi_{I^\sharp}(i) \frac{\partial}{\partial x^\sharp} \frac{\partial w}{\partial x^\sharp} + \chi_{I^\sharp}(i) \frac{2}{\varepsilon} \frac{\partial}{\partial x^\sharp} \frac{\partial w}{\partial x_i^1} + \frac{1}{\varepsilon^2} \frac{\partial}{\partial x_i^1} \frac{\partial w}{\partial x_i^1} \right), \end{aligned}$$

for all $i \in I = \{1, 2, 3\}$ where $I^\sharp = \{1\}$.

We check that $B_{in,1}^\varepsilon(\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon = 0$ on $\Gamma_{in,1,ext}^{\alpha\varepsilon,vac}$ and a straightforward calculation reveals that

$$\frac{1}{\varepsilon^2} \int_{\Omega_{in,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B_{in,1}^\varepsilon(\Delta_{x^1} w) dx^\varepsilon = \frac{1}{\varepsilon} \int_{\Gamma_{in,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon B_{in,1}^\varepsilon(\nabla_{x^1} w \cdot \mathbf{n}^1) ds(x^\varepsilon) + O(\varepsilon),$$

where

$$\begin{aligned} O(\varepsilon) &= \int_{\Omega_{in,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B_{in,1}^\varepsilon \left(\frac{\partial}{\partial x^\sharp} \frac{\partial w}{\partial x^\sharp} \right) dx^\varepsilon + \frac{2}{\varepsilon} \int_{\Omega_{in,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B_{in,1}^\varepsilon \left(\frac{\partial}{\partial x^\sharp} \frac{\partial w}{\partial x_i^1} \right) dx^\varepsilon \\ &\quad - \int_{\partial\Omega_{in,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B_{in,1}^\varepsilon \left(\frac{\partial w}{\partial x^\sharp} \right) n_1^\varepsilon ds(x^\varepsilon). \end{aligned}$$

Thanks to Proposition 17, we have

$$\frac{1}{\varepsilon^2} \int_{\Omega_{in,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon T_{in,1}^{\varepsilon*}(\Delta_{x^1} w) dx^\varepsilon = \frac{1}{\varepsilon} \int_{\Gamma_{in,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon T_{in,1}^{\varepsilon*}(\nabla_{x^1} w \cdot \mathbf{n}^1) ds(x^\varepsilon) + O(\varepsilon). \quad (2.6.12)$$

By the definition of $T_{in,1}^{\varepsilon*}$, it follows that

$$\int_{\Omega_{in,1}^\sharp \times \Omega_{in,1}^{1,vac}} T_{in,1}^\varepsilon(\phi^\varepsilon) \Delta_{x^1} w dx^\sharp dx^1 = \int_{\Omega_{in,1}^\sharp \times \Gamma_{in,1,int}^{\alpha\varepsilon,vac}} T_{in,1}^\varepsilon(V^\varepsilon) \nabla_{x^1} w \cdot \mathbf{n}^1 dx^\sharp ds(x^1) + O(\varepsilon).$$

Passing ε to 0, combined with Proposition 19, we obtain

$$\int_{\Omega_{in,1}^\sharp \times \Omega_{in,1}^{1,vac}} (\phi_{in}^{1,\alpha} + \tilde{\phi}^0) \Delta_{x^1} w dx^\sharp dx^1 = \int_{\Omega_{in,1}^\sharp \times \Gamma_{in,1,int}^{1,vac}} (v_{in}^{1,\alpha} + \tilde{V}^0) \nabla_{x^1} w \cdot \mathbf{n}^1 dx^\sharp ds(x^1).$$

for each α .

It follows that the above equality still holds if w is taken on the form of $\tau_\alpha v$, where τ_α is a smooth truncation function with compact support $\Omega_{in,1}^\sharp \times \Omega_{in,1}^{1,vac}$ and $v \in C^\infty(\Omega_{in,1}^\sharp \times \overline{\Omega_{in,1}^{\infty,vac}}) \cap H^2(\Omega_{in,1}^\sharp \times \overline{\Omega_{in,1}^{\infty,vac}})$, $\Omega_{in,1}^{\infty,vac}$ - periodic in the directions x_1^1, x_2^1 , $v = 0$ on $\Gamma_{in,1,int}^{\infty,vac\pm}$, $\nabla_{x^1} v \cdot \mathbf{n}^1 = 0$ on

$\Gamma_{in,1,top}^{\infty,vac,\pm} \cup \Gamma_{in,1,per}^{\infty,vac,\pm}$, $|v|$, $|\nabla_{x^1} v|$, and $|\Delta_{x^1} v|$ exponentially decrease to 0 when $|x_2^1| \rightarrow +\infty$, and $\tau_\alpha v \rightarrow v$ in $H^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{\infty,vac})$ when α tends to infinity. Then

$$\int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{\infty,vac}} (\phi_{in}^{1,\alpha} + \widetilde{\phi}^0) \chi_{\Omega_{in,1}^{1,vac}} \Delta_{x^1} w \, dx^\# dx^1 = \int_{\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{\infty,vac}} (v_{in}^{1,\alpha} + \widetilde{V}^0) \chi_{\Omega_{in,1}^{1,vac}} \nabla_{x^1} w \cdot \mathbf{n}^1 \, dx^\# ds(x^1).$$

Passing α to $+\infty$, by Assumption 2, we get

$$\int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{\infty,vac}} (\phi_{in}^1 + \widetilde{\phi}^0) \Delta_{x^1} v \, dx^\# dx^1 = \int_{\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{\infty,vac}} (v_{in}^1 + \widetilde{V}^0) \chi_{\Omega_{in,1}^{1,vac}} \nabla_{x^1} v \cdot \mathbf{n}^1 \, dx^\# ds(x^1).$$

Now, we choose v vanishing out of $\Omega_{in,1}^\# \times \Omega_{in,1}^{1,vac}$ for a given α ,

$$\int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,vac}} (\phi_{in}^1 + \widetilde{\phi}^0) \Delta_{x^1} v \, dx^\# dx^1 = \int_{\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{1,vac}} (v_{in}^1 + \widetilde{V}^0) \nabla_{x^1} v \cdot \mathbf{n}^1 \, dx^\# ds(x^1).$$

Applying Green's formula twice, then

$$\begin{aligned} & \sum_{\pm} \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,vac,\pm}} \Delta_{x^1} (\phi_{in}^{1,\pm} + \widetilde{\phi}^{0,\pm}) v \, dx^\# dx^1 \\ & - \sum_{\pm} \int_{\Omega_{in,1}^\# \times \partial\Omega_{in,1}^{1,vac,\pm}} \nabla_{x^1} (\phi_{in}^{1,\pm} + \widetilde{\phi}^{0,\pm}) \cdot \mathbf{n}^{1,\pm} v \, dx^\# ds(x^1) \\ & + \sum_{\pm} \int_{\Omega_{in,1}^\# \times \partial\Omega_{in,1}^{1,vac,\pm}} (\phi_{in}^{1,\pm} + \widetilde{\phi}^{0,\pm}) \nabla_{x^1} v \cdot \mathbf{n}^{1,\pm} \, dx^\# ds(x^1) \\ & = \sum_{\pm} \int_{\Omega_{in,1}^\# \times \Gamma_{in,1,int}^{1,vac,\pm}} (v_{in}^{1,\pm} + \widetilde{V}^{0,\pm}) \nabla_{x^1} v \cdot \mathbf{n}^{1,\pm} \, dx^\# ds(x^1). \end{aligned}$$

Decomposing $\Omega_{in,1}^{1,vac}$ into two parts $\Omega_{in,1}^{1,vac,\pm}$ with their boundaries $\partial\Omega_{in,1}^{1,vac,\pm} = \Gamma_{in,1,int}^{1,vac,\pm} \cup \Gamma_{in,1,top}^{1,vac,\pm} \cup \Gamma_{bl,1,per}^{1,vac,\pm} \cup \Gamma_{in,1,\alpha}^{1,vac,\pm} \cup \Gamma_{in,1,interf}^{1,vac}$, combining with the results of Proposition 9, $\Delta_{x^1} \widetilde{\phi}^{0,\pm} = 0$ in $\Omega_{in,1}^{1,vac,\pm}$, $\widetilde{\phi}^{0,\pm} = \widetilde{V}^{0,\pm}$ on $\Gamma_{in,1,int}^{1,vac,\pm}$, $\nabla_{x^1} \widetilde{\phi}^{0,\pm} \cdot \mathbf{n}^{1,\pm} = 0$ on $\Gamma_{in,1,top}^{1,vac,\pm}$, $\nabla_{x^1} \widetilde{\phi}^{0,\pm} \cdot \mathbf{n}^{1,\pm}$ is $\Gamma_{bl,1,per}^{1,vac,\pm}$ -antiperiodic, and

from the conditions satisfied by v it remains

$$\begin{aligned}
& \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Omega_{in,1}^{1,vac\pm}} \Delta_{x^1}(\phi_{in}^{1\pm}) v \, dx^{\#} dx^1 \\
& - \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times (\Gamma_{in,1,top}^{1,vac\pm} \cup \Gamma_{in,1,per}^{1,vac\pm})} \nabla_{x^1} \phi_{in}^{1\pm} \cdot \mathbf{n}^{1\pm} v \, dx^{\#} ds(x^1) \\
& - \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,interf}^{1,vac}} \left[\nabla_{x^1} (\phi_{in}^{1+} + \tilde{\phi}^{0+}) - \nabla_{x^1} (\phi_{in}^{1-} + \tilde{\phi}^{0-}) \right] \cdot \mathbf{n}^{1+} v \, dx^{\#} ds(x^1) \\
& + \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,int}^{1,vac\pm}} \phi_{in}^{1\pm} \nabla_{x^1} v \cdot \mathbf{n}^{1\pm} \, dx^{\#} ds(x^1) \\
& + \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,interf}^{1,vac}} \left[(\phi_{in}^{1+} + \tilde{\phi}^{0+}) - (\phi_{in}^{1-} + \tilde{\phi}^{0-}) \right] \nabla_{x^1} v \cdot \mathbf{n}^{1+} \, dx^{\#} ds(x^1) \\
& = \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,int}^{1,vac\pm}} v_{in}^{1\pm} \nabla_{x^1} v \cdot \mathbf{n}^{1\pm} \, dx^{\#} ds(x^1).
\end{aligned}$$

The rest of proof runs as the previous proof. \square

2.6.5/ INTERNAL EDGE MODEL

Let us assume that all interface models are yet built with the index $i = 1, 2, 3, 4$ as in Figure 2.1. We consider the contributions of two interface models $i = 1$ and $i = 2$ at the first internal edge zone. Obviously, the sum of contributions is not continuous at internal edges which are interfaces between subdomains of the first physical internal edged domain, see Figure 2.11, then it leads to propose an internal edge corrector to overcome this problem. We now introduce terms

$$\begin{aligned}
\phi_{ine}^{\varepsilon} &= \phi^{\varepsilon} - B^{\varepsilon} \phi^0 - B_{in,2}^{\varepsilon} \phi_{in}^{2-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,vac,2}} - (B_{in,1}^{\varepsilon} \phi_{in}^{1+} + B_{in,2}^{\varepsilon} \phi_{in}^{2+}) \chi_{\Omega_{ine,1}^{\alpha\varepsilon,vac,3}} - B_{in,1}^{\varepsilon} \phi_{in}^{1-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,vac,4}}, \\
v_{ine}^{\varepsilon} &= V^{\varepsilon} - B^{\varepsilon} V^0 - B_{in,2}^{\varepsilon} v_{in}^{2-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,vac,2}} - (B_{in,1}^{\varepsilon} v_{in}^{1+} + B_{in,2}^{\varepsilon} v_{in}^{2+}) \chi_{\Omega_{ine,1}^{\alpha\varepsilon,vac,3}} - B_{in,1}^{\varepsilon} v_{in}^{1-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,vac,4}},
\end{aligned}$$

where ϕ^0 is the solution of the periodic model, $\phi_{in}^{1\pm}$ and $\phi_{in}^{2\pm}$ are the solutions of the first and second interface models in the interface zones near the first internal edge zone, $B_{in,1}^{\varepsilon}$ and $B_{in,2}^{\varepsilon}$ are the smooth approximation operators of the first and second adjoint interface two-scale operators $T_{in,1}^{\varepsilon*}$ and $T_{in,2}^{\varepsilon*}$, $v_{in}^{1\pm}$ and $v_{in}^{2\pm}$ are the weak limits of $v_{in}^{1,\alpha\pm} \chi_{\Omega_{in,1}^{1,vac\pm}}$ in $L^2(\Omega_{in,1}^{\#} \times \Gamma_{in,1,int}^{\infty,vac\pm})$ and of $v_{in}^{2,\alpha\pm} \chi_{\Omega_{in,2}^{1,vac\pm}}$ in $L^2(\Omega_{in,2}^{\#} \times \Gamma_{in,2,int}^{\infty,vac\pm})$ when α tends to $+\infty$, $v_{in}^{1,\alpha\pm}$ and $v_{in}^{2,\alpha\pm}$ being the weak limits of $T_{in,1}^{\varepsilon}(v_{in}^{\varepsilon})$ in $L^2(\Omega_{in,1}^{\#} \times \Gamma_{in,1,int}^{1,vac\pm})$ and of $T_{in,2}^{\varepsilon}(v_{in}^{\varepsilon})$ in $L^2(\Omega_{in,2}^{\#} \times \Gamma_{in,2,int}^{1,vac\pm})$ when ε tends to 0, $\chi_{\Omega_{ine,1}^{\alpha\varepsilon,vac,i}}$ are the characteristic functions of the domains $\Omega_{ine,1}^{\alpha\varepsilon,vac,i}$ introduced in the following.

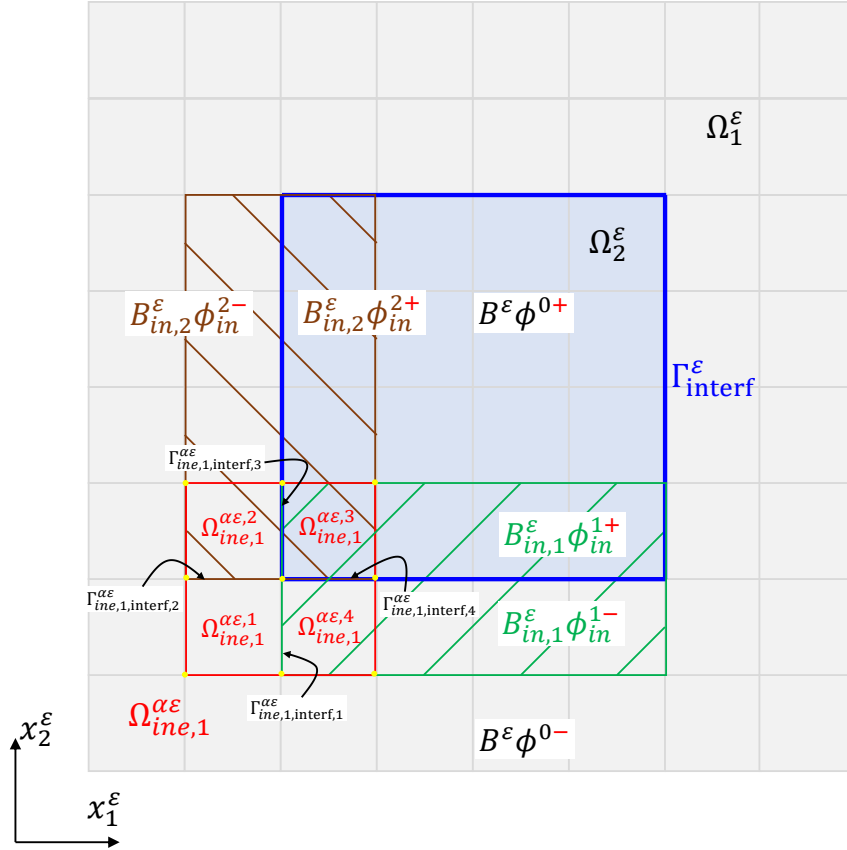


Figure 2.11: Description of the geometry of the internal edge problem. The green and maroon colors represent the zones of the first and the second interface models. The red region is the zone of the first internal edge model made with four subregions. The electrostatic potential has a different approximation in each of these subregions.

2.6.5.1/ GEOMETRY NOTATIONS

We begin to define the whole internal edge boundary layer domain $\Omega_{ine,1}^{\alpha\epsilon}$, which subscript $ine,1$ refers to the first internal edge, as a subdomain of $\Omega_1^\epsilon \cup \Omega_2^\epsilon$ and constructed by $\Omega_{ine,1}^{\alpha\epsilon} = \cup_{c \in I_{ine,1}} \Omega_c^\epsilon$. Here $I_{ine,1}$ is a set of multi-indices $c = (c_1, c_2) : c_1 \in \overline{i_1 - \alpha, i_1 + \alpha - 1}$, and $c_2 \in \overline{i_2 - \alpha, i_2 + \alpha - 1}$, i_1, i_2 being such that $\Omega_{(i_1, i_2)}^\epsilon$ is the first internal edge cell, see Figure 2.12.

The domain $\Omega_{ine,1}^{\alpha\epsilon}$ is decomposed into four nonoverlapping subdomains $\Omega_{ine,1}^{\alpha\epsilon, i}$ defined as $\Omega_{ine,1}^{\alpha\epsilon, i} = \cup_{c \in I_{ine,1}^i} \Omega_c^\epsilon$ with the multi-index sets $I_{ine,1}^i$

$$\begin{aligned} I_{ine,1}^1 &= \{(c_1, c_2) : c_1 \in \overline{i_1 - \alpha, i_1 - 1}, c_2 \in \overline{i_2 - \alpha, i_2 - 1}\}, \\ I_{ine,1}^2 &= \{(c_1, c_2) : c_1 \in \overline{i_1 - \alpha, i_1 - 1}, c_2 \in \overline{i_2, i_2 + \alpha - 1}\}, \\ I_{ine,1}^3 &= \{(c_1, c_2) : c_1 \in \overline{i_1, i_1 + \alpha - 1}, c_2 \in \overline{i_2, i_2 + \alpha - 1}\}, \\ I_{ine,1}^4 &= \{(c_1, c_2) : c_1 \in \overline{i_1, i_1 + \alpha - 1}, c_2 \in \overline{i_2 - \alpha, i_2 - 1}\}. \end{aligned}$$

We observe that $\Omega_{ine,1}^{\alpha\varepsilon,i}$ is a subdomain of Ω_1^ε for $i = 1, 2, 4$ and of Ω_2^ε for $i = 3$. For the sake of concision, interface numbering is with indices modulo 4, e.g. 5 plays the role of 1. Precisely, the interface between $\Omega_{ine,1}^{\alpha\varepsilon,i}$ and $\Omega_{ine,1}^{\alpha\varepsilon,i+1}$ is noted $\Gamma_{ine,1,interf,i+1}^{\alpha\varepsilon}$ for $i = 1, 2, 3, 4$ and $\Gamma_{ine,1,interf,5}^{\alpha\varepsilon}$ or $\Gamma_{ine,1,interf,1}^{\alpha\varepsilon}$ for $i = 4$.

The vacuum parts of all components of $\Omega_{ine,1}^{\alpha\varepsilon}$ are decorated with the superscript *vac*. The boundary $\partial\Omega_{ine,1}^{\alpha\varepsilon,vac,i}$ of each $\Omega_{ine,1}^{\alpha\varepsilon,vac,i}$ is decomposed as $\Gamma_{ine,1,int}^{\alpha\varepsilon,vac,i} \cup \Gamma_{ine,1,ext}^{\alpha\varepsilon,vac,i} \cup \Gamma_{ine,1,interf,i}^{\alpha\varepsilon,vac}$ with $\Gamma_{ine,1,interf,i+1}^{\alpha\varepsilon,vac} = \Gamma_{ine,1,top}^{\alpha\varepsilon,vac,i} \cup \Gamma_{ine,1,\alpha}^{\alpha\varepsilon,vac,i}$. The whole interface between these domains is $\Gamma_{ine,1,interf}^{\alpha\varepsilon,vac} = \bigcup_{i=1}^4 \Gamma_{ine,1,interf,i}^{\alpha\varepsilon,vac}$. Thus, the whole boundary $\partial\Omega_{ine,1}^{\alpha\varepsilon,vac}$ is $(\bigcup_{i=1}^4 \partial\Omega_{ine,1}^{\alpha\varepsilon,vac,i}) \setminus \Gamma_{ine,1,interf}^{\alpha\varepsilon,vac}$.

The mechanical parts of all components of $\Omega_{ine,1}^{\alpha\varepsilon}$ are decorated with the superscript *mec*. The principle of notations is similar to this of the vacuum, but with taking into account the special features of the mechanical boundaries. It is skipped.

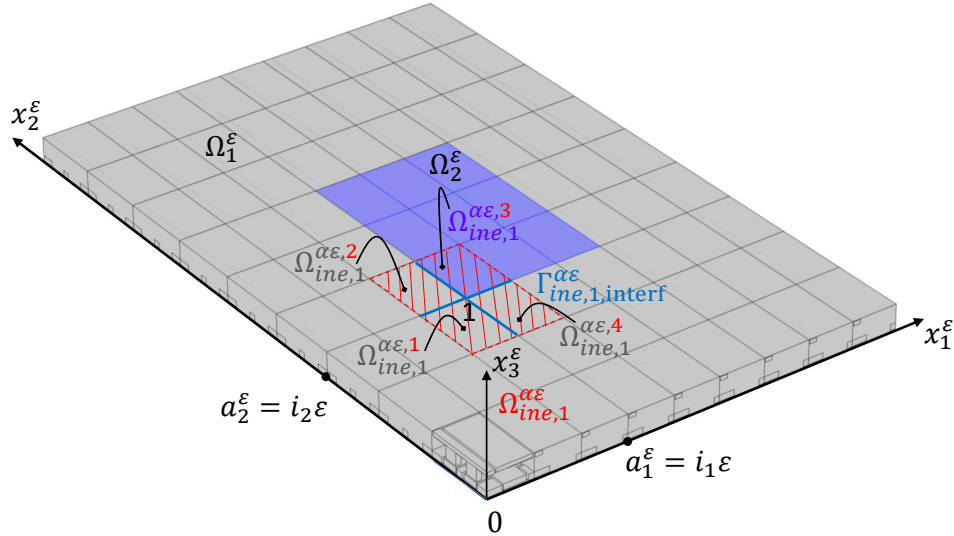


Figure 2.12: The first internal edge $\Omega_{ine,1}^{\alpha\varepsilon}$ in the physical domain with $\alpha = 1$.

The finite microscopic domain $\Omega_{ine,1}^1$, parametrized by α , is similarly defined as $\bigcup_{i=1}^4 \Omega_{ine,1}^{1,i}$, see Figure 2.13, with

$$\begin{aligned}\Omega_{ine,1}^{1,1} &= \bigcup_{\xi,\eta=0,\alpha-1}(\Omega^1 + (-\xi - 1/2, -\eta - 1/2, 1/2)), \\ \Omega_{ine,1}^{1,2} &= \bigcup_{\xi,\eta=0,\alpha-1}(\Omega^1 + (-\xi - 1/2, \eta + 1/2, 1/2)), \\ \Omega_{ine,1}^{1,3} &= \bigcup_{\xi,\eta=0,\alpha-1}(\Omega^1 + (\xi + 1/2, \eta + 1/2, 1/2)), \\ \Omega_{ine,1}^{1,4} &= \bigcup_{\xi,\eta=0,\alpha-1}(\Omega^1 + (\xi + 1/2, -\eta - 1/2, 1/2)).\end{aligned}$$

The notation system built for the physical domain is transposed to the microscopic domain without the need to detail it. The infinite microscopic domain $\Omega_{ine,1}^\infty$ is defined as the limit

of $\Omega_{ine,1}^1$ when α tends to infinity.

Finally, for all regular function v defined in $\Omega_{in,1}^1$, we denote v^i the restriction of v to $\Omega_{in,1}^{1,i}$ and $[[v]]$ stands for a jump of v at the interface defined by the following formula

$$[[v]] = \begin{cases} v^1 - v^4 & \text{at } \Gamma_{ine,1,interf,1}^1 \\ v^1 - v^2 & \text{at } \Gamma_{ine,1,interf,2}^{1,vac} \\ v^3 - v^2 & \text{at } \Gamma_{ine,1,interf,3}^{1,vac} \\ v^3 - v^4 & \text{at } \Gamma_{ine,1,interf,4}^{1,vac} \end{cases}$$

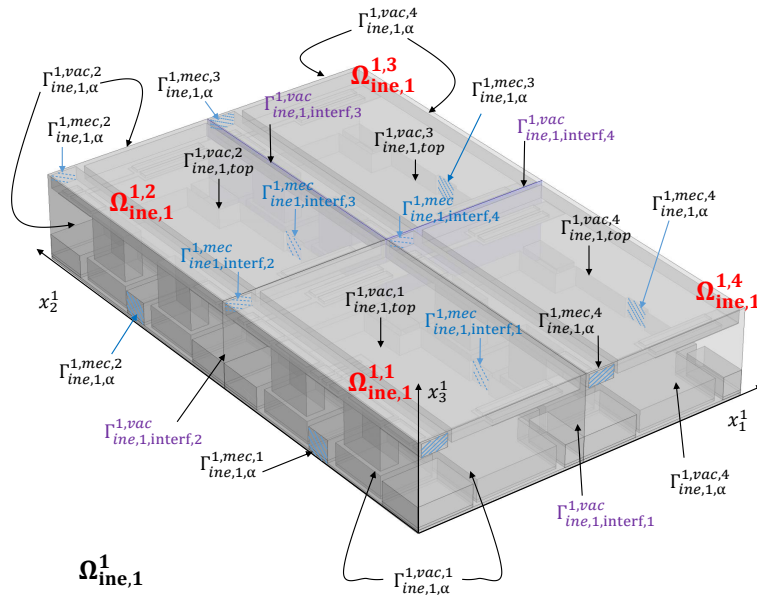


Figure 2.13: The first internal edge $\Omega_{ine,1}^1$ in the microscopic domain with $\alpha = 1$.

2.6.5.2/ INTERNAL EDGE BOUNDARY LAYER TWO-SCALE OPERATOR

Now we introduce the dilation operator $T_{ine,1}^\varepsilon$ at the first internal edge.

Definition 19:

For all functions $\varphi \in L^2(\Omega_{ine,1}^{\alpha\varepsilon})$ (resp., $\varphi \in L^2(\Gamma_{ine,1}^{\alpha\varepsilon})$), we define $T_{ine,1}^\varepsilon$ as

$$T_{ine,1}^\varepsilon(\varphi)(x^1) = \varphi(\varepsilon x_1^1 + L_1^1, \varepsilon x_2^1 + L_1^2, \varepsilon x_3^1)$$

for a.e. x^1 in $\Omega_{ine,1}^1$ (resp., x^1 in $\Gamma_{ine,1}^1$), where $L_1^1 = i_1\varepsilon$ and $L_1^2 = i_2\varepsilon$, $i_1, i_2 \in \mathbb{Z}^+$.

Definition 20:

We define the adjoint operator $T_{ine,1}^{\varepsilon^*}$ of $T_{ine,1}^{\varepsilon}$ by

$$\frac{1}{\varepsilon^3} \int_{\Omega_{ine,1}^{\alpha\varepsilon}} \varphi T_{ine,1}^{\varepsilon^*}(\psi) dx^\varepsilon = \int_{\Omega_{ine,1}^1} T_{ine,1}^{\varepsilon}(\varphi)\psi dx^1,$$

for all $\psi \in L^2(\Omega_{ine,1}^1)$, $\varphi \in L^2(\Omega_{ine,1}^{\alpha\varepsilon})$ and

$$\frac{1}{\varepsilon^2} \int_{\Gamma_{ine,1}^{\alpha\varepsilon}} \varphi T_{ine,1}^{\varepsilon^*}(\psi) ds(x^\varepsilon) = \int_{\Gamma_{ine,1}^1} T_{ine,1}^{\varepsilon}(\varphi)\psi ds(x^1),$$

for all $\psi \in L^2(\Gamma_{ine,1}^1)$, $\varphi \in L^2(\Gamma_{ine,1}^{\alpha\varepsilon})$.

From the definitions, we obtain an explicit formula of $T_{ine,1}^{\varepsilon^*}$ as

$$T_{ine,1}^{\varepsilon^*}(\psi)(x^\varepsilon) = \psi\left(\frac{x_1^\varepsilon - L_1^1}{\varepsilon}, \frac{x_2^\varepsilon - L_1^2}{\varepsilon}, \frac{x_3^\varepsilon}{\varepsilon}\right),$$

valid for ψ in $L^2(\Omega_{ine,1}^1)$ and in $L^2(\Gamma_{ine,1}^1)$.

In this internal edge case, the operator $T_{ine,1}^{\varepsilon^*}$ and its approximation $B_{ine,1}^{\varepsilon}$ are identical however both will be used to follow the algorithm of Section 2.5.3.

Proposition 21:

Let B^ε , $B_{in,1}^\varepsilon$, and $B_{in,2}^\varepsilon$ be smooth approximate operators of $T^{\varepsilon*}$, $T_{in,1}^{\varepsilon*}$, resp. $T_{in,2}^{\varepsilon*}$, then

1. For all $\psi \in C^1(\Omega^\sharp; L^2(\Omega^1)) \cap C^0(\Omega^\sharp; H^1(\Omega^1))$ and Ω^1 - periodic in the directions x_1^1, x_2^1 .

$$T_{ine,1}^\varepsilon(B^\varepsilon\psi) \rightarrow \widetilde{\psi} \text{ in } L^2(\Omega_{ine,1}^1) \text{ and in } L^2(\Gamma_{ine,1}^1),$$

where $\widetilde{\psi}(x^1) = \psi((L_1^1, L_1^2), x^1 - 1/2)$.

2. For all $\psi^\pm \in C^1(\Omega_{in,1}^\sharp; L^2(\Omega_{in,1}^{\infty\pm})) \cap C^0(\Omega_{in,1}^\sharp; H^1(\Omega_{in,1}^{\infty\pm}))$ and $\Omega_{in,1}^{\infty\pm}$ - periodic in the directions $x_1^{1\pm}$,

$$T_{ine,1}^\varepsilon(B_{in,1}^\varepsilon\psi^\pm) \rightarrow \widetilde{\psi}^\pm \text{ in } L^2(\Omega_{ine,1}^{1,3} \cup \Omega_{ine,1}^{1,4}) \text{ and in } L^2(\Gamma_{ine,1}^{1,3} \cup \Gamma_{ine,1}^{1,4}),$$

where $\widetilde{\psi}^\pm(x^1) = \psi^\pm(L_1^1, (x_1^1 - 1/2, x_2^1, x_3^1))$.

3. For all $\psi^\pm \in C^1(\Omega_{in,2}^\sharp; L^2(\Omega_{in,2}^{\infty\pm})) \cap C^0(\Omega_{in,2}^\sharp; H^1(\Omega_{in,2}^{\infty\pm}))$ and $\Omega_{in,2}^{\infty\pm}$ - periodic in the directions $x_2^{1\pm}$,

$$T_{ine,1}^\varepsilon(B_{in,2}^\varepsilon\psi^\pm) \rightarrow \widetilde{\psi}^\pm \text{ in } L^2(\Omega_{ine,1}^{1,2} \cup \Omega_{ine,1}^{1,3}) \text{ and in } L^2(\Gamma_{ine,1}^{1,2} \cup \Gamma_{ine,1}^{1,3}),$$

where $\widetilde{\psi}^\pm(x^1) = \psi^\pm(L_1^2, (x_1^1, x_2^1 - 1/2, x_3^1))$.

2.6.5.3/ DERIVATION OF AN INTERNAL EDGE MODEL**Assumption 5:**

We assume that

1. For each α , there exist $\phi_{ine}^{1,\alpha}$ in $H^1(\Omega_{ine,1}^{1,vac})$ and $v_{ine}^{1,\alpha}$ in $L^2(\Gamma_{ine,1,int}^{1,vac})$ such that $T_{ine,1}^\varepsilon(\phi_{ine}^\varepsilon) \rightharpoonup \phi_{ine}^{1,\alpha}$ weakly in $L^2(\Omega_{ine,1}^{1,vac})$ and $T_{ine,1}^\varepsilon(v_{ine}^\varepsilon) \rightharpoonup v_{ine}^{1,\alpha}$ weakly in $L^2(\Gamma_{ine,1,int}^{1,vac})$.
2. There exist ϕ_{ine}^1 in $H^1(\Omega_{ine,1}^{\infty,vac})$, ϕ_{ine}^1 and its gradient converge exponentially fast to zero when $|x_1^1| + |x_2^1| \rightarrow +\infty$, and v_{ine}^1 in $L^2(\Gamma_{ine,1,int}^{\infty,vac})$ such that $\phi_{ine}^{1,\alpha} \chi_{\Omega_{ine,1}^{1,vac}} \rightharpoonup \phi_{ine}^1$ weakly in $L^2(\Omega_{ine,1}^{\infty,vac})$ and $v_{ine}^{1,\alpha} \chi_{\Omega_{ine,1}^{1,vac}} \rightharpoonup v_{ine}^1$ weakly in $L^2(\Gamma_{ine,1,int}^{\infty,vac})$.

The next proposition results from using Proposition 21.

Proposition 22:

Assuming $\phi^0 \in C^1(\Omega^\sharp; L^2(\Omega^1)) \cap C^0(\Omega^\sharp; H^1(\Omega^1))$, $\phi_{in}^{1-} \in C^1(\Omega_{in,1}^\sharp; L^2(\Omega_{in,1}^{\infty\pm})) \cap C^0(\Omega_{in,1}^\sharp; H^1(\Omega_{in,1}^{\infty\pm}))$, $\phi_{in}^{2-} \in C^1(\Omega_{in,2}^\sharp; L^2(\Omega_{in,2}^{\infty\pm})) \cap C^0(\Omega_{in,2}^\sharp; H^1(\Omega_{in,2}^{\infty\pm}))$ **Asumption 5** then

$$T_{ine,1}^\varepsilon(\phi^\varepsilon) \rightharpoonup \phi_{ine}^{1,\alpha} + \widetilde{\phi}^0 + \widetilde{\phi_{in}^{2-}} \chi_{\Omega_{ine,1}^{1,vac,2}} + (\widetilde{\phi_{in}^{1+}} + \widetilde{\phi_{in}^{2+}}) \chi_{\Omega_{ine,1}^{1,vac,3}} + \widetilde{\phi_{in}^{1-}} \chi_{\Omega_{ine,1}^{1,vac,4}}$$

weakly in $L^2(\Omega_{ine,1}^{1,vac})$ and

$$T_{ine,1}^\varepsilon(v^\varepsilon) \rightharpoonup v_{ine}^{1,\alpha} + \widetilde{V}^0 + \widetilde{v_{in}^{2-}} \chi_{\Omega_{ine,1}^{1,vac,2}} + (\widetilde{v_{in}^{1+}} + \widetilde{v_{in}^{2+}}) \chi_{\Omega_{ine,1}^{1,vac,3}} + \widetilde{v_{in}^{1-}} \chi_{\Omega_{ine,1}^{1,vac,4}}$$

weakly in $L^2(\Gamma_{ine,1,int}^{1,vac})$, where $\widetilde{\phi^{0\pm}}(x^1) = \phi^{0\pm}((L_1^1, L_1^2), x^1 - 1/2)$, $\widetilde{\phi_{in}^{1\pm}}(x^1) = \phi_{in}^{1\pm}(L_1^2, (x_1^1 - 1/2, x_2^1, x_3^1))$, and $\widetilde{\phi_{in}^{2\pm}}(x^1) = \phi_{in}^{2\pm}(L_1^1, (x_1^1, x_2^1 - 1/2, x_3^1))$.

Proposition 23:

The limit ϕ_{ine}^1 is a solution to

$$\left\{ \begin{array}{ll} -\Delta_{x^1} \phi_{ine}^1 = 0 & \text{in } \Omega_{ine,1}^{\infty,vac} \\ \phi_{ine}^1 = v_{ine}^1 & \text{on } \Gamma_{ine,1,int}^{\infty,vac} \\ \nabla_{x^1} \phi_{ine}^1 \cdot \mathbf{n}^1 = 0 & \text{on } \Gamma_{ine,1,top}^{\infty,vac} \\ [[\phi_{ine}^1]] = \widetilde{\phi_{in}^{1-}} & \text{on } \Gamma_{ine,1,interf,1}^{\infty,vac} \\ [[\nabla_{x^1} \phi_{ine}^1]] \cdot \mathbf{n}^1 = \nabla_{x^1} \widetilde{\phi_{in}^{1-}} \cdot \mathbf{n}^1 & \text{on } \Gamma_{ine,1,interf,1}^{\infty,vac} \\ [[\phi_{ine}^1]] = \widetilde{\phi_{in}^{2-}} & \text{on } \Gamma_{ine,1,interf,2}^{\infty,vac} \\ [[\nabla_{x^1} \phi_{ine}^1]] \cdot \mathbf{n}^1 = \nabla_{x^1} \widetilde{\phi_{in}^{2-}} \cdot \mathbf{n}^1 & \text{on } \Gamma_{ine,1,interf,2}^{\infty,vac} \\ [[\phi_{ine}^1]] = -\widetilde{\phi_{in}^{1+}} & \text{on } \Gamma_{ine,1,interf,3}^{\infty,vac} \\ [[\nabla_{x^1} \phi_{ine}^1]] \cdot \mathbf{n}^1 = -\nabla_{x^1} \widetilde{\phi_{in}^{1+}} \cdot \mathbf{n}^1 & \text{on } \Gamma_{ine,1,interf,3}^{\infty,vac} \\ [[\phi_{ine}^1]] = -\widetilde{\phi_{in}^{2+}} & \text{on } \Gamma_{ine,1,interf,4}^{\infty,vac} \\ [[\nabla_{x^1} \phi_{ine}^1]] \cdot \mathbf{n}^1 = -\nabla_{x^1} \widetilde{\phi_{in}^{2+}} \cdot \mathbf{n}^1 & \text{on } \Gamma_{ine,1,interf,4}^{\infty,vac} \end{array} \right.$$

Proof. The main idea of the proof is the same as the previous proofs. Firstly, we replace v^ε in (2.6.3) by a smooth function v_{ine}^ε defined in $\Omega_{ine,1}^{\alpha\varepsilon,vac}$ and vanishing out of $\Omega_{ine,1}^{\alpha\varepsilon,vac}$, then

$$\int_{\Omega_{ine,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} v_{ine}^\varepsilon \, dx^\varepsilon = \int_{\Gamma_{ine,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} v_{ine}^\varepsilon \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon) + \int_{\Gamma_{ine,1,ext}^{\alpha\varepsilon,vac}} \varphi^\varepsilon \nabla_{x^\varepsilon} v_{ine}^\varepsilon \cdot \mathbf{n}^\varepsilon \, ds(x^\varepsilon).$$

After that, we substitute v_{ine}^ε by $\varepsilon^{-1} B_{ine,1}^\varepsilon(w)$ where w is in $C^\infty(\overline{\Omega_{ine,1}^{1,vac}})$ such that $w = 0$ on

$\Gamma_{ine,1,int}^{1,vac} \cup \Gamma_{ine,1,\alpha}^{1,vac}$ and $\nabla_{x^1} w \cdot \mathbf{n}^1 = 0$ on $\Gamma_{ine,1,top}^{1,vac} \cup \Gamma_{ine,1,\alpha}^{1,vac}$, hence

$$\frac{1}{\varepsilon} \int_{\Omega_{ine,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon \Delta_{x^\varepsilon} B_{ine,1}^\varepsilon(w) dx^\varepsilon = \frac{1}{\varepsilon} \int_{\Gamma_{ine,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon \nabla_{x^\varepsilon} B_{ine,1}^\varepsilon(w) \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon) + \frac{1}{\varepsilon} \int_{\Gamma_{ine,1,ext}^{\alpha\varepsilon,vac}} \phi^\varepsilon \nabla_{x^\varepsilon} B_{ine,1}^\varepsilon(w) \cdot \mathbf{n}^\varepsilon ds(x^\varepsilon).$$

Obviously,

$$\frac{\partial B_{ine,1}^\varepsilon w}{\partial x_i^\varepsilon} = \frac{1}{\varepsilon} B_{ine,1}^\varepsilon \left(\frac{\partial w}{\partial x_i^1} \right) \text{ and } \frac{\partial}{\partial x_i^\varepsilon} \frac{\partial B_{ine,1}^\varepsilon w}{\partial x_i^\varepsilon} = \frac{1}{\varepsilon^2} B_{ine,1}^\varepsilon \left(\frac{\partial}{\partial x_i^1} \frac{\partial w}{\partial x_i^1} \right),$$

for all $i = 1, 2, 3$, and $B_{ine,1}^\varepsilon(\nabla_{x^1} w) \cdot \mathbf{n}^\varepsilon = 0$ on $\Gamma_{ine,1,ext}^{\alpha\varepsilon,vac}$. Thus,

$$\frac{1}{\varepsilon^3} \int_{\Omega_{ine,1}^{\alpha\varepsilon,vac}} \phi^\varepsilon B_{ine,1}^\varepsilon(\Delta_{x^1} w) dx^\varepsilon = \frac{1}{\varepsilon^2} \int_{\Gamma_{ine,1,int}^{\alpha\varepsilon,vac}} V^\varepsilon B_{ine,1}^\varepsilon(\nabla_{x^1} w \cdot \mathbf{n}^1) ds(x^\varepsilon).$$

Replacing $B_{ine,1}^\varepsilon$ by $T_{ine,1}^{\varepsilon*}$, then transposing $T_{ine,1}^{\varepsilon*}$ to $T_{ine,1}^\varepsilon$, we have

$$\int_{\Omega_{ine,1}^{1,vac}} T_{ine,1}^\varepsilon(\phi^\varepsilon) \Delta_{x^1} w dx^1 = \int_{\Gamma_{ine,1,int}^{1,vac}} T_{ine,1}^\varepsilon(V^\varepsilon) \nabla_{x^1} w \cdot \mathbf{n}^1 ds(x^1).$$

Decomposing $\Omega_{ine,1}^{1,vac} = \cup_{i=1}^4 \Omega_{ine,1}^{1,vac,i}$ and $\Gamma_{ine,1}^{1,vac} = \cup_{i=1}^4 \Gamma_{ine,1,int}^{1,vac,i}$ the above equality becomes

$$\sum_{i=1}^4 \int_{\Omega_{ine,1}^{1,vac,i}} T_{ine,1}^\varepsilon(\phi^\varepsilon) \Delta_{x^1} w dx^1 = \sum_{i=1}^4 \int_{\Gamma_{ine,1,int}^{1,vac,i}} T_{ine,1}^\varepsilon(V^\varepsilon) \nabla_{x^1} w \cdot \mathbf{n}^{1,i} ds(x^1).$$

Passing ε to 0, and combining with Proposition 22, gives

$$\begin{aligned} l.h.s &= \int_{\Omega_{ine,1}^{1,vac,1}} \left(\phi_{ine}^{1,\alpha,1} + \widetilde{\phi}^{0-} \right) \Delta_{x^1} w dx^1 + \int_{\Omega_{ine,1}^{1,vac,2}} \left(\phi_{ine}^{1,\alpha,2} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{2-} \right) \Delta_{x^1} w dx^1 \\ &+ \int_{\Omega_{ine,1}^{1,vac,3}} \left(\phi_{ine}^{1,\alpha,3} + \widetilde{\phi}^{0+} + \widetilde{\phi}_{in}^{1+} + \widetilde{\phi}_{in}^{2+} \right) \Delta_{x^1} w dx^1 + \int_{\Omega_{ine,1}^{1,vac,4}} \left(\phi_{ine}^{1,\alpha,4} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{1-} \right) \Delta_{x^1} w dx^1 \end{aligned}$$

and

$$\begin{aligned} r.h.s &= \int_{\Gamma_{ine,1}^{1,vac,1}} \left(v_{ine}^{1,\alpha,1} + \widetilde{V}^{0-} \right) \nabla_{x^1} w \cdot \mathbf{n}^1 ds(x^1) + \int_{\Gamma_{ine,1}^{1,vac,2}} \left(v_{ine}^{1,\alpha,2} + \widetilde{V}^{0-} + \widetilde{v}_{in}^{2-} \right) \nabla_{x^1} w \cdot \mathbf{n}^1 ds(x^1) \\ &+ \int_{\Gamma_{ine,1}^{1,vac,3}} \left(v_{ine}^{1,\alpha,3} + \widetilde{V}^{0+} + \widetilde{v}_{in}^{1+} + \widetilde{v}_{in}^{2+} \right) \nabla_{x^1} w \cdot \mathbf{n}^1 ds(x^1) + \int_{\Gamma_{ine,1}^{1,vac,4}} \left(v_{ine}^{1,\alpha,4} + \widetilde{V}^{0-} + \widetilde{v}_{in}^{1-} \right) \nabla_{x^1} w \cdot \mathbf{n}^1 ds(x^1). \end{aligned}$$

It follows that these above equalities still hold if w is taken on the form of $\tau_\alpha v$, where $v \in C^\infty(\overline{\Omega_{ine,1}^{\infty,vac}}) \cap H^2(\overline{\Omega_{ine,1}^{\infty,vac}})$, $v = 0$ on $\Gamma_{ine,1,int}^{\infty,vac}$, $v = 0$ on $\Gamma_{ine,1,int}^{\infty,vac}$ and $\nabla_{x^1} v \cdot \mathbf{n}^1 = 0$ on $\Gamma_{ine,1,top}^{\infty,vac}$, $|v|$, $|\nabla_{x^1} v|$, and $|\Delta_{x^1} v|$ exponentially decrease to 0 when $|x_1^1| + |x_2^1| \rightarrow +\infty$, and τ_α is a smooth

truncation function with compact support $\Omega_{ine,1}^{1,vac}$. Then

$$\begin{aligned} l.h.s = & \int_{\Omega_{ine,1}^{\infty,vac,1}} \left(\phi_{ine}^{1,\alpha,1} + \overline{\phi}^{0-} \right) \chi_{\Omega_{ine,1}^{1,vac}} \Delta_{x^1} \tau_\alpha v \, dx^1 + \int_{\Omega_{ine,1}^{\infty,vac,2}} \left(\phi_{ine}^{1,\alpha,2} + \overline{\phi}^{0-} + \overline{\phi}_{in}^{2-} \right) \chi_{\Omega_{ine,1}^{1,vac}} \Delta_{x^1} \tau_\alpha v \, dx^1 \\ & + \int_{\Omega_{ine,1}^{\infty,vac,3}} \left(\phi_{ine}^{1,\alpha,3} + \overline{\phi}^{0+} + \overline{\phi}_{in}^{1+} + \overline{\phi}_{in}^{2+} \right) \chi_{\Omega_{ine,1}^{1,vac}} \Delta_{x^1} \tau_\alpha v \, dx^1 \\ & + \int_{\Omega_{ine,1}^{\infty,vac,4}} \left(\phi_{ine}^{1,\alpha,4} + \overline{\phi}^{0-} + \overline{\phi}_{in}^{1-} \right) \chi_{\Omega_{ine,1}^{1,vac}} \Delta_{x^1} \tau_\alpha v \, dx^1, \end{aligned}$$

and

$$\begin{aligned} r.h.s = & \int_{\Gamma_{ine,1}^{\infty,vac,1}} \left(v_{ine}^{1,\alpha,1} + \overline{V}^{0-} \right) \chi_{\Omega_{ine,1}^{1,vac}} \nabla_{x^1} \tau_\alpha v \cdot \mathbf{n}^1 \, ds(x^1) \\ & + \int_{\Gamma_{ine,1}^{\infty,vac,2}} \left(v_{ine}^{1,\alpha,2} + \overline{V}^{0-} + \overline{v}_{in}^{2-} \right) \chi_{\Omega_{ine,1}^{1,vac}} \nabla_{x^1} \tau_\alpha v \cdot \mathbf{n}^1 \, ds(x^1) \\ & + \int_{\Gamma_{ine,1}^{\infty,vac,3}} \left(v_{ine}^{1,\alpha,3} + \overline{V}^{0+} + \overline{v}_{in}^{1+} + \overline{v}_{in}^{2+} \right) \chi_{\Omega_{ine,1}^{1,vac}} \nabla_{x^1} \tau_\alpha v \cdot \mathbf{n}^1 \, ds(x^1) \\ & + \int_{\Gamma_{ine,1}^{\infty,vac,4}} \left(v_{ine}^{1,\alpha,4} + \overline{V}^{0-} + \overline{v}_{in}^{1-} \right) \chi_{\Omega_{ine,1}^{1,vac}} \nabla_{x^1} \tau_\alpha v \cdot \mathbf{n}^1 \, ds(x^1). \end{aligned}$$

Passing α to $+\infty$, by Assumption 5,

$$\begin{aligned} l.h.s = & \int_{\Omega_{ine,1}^{\infty,vac,1}} \left(\phi_{ine}^{1,1} + \overline{\phi}^{0-} \right) \Delta_{x^1} v \, dx^1 + \int_{\Omega_{ine,1}^{\infty,vac,2}} \left(\phi_{ine}^{1,2} + \overline{\phi}^{0-} + \overline{\phi}_{in}^{2-} \right) \Delta_{x^1} v \, dx^1 \\ & + \int_{\Omega_{ine,1}^{\infty,vac,3}} \left(\phi_{ine}^{1,3} + \overline{\phi}^{0+} + \overline{\phi}_{in}^{1+} + \overline{\phi}_{in}^{2+} \right) \Delta_{x^1} v \, dx^1 + \int_{\Omega_{ine,1}^{\infty,vac,4}} \left(\phi_{ine}^{1,4} + \overline{\phi}^{0-} + \overline{\phi}_{in}^{1-} \right) \Delta_{x^1} v \, dx^1, \end{aligned}$$

and

$$\begin{aligned} r.h.s = & \int_{\Gamma_{ine,1}^{\infty,vac,1}} \left(v_{ine}^{1,1} + \overline{V}^{0-} \right) \nabla_{x^1} v \cdot \mathbf{n}^1 \, ds(x^1) + \int_{\Gamma_{ine,1}^{\infty,vac,2}} \left(v_{ine}^{1,2} + \overline{V}^{0-} + \overline{v}_{in}^{2-} \right) \nabla_{x^1} v \cdot \mathbf{n}^1 \, ds(x^1) \\ & + \int_{\Gamma_{ine,1}^{\infty,vac,3}} \left(v_{ine}^{1,3} + \overline{V}^{0+} + \overline{v}_{in}^{1+} + \overline{v}_{in}^{2+} \right) \nabla_{x^1} v \cdot \mathbf{n}^1 \, ds(x^1) + \int_{\Gamma_{ine,1}^{\infty,vac,4}} \left(v_{ine}^{1,4} + \overline{V}^{0-} + \overline{v}_{in}^{1-} \right) \nabla_{x^1} v \cdot \mathbf{n}^1 \, ds(x^1). \end{aligned}$$

Now, we choose v vanishing out of $\Omega_{ine,1}^{1,vac}$ for a given α ,

$$\begin{aligned} l.h.s = & \int_{\Omega_{ine,1}^{1,vac,1}} \left(\phi_{ine}^{1,1} + \overline{\phi}^{0-} \right) \Delta_{x^1} v \, dx^1 + \int_{\Omega_{ine,1}^{1,vac,2}} \left(\phi_{ine}^{1,2} + \overline{\phi}^{0-} + \overline{\phi}_{in}^{2-} \right) \Delta_{x^1} v \, dx^1 \\ & + \int_{\Omega_{ine,1}^{1,vac,3}} \left(\phi_{ine}^{1,3} + \overline{\phi}^{0+} + \overline{\phi}_{in}^{1+} + \overline{\phi}_{in}^{2+} \right) \Delta_{x^1} v \, dx^1 + \int_{\Omega_{ine,1}^{1,vac,4}} \left(\phi_{ine}^{1,4} + \overline{\phi}^{0-} + \overline{\phi}_{in}^{1-} \right) \Delta_{x^1} v \, dx^1 \end{aligned}$$

that we note

$$= T_1 + T_2 + T_3 + T_4,$$

and

$$\begin{aligned} r.h.s = & \int_{\Gamma_{ine,1}^{1,vac,1}} \left(v_{ine}^{1,1} + \widetilde{V}^{0-} \right) \nabla_{x^1} v \cdot \mathbf{n}^1 ds(x^1) + \int_{\Gamma_{ine,1}^{1,vac,2}} \left(v_{ine}^{1,2} + \widetilde{V}^{0-} + \widetilde{v}_{in}^{2-} \right) \nabla_{x^1} v \cdot \mathbf{n}^1 ds(x^1) \\ & + \int_{\Gamma_{ine,1}^{1,vac,3}} \left(v_{ine}^{1,3} + \widetilde{V}^{0+} + \widetilde{v}_{in}^{1+} + \widetilde{v}_{in}^{2+} \right) \nabla_{x^1} v \cdot \mathbf{n}^1 ds(x^1) + \int_{\Gamma_{ine,1}^{1,vac,4}} \left(v_{ine}^{1,4} + \widetilde{V}^{0-} + \widetilde{v}_{in}^{1-} \right) \nabla_{x^1} v \cdot \mathbf{n}^1 ds(x^1). \end{aligned}$$

Applying Green's formula twice to each term T_i yields,

$$\begin{aligned} T_1 &= \int_{\Omega_{ine,1}^{1,vac,1}} \left(\phi_{ine}^{1,1} + \widetilde{\phi}^{0-} \right) \Delta_{x^1} v dx^1 \\ &= \int_{\Omega_{ine,1}^{1,vac,1}} \Delta_{x^1} \left(\phi_{ine}^{1,1} + \widetilde{\phi}^{0-} \right) v dx^1 + \int_{\partial\Omega_{ine,1}^{1,vac,1}} \left(\phi_{ine}^{1,1} + \widetilde{\phi}^{0-} \right) \nabla_{x^1} v \cdot \mathbf{n}^{1,1} ds(x^1) \\ &\quad - \int_{\partial\Omega_{ine,1}^{1,vac,1}} v \nabla_{x^1} \left(\phi_{ine}^{1,1} + \widetilde{\phi}^{0-} \right) \cdot \mathbf{n}^{1,1} ds(x^1), \end{aligned}$$

$$\begin{aligned} T_2 &= \int_{\Omega_{ine,1}^{1,vac,2}} \left(\phi_{ine}^{1,2} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{2-} \right) \Delta_{x^1} v dx^1 \\ &= \int_{\Omega_{ine,1}^{1,vac,2}} \Delta_{x^1} \left(\phi_{ine}^{1,2} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{2-} \right) v dx^1 + \int_{\partial\Omega_{ine,1}^{1,vac,2}} \left(\phi_{ine}^{1,2} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{2-} \right) \nabla_{x^1} v \cdot \mathbf{n}^{1,2} ds(x^1) \\ &\quad - \int_{\partial\Omega_{ine,1}^{1,vac,2}} v \nabla_{x^1} \left(\phi_{ine}^{1,2} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{2-} \right) \cdot \mathbf{n}^{1,2} ds(x^1), \end{aligned}$$

$$\begin{aligned} T_3 &= \int_{\Omega_{ine,1}^{1,vac,3}} \left(\phi_{ine}^{1,3} + \widetilde{\phi}^{0+} + \widetilde{\phi}_{in}^{1+} + \widetilde{\phi}_{in}^{2+} \right) \Delta_{x^1} v dx^1 \\ &= \int_{\Omega_{ine,1}^{1,vac,3}} \Delta_{x^1} \left(\phi_{ine}^{1,3} + \widetilde{\phi}^{0+} + \widetilde{\phi}_{in}^{1+} + \widetilde{\phi}_{in}^{2+} \right) v dx^1 + \int_{\partial\Omega_{ine,1}^{1,vac,3}} \left(\phi_{ine}^{1,3} + \widetilde{\phi}^{0+} + \widetilde{\phi}_{in}^{1+} + \widetilde{\phi}_{in}^{2+} \right) \nabla_{x^1} v \cdot \mathbf{n}^{1,3} ds(x^1) \\ &\quad - \int_{\partial\Omega_{ine,1}^{1,vac,3}} v \nabla_{x^1} \left(\phi_{ine}^{1,3} + \widetilde{\phi}^{0+} + \widetilde{\phi}_{in}^{1+} + \widetilde{\phi}_{in}^{2+} \right) \cdot \mathbf{n}^{1,3} ds(x^1), \end{aligned}$$

and

$$\begin{aligned} T_4 &= \int_{\Omega_{ine,1}^{1,vac,4}} \left(\phi_{ine}^{1,4} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{1-} \right) \Delta_{x^1} v dx^1 \\ &= \int_{\Omega_{ine,1}^{1,vac,4}} \Delta_{x^1} \left(\phi_{ine}^{1,4} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{1-} \right) v dx^1 + \int_{\partial\Omega_{ine,1}^{1,vac,4}} \left(\phi_{ine}^{1,4} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{1-} \right) \nabla_{x^1} v \cdot \mathbf{n}^{1,4} ds(x^1) \\ &\quad - \int_{\partial\Omega_{ine,1}^{1,vac,4}} v \nabla_{x^1} \left(\phi_{ine}^{1,4} + \widetilde{\phi}^{0-} + \widetilde{\phi}_{in}^{1-} \right) \cdot \mathbf{n}^{1,4} ds(x^1). \end{aligned}$$

Decomposing each $\partial\Omega_{ine,1}^{1,vac,i} = \Gamma_{ine,1,int}^{1,vac,i} \cup \Gamma_{ine,1,top}^{1,vac,i} \cup \Gamma_{ine,1,\alpha}^{1,vac,i} \cup \Gamma_{ine,1,interf,i}^{1,vac} \cup \Gamma_{ine,1,interf,i+1}^{1,vac}$ for $i = 1, 2, 3, 4$ and combining with the conditions satisfied by v , with the results from Proposition 9 and with Proposition 20 it follows that $\Delta_{x^1} \widetilde{\phi}^{0\pm} = 0$ in $\Omega_{ine,1}^{1,vac}$, $\Delta_{x^1} \widetilde{\phi}_{in}^{1+} = 0$ in $\Omega_{ine,1}^{1,vac,3}$, $\Delta_{x^1} \widetilde{\phi}_{in}^{1-} =$

0 in $\Omega_{ine,1}^{1,vac,4}$, $\Delta_{x^1} \widetilde{\phi}_{in}^{2+} = 0$ in $\Omega_{ine,1}^{1,vac,3}$, $\Delta_{x^1} \widetilde{\phi}_{in}^{2-} = 0$ in $\Omega_{ine,1}^{1,vac,2}$, $\nabla_{x^1} \widetilde{\phi}^0 \cdot \mathbf{n}^1 = 0$ on $\Gamma_{ine,1,top}^{1,vac}$, $\nabla_{x^1} \widetilde{\phi}_{in}^{1+} \cdot \mathbf{n}^1 = 0$ on $\Gamma_{ine,1,top}^{1,vac,3}$, $\nabla_{x^1} \widetilde{\phi}_{in}^{1-} \cdot \mathbf{n}^1 = 0$ on $\Gamma_{ine,1,top}^{1,vac,4}$, $\nabla_{x^1} \widetilde{\phi}_{in}^{2+} \cdot \mathbf{n}^1 = 0$ on $\Gamma_{ine,1,top}^{1,vac,3}$, $\nabla_{x^1} \widetilde{\phi}_{in}^{2-} \cdot \mathbf{n}^1 = 0$ on $\Gamma_{ine,1,top}^{1,vac,2}$, $\left[\left[\widetilde{\phi}^0 + \widetilde{\phi}_{in}^1 \right] \right] = \left[\left[\nabla_{x^1} \widetilde{\phi}^0 + \nabla_{x^1} \widetilde{\phi}_{in}^1 \right] \right] \cdot \mathbf{n}^{1,3} = 0$ on $\Gamma_{ine,1,interf,4}^{1,vac}$, $\left[\left[\widetilde{\phi}^0 + \widetilde{\phi}_{in}^2 \right] \right] = \left[\left[\nabla_{x^1} \widetilde{\phi}^0 + \nabla_{x^1} \widetilde{\phi}_{in}^2 \right] \right] \cdot \mathbf{n}^{1,3} = 0$ on $\Gamma_{ine,1,interf,3}^{1,vac}$, thus we get

$$\begin{aligned}
& \sum_{i=1}^4 \int_{\Omega_{ine,1}^{1,vac,i}} \Delta_{x^1} (\phi_{ine}^{1,i}) v \, dx^1 - \sum_{i=1}^4 \int_{\Gamma_{ine,1,top}^{1,vac,i}} v \nabla_{x^1} \phi_{ine}^{1,i} \cdot \mathbf{n}^{1,i} \, ds(x^1) \\
& + \int_{\Gamma_{ine,1,interf,1}^{1,vac}} (\phi_{ine}^{1,1} - \phi_{ine}^{1,4} - \widetilde{\phi}_{in}^{1-}) \nabla_{x^1} v \cdot \mathbf{n}^{1,1} \, ds(x^1) - \int_{\Gamma_{ine,1,interf,1}^{1,vac}} v \left[\nabla_{x^1} (\phi_{ine}^{1,1} - \phi_{ine}^{1,4}) - \nabla_{x^1} \widetilde{\phi}_{in}^{1-} \right] \cdot \mathbf{n}^{1,1} \, ds(x^1) \\
& + \int_{\Gamma_{ine,1,interf,2}^{1,vac}} (\phi_{ine}^{1,1} - \phi_{ine}^{1,2} - \widetilde{\phi}_{in}^{2-}) \nabla_{x^1} v \cdot \mathbf{n}^{1,1} \, ds(x^1) - \int_{\Gamma_{ine,1,interf,2}^{1,vac}} v \left[\nabla_{x^1} (\phi_{ine}^{1,1} - \phi_{ine}^{1,2}) - \nabla_{x^1} \widetilde{\phi}_{in}^{2-} \right] \cdot \mathbf{n}^{1,1} \, ds(x^1) \\
& + \int_{\Gamma_{ine,1,interf,3}^{1,vac}} (\phi_{ine}^{1,3} - \phi_{ine}^{1,2} + \widetilde{\phi}_{in}^{1+}) \nabla_{x^1} v \cdot \mathbf{n}^{1,3} \, ds(x^1) - \int_{\Gamma_{ine,1,interf,3}^{1,vac}} v \left[\nabla_{x^1} (\phi_{ine}^{1,3} - \phi_{ine}^{1,2}) + \nabla_{x^1} \widetilde{\phi}_{in}^{1+} \right] \cdot \mathbf{n}^{1,3} \, ds(x^1) \\
& + \int_{\Gamma_{ine,1,interf,4}^{1,vac}} (\phi_{ine}^{1,3} - \phi_{ine}^{1,4} + \widetilde{\phi}_{in}^{2+}) \nabla_{x^1} v \cdot \mathbf{n}^{1,3} \, ds(x^1) - \int_{\Gamma_{ine,1,interf,4}^{1,vac}} v \left[\nabla_{x^1} (\phi_{ine}^{1,3} - \phi_{ine}^{1,4}) + \nabla_{x^1} \widetilde{\phi}_{in}^{2+} \right] \cdot \mathbf{n}^{1,3} \, ds(x^1) \\
& + \sum_{i=1}^4 \int_{\Gamma_{ine,1,int}^{1,vac,i}} \phi_{ine}^{1,i} \nabla_{x^1} v \cdot \mathbf{n}^{1,i} \, ds(x^1) = \sum_{i=1}^4 \int_{\Gamma_{ine,1,int}^{1,vac,i}} v_{ine}^{1,i} \nabla_{x^1} v \cdot \mathbf{n}^{1,i} \, ds(x^1)
\end{aligned}$$

The rest of the proof runs similarly as the proofs of the previous models. \square

2.7/ LINEAR ELASTICITY MODELS

Let us move to the linear elasticity part. In this part, we use the superscripts ε , \sharp , and 1 on derivation operators related to the variables x^ε , x^\sharp and x^1 , e.g. div^ε and σ^ε represent div and σ with respect to the variable x^ε .

We now recall the system 2.4.2 of linear elasticity,

$$\begin{cases} -div^\varepsilon \sigma^\varepsilon(\mathbf{u}^\varepsilon) & = \varepsilon^{-2} \mathbf{f}^\varepsilon \text{ in } \Omega^{\varepsilon,mec} \\ \mathbf{u}^\varepsilon & = \mathbf{0} \text{ on } \Gamma_0^{\varepsilon,mec} \\ \sigma^\varepsilon(\mathbf{u}^\varepsilon) \mathbf{n}^\varepsilon & = \varepsilon^{-1} \mathbf{g}^\varepsilon \text{ on } \Gamma_1^{\varepsilon,mec}. \end{cases} \quad (2.7.1)$$

We recall that there exists a unique $\mathbf{u}^\varepsilon \in \mathcal{H}_{\Gamma_0^{\varepsilon,mec},0}^1(\Omega^{\varepsilon,mec}) \doteq \{\mathbf{v}^\varepsilon = (v_i^\varepsilon) : v_i^\varepsilon \in H^1(\Omega^{\varepsilon,mec}), v_i^\varepsilon = 0 \text{ on } \Gamma_0^{\varepsilon,mec}, i = 1, 2, 3\}$ such that

$$\int_{\Omega^{\varepsilon,mec}} \sigma_{ij}^\varepsilon(\mathbf{u}^\varepsilon) \partial_j^\varepsilon v_i^\varepsilon \, dx^\varepsilon = \frac{1}{\varepsilon^2} \int_{\Omega^{\varepsilon,mec}} f_i^\varepsilon v_i^\varepsilon \, dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_1^{\varepsilon,mec}} g_i^\varepsilon v_i^\varepsilon \, ds(x^\varepsilon), \quad (2.7.2)$$

for all $\mathbf{v}^\varepsilon \in \mathcal{H}_{\Gamma_0^{\varepsilon,mec},0}^1(\Omega^{\varepsilon,mec})$, see [35].

We now develop the stress expression in (2.7.2),

$$\int_{\Omega^{\varepsilon,mec}} (\lambda e_{pp}^\varepsilon(\mathbf{u}^\varepsilon) e_{qq}^\varepsilon(\mathbf{v}^\varepsilon) + 2\mu e_{ij}^\varepsilon(\mathbf{u}^\varepsilon) e_{ij}^\varepsilon(\mathbf{v}^\varepsilon)) dx^\varepsilon = \frac{1}{\varepsilon^2} \int_{\Omega^{\varepsilon,mec}} f_i^\varepsilon v_i^\varepsilon dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_1^{\varepsilon,mec}} g_i^\varepsilon v_i^\varepsilon ds(x^\varepsilon).$$

But $e_{pp}^\varepsilon(\mathbf{u}^\varepsilon) = \delta_{ij} \partial_j^\varepsilon u_i^\varepsilon$ and $e_{ij}^\varepsilon(\mathbf{u}^\varepsilon) e_{ij}^\varepsilon(\mathbf{v}^\varepsilon) = \partial_j^\varepsilon (u_i^\varepsilon) e_{ij}^\varepsilon(\mathbf{v}^\varepsilon)$ implies

$$\int_{\Omega^{\varepsilon,mec}} \partial_j^\varepsilon (u_i^\varepsilon) (\lambda \delta_{ij} e_{qq}^\varepsilon(\mathbf{v}^\varepsilon) + 2\mu e_{ij}^\varepsilon(\mathbf{v}^\varepsilon)) dx^\varepsilon = \frac{1}{\varepsilon^2} \int_{\Omega^{\varepsilon,mec}} f_i^\varepsilon v_i^\varepsilon dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_1^{\varepsilon,mec}} g_i^\varepsilon v_i^\varepsilon ds(x^\varepsilon).$$

It could be read under the following equivalent form

$$\int_{\Omega^{\varepsilon,mec}} \partial_j^\varepsilon (u_i^\varepsilon) \sigma_{ij}^\varepsilon(\mathbf{v}^\varepsilon) dx^\varepsilon = \frac{1}{\varepsilon^2} \int_{\Omega^{\varepsilon,mec}} f_i^\varepsilon v_i^\varepsilon dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_1^{\varepsilon,mec}} g_i^\varepsilon v_i^\varepsilon ds(x^\varepsilon).$$

Applying the Green's formula, we obtain a very weak form of Problem 2.7.1,

$$\begin{aligned} & - \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon \partial_j^\varepsilon \sigma_{ij}^\varepsilon(\mathbf{v}^\varepsilon) dx^\varepsilon + \int_{\Gamma_1^{\varepsilon,mec}} u_i^\varepsilon \sigma_{ij}^\varepsilon(\mathbf{v}^\varepsilon) n_j^\varepsilon ds(x^\varepsilon) \\ & = \frac{1}{\varepsilon^2} \int_{\Omega^{\varepsilon,mec}} f_i^\varepsilon v_i^\varepsilon dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_1^{\varepsilon,mec}} g_i^\varepsilon v_i^\varepsilon ds(x^\varepsilon), \end{aligned} \quad (2.7.3)$$

for all $\mathbf{v}^\varepsilon \in \mathcal{H}_{\Gamma_0^{\varepsilon,mec},0}^2(\Omega^{\varepsilon,mec}) \doteq \{\mathbf{v}^\varepsilon = (v_i^\varepsilon) : v_i^\varepsilon \in H^2(\Omega^{\varepsilon,mec}), v_i^\varepsilon = 0 \text{ on } \Gamma_0^{\varepsilon,mec}, i = 1, 2, 3\}$.

Remark 6:

In this part, we do not redefine two-scale transform operators of vector valued functions, instead we use a bold notation such as \mathbf{T}^ε with $\mathbf{T}^\varepsilon \mathbf{v} = (T^\varepsilon v_1, T^\varepsilon v_2, T^\varepsilon v_3)$ built from the two-scale transform operators T^ε defined for scalar functions in the previous sections. Similarly, bold notations are use for function spaces to indicate vector valued function spaces, e.g. $\mathbf{L}^2(\Omega^{\varepsilon,mec}) = [L^2(\Omega^{\varepsilon,mec})]^3$ and $\mathbf{L}^2(\Gamma_1^{\varepsilon,mec}) = [L^2(\Gamma_1^{\varepsilon,mec})]^3$.

Assumption 6:

Let \mathbf{u}^ε be a solution of (2.7.2), and assume that the norms $\|\mathbf{u}^\varepsilon\|_{\mathbf{L}^2(\Omega^{\varepsilon,mec})}$, $\varepsilon \|\nabla^\varepsilon \mathbf{u}^\varepsilon\|_{\mathbf{L}^2(\Omega^{\varepsilon,mec})}$, $\|\mathbf{f}^\varepsilon\|_{\mathbf{L}^2(\Omega^{\varepsilon,mec})}$ and $\|\mathbf{g}^\varepsilon\|_{\mathbf{L}^2(\Gamma_1^{\varepsilon,mec})}$ are bounded uniformly with respect to ε .

2.7.1/ PERIODIC MODEL

Proposition 24:

Let \mathbf{u}^ε be a solution of (2.7.2) and satisfies Assumption 6, then there exist $\mathbf{u}^0 = (u_1^0, u_2^0, u_3^0)$ in $\mathbf{L}^2(\Omega^\sharp, \mathbf{H}^1(\Omega^{1,mec})), \Omega^{1,mec}$ - periodic in the directions x_1^1 and x_2^1 , $\mathbf{f}^0 = (f_1^0, f_2^0, f_3^0)$ in $\mathbf{L}^2(\Omega^\sharp \times \Omega^{1,mec})$, and $\mathbf{g}^0 = (g_1^0, g_2^0, g_3^0)$ in $\mathbf{L}^2(\Omega^\sharp \times \Gamma_1^{1,mec})$, such that $\mathbf{T}^\varepsilon \mathbf{u}^\varepsilon \rightharpoonup \mathbf{u}^0$, $\mathbf{T}^\varepsilon(\mathbf{f}^\varepsilon) \rightharpoonup \mathbf{f}^0$ weakly in $\mathbf{L}^2(\Omega^\sharp \times \Omega^{1,mec})$, and $\mathbf{T}^\varepsilon \mathbf{g}^\varepsilon \rightharpoonup \mathbf{g}^0$ weakly in $\mathbf{L}^2(\Omega^\sharp \times \Gamma_1^{1,mec})$. Moreover, for a.e. $x^\sharp \in \Omega^\sharp$, \mathbf{u}^0 is a solution to

$$\left\{ \begin{array}{ll} -\operatorname{div}^1 \sigma^1(\mathbf{u}^0) = \mathbf{f}^0 & \text{in } \Omega^{1,mec} \\ \mathbf{u}^0 = \mathbf{0} & \text{on } \Gamma_0^{1,mec} \\ \sigma^1(\mathbf{u}^0) \mathbf{n}^1 = \mathbf{g}^0 & \text{on } \Gamma_1^{1,mec} \\ \sigma^1(\mathbf{u}^0) \mathbf{n}^1 \text{ is } & \Gamma_{per}^{1,mec} \text{- antiperiodic} \\ \mathbf{u}^0 \text{ is } & \Gamma_{per}^{1,mec} \text{- periodic.} \end{array} \right.$$

Proof. By Proposition 3, we obtain the existence and the periodicity of \mathbf{u}^0 . The proof is completed by showing that \mathbf{u}^0 satisfies the above equation. We now replace \mathbf{v}^ε by $\varepsilon \mathbf{B}^\varepsilon(\mathbf{w}) = (\varepsilon B^\varepsilon w_1, \varepsilon B^\varepsilon w_2, \varepsilon B^\varepsilon w_3)$ in (2.7.3), where $\mathbf{w} \in \mathbf{C}^\infty(\overline{\Omega^\sharp} \times \overline{\Omega^{1,mec}})$, $\Omega^{1,mec}$ - periodic in the directions x_1^1 and x_2^1 , $\mathbf{w} = \mathbf{0}$ on $\Gamma_0^{1,mec}$ and $\sigma^1(\mathbf{w}) \mathbf{n}^1 = 0$ on $\Gamma_1^{1,mec} \cup \Gamma_{per}^{1,mec}$. Then, we can assert that

$$\begin{aligned} & -\varepsilon \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon \partial_j^\varepsilon \sigma_{ij}^\varepsilon(\mathbf{B}^\varepsilon \mathbf{w}) dx^\varepsilon + \varepsilon \int_{\Gamma_1^{\varepsilon,mec}} u_i^\varepsilon \sigma_{ij}^\varepsilon(\mathbf{B}^\varepsilon \mathbf{w}) n_j^\varepsilon ds(x^\varepsilon) \\ & = \frac{1}{\varepsilon} \int_{\Omega^{\varepsilon,mec}} f_i^\varepsilon B^\varepsilon w_i dx^\varepsilon + \int_{\Gamma_1^{\varepsilon,mec}} g_i^\varepsilon B^\varepsilon w_i ds(x^\varepsilon). \end{aligned} \quad (2.7.4)$$

We check that

$$\partial_j^\varepsilon(\mathbf{B}^\varepsilon \mathbf{w}) = \mathbf{B}^\varepsilon(\partial_j^\sharp \mathbf{w}) \chi_{I^\sharp}(j) + \frac{1}{\varepsilon} \mathbf{B}^\varepsilon(\partial_j^1 \mathbf{w}) \text{ and } e^\varepsilon(\mathbf{B}^\varepsilon \mathbf{w}) = \mathbf{B}^\varepsilon(e^\sharp(\mathbf{w})) + \frac{1}{\varepsilon} \mathbf{B}^\varepsilon(e^1(\mathbf{w})),$$

where

$$e_{ij}^\sharp(\mathbf{w}) = \frac{1}{2} (\partial_i^\sharp w_j \chi_{I^\sharp}(i) + \partial_j^\sharp w_i \chi_{I^\sharp}(j)) \text{ and } e_{ij}^1(\mathbf{w}) = \frac{1}{2} (\partial_i^1 w_j + \partial_j^1 w_i) \text{ with } I^\sharp = \{1, 2\},$$

and we denote

$$\sigma_{ij}^\sharp(\mathbf{w}) = \lambda e_{pp}^\sharp(\mathbf{w}) \delta_{ij} + 2\mu e_{ij}^\sharp(\mathbf{w}) \text{ and } \sigma_{ij}^1(\mathbf{w}) = \lambda e_{pp}^1(\mathbf{w}) \delta_{ij} + 2\mu e_{ij}^1(\mathbf{w}).$$

And it is clear that $B^\varepsilon(\sigma^1(\mathbf{w})) \mathbf{n}^\varepsilon = \mathbf{0}$ on $\Gamma_1^{\varepsilon,mec}$. Then, a direct computation shows that *l.h.s*

of (2.7.4) becomes,

$$\begin{aligned} l.h.s &= -\varepsilon \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon \partial_j^\varepsilon \left(B^\varepsilon(\sigma_{ij}^\sharp(\mathbf{w})) + \frac{1}{\varepsilon} B^\varepsilon(\sigma_{ij}^1(\mathbf{w})) \right) dx^\varepsilon \\ &+ \varepsilon \int_{\Gamma_1^{\varepsilon,mec}} u_i^\varepsilon \left(B^\varepsilon(\sigma_{ij}^\sharp(\mathbf{w})) + \frac{1}{\varepsilon} B^\varepsilon(\sigma_{ij}^1(\mathbf{w})) \right) n_j^\varepsilon ds(x^\varepsilon) \\ &= -\frac{1}{\varepsilon} \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon B^\varepsilon(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon + O(\varepsilon), \end{aligned}$$

where

$$\begin{aligned} O(\varepsilon) &= -\varepsilon \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon B^\varepsilon(\partial_j^\sharp \sigma_{ij}^\sharp(\mathbf{w}) \chi_{I^\sharp}(j)) dx^\varepsilon - \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon B^\varepsilon(\partial_j^1 \sigma_{ij}^\sharp(\mathbf{w})) dx^\varepsilon \\ &- \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon B^\varepsilon(\partial_j^\sharp \sigma_{ij}^1(\mathbf{w}) \chi_{I^\sharp}(j)) dx^\varepsilon + \varepsilon \int_{\Gamma_1^{\varepsilon,mec}} u_i^\varepsilon B^\varepsilon(\sigma_{ij}^\sharp(\mathbf{w})) n_j^\varepsilon \chi_{I^\sharp}(j) ds(x^\varepsilon). \end{aligned}$$

Thus (2.7.4) changes into

$$-\frac{1}{\varepsilon} \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon B^\varepsilon(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon + O(\varepsilon) = \frac{1}{\varepsilon} \int_{\Omega^{\varepsilon,mec}} f_i^\varepsilon B^\varepsilon(w_i) dx^\varepsilon + \int_{\Gamma_1^{\varepsilon,mec}} g_i^\varepsilon B^\varepsilon(w_i) ds(x^\varepsilon).$$

Approximating B^ε by $T^{\varepsilon*}$, we obtain

$$-\frac{1}{\varepsilon} \int_{\Omega^{\varepsilon,mec}} u_i^\varepsilon T^{\varepsilon*}(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon + O(\varepsilon) = \frac{1}{\varepsilon} \int_{\Omega^{\varepsilon,mec}} f_i^\varepsilon T^{\varepsilon*}(w_i) dx^\varepsilon + \int_{\Gamma_1^{\varepsilon,mec}} g_i^\varepsilon T^{\varepsilon*}(w_i) ds(x^\varepsilon).$$

Using the adjoint operator definition gives

$$\begin{aligned} - \int_{\Omega^\sharp \times \Omega^{1,mec}} T^\varepsilon(u_i^\varepsilon) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^\sharp dx^1 &= \int_{\Omega^\sharp \times \Omega^{1,mec}} T^\varepsilon(f_i^\varepsilon) w_i dx^\sharp dx^1 \\ &+ \int_{\Omega^\sharp \times \Gamma_1^{1,mec}} T^\varepsilon(g_i^\varepsilon) w_i dx^\sharp ds(x^1) + O(\varepsilon). \end{aligned}$$

Passing ε to 0 and by Assumption 6, then we get

$$- \int_{\Omega^\sharp \times \Omega^{1,mec}} u_i^0 \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^\sharp dx^1 = \int_{\Omega^\sharp \times \Omega^{1,mec}} f_i^0 w_i dx^\sharp dx^1 + \int_{\Omega^\sharp \times \Gamma_1^{1,mec}} g_i^0 w_i dx^\sharp ds(x^1).$$

Applying the Green's formula twice for l.h.s,

$$\begin{aligned} &- \int_{\Omega^\sharp \times \Omega^{1,mec}} \partial_j^1(\sigma_{ij}^1(\mathbf{u}^0)) w_i dx^\sharp dx^1 + \int_{\Omega^\sharp \times \partial\Omega^{1,mec}} \sigma_{ij}^1(\mathbf{u}^0) n_j^1 w_i dx^\sharp ds(x^1) - \int_{\Omega^\sharp \times \partial\Omega^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^1 u_i^0 dx^\sharp ds(x^1) \\ &= \int_{\Omega^\sharp \times \Omega^{1,mec}} f_i^0 w_i dx^\sharp dx^1 + \int_{\Omega^\sharp \times \Gamma_1^{1,mec}} g_i^0 w_i dx^\sharp ds(x^1). \end{aligned}$$

Decomposing $\partial\Omega^{1,mec}$ into $\Gamma_0^{1,mec} \cup \Gamma_1^{1,mec} \cup \Gamma_{per}^{1,mec}$ and combining with conditions satisfied w

yields

$$\begin{aligned} & - \int_{\Omega^\# \times \Omega^{1,mec}} \partial_j^1 (\sigma_{ij}^1(\mathbf{u}^0)) w_i \, dx^\# dx^1 + \int_{\Omega^\# \times (\Gamma_{per}^{1,mec} \cup \Gamma_1^{1,mec})} \sigma_{ij}^1(\mathbf{u}^0) n_j^1 w_i \, dx^\# ds(x^1) \\ & - \int_{\Omega^\# \times \Gamma_0^1} \sigma_{ij}^1(\mathbf{w}) n_j^1 u_i^0 \, dx^\# ds(x^1) = \int_{\Omega^\# \times \Omega^{1,mec}} f_i^0 w_i \, dx^\# dx^1 + \int_{\Omega^\# \times \Gamma_1^{1,mec}} g_i^0 w_i \, dx^\# ds(x^1). \end{aligned} \quad (2.7.5)$$

i) If we choose $\mathbf{w} = \mathbf{0}$ on $\Gamma_{per}^{1,mec} \cup \Gamma_1^{1,mec}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\Gamma_0^{1,mec}$, then

$$- \int_{\Omega^\# \times \Omega^1} \partial_j^1 (\sigma_{ij}^1(\mathbf{u}^0)) w_i \, dx^\# dx^1 = \int_{\Omega^\# \times \Omega^1} f_i^0 w_i \, dx^\# dx^1.$$

We can deduce that

$$-\partial_j^1 \sigma_{ij}^1(\mathbf{u}^0) = f_i^0 \text{ in } \Omega^{1,mec}.$$

Then (2.7.5) changes into

$$\int_{\Omega^\# \times (\Gamma_{per}^{1,mec} \cup \Gamma_1^{1,mec})} \sigma_{ij}^1(\mathbf{u}^0) n_j^1 w_i \, dx^\# ds(x^1) + \int_{\Omega^\# \times \Gamma_0^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^1 u_i^0 \, dx^\# ds(x^1) = \int_{\Omega^\# \times \Gamma_1^{1,mec}} g_i^0 w_i \, dx^\# ds(x^1).$$

ii) If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{per}^{1,mec}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\Gamma_0^{1,mec}$

$$\int_{\Omega^\# \times \Gamma_1^{1,mec}} \sigma_{ij}^1(\mathbf{u}^0) n_j^1 w_i \, dx^\# ds(x^1) = \int_{\Omega^\# \times \Gamma_1^{1,mec}} g_i^0 w_i \, dx^\# ds(x^1).$$

We can assert that

$$\sigma_{ij}^1(\mathbf{u}^0) n_j^1 = g_i^0 \text{ on } \Gamma_1^{1,mec}.$$

Hence,

$$\int_{\Omega^\# \times \Gamma_{per}^{1,mec}} \sigma_{ij}^1(\mathbf{u}^0) n_j^1 w_i \, dx^\# ds(x^1) - \int_{\Omega^\# \times \Gamma_0^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^1 u_i^0 \, dx^\# ds(x^1) = 0.$$

iii) If $\mathbf{w} = \mathbf{0}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\Gamma_{per}^{1,mec}$, then

$$- \int_{\Omega^\# \times \Gamma_0^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^1 u_i^0 \, dx^\# ds(x^1) = 0.$$

we obtain

$$\mathbf{u} = \mathbf{0} \text{ on } \Gamma_0^{1,mec}.$$

Lastly, we have

$$\int_{\Omega^\# \times \Gamma_{per}^1} \sigma_{ij}^1(\mathbf{u}^0) n_j^1 w_i \, dx^\# ds(x^1) = 0.$$

It leads

$$\sigma_{ij}^1(\mathbf{u}^0) n_j^1 \text{ is } \Gamma_{per}^{1,mec} \text{- antiperiodic.}$$

□

2.7.2/ LATERAL BOUNDARY MODEL

We now investigate the first lateral boundary model by introducing some terms $\mathbf{u}_{bl}^\varepsilon = \mathbf{u}^\varepsilon - \mathbf{B}^\varepsilon \mathbf{u}^0$, $\mathbf{f}_{bl}^\varepsilon = \mathbf{f}^\varepsilon - \mathbf{B}^\varepsilon \mathbf{f}^0$, $\mathbf{g}_{bl}^\varepsilon = \mathbf{g}^\varepsilon - \mathbf{B}^\varepsilon \mathbf{g}^0$, and make the following assumptions.

Assumption 7:

We assume that:

1. For each α , there exist $\mathbf{u}_{bl}^{1,\alpha} = (u_{bl,1}^{1,\alpha}, u_{bl,2}^{1,\alpha}, u_{bl,3}^{1,\alpha}) \in \mathbf{L}^2(\Omega_{bl,1}^\#, \mathbf{H}^1(\Omega_{bl,1}^{1,mec}))$, $\Omega_{bl,1}^{1,mec}$ -periodic in the direction x_1^1 , $\mathbf{f}_{bl}^{1,\alpha} = (f_{bl,1}^{1,\alpha}, f_{bl,2}^{1,\alpha}, f_{bl,3}^{1,\alpha}) \in \mathbf{L}^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec})$, and $\mathbf{g}_{bl}^{1,\alpha} = (g_{bl,1}^{1,\alpha}, g_{bl,2}^{1,\alpha}, g_{bl,3}^{1,\alpha}) \in \mathbf{L}^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{1,mec})$ such that $\mathbf{T}_{bl,1}^\varepsilon \mathbf{u}_{bl}^\varepsilon \rightharpoonup \mathbf{u}_{bl}^{1,\alpha}$, $\mathbf{T}_{bl,1}^\varepsilon \mathbf{f}_{bl}^\varepsilon \rightharpoonup \mathbf{f}_{bl}^{1,\alpha}$ weakly in $\mathbf{L}^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec})$, and $\mathbf{T}_{bl,1}^\varepsilon (\mathbf{g}_{bl}^\varepsilon) \rightharpoonup \mathbf{g}_{bl}^{1,\alpha}$ weakly in $\mathbf{L}^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{1,mec})$ when $\varepsilon \rightarrow 0$.
2. There exist $\mathbf{u}_{bl}^1 = (u_{bl,1}^1, u_{bl,2}^1, u_{bl,3}^1) \in \mathbf{L}^2(\Omega_{bl,1}^\#, \mathbf{H}^1(\Omega_{bl,1}^{\infty,mec}))$, $\Omega_{bl,1}^{\infty,mec}$ -periodic in the direction x_1^1 , $u_{bl,i}^1$ and $\sigma_{ij}^1(\mathbf{u}_{bl}^1)$ converges exponentially fast to 0 when $x_2^1 \rightarrow +\infty$ for all $i, j \in \{1, 2, 3\}$, $\mathbf{f}_{bl}^1 = (f_{bl,1}^1, f_{bl,2}^1, f_{bl,3}^1) \in \mathbf{L}^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^{\infty,mec})$, and $\mathbf{g}_{bl}^1 = (g_{bl,1}^1, g_{bl,2}^1, g_{bl,3}^1) \in \mathbf{L}^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{\infty,mec})$ such that $\mathbf{u}_{bl}^{1,\alpha} \chi_{\Omega_{bl,1}^{1,mec}} \rightarrow \mathbf{u}_{bl}^1$, $\mathbf{f}_{bl}^{1,\alpha} \chi_{\Omega_{bl,1}^{1,mec}} \rightarrow \mathbf{f}_{bl}^1$ weakly in $\mathbf{L}^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^{\infty,mec})$, and $\mathbf{g}_{bl}^{1,\alpha} \chi_{\Omega_{bl,1}^{1,mec}} \rightarrow \mathbf{g}_{bl}^1$ weakly in $\mathbf{L}^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{\infty,mec})$ when $\alpha \rightarrow \infty$.

Proposition 25:

By Assumption 7 and using Proposition 11, then $\mathbf{T}_{bl,1}^\varepsilon \mathbf{u}^\varepsilon \rightharpoonup \mathbf{u}_{bl}^{1,\alpha} + \widetilde{\mathbf{u}}^0$, $\mathbf{T}_{bl,1}^\varepsilon (\mathbf{f}^\varepsilon) \rightharpoonup \mathbf{f}_{bl}^{1,\alpha} + \widetilde{\mathbf{f}}^0$ weakly in $\mathbf{L}^2(\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec})$ and $\mathbf{T}_{bl,1}^\varepsilon \mathbf{g}^\varepsilon \rightharpoonup \mathbf{g}_{bl}^{1,\alpha} + \widetilde{\mathbf{g}}^0$ weakly in $\mathbf{L}^2(\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{1,mec})$ where $\widetilde{\varphi}(x^\#, x^1) = \varphi((x^\#, 0), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2}))$, for $(x^\#, x^1) \in \Omega_{bl,1}^\# \times \overline{\Omega_{bl,1}^{1,mec}}$.

Proposition 26:

The limit \mathbf{u}_{bl}^1 is a solution to

$$\left\{ \begin{array}{ll} -\operatorname{div}^1 \sigma^1(\mathbf{u}_{bl}^1) = & \mathbf{f}_{bl}^1 & \text{in } \Omega_{bl,1}^{\infty,mec} \\ \mathbf{u}_{bl}^1 = & \mathbf{0} & \text{on } \Gamma_{bl,1,0}^{\infty,mec} \\ \mathbf{u}_{bl}^1 & \text{is } \Gamma_{bl,1,per}^{\infty,mec} \text{ - periodic} & \\ \sigma^1(\mathbf{u}_{bl}^1) \mathbf{n}^1 = & \mathbf{g}_{bl}^1 & \text{on } \Gamma_{bl,1,1}^{\infty,mec} \\ \sigma^1(\mathbf{u}_{bl}^1) \mathbf{n}^1 & \text{is } \Gamma_{bl,1,per}^{\infty,mec} \text{ - antiperiodic} & \\ \sigma^1(\mathbf{u}_{bl}^1) \mathbf{n}^1 = & -\sigma^1(\widetilde{\mathbf{u}}^0) \mathbf{n}^1 & \text{on } \Gamma_{bl,1,Or}^{\infty,mec}. \end{array} \right. \quad (2.7.6)$$

Proof. We replace \mathbf{v}^ε in (2.7.3) by a smooth vector valued function $\mathbf{v}_{bl}^\varepsilon$ defined in $\Omega_{bl,1}^{\alpha\varepsilon,mec}$ and vanishing out of $\overline{\Omega_{bl,1}^{\alpha\varepsilon,mec}}$. Then, we substitute $\mathbf{v}_{bl}^\varepsilon$ by $\mathbf{B}_{bl,1}^\varepsilon(\mathbf{w})$ where $\mathbf{w} \in \mathbf{C}^\infty(\Omega_{bl,1}^\# \times$

$\overline{\Omega_{bl,1}^{1,mec}}$, is $\Omega_{bl,1}^{1,mec}$ - periodic in the direction x_1^1 , $\mathbf{w} = 0$ on $\Gamma_{bl,1,0}^{1,mec} \cup \Gamma_{bl,1,\alpha}^{1,mec}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = 0$ on $\Gamma_{bl,1,per}^{1,mec} \cup \Gamma_{bl,1,1}^{1,mec} \cup \Gamma_{bl,1,Or}^{1,mec} \cup \Gamma_{bl,1,\alpha}^{1,mec}$, we get

$$\begin{aligned} & - \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon \partial_j^\varepsilon \sigma_{ij}^1(\mathbf{B}_{bl,1}^\varepsilon(\mathbf{w})) dx^\varepsilon + \int_{\Gamma_{bl,1,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon \sigma_{ij}^\varepsilon(\mathbf{B}_{bl,1}^\varepsilon(\mathbf{w})) n_j^\varepsilon ds(x^\varepsilon) \\ & = \frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon B_{bl,1}^\varepsilon(w_i) dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_{bl,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon B_{bl,1}^\varepsilon(w_i) ds(x^\varepsilon). \end{aligned} \quad (2.7.7)$$

Then

$$\partial_j^\varepsilon \mathbf{B}_{bl,1}^\varepsilon(\mathbf{w}) = \mathbf{B}_{bl,1}^\varepsilon(\partial_j^\# \mathbf{w}) \chi_{I^\#}(j) + \frac{1}{\varepsilon} \mathbf{B}_{bl,1}^\varepsilon(\partial_j^1 \mathbf{w}) \text{ and } e^\varepsilon(\mathbf{B}_{bl,1}^\varepsilon(\mathbf{w})) = \mathbf{B}_{bl,1}^\varepsilon(e^\#(\mathbf{w})) + \frac{1}{\varepsilon} \mathbf{B}_{bl,1}^\varepsilon(e^1(\mathbf{w})),$$

where

$$e^\#(\mathbf{w}) = \frac{1}{2} (\partial_i^\# w_j \chi_{I^\#}(i) + \partial_j^\# w_i \chi_{I^\#}(j)) \text{ and } e_{ij}^1(\mathbf{w}) = \frac{1}{2} (\partial_i^1 w_j + \partial_j^1 w_i), \text{ with } I^\# = \{1\},$$

and we denote

$$\sigma_{ij}^\#(\mathbf{w}) = \lambda e_{pp}^\#(\mathbf{w}) \delta_{ij} + 2\mu e_{ij}^\#(\mathbf{w}) \text{ and } \sigma_{ij}^1(\mathbf{w}) = \lambda e_{pp}^1(\mathbf{w}) \delta_{ij} + 2\mu e_{ij}^1(\mathbf{w}).$$

One checks that $B_{bl,1}^\varepsilon(\sigma_{ij}^1(\mathbf{w})) \mathbf{n}^\varepsilon = \mathbf{0}$ on $\Gamma_{bl,1,1}^{\alpha\varepsilon,mec}$. Plugging into (2.7.7), we can assert that

$$-\frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{bl,1}^\varepsilon(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon + O(\varepsilon) = \frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon B_{bl,1}^\varepsilon(w_i) dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_{bl,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon B_{bl,1}^\varepsilon(w_i) ds(x^\varepsilon).$$

where

$$\begin{aligned} O(\varepsilon) & = - \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{bl,1}^\varepsilon(\partial_j^\# \sigma_{ij}^\#(\mathbf{w}) \chi_{I^\#}(j)) dx^\varepsilon - \frac{1}{\varepsilon} \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{bl,1}^\varepsilon(\partial_j^1 \sigma_{ij}^\#(\mathbf{w})) dx^\varepsilon \\ & \quad - \frac{1}{\varepsilon} \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{bl,1}^\varepsilon(\partial_j^\# \sigma_{ij}^1(\mathbf{w}) \chi_{I^\#}(j)) dx^\varepsilon + \int_{\Gamma_{bl,1,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{bl,1}^\varepsilon(\sigma_{ij}^\#(\mathbf{w})) n_j^\varepsilon \chi_{I^\#}(j) ds(x^\varepsilon). \end{aligned}$$

Replacing $B_{bl,1}^\varepsilon$ by $T_{bl,1}^{\varepsilon*}$, then

$$-\frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon T_{bl,1}^{\varepsilon*}(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon = \frac{1}{\varepsilon^2} \int_{\Omega_{bl,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon T_{bl,1}^{\varepsilon*}(w_i) dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_{bl,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon T_{bl,1}^{\varepsilon*}(w_i) ds(x^\varepsilon) + O(\varepsilon).$$

By the definition of the adjoint operator $T_{bl,1}^{\varepsilon*}$, we have

$$\begin{aligned} - \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} T_{bl,1}^\varepsilon(u_i^\varepsilon) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^\# dx^1 & = \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} T_{bl,1}^\varepsilon(f_i^\varepsilon) w_i dx^\# dx^1 \\ & \quad + \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{1,mec}} T_{bl,1}^\varepsilon(g_i^\varepsilon) w_i dx^\# ds(x^1) + O(\varepsilon). \end{aligned}$$

Passing ε to 0 and combining with Proposition 25 yields

$$\begin{aligned} - \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} \left(u_{bl,i}^{1,\alpha} + \widetilde{u}_i^0 \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^\# dx^1 &= \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} \left(f_{bl,i}^{1,\alpha} + \widetilde{f}_i^0 \right) w_i dx^\# dx^1 \\ &+ \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{1,mec}} \left(g_{bl,i}^{1,\alpha} + \widetilde{g}_i^0 \right) w_i dx^\# ds(x^1). \end{aligned}$$

It follows by the same method in the proof of the lateral boundary layer in the electrostatic model part that

$$\begin{aligned} - \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} \left(u_{bl,i}^1 + \widetilde{u}_i^0 \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^\# dx^1 &= \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} \left(f_{bl,i}^1 + \widetilde{f}_i^0 \right) w_i dx^\# dx^1 \\ &+ \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{1,mec}} \left(g_{bl,i}^1 + \widetilde{g}_i^0 \right) w_i dx^\# ds(x^1). \end{aligned}$$

for each α .

Applying the Green's formula twice in the l.h.s, we obtain

$$\begin{aligned} - \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} \partial_j^1 \left(\sigma_{ij}^1(\mathbf{u}_{bl}^1 + \widetilde{\mathbf{u}}^0) \right) w_i dx^\# dx^1 &+ \int_{\Omega_{bl,1}^\# \times \partial \Omega_{bl,1}^{1,mec}} \sigma_{ij}^1(\mathbf{u}_{bl}^1 + \widetilde{\mathbf{u}}^0) n_j^1 w_i dx^\# ds(x^1) \\ - \int_{\Omega_{bl,1}^\# \times \partial \Omega_{bl,1}^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^1 \left(u_{bl,i}^1 + \widetilde{u}_i^0 \right) dx^\# ds(x^1) &= \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} \left(f_{bl,i}^1 + \widetilde{f}_i^0 \right) w_i dx^\# dx^1 \\ &+ \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{1,mec}} \left(g_{bl,i}^1 + \widetilde{g}_i^0 \right) w_i dx^\# ds(x^1). \end{aligned}$$

Decomposing $\partial \Omega_{bl,1}^{1,mec}$ into $\Gamma_{bl,1,0}^{1,mec} \cup \Gamma_{bl,1,1}^{1,mec} \cup \Gamma_{bl,1,per}^{1,mec} \cup \Gamma_{bl,1,Or}^{1,mec} \cup \Gamma_{bl,1,\alpha}^{1,mec}$ and combining with Proposition 24 and the conditions satisfied \mathbf{w} gives

$$\begin{aligned} - \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} \partial_j^1 \left(\sigma_{ij}^1(\mathbf{u}_{bl}^1) \right) w_i dx^\# dx^1 &+ \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,Or}^{1,mec}} \sigma_{ij}^1(\mathbf{u}_{bl}^1) n_j^1 w_i dx^\# ds(x^1) \\ + \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,per}^{1,mec}} \sigma_{ij}^1(\mathbf{u}_{bl}^1) n_j^1 w_i dx^\# ds(x^1) &- \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,0}^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^1 u_{bl,i}^1 dx^\# ds(x^1) \\ = \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} f_{bl,i}^1 w_i dx^\# dx^1 &+ \int_{\Omega_{bl,1}^\# \times \Gamma_{bl,1,1}^{1,mec}} g_{bl,i}^1 w_i dx^\# ds(x^1). \end{aligned}$$

If we choose $\mathbf{w} = 0$ on $\Gamma_{bl,1,1}^{1,mec} \cup \Gamma_{bl,1,per}^{1,mec} \cup \Gamma_{bl,1,Or}^{1,mec}$ and $\sigma^1(\mathbf{w})\mathbf{n} = \mathbf{0}$ on $\Gamma_{bl,1,0}^{1,mec}$,

$$- \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} \partial_j^1 \left(\sigma_{ij}^1(\mathbf{u}_{bl}^1) \right) w_i dx^\# dx^1 = \int_{\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}} f_{bl,i}^1 w_i dx^\# dx^1.$$

Hence, $-\partial_j^1 \sigma_{ij}^1(\mathbf{u}_{bl}^1) = f_{bl,i}^1$ in $\Omega_{bl,1}^\# \times \Omega_{bl,1}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{bl,1,1}^{1,mec} \cup \Gamma_{bl,1,per}^{1,mec} \cup \Gamma_{bl,1,Or}^{1,mec}$, we get $u_{bl,i}^1 = 0$ on $\Gamma_{bl,1,0}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{bl,1,per}^{1,mec} \cup \Gamma_{bl,1,Or}^{1,mec}$, so $\sigma_{ij}^1(\mathbf{u}_{bl}^1)n_j^1 = g_{bl,i}^1$ on $\Gamma_{bl,1,1}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{bl,1,per}^{1,mec}$. Thus, $\sigma_{ij}^1(\mathbf{u}_{bl}^1)n_j^1 = -\sigma_{ij}^1(\widetilde{\mathbf{u}}^0)n_j^1$ on $\Gamma_{bl,1,Or}^{1,mec}$.

Lastly, we obtain $\sigma_{ij}^1(\mathbf{u}_{bl}^1)n_j^1$ is $\Gamma_{bl,1,per}^{1,mec}$ - antiperiodic.

Since this is true for each α then the above equations hold in the infinite domain and the proof is completed. \square

2.7.3/ EXTERIOR EDGE MODEL

We now introduce $\mathbf{u}_{exe}^\varepsilon = \mathbf{u}^\varepsilon - (\mathbf{B}^\varepsilon \mathbf{u}^0 + \mathbf{B}_{bl,1}^\varepsilon \mathbf{u}_{bl}^1 + \mathbf{B}_{bl,2}^\varepsilon \mathbf{u}_{bl}^2)$, $\mathbf{f}_{exe}^\varepsilon = \mathbf{f}^\varepsilon - (\mathbf{B}^\varepsilon \mathbf{f}^0 + \mathbf{B}_{bl,1}^\varepsilon \mathbf{f}_{bl}^1 + \mathbf{B}_{bl,2}^\varepsilon \mathbf{f}_{bl}^2)$ and $\mathbf{g}_{exe}^\varepsilon = \mathbf{g}^\varepsilon - (\mathbf{B}^\varepsilon \mathbf{g}^0 + \mathbf{B}_{bl,1}^\varepsilon \mathbf{g}_{bl}^1 + \mathbf{B}_{bl,2}^\varepsilon \mathbf{g}_{bl}^2)$, and make the following assumptions.

Assumption 8:

We assume that

1. For each α , there exist $\mathbf{u}_{exe}^{1,\alpha} = (u_{exe,1}^{1,\alpha}, u_{exe,2}^{1,\alpha}, u_{exe,3}^{1,\alpha})$ in $\mathbf{H}^1(\Omega_{exe,1}^{1,mec})$, $\mathbf{f}_{exe}^1 = (f_{exe,1}^{1,\alpha}, f_{exe,2}^{1,\alpha}, f_{exe,3}^{1,\alpha})$ in $\mathbf{L}^2(\Omega_{exe,1}^{1,mec})$, and $\mathbf{g}_{exe}^1 = (g_{exe,1}^{1,\alpha}, g_{exe,2}^{1,\alpha}, g_{exe,3}^{1,\alpha})$ in $\mathbf{L}^2(\Gamma_{exe,1,1}^{1,mec})$ such that $\mathbf{T}_{exe,1}^\varepsilon(\mathbf{u}_{exe}^\varepsilon) \rightharpoonup \mathbf{u}_{exe}^{1,\alpha}$, $\mathbf{T}_{exe,1}^\varepsilon \mathbf{f}_{exe}^\varepsilon \rightharpoonup \mathbf{f}_{exe}^{1,\alpha}$ weakly in $\mathbf{L}^2(\Omega_{exe,1}^{1,mec})$, and $\mathbf{T}_{exe,1}^\varepsilon \mathbf{g}_{exe}^\varepsilon \rightharpoonup \mathbf{g}_{exe}^{1,\alpha}$ weakly in $\mathbf{L}^2(\Gamma_{exe,1,1}^{1,mec})$ when $\varepsilon \rightarrow 0$.
2. There exist $\mathbf{u}_{exe}^1 = (u_{exe,1}^1, u_{exe,2}^1, u_{exe,3}^1)$ in $\mathbf{H}^1(\Omega_{exe,1}^{\infty,mec})$, $u_{exe,i}^1$ and $\sigma_{ij}^1(\mathbf{u}_{exe}^1)$ converges exponentially fast to 0 when $x_1^1 + x_2^1 \rightarrow +\infty$ for $i, j \in \{1, 2, 3\}$, $\mathbf{f}_{exe}^1 = (f_{exe,1}^1, f_{exe,2}^1, f_{exe,3}^1)$ in $\mathbf{L}^2(\Omega_{exe,1}^{\infty,mec})$, and $\mathbf{g}_{exe}^1 = (g_{exe,1}^1, g_{exe,2}^1, g_{exe,3}^1)$ in $\mathbf{L}^2(\Gamma_{exe,1,1}^{\infty,mec})$ such that $\mathbf{u}_{exe}^{1,\alpha} \chi_{\Omega_{exe,1}^{1,mec}} \rightarrow \mathbf{u}_{exe}^1$, $\mathbf{f}_{exe}^{1,\alpha} \chi_{\Omega_{exe,1}^{1,mec}} \rightarrow \mathbf{f}_{exe}^1$ weakly in $\mathbf{L}^2(\Omega_{exe,1}^{1,mec})$, and $\mathbf{g}_{exe}^{1,\alpha} \chi_{\Omega_{exe,1}^{1,mec}} \rightarrow \mathbf{g}_{exe}^1$ weakly in $\mathbf{L}^2(\Gamma_{exe,1,1}^{1,mec})$ when $\alpha \rightarrow +\infty$.

Proposition 27:

Under Assumption 8 and using Proposition 14, then we obtain $\mathbf{T}_{exe,1}^\varepsilon \mathbf{u}_{exe}^\varepsilon \rightharpoonup \mathbf{u}_{exe}^{1,\alpha} + \widetilde{\mathbf{u}}^0 + \widetilde{\mathbf{u}}_{bl}^1 + \widetilde{\mathbf{u}}_{bl}^2$, $\mathbf{T}_{exe,1}^\varepsilon(\mathbf{f}_{exe}^\varepsilon) \rightharpoonup \mathbf{f}_{exe}^{1,\alpha} + \widetilde{\mathbf{f}}^0 + \widetilde{\mathbf{f}}_{bl}^1 + \widetilde{\mathbf{f}}_{bl}^2$ weakly in $\mathbf{L}^2(\Omega_{exe,1}^{1,mec})$ and $\mathbf{T}_{exe,1}^\varepsilon \mathbf{g}_{exe}^\varepsilon \rightharpoonup \mathbf{g}_{exe}^{1,\alpha} + \widetilde{\mathbf{g}}^0 + \widetilde{\mathbf{g}}_{bl}^1 + \widetilde{\mathbf{g}}_{bl}^2$ weakly in $\mathbf{L}^2(\Gamma_{exe,1,1}^{1,mec})$, where $\widetilde{\varphi}^0(x^1) = \varphi^0(0, x^1 - 1/2)$, $\widetilde{\varphi}_{bl}^1(x^1) = \varphi_{bl}^1(0, (x_1^1 - 1/2, x_2^1, x_3^1))$, and $\widetilde{\varphi}_{bl}^2(x^1) = \varphi_{bl}^2(0, (x_1^1, x_2^1 - 1/2, x_3^1))$ for $x^1 \in \Omega_{exe,1}^{1,mec}$, resp. $x^1 \in \Gamma_{exe,1,1}^{1,mec}$.

Proposition 28:

The limit \mathbf{u}_{exe}^1 is a solution to

$$\left\{ \begin{array}{lll} -\operatorname{div}^1 \sigma^1(\mathbf{u}_{exe}^1) = \mathbf{f}_{exe}^1 & \text{in} & \Omega_{exe,1}^{\infty,mec} \\ \mathbf{u}_{exe}^1 = \mathbf{0} & \text{on} & \Gamma_{exe,1,0}^{\infty,mec} \\ \sigma^1(\mathbf{u}_{exe}^1) \mathbf{n}^1 = \mathbf{g}_{exe}^1 & \text{on} & \Gamma_{exe,1,1}^{\infty,mec} \\ \sigma^1(\mathbf{u}_{exe}^1) \mathbf{n}^1 = -\sigma^1(\mathbf{u}_{bl}^2) \mathbf{n}^1 & \text{on} & \Gamma_{exe,1,bl1}^{\infty,mec} \\ \sigma^1(\mathbf{u}_{exe}^1) \mathbf{n}^1 = -\sigma^1(\mathbf{u}_{bl}^1) \mathbf{n}^1 & \text{on} & \Gamma_{exe,1,bl2}^{\infty,mec} \end{array} \right.$$

Proof. We replace \mathbf{v}^ε in (2.7.3) by a smooth function $\mathbf{v}_{exe}^\varepsilon$ defined in $\Omega_{exe,1}^{\alpha\varepsilon,mec}$ and vanishing out of $\overline{\Omega_{exe,1}^{\alpha\varepsilon,mec}}$. Then we substitute $\mathbf{v}_{exe}^\varepsilon$ by $\varepsilon^{-1} \mathbf{B}_{exe,1}^\varepsilon \mathbf{w}$, where $\mathbf{w} \in \mathbf{C}^\infty(\Omega_{exe,1}^\# \times \overline{\Omega_{exe,1}^{1,mec}})$, $\mathbf{w} = \mathbf{0}$ on $\Gamma_{exe,1,0}^{1,mec} \cup \Gamma_{exe,1,\alpha}^{1,mec}$ and $\sigma^1(\mathbf{w}) \mathbf{n}^1 = \mathbf{0}$ on $\Gamma_{exe,1,1}^{1,mec} \cup \Gamma_{exe,1,\alpha}^{1,mec} \cup \Gamma_{exe,1,bl1}^{1,mec} \cup \Gamma_{exe,1,bl2}^{1,mec}$, then

$$\begin{aligned} & -\frac{1}{\varepsilon} \int_{\Omega_{exe,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon \partial_j^\varepsilon \sigma_{ij}^1(\mathbf{B}_{exe,1}^\varepsilon(\mathbf{w})) \, dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_{exe,1,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon \sigma_{ij}^\varepsilon(\mathbf{B}_{exe,1}^\varepsilon(\mathbf{w})) n_j^\varepsilon \, ds(x^\varepsilon) \\ & = \frac{1}{\varepsilon^3} \int_{\Omega_{exe,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon B_{exe,1}^\varepsilon(w_i) \, dx^\varepsilon + \frac{1}{\varepsilon^2} \int_{\Gamma_{exe,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon B_{exe,1}^\varepsilon(w_i) \, ds(x^\varepsilon). \end{aligned}$$

A straightforward calculation reveals that

$$-\frac{1}{\varepsilon^3} \int_{\Omega_{exe,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{exe,1}^\varepsilon(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) \, dx^\varepsilon = \frac{1}{\varepsilon^3} \int_{\Omega_{exe,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon B_{exe,1}^\varepsilon(w_i) \, dx^\varepsilon + \frac{1}{\varepsilon^2} \int_{\Gamma_{exe,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon B_{exe,1}^\varepsilon(w_i) \, ds(x^\varepsilon).$$

Replacing $B_{exe,1}^\varepsilon$ by $T_{exe,1}^{\varepsilon*}$ and combining with the definition of $T_{exe,1}^{\varepsilon*}$, we can write that

$$-\int_{\Omega_{exe,1}^{1,mec}} T_{exe,1}^\varepsilon(u_i^\varepsilon) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) \, dx^1 = \int_{\Omega_{exe,1}^{1,mec}} T_{exe,1}^\varepsilon(f_i^\varepsilon) w_i \, dx^1 + \int_{\Gamma_{exe,1,1}^{1,mec}} T_{exe,1}^\varepsilon(g_i^\varepsilon) w_i \, ds(x^1).$$

Passing ε to 0 and by Proposition 27, yields

$$\begin{aligned} & -\int_{\Omega_{exe,1}^{1,mec}} \left(u_{exe,i}^1 + \widetilde{u}_i^0 + \widetilde{u}_{bl,i}^1 + \widetilde{u}_{bl,i}^2 \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) \, dx^1 \\ & = \int_{\Omega_{exe,1}^{1,mec}} \left(f_{exe,i}^{1,\alpha} + \widetilde{f}_i^0 + \widetilde{f}_{bl,i}^1 + \widetilde{f}_{bl,i}^2 \right) w_i \, dx^1 + \int_{\Gamma_{exe,1,1}^{1,mec}} \left(g_{exe,i}^{1,\alpha} + \widetilde{g}_i^0 + \widetilde{g}_{bl,i}^1 + \widetilde{g}_{bl,i}^2 \right) w_i \, ds(x^1). \end{aligned}$$

In the same manner as the proof of the electrostatic edge model, we can assert that

$$\begin{aligned} & -\int_{\Omega_{exe,1}^{1,mec}} \left(u_{exe,i}^1 + \widetilde{u}_i^0 + \widetilde{u}_{bl,i}^1 + \widetilde{u}_{bl,i}^2 \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) \, dx^1 \\ & = \int_{\Omega_{exe,1}^{1,mec}} \left(f_{exe,i}^1 + \widetilde{f}_i^0 + \widetilde{f}_{bl,i}^1 + \widetilde{f}_{bl,i}^2 \right) w_i \, dx^1 + \int_{\Gamma_{exe,1,1}^{1,mec}} \left(g_{exe,i}^1 + \widetilde{g}_i^0 + \widetilde{g}_{bl,i}^1 + \widetilde{g}_{bl,i}^2 \right) w_i \, ds(x^1), \end{aligned}$$

for each α .

Applying Green's formula twice and decomposing $\partial\Omega_{exe,1}^{1,mec}$ into $\Gamma_{exe,1,0}^{1,mec} \cup \Gamma_{exe,1,1}^{1,mec} \cup \Gamma_{exe,1,bl1}^{1,mec} \cup$

$\Gamma_{exe,1,b12}^{1,mec} \cup \Gamma_{exe,1,\alpha}^{1,mec}$, combining with conditions satisfied by \mathbf{w} , Proposition 24, and Proposition 26, we get

$$\begin{aligned} & - \int_{\Omega_{exe,1}^{1,mec}} \partial_j^1(\sigma_{ij}^1(\mathbf{u}_{exe}^1)) w_i dx^1 + \int_{\Gamma_{exe,1,1}^{1,mec}} \sigma_{ij}^1(\mathbf{u}_{exe}^1) n_j^1 w_i ds(x^1) \\ & + \int_{\Gamma_{exe,1,b11}^{1,mec}} \sigma_{ij}^1(\mathbf{u}_{exe}^1 + \widetilde{\mathbf{u}}_{bl}^2) n_j^1 w_i ds(x^1) + \int_{\Gamma_{exe,1,b12}^{1,mec}} \sigma_{ij}^1(\mathbf{u}_{exe}^1 + \widetilde{\mathbf{u}}_{bl}^1) n_j^1 w_i ds(x^1) \\ & - \int_{\Gamma_{exe,1,0}^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^1 u_{exe,i}^1 ds(x^1) = \int_{\Omega_{exe,1}^{1,mec}} f_{exe,i}^1 w_i dx^1 + \int_{\Gamma_{exe,1,1}^{1,mec}} g_{exe,i}^1 w_i ds(x^1). \end{aligned}$$

If $\mathbf{w} = \mathbf{0}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\partial\Omega_{exe,1}^{1,mec}$, we obtain $-\partial_j^1\sigma_{ij}^1(\mathbf{u}_{exe}^1) = \mathbf{f}_{exe}^1$ in $\Omega_{exe,1}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\partial\Omega_{exe,1}^{1,mec}$, thus $\mathbf{u}_{exe}^1 = \mathbf{0}$ on $\Gamma_{exe,1,0}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{exe,1,b11}^{1,mec} \cup \Gamma_{exe,1,b12}^{1,mec}$, we assert that $\sigma_{ij}^1(\mathbf{u}_{exe}^1) n_j^1 = g_{exe,i}^1$ on $\Gamma_{exe,1,1}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{exe,1,b12}^{1,mec}$, we deduce $\sigma_{ij}^1(\mathbf{u}_{exe}^1) n_j^1 = -\sigma_{ij}^1(\widetilde{\mathbf{u}}_{bl}^1) n_j^1$ on $\Gamma_{exe,1,b11}^{1,mec}$.

Lastly, we get $\sigma_{ij}^1(\mathbf{u}_{exe}^1) n_j^1 = -\sigma_{ij}^1(\widetilde{\mathbf{u}}_{bl}^2) n_j^1$ on $\Gamma_{exe,1,b11}^{1,mec}$.

Since this is true for each α then the above equations hold in the infinite domain and the proof is completed. \square

2.7.4/ INTERFACE MODEL

Let us recall that $\mathbf{u}_{bl}^\varepsilon = \mathbf{u}^\varepsilon - \mathbf{B}^\varepsilon(\mathbf{u}^0)$, $\mathbf{f}_{bl}^\varepsilon = \mathbf{f}^\varepsilon - \mathbf{B}^\varepsilon(\mathbf{f}^0)$, $\mathbf{g}_{bl}^\varepsilon = \mathbf{g}^\varepsilon - \mathbf{B}^\varepsilon(\mathbf{g}^0)$, and make the following assumptions

Assumption 9:

We assume that

1. For each α , there exist $\mathbf{u}_{in}^{1,\alpha} = (u_{in,1}^{1,\alpha}, u_{in,2}^{1,\alpha}, u_{in,3}^{1,\alpha}) \in \mathbf{L}^2(\Omega_{in,1}^\#, \mathbf{H}^1(\Omega_{in,1}^{1,mec}))$, is $\Omega_{in,1}^{1,mec}$ -periodic in the direction x_2^1 , $\mathbf{f}_{in}^1 = (f_{in,1}^1, f_{in,2}^1, f_{in,3}^1) \in \mathbf{L}^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec})$, and $\mathbf{g}_{in}^1 = (g_{in,1}^1, g_{in,2}^1, g_{in,3}^1) \in \mathbf{L}^2(\Omega_{in,1}^\# \times \Gamma_{in,1,1}^{1,mec})$ such that $\mathbf{T}_{in,1}^\varepsilon \mathbf{u}_{bl}^\varepsilon \rightharpoonup \mathbf{u}_{in}^1$, $\mathbf{T}_{in,1}^\varepsilon \mathbf{f}_{bl}^\varepsilon \rightharpoonup \mathbf{f}_{in}^1$ weakly in $\mathbf{L}^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec})$, and $\mathbf{T}_{in,1}^\varepsilon \mathbf{g}_{bl}^\varepsilon \rightharpoonup \mathbf{g}_{in}^1$ weakly in $\mathbf{L}^2(\Omega_{in,1}^\# \times \Gamma_{in,1,1}^{1,mec})$ when $\varepsilon \rightarrow 0$.
2. There exist $\mathbf{u}_{in}^1 = (u_{in,1}^1, u_{in,2}^1, u_{in,3}^1) \in \mathbf{L}^2(\Omega_{in,1}^\#, H^1(\Omega_{in,1}^{\infty,mec}))$, $\Omega_{in,1}^{\infty,mec}$ -periodic in the direction x_2^1 , $u_{in,i}^1$ and $\sigma_{ij}^1(\mathbf{u}_{in}^1)$ converge exponentially to zero when $|x_2^1| \rightarrow +\infty$ for all $i, j \in \{1, 2, 3\}$, $\mathbf{f}_{in}^1 = (f_{in,1}^1, f_{in,2}^1, f_{in,3}^1) \in \mathbf{L}^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{\infty,mec})$, and $\mathbf{g}_{in}^1 = (g_{in,1}^1, g_{in,2}^1, g_{in,3}^1) \in \mathbf{L}^2(\Omega_{in,1}^\# \times \Gamma_{in,1,1}^{\infty,mec})$ such that $\mathbf{u}_{in}^{1,\alpha} \chi_{\Omega_{in,1}^{1,mec}} \rightharpoonup \mathbf{u}_{in}^1$, $\mathbf{f}_{in}^{1,\alpha} \chi_{\Omega_{in,1}^{1,mec}} \rightharpoonup \mathbf{f}_{in}^1$ weakly in $\mathbf{L}^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{\infty,mec})$, and $\mathbf{g}_{in}^{1,\alpha} \chi_{\Omega_{in,1}^{1,mec}} \rightharpoonup \mathbf{g}_{in}^1$ weakly in $\mathbf{L}^2(\Omega_{in,1}^\# \times \Gamma_{in,1,1}^{\infty,mec})$ when $\alpha \rightarrow +\infty$.

Proposition 29:

Under Assumption 9 and using Proposition 18, then $\mathbf{T}_{in,1}^\varepsilon \mathbf{u}^\varepsilon \rightharpoonup \mathbf{u}_{in}^{1,\alpha} + \widetilde{\mathbf{u}}^0$, $\mathbf{T}_{in,1}^\varepsilon (\mathbf{f}^\varepsilon) \rightharpoonup \mathbf{f}_{in}^{1,\alpha} + \widetilde{\mathbf{f}}^0$ weakly in $\mathbf{L}^2(\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec})$ and $\mathbf{T}_{in,1}^\varepsilon \mathbf{g}^\varepsilon \rightharpoonup \mathbf{g}_{in}^{1,\alpha} + \widetilde{\mathbf{g}}^0$ weakly in $\mathbf{L}^2(\Omega_{in,1}^\# \times \Gamma_{in,1,1}^{1,mec})$, where $\widetilde{\varphi}(x^\#, x^1) = \varphi(x^\#, L_1^2), (x_1^1, x_2^1 - \frac{1}{2}, x_3^1 - \frac{1}{2})$ for $(x^\#, x^1) \in \Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec}$, (resp. $(x^\#, x^1) \in \Omega_{in,1}^\# \times \Gamma_{in,1,1}^{1,mec}$).

Proposition 30:

The limit \mathbf{u}_{in}^1 is a solution to

$$\left\{ \begin{array}{ll} -div^1 \sigma^1(\mathbf{u}_{in}^1) = \mathbf{f}_{in}^1 & \text{in } \Omega_{in,1}^{\infty,mec} \\ \mathbf{u}_{in}^1 = \mathbf{0} & \text{on } \Gamma_{in,1,0}^{\infty,mec} \\ \mathbf{u}_{in}^1 & \text{is } \Gamma_{in,1,per}^{\infty,mec} \text{ - periodic} \\ \sigma^1(\mathbf{u}_{in}^1) \mathbf{n}^1 = \mathbf{g}_{in}^1 & \text{on } \Gamma_{in,1,1}^{\infty,mec} \\ \sigma^1(\mathbf{u}_{in}^1) \mathbf{n}^1 & \text{is } \Gamma_{in,1,per}^{\infty,mec} \text{ - antiperiodic} \\ \left[\left[\sigma^1(\mathbf{u}_{in}^1) \right] \right] \mathbf{n}^1 = - \left[\left[\sigma^1(\widetilde{\mathbf{u}}^0) \right] \right] \mathbf{n}^1 & \text{on } \Gamma_{in,1,interf}^{\infty,mec} \\ \left[\left[\mathbf{u}_{in}^1 \right] \right] = - \left[\left[\widetilde{\mathbf{u}}^0 \right] \right] & \text{on } \Gamma_{in,1,interf}^{\infty,mec} \end{array} \right. \quad (2.7.8)$$

Proof. We replace \mathbf{v}^ε in (2.7.3) by a smooth function $\mathbf{v}_{in}^\varepsilon$ defined in $\overline{\Omega_{in,1}^{\alpha\varepsilon,mec}}$ and vanishing out of $\overline{\Omega_{in,1}^{\alpha\varepsilon,mec}}$. Then, we substitute $\mathbf{v}_{in}^\varepsilon$ by $\mathbf{B}_{in,1}^\varepsilon(\mathbf{w})$, where \mathbf{w} is in $\mathbf{C}^\infty(\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec})$, is $\Omega_{in,1}^{1,mec}$ -periodic in the directions x_1^1, x_2^1 , $\mathbf{w} = 0$ on $\Gamma_{in,1,0}^{1,mec} \cup \Gamma_{in,1,\alpha}^{1,mec}$ and $\sigma^1(\mathbf{w}) \mathbf{n}^1 = \mathbf{0}$ on $\Gamma_{in,1,1}^{1,mec} \cup \Gamma_{in,1,per}^{1,mec} \cup \Gamma_{in,1,\alpha}^{1,mec}$. It follows that

$$\begin{aligned} & - \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon \partial_j^\varepsilon \sigma_{ij}^1 \mathbf{B}_{in,1}^\varepsilon(\mathbf{w}) n_j^\varepsilon dx^\varepsilon + \int_{\Gamma_{in,1,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon \sigma_{ij}^\varepsilon (\mathbf{B}_{in,1}^\varepsilon(\mathbf{w})) n_j^\varepsilon ds(x^\varepsilon) \\ & = \frac{1}{\varepsilon^2} \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon B_{in,1}^\varepsilon(w_i) dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_{in,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon B_{in,1}^\varepsilon(w_i) ds(x^\varepsilon). \end{aligned}$$

After some calculations, we obtain

$$-\frac{1}{\varepsilon^2} \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{in,1}^\varepsilon(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon + O(\varepsilon) = \frac{1}{\varepsilon^2} \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon B_{in,1}^\varepsilon(w_i) dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_{in,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon B_{in,1}^\varepsilon(w_i) ds(x^\varepsilon),$$

where

$$\begin{aligned} O(\varepsilon) & = - \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{in,1}^\varepsilon(\partial_j^\# \sigma_{ij}^\#(\mathbf{w}) \chi_{I^\#}(j)) dx^\varepsilon - \frac{1}{\varepsilon} \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{in,1}^\varepsilon(\partial_j^1 \sigma_{ij}^\#(\mathbf{w})) dx^\varepsilon \\ & \quad - \frac{1}{\varepsilon} \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{in,1}^\varepsilon(\partial_j^\# \sigma_{ij}^1(\mathbf{w}) \chi_{I^\#}(j)) dx^\varepsilon + \int_{\Gamma_{in,1,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{in,1}^\varepsilon(\sigma_{ij}^\#(\mathbf{w})) n_j^\varepsilon ds(x^\varepsilon) \text{ with } I^\# = \{1\}. \end{aligned}$$

Replacing $B_{in,1}^\varepsilon$ by $T_{in,1}^{\varepsilon,*}$, the above equality becomes

$$-\frac{1}{\varepsilon^2} \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon T_{in,1}^{\varepsilon,*} (\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon = \frac{1}{\varepsilon^2} \int_{\Omega_{in,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon T_{in,1}^{\varepsilon,*} (w_i) dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_{in,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon T_{in,1}^{\varepsilon,*} (w_i) ds(x^\varepsilon) + O(\varepsilon).$$

By the definition of the adjoint operator $T_{in,1}^{\varepsilon,*}$, it follows that

$$\begin{aligned} - \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec}} T_{in,1}^\varepsilon (u_i^\varepsilon) \partial_j^1 (\sigma_{ij}^1(\mathbf{w})) dx^\# dx^1 &= \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec}} T_{in,1}^\varepsilon (f_i^\varepsilon) w_i dx^\# dx^1 \\ &+ \int_{\Omega_{in,1}^\# \times \Gamma_{in,1,1}^{1,mec}} T_{in,1}^\varepsilon (g_i^\varepsilon) w_i dx^\# ds(x^1) + O(\varepsilon). \end{aligned}$$

Passing ε to 0 and combining with Proposition 29 gives

$$\begin{aligned} - \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec}} (u_{in,i}^{1,\alpha} + \tilde{u}_i^0) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^\# dx^1 &= \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec}} (f_{in,i}^{1,\alpha} + \tilde{f}_i^0) w_i dx^\# dx^1 \\ &+ \int_{\Omega_{in,1}^\# \times \Gamma_{in,1,1}^{1,mec}} (g_{in,i}^{1,\alpha} + \tilde{g}_i^0) w_i dx^\# ds(x^1). \end{aligned}$$

In the same manner with the previous proofs, we can assert that for each α

$$\begin{aligned} - \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec}} (u_{in,i}^1 + \tilde{u}_i^0) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^\# dx^1 &= \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec}} (f_{in,i}^1 + \tilde{f}_i^0) w_i dx^\# dx^1 \\ &+ \int_{\Omega_{in,1}^\# \times \Gamma_{in,1,1}^{1,mec}} (g_{in,i}^1 + \tilde{g}_i^0) w_i dx^\# ds(x^1). \end{aligned}$$

Applying Green's formula twice in the l.h.s,

$$\begin{aligned} l.h.s &= - \sum_{\pm} \int_{\Omega_{in,1}^\# \times \Omega_{in,1}^{1,mec\pm}} \partial_j^1 (\sigma_{ij}^1(\mathbf{u}_{in}^{1\pm} + \tilde{\mathbf{u}}^{0\pm})) w_i dx^\# dx^1 \\ &+ \sum_{\pm} \int_{\Omega_{in,1}^\# \times \partial\Omega_{in,1}^{1,mec\pm}} \sigma_{ij}^1(\mathbf{u}_{in}^{1\pm} + \tilde{\mathbf{u}}^{0\pm}) n_j^{1\pm} w_i dx^\# ds(x^1) \\ &- \sum_{\pm} \int_{\Omega_{in,1}^\# \times \partial\Omega_{in,1}^{1,mec\pm}} \sigma_{ij}^1(\mathbf{w}) n_j^\pm (u_{in,i}^{1\pm} + \tilde{u}_i^{0\pm}) dx^\# ds(x^1). \end{aligned}$$

Decomposing $\Omega_{in,1}^{1,mec}$ into two nonoverlapping subdomains $\Omega_{in,1}^{1,mec+}$ and $\Omega_{in,1}^{1,mec-}$ with their boundaries $\Gamma_{in,1,0}^{1,mec,\pm} \cup \Gamma_{in,1,1}^{1,mec,\pm} \cup \Gamma_{in,1,per}^{1,mec,\pm} \cup \Gamma_{in,1,\alpha}^{1,mec,\pm} \cup \Gamma_{in,1,interf}^{1,mec}$, combing with the conditions

satisfied \mathbf{w} , and Proposition 24, we obtain

$$\begin{aligned}
& - \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Omega_{in,1}^{1,mec\pm}} \partial_j^1(\sigma_{ij}^1(\mathbf{u}_{in}^{1\pm})) w_i \, dx^{\#} dx^1 + \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,1}^{1,mec\pm}} \sigma_{ij}^1(\mathbf{u}_{in}^{1\pm}) n_j^{1\pm} w_i \, dx^{\#} ds(x^1) \quad (2.7.9) \\
& + \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,per}^{1,mec\pm}} \sigma_{ij}^1(\mathbf{u}_{in}^{1\pm}) n_j^{1\pm} w_i \, dx^{\#} ds(x^1) \\
& + \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,interf}^{1,mec}} \left[\sigma_{ij}^1(\mathbf{u}_{in}^{1+} + \tilde{\mathbf{u}}^{0+}) - \sigma_{ij}^1(\mathbf{u}_{in}^{1-} + \tilde{\mathbf{u}}^{0-}) \right] n_j^{1+} w_i \, dx^{\#} ds(x^1) \\
& - \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,interf}^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^{1+} \left[(u_{in,i}^{1+} + \tilde{u}_i^{0+}) - (u_{in,i}^{1-} + \tilde{u}_i^{0-}) \right] \, dx^{\#} ds(x^1) \\
& - \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,0}^{1,mec\pm}} \sigma_{ij}^1(\mathbf{w}) n_j^{1\pm} u_{in,i}^{1\pm} \, dx^{\#} ds(x^1) \\
& = \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Omega_{in,1}^{1,mec\pm}} f_{in,i}^{1\pm} w_i \, dx^{\#} dx^1 + \sum_{\pm} \int_{\Omega_{in,1}^{\#} \times \Gamma_{in,1,1}^{1,mec\pm}} g_{in,i}^{1\pm} w_i \, dx^{\#} ds(x^1).
\end{aligned}$$

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{in,1,1}^{1,mec} \cup \Gamma_{in,1,per}^{1,mec} \cup \Gamma_{in,1,interf}^{1,mec}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\Gamma_{in,1,0}^{1,mec} \cup \Gamma_{in,1,interf}^{1,mec}$, then $\partial_j^1 \sigma_{ij}^1(\mathbf{u}_{in}^1) = f_{in,i}^1$ in $\Omega_{in,1}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{in,1,per}^{1,mec} \cup \Gamma_{in,1,interf}^{1,mec}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\Gamma_{in,1,interf}^{1,mec} \cup \Gamma_{in,1,0}^{1,mec}$, we get $\sigma_{ij}^1(\mathbf{u}_{in}^1) n_j^1 = g_{in,i}^1$ on $\Gamma_{in,1,1}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{in,1,per}^{1,mec} \cup \Gamma_{in,1,interf}^{1,mec}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\Gamma_{in,1,interf}^{1,mec}$, thus $\mathbf{u}_{in}^1 = \mathbf{0}$ on $\Gamma_{in,1,0}^{1,mec}$.

If $\mathbf{w} = \mathbf{0}$ on $\Gamma_{in,1,interf}^{1,mec}$ and $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\Gamma_{in,1,interf}^{1,mec}$, we have $\sigma_{ij}^1(\mathbf{u}_{in}^1) n_j^1$ is $\Gamma_{in,1,per}^{1,mec}$ -antiperiodic.

If $\sigma^1(\mathbf{w})\mathbf{n}^1 = \mathbf{0}$ on $\Gamma_{in,1,interf}^{1,mec}$, it follows that $\left[\sigma_{ij}^1(\mathbf{u}_{in}^{1+} + \tilde{\mathbf{u}}^{0+}) - \sigma_{ij}^1(\mathbf{u}_{in}^{1-} + \tilde{\mathbf{u}}^{0-}) \right] n_j^{1+} = 0$ or $\left[\sigma_{ij}^1(\mathbf{u}_{in}^1) \right] n_j^1 = - \left[\sigma_{ij}^1(\tilde{\mathbf{u}}^0) \right] n_j^1$ on $\Gamma_{in,1,interf}^{1,mec}$.

Finally, we also obtain $\left[\mathbf{u}_{in}^1 \right] = - \left[\tilde{\mathbf{u}}^0 \right]$ on $\Gamma_{in,1,interf}^{1,mec}$.

Since this is true for each α then the above equations hold in the infinite domain and the proof is completed. \square

2.7.5/ INTERNAL EDGE MODEL

To investigate the internal edge model, we first introduce some terms

$$\begin{aligned}
\mathbf{u}_{ine}^{\varepsilon} &= \mathbf{u}^{\varepsilon} - \mathbf{B}^{\varepsilon} \mathbf{u}^0 - \mathbf{B}_{in,2}^{\varepsilon} \mathbf{u}_{in}^{2-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,2}} - (\mathbf{B}_{in,1}^{\varepsilon} \mathbf{u}_{in}^{1+} + \mathbf{B}_{in,2}^{\varepsilon} \mathbf{u}_{in}^{2+}) \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,3}} - \mathbf{B}_{in,2}^{\varepsilon} \mathbf{u}_{in}^{1-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,4}}, \\
\mathbf{f}_{ine}^{\varepsilon} &= \mathbf{f}^{\varepsilon} - \mathbf{B}^{\varepsilon} \mathbf{f}^0 - \mathbf{B}_{in,2}^{\varepsilon} \mathbf{f}_{in}^{2-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,2}} - (\mathbf{B}_{in,1}^{\varepsilon} \mathbf{f}_{in}^{1+} + \mathbf{B}_{in,2}^{\varepsilon} \mathbf{f}_{in}^{2+}) \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,3}} - \mathbf{B}_{in,2}^{\varepsilon} \mathbf{f}_{in}^{1-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,4}}, \\
\mathbf{g}_{ine}^{\varepsilon} &= \mathbf{g}^{\varepsilon} - \mathbf{B}^{\varepsilon} \mathbf{g}^0 - \mathbf{B}_{in,2}^{\varepsilon} \mathbf{g}_{in}^{2-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,2}} - (\mathbf{B}_{in,1}^{\varepsilon} \mathbf{g}_{in}^{1+} + \mathbf{B}_{in,2}^{\varepsilon} \mathbf{g}_{in}^{2+}) \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,3}} - \mathbf{B}_{in,2}^{\varepsilon} \mathbf{g}_{in}^{1-} \chi_{\Omega_{ine,1}^{\alpha\varepsilon,mec,4}},
\end{aligned}$$

and make some the following assumptions.

Assumption 10:

We assume that

1. For each α , there exist $\mathbf{u}_{ine}^{1,\alpha} = (u_{ine,1}^{1,\alpha}, u_{ine,2}^{1,\alpha}, u_{ine,3}^{1,\alpha})$ in $\mathbf{H}^1(\Omega_{ine,1}^{1,mec})$, $\mathbf{f}_{ine}^{1,\alpha} = (f_{ine,1}^{1,\alpha}, f_{ine,2}^{1,\alpha}, f_{ine,3}^{1,\alpha})$ in $\mathbf{L}^2(\Omega_{ine,1}^{1,mec})$, and $\mathbf{g}_{ine}^{1,\alpha} = (g_{ine,1}^{1,\alpha}, g_{ine,2}^{1,\alpha}, g_{ine,3}^{1,\alpha})$ in $\mathbf{L}^2(\Gamma_{ine,1,1}^{1,mec})$ such that $\mathbf{T}_{ine,1}^\varepsilon \mathbf{u}_{ine}^\varepsilon \rightharpoonup \mathbf{u}_{ine}^{1,\alpha}$ and $\mathbf{T}_{ine,1}^\varepsilon \mathbf{f}_{ine}^\varepsilon \rightharpoonup \mathbf{f}_{ine}^{1,\alpha}$ weakly in $\mathbf{L}^2(\Omega_{ine,1}^{1,mec})$, and $\mathbf{T}_{ine,1}^\varepsilon \mathbf{g}_{ine}^\varepsilon \rightharpoonup \mathbf{g}_{ine}^{1,\alpha}$ weakly in $\mathbf{L}^2(\Gamma_{ine,1,1}^{1,mec})$ when $\varepsilon \rightarrow 0$.
2. There exist $\mathbf{u}_{ine}^1 = (u_{ine,1}^1, u_{ine,2}^1, u_{ine,3}^1)$ in $\mathbf{H}^1(\Omega_{ine,i}^{\infty,mec})$, $u_{ine,i}^1$ and $\sigma_{ij}^1(\mathbf{u}_{ine}^1)$ converge exponentially to zero when $|x_1^1| + |x_2^1| \rightarrow +\infty$ for all $i, j \in \{1, 2, 3\}$, $\mathbf{f}_{ine}^1 = (f_{ine,1}^1, f_{ine,2}^1, f_{ine,3}^1)$ in $\mathbf{L}^2(\Omega_{ine,1}^{\infty,mec})$, and $\mathbf{g}_{ine}^1 = (g_{ine,1}^1, g_{ine,2}^1, g_{ine,3}^1)$ in $\mathbf{L}^2(\Gamma_{ine,1,1}^{\infty,mec})$ such that $\mathbf{u}_{ine}^{1,\alpha} \chi_{\Omega_{ine,1}^{1,mec}} \rightharpoonup \mathbf{u}_{ine}^1$ and $\mathbf{f}_{ine}^{1,\alpha} \chi_{\Omega_{ine,1}^{1,mec}} \rightharpoonup \mathbf{f}_{ine}^1$ weakly in $\mathbf{L}^2(\Omega_{ine,1}^{\infty,mec})$, and $\mathbf{g}_{ine}^{1,\alpha} \chi_{\Omega_{ine,1}^{1,mec}} \rightharpoonup \mathbf{g}_{ine}^1$ weakly in $\mathbf{L}^2(\Gamma_{ine,1,1}^{\infty,mec})$.

Proposition 31:

Under Assumption 10 and using Proposition 22, then we obtain

$$T_{ine,1}^\varepsilon(\mathbf{u}^\varepsilon) \rightharpoonup \mathbf{u}_{ine}^{1,\alpha} + \widetilde{\mathbf{u}}^0 + \widetilde{\mathbf{u}}_{in}^{2-} \chi_{\Omega_{ine,1}^{1,mec,2}} + (\widetilde{\mathbf{u}}_{in}^{1+} + \widetilde{\mathbf{u}}_{in}^{2+}) \chi_{\Omega_{ine,1}^{1,mec,3}} + \widetilde{\mathbf{u}}_{in}^{1-} \chi_{\Omega_{ine,1}^{1,mec,4}},$$

$$T_{ine,1}^\varepsilon(\mathbf{f}^\varepsilon) \rightharpoonup \mathbf{f}_{ine}^{1,\alpha} + \widetilde{\mathbf{f}}^0 + \widetilde{\mathbf{f}}_{in}^{2-} \chi_{\Omega_{ine,1}^{1,mec,2}} + (\widetilde{\mathbf{f}}_{in}^{1+} + \widetilde{\mathbf{f}}_{in}^{2+}) \chi_{\Omega_{ine,1}^{1,mec,3}} + \widetilde{\mathbf{f}}_{in}^{1-} \chi_{\Omega_{ine,1}^{1,mec,4}},$$

weakly in $\mathbf{L}^2(\Omega_{ine,1}^{1,mec})$ and

$$T_{ine,1}^\varepsilon(\mathbf{g}^\varepsilon) \rightharpoonup \mathbf{g}_{ine}^{1,\alpha} + \widetilde{\mathbf{g}}^0 + \widetilde{\mathbf{g}}_{in}^{2-} \chi_{\Omega_{ine,1}^{1,mec,2}} + (\widetilde{\mathbf{g}}_{in}^{1+} + \widetilde{\mathbf{g}}_{in}^{2+}) \chi_{\Omega_{ine,1}^{1,mec,3}} + \widetilde{\mathbf{g}}_{in}^{1-} \chi_{\Omega_{ine,1}^{1,mec,4}},$$

weakly in $\mathbf{L}^2(\Gamma_{ine,1,1}^{1,mec})$, where $\widetilde{\varphi}^0(x^1) = \varphi^0((L_1^1, L_1^2), x^1 - 1/2)$ for $x^1 \in \Omega_{ine,1}^{1,mec}$, $\widetilde{\varphi}_{in}^{1\pm}(x^1) = \varphi_{in}^{1\pm}(L_1^2, (x_1^1 - 1/2, x_2^1, x_3^1))$ for $x^1 \in \Omega_{ine,1}^{1,mec,3} \cup \Omega_{ine,1}^{1,mec,4}$, and $\widetilde{\varphi}_{in}^{2\pm}(x^1) = \varphi_{in}^{2\pm}(L_1^1, (x_1^1, x_2^1 - 1/2, x_3^1))$ for $x^1 \in \Omega_{ine,1}^{1,mec,2} \cup \Omega_{ine,1}^{1,mec,3}$.

Proposition 32:

The limit \mathbf{u}_{ine}^1 is a solution to

$$\left\{ \begin{array}{ll} -\operatorname{div}^1 \sigma^1(\mathbf{u}_{ine}^1) = \mathbf{f}_{ine}^1 & \text{in } \Omega_{ine,1}^{\infty,mec} \\ \mathbf{u}_{ine}^1 = \mathbf{0} & \text{on } \Gamma_{ine,1,0}^{\infty,mec} \\ \sigma^1(\mathbf{u}_{ine}^1) \mathbf{n}^1 = \mathbf{g}_{ine}^1 & \text{on } \Gamma_{ine,1,1}^{\infty,mec} \\ \llbracket \mathbf{u}_{ine}^1 \rrbracket = \widetilde{\mathbf{u}_{in}^{1-}} & \text{on } \Gamma_{ine,1,interf,1}^{\infty,mec} \\ \llbracket \sigma^1(\mathbf{u}_{ine}^1) \rrbracket \mathbf{n}^1 = \sigma^1(\widetilde{\mathbf{u}_{in}^{1-}}) \mathbf{n}^1 & \text{on } \Gamma_{ine,1,interf,1}^{\infty,mec} \\ \llbracket \mathbf{u}_{ine}^1 \rrbracket = \widetilde{\mathbf{u}_{in}^{2-}} & \text{on } \Gamma_{ine,1,interf,2}^{\infty,mec} \\ \llbracket \sigma^1(\mathbf{u}_{ine}^1) \rrbracket \mathbf{n}^1 = \sigma^1(\widetilde{\mathbf{u}_{in}^{2-}}) \mathbf{n}^1 & \text{on } \Gamma_{ine,1,interf,2}^{\infty,mec} \\ \llbracket \mathbf{u}_{ine}^1 \rrbracket = -\widetilde{\mathbf{u}_{in}^{1+}} & \text{on } \Gamma_{ine,1,interf,3}^{\infty,mec} \\ \llbracket \sigma^1(\mathbf{u}_{ine}^1) \rrbracket \mathbf{n}^1 = -\sigma^1(\widetilde{\mathbf{u}_{in}^{1+}}) \mathbf{n}^1 & \text{on } \Gamma_{ine,1,interf,3}^{\infty,mec} \\ \llbracket \mathbf{u}_{ine}^1 \rrbracket = -\widetilde{\mathbf{u}_{in}^{2+}} & \text{on } \Gamma_{ine,1,interf,4}^{\infty,mec} \\ \llbracket \sigma^1(\mathbf{u}_{ine}^1) \rrbracket \mathbf{n}^1 = -\sigma^1(\widetilde{\mathbf{u}_{in}^{2+}}) \mathbf{n}^1 & \text{on } \Gamma_{ine,1,interf,4}^{\infty,mec} \end{array} \right. \quad (2.7.10)$$

Proof. We replace \mathbf{v}^ε in (2.7.3) by a smooth function $\mathbf{v}_{ine}^\varepsilon$ defined in $\overline{\Omega_{ine,1}^{\alpha\varepsilon,mec}}$ and vanishing out of $\overline{\Omega_{ine,1}^{\alpha\varepsilon,mec}}$. Then we substitute $\mathbf{v}_{ine}^\varepsilon$ by $\varepsilon^{-1} \mathbf{B}_{ine,1}^\varepsilon(\mathbf{w})$, where \mathbf{w} is in $\mathbf{C}^\infty(\Omega_{ine,1}^\sharp \times \overline{\Omega_{ine,1}^{1,mec}})$ satisfied $\mathbf{w} = 0$ on $\Gamma_{ine,1,0}^{1,mec} \cup \Gamma_{ine,1,\alpha}^{1,mec}$ and $\sigma^1(\mathbf{w}) \mathbf{n}^1 = 0$ on $\Gamma_{ine,1,1}^{1,mec,i} \cup \Gamma_{ine,1,\alpha}^{1,mec,i}$. It follows that

$$\begin{aligned} & -\frac{1}{\varepsilon} \int_{\Omega_{ine,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon \partial_j^\varepsilon (\sigma_{ij}^1(\mathbf{B}_{ine,1}^\varepsilon(\mathbf{w}))) dx^\varepsilon + \frac{1}{\varepsilon} \int_{\Gamma_{ine,1,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon \sigma_{ij}^\varepsilon(\mathbf{B}_{ine,1}^\varepsilon(\mathbf{w})) n_j^\varepsilon ds(x^\varepsilon) \\ & = \frac{1}{\varepsilon^3} \int_{\Omega_{ine,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon B_{ine,1}^\varepsilon(w_i) dx^\varepsilon + \frac{1}{\varepsilon^2} \int_{\Gamma_{ine,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon B_{ine,1}^\varepsilon(w_i) ds(x^\varepsilon). \end{aligned}$$

A straightforward calculation reveals

$$-\frac{1}{\varepsilon^3} \int_{\Omega_{ine,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon B_{ine,1}^\varepsilon(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon = \frac{1}{\varepsilon^3} \int_{\Omega_{ine,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon B_{ine,1}^\varepsilon(w_i) dx^\varepsilon + \frac{1}{\varepsilon^2} \int_{\Gamma_{ine,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon B_{ine,1}^\varepsilon(w_i) ds(x^\varepsilon).$$

Replacing $B_{ine,1}^\varepsilon$ by $T_{ine,1}^{\varepsilon*}$, thus

$$-\frac{1}{\varepsilon^3} \int_{\Omega_{ine,1}^{\alpha\varepsilon,mec}} u_i^\varepsilon T_{ine,1}^{\varepsilon*}(\partial_j^1 \sigma_{ij}^1(\mathbf{w})) dx^\varepsilon = \frac{1}{\varepsilon^3} \int_{\Omega_{ine,1}^{\alpha\varepsilon,mec}} f_i^\varepsilon T_{ine,1}^{\varepsilon*}(w_i) dx^\varepsilon + \frac{1}{\varepsilon^2} \int_{\Gamma_{ine,1,1}^{\alpha\varepsilon,mec}} g_i^\varepsilon T_{ine,1}^{\varepsilon*}(w_i) ds(x^\varepsilon).$$

By definition of the adjoint operator $T_{ine,1}^{\varepsilon*}$, we obtain

$$-\int_{\Omega_{ine,1}^{1,mec}} T_{ine,1}^\varepsilon(u_i^\varepsilon) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 = \int_{\Omega_{ine,1}^{1,mec}} T_{ine,1}^\varepsilon(f_i^\varepsilon) w_i dx^1 + \int_{\Gamma_{ine,1,1}^{1,mec}} T_{ine,1}^\varepsilon(g_i^\varepsilon) w_i ds(x^1).$$

Decomposing the integral formula into subregions $\Omega_{ine,1}^{1,mec} = \cup_{k=4}^4 \Omega_{ine,1}^{1,mec,k}$ and $\Gamma_{ine,1,1}^{1,mec} =$

$\cup_{k=1}^4 \Gamma_{ine,1,1}^{1,mec,k}$ gives

$$-\sum_{k=1}^4 \int_{\Omega_{ine,1}^{1,mec,k}} T_{ine,1}^\varepsilon(u_i^\varepsilon) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 = \sum_{k=1}^4 \int_{\Omega_{ine,1}^{1,mec,k}} T_{ine,1}^\varepsilon(f_i^\varepsilon) w_i dx^1 + \sum_{k=1}^4 \int_{\Gamma_{ine,1,1}^{1,mec,k}} T_{ine,1}^\varepsilon(g_i^\varepsilon) w_i ds(x^1).$$

Passing ε to 0 and combining with Proposition 31 gives

$$\begin{aligned} l.h.s = & - \int_{\Omega_{ine,1}^{1,mec,1}} \left(u_{ine}^{1,\alpha,1} + \widetilde{u}_i^{0-} \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 - \int_{\Omega_{ine,1}^{1,mec,2}} \left(u_{ine}^{1,\alpha,2} + \widetilde{u}_i^{0-} + \widetilde{u}_i^{2-} \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 \\ & - \int_{\Omega_{ine,1}^{1,mec,3}} \left(u_{ine}^{1,\alpha,3} + \widetilde{u}_i^{0+} + \widetilde{u}_i^{1+} + \widetilde{u}_i^{2+} \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 - \int_{\Omega_{ine,1}^{1,mec,4}} \left(u_{ine}^{1,\alpha,4} + \widetilde{u}_i^{0-} + \widetilde{u}_i^{1-} \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1, \end{aligned}$$

and

$$\begin{aligned} r.h.s = & \int_{\Omega_{ine,1}^{1,mec,1}} \left(f_{ine}^{1,\alpha,1} + \widetilde{f}_i^{0-} \right) w_i dx^1 + \int_{\Omega_{ine,1}^{1,mec,2}} \left(f_{ine}^{1,\alpha,2} + \widetilde{f}_i^{0-} + \widetilde{f}_i^{2-} \right) w_i dx^1 \\ & + \int_{\Omega_{ine,1}^{1,mec,3}} \left(f_{ine}^{1,\alpha,3} + \widetilde{f}_i^{0+} + \widetilde{f}_i^{1+} + \widetilde{f}_i^{2+} \right) w_i dx^1 + \int_{\Omega_{ine,1}^{1,mec,4}} \left(f_{ine}^{1,\alpha,4} + \widetilde{f}_i^{0-} + \widetilde{f}_i^{1-} \right) w_i dx^1 \\ & + \int_{\Gamma_{ine,1,1}^{1,mec,1}} \left(g_{ine}^{1,\alpha,1} + \widetilde{g}_i^{0-} \right) w_i ds(x^1) + \int_{\Gamma_{ine,1,1}^{1,mec,2}} \left(g_{ine}^{1,\alpha,2} + \widetilde{g}_i^{0-} + \widetilde{g}_i^{2-} \right) w_i ds(x^1) \\ & + \int_{\Gamma_{ine,1,1}^{1,mec,3}} \left(g_{ine}^{1,\alpha,3} + \widetilde{g}_i^{0+} + \widetilde{g}_i^{1+} + \widetilde{g}_i^{2+} \right) w_i ds(x^1) + \int_{\Gamma_{ine,1,1}^{1,mec,4}} \left(g_{ine}^{1,\alpha,4} + \widetilde{g}_i^{0-} + \widetilde{g}_i^{1-} \right) w_i ds(x^1). \end{aligned}$$

As in the previous proofs, we can assert that for each α ,

$$\begin{aligned} l.h.s = & - \int_{\Omega_{ine,1}^{1,mec,1}} \left(u_{ine}^{1,1} + \widetilde{u}_i^{0-} \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 - \int_{\Omega_{ine,1}^{1,mec,2}} \left(u_{ine}^{1,2} + \widetilde{u}_i^{0-} + \widetilde{u}_i^{2-} \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 \\ & - \int_{\Omega_{ine,1}^{1,mec,3}} \left(u_{ine}^{1,3} + \widetilde{u}_i^{0+} + \widetilde{u}_i^{1+} + \widetilde{u}_i^{2+} \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 - \int_{\Omega_{ine,1}^{1,mec,4}} \left(u_{ine}^{1,4} + \widetilde{u}_i^{0-} + \widetilde{u}_i^{1-} \right) \partial_j^1 \sigma_{ij}^1(\mathbf{w}) dx^1 \\ = & T_1 + T_2 + T_3 + T_4, \end{aligned}$$

$$\begin{aligned} r.h.s = & \int_{\Omega_{ine,1}^{1,mec,1}} \left(f_{ine}^{1,1} + \widetilde{f}_i^{0-} \right) w_i dx^1 + \int_{\Omega_{ine,1}^{1,mec,2}} \left(f_{ine}^{1,2} + \widetilde{f}_i^{0-} + \widetilde{f}_i^{2-} \right) w_i dx^1 \\ & + \int_{\Omega_{ine,1}^{1,mec,3}} \left(f_{ine}^{1,3} + \widetilde{f}_i^{0+} + \widetilde{f}_i^{1+} + \widetilde{f}_i^{2+} \right) w_i dx^1 + \int_{\Omega_{ine,1}^{1,mec,4}} \left(f_{ine}^{1,4} + \widetilde{f}_i^{0-} + \widetilde{f}_i^{1-} \right) w_i dx^1 \\ & + \int_{\Gamma_{ine,1,1}^{1,mec,1}} \left(g_{ine}^{1,1} + \widetilde{g}_i^{0-} \right) w_i ds(x^1) + \int_{\Gamma_{ine,1,1}^{1,mec,2}} \left(g_{ine}^{1,2} + \widetilde{g}_i^{0-} + \widetilde{g}_i^{2-} \right) w_i ds(x^1) \\ & + \int_{\Gamma_{ine,1,1}^{1,mec,3}} \left(g_{ine}^{1,3} + \widetilde{g}_i^{0+} + \widetilde{g}_i^{1+} + \widetilde{g}_i^{2+} \right) w_i ds(x^1) + \int_{\Gamma_{ine,1,1}^{1,mec,4}} \left(g_{ine}^{1,4} + \widetilde{g}_i^{0-} + \widetilde{g}_i^{1-} \right) w_i ds(x^1) \end{aligned}$$

Applying the Green's formula twice for each teams T_i yields

$$T_1 = - \int_{\Omega_{ine,1}^{1,vac,1}} \partial_j^1 \left(\sigma_{ij}^1 \left(u_{ine}^{1,1} + \widetilde{u}_i^{0-} \right) \right) w_i dx^1 + \int_{\partial\Omega_{ine,1}^{1,mec,1}} \sigma_{ij}^1 \left(\mathbf{u}_{ine}^{1,1} + \widetilde{\mathbf{u}}^{0-} \right) n_j^{1,1} w_i ds(x^1) \\ - \int_{\partial\Omega_{ine,1}^{1,1}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,1} \left(u_{ine}^{1,1} + \widetilde{u}_i^{0-} \right) ds(x^1),$$

$$T_2 = - \int_{\Omega_{ine,1}^{1,vac,2}} \partial_j^1 \left(\sigma_{ij}^1 \left(u_{ine}^{1,2} + \widetilde{u}_i^{0-} + \widetilde{u}_i^{2-} \right) \right) w_i dx^1 + \int_{\partial\Omega_{ine,1}^{1,mec,2}} \sigma_{ij}^1 \left(\mathbf{u}_{ine}^{1,2} + \widetilde{\mathbf{u}}^{0-} + \widetilde{\mathbf{u}}^{2-} \right) n_j^{1,2} w_i ds(x^1) \\ - \int_{\partial\Omega_{ine,1}^{1,2}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,2} \left(u_{ine}^{1,2} + \widetilde{u}_i^{0-} + \widetilde{u}_i^{2-} \right) ds(x^1),$$

$$T_3 = - \int_{\Omega_{ine,1}^{1,vac,3}} \partial_j^1 \left(\sigma_{ij}^1 \left(u_{ine}^{1,3} + \widetilde{u}_i^{0+} + \widetilde{u}_i^{1+} + \widetilde{u}_i^{2+} \right) \right) w_i dx^1 + \int_{\partial\Omega_{ine,1}^{1,mec,3}} \sigma_{ij}^1 \left(\mathbf{u}_{ine}^{1,3} + \widetilde{\mathbf{u}}^{0+} + \widetilde{\mathbf{u}}^{1+} + \widetilde{\mathbf{u}}^{2+} \right) n_j^{1,3} w_i ds(x^1) \\ - \int_{\partial\Omega_{ine,1}^{1,3}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,3} \left(u_{ine}^{1,3} + \widetilde{u}_i^{0+} + \widetilde{u}_i^{1+} + \widetilde{u}_i^{2+} \right) ds(x^1),$$

$$T_4 = - \int_{\Omega_{ine,1}^{1,vac,3}} \partial_j^1 \left(\sigma_{ij}^1 \left(u_{ine}^{1,4} + \widetilde{u}_i^{0-} + \widetilde{u}_i^{1-} \right) \right) w_i dx^1 + \int_{\partial\Omega_{ine,1}^{1,mec,3}} \sigma_{ij}^1 \left(\mathbf{u}_{ine}^{1,4} + \widetilde{\mathbf{u}}^{0-} + \widetilde{\mathbf{u}}^{1-} \right) n_j^{1,4} w_i ds(x^1) \\ - \int_{\partial\Omega_{ine,1}^{1,3}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,4} \left(u_{ine}^{1,4} + \widetilde{u}_i^{0-} + \widetilde{u}_i^{1-} \right) ds(x^1).$$

Decomposing $\partial\Omega_{ine,1}^{1,mec,i} = \Gamma_{ine,1,0}^{1,mec,i} \cup \Gamma_{ine,1,1}^{1,mec,i} \cup \Gamma_{ine,1,\alpha}^{1,mec,i} \cup \Gamma_{ine,1,interf,i}^{1,mec} \cup \Gamma_{ine,1,interf,i+1}^{1,mec}$ and combining with conditions satisfied by \mathbf{w} , Proposition 24 and Proposition 30, we assert that

$$- \sum_{k=1}^4 \int_{\Omega_{ine,1}^{1,mec,k}} \partial_j^1 (\sigma_{ij}^1(\mathbf{u}_{ine}^{1,k})) w_i dx^1 + \sum_{i=1}^4 \int_{\Gamma_{ine,1,1}^{1,mec,k}} \sigma_{ij}^1(\mathbf{u}_{ine}^{1,k}) n_j^{1,k} w_i ds(x^1) \\ + \int_{\Gamma_{ine,1,interf,1}^{1,mec}} \sigma_{ij}^1 \left(\mathbf{u}_{ine}^{1,1} - \mathbf{u}_{ine}^{1,4} - \widetilde{\mathbf{u}}_i^{1-} \right) n_j^{1,1} w_i ds(x^1) - \int_{\Gamma_{ine,1,interf,1}^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,1} \left(u_{ine,i}^{1,1} - u_{ine,i}^{1,4} - \widetilde{u}_i^{1-} \right) ds(x^1) \\ + \int_{\Gamma_{ine,1,interf,2}^{1,mec}} \sigma_{ij}^1 \left(\mathbf{u}_{ine}^{1,1} - \mathbf{u}_{ine}^{1,2} - \widetilde{\mathbf{u}}_i^{2-} \right) n_j^{1,1} w_i ds(x^1) - \int_{\Gamma_{ine,1,interf,2}^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,1} \left(u_{ine,i}^{1,1} - u_{ine,i}^{1,2} - \widetilde{u}_i^{2-} \right) ds(x^1) \\ + \int_{\Gamma_{ine,1,interf,3}^{1,mec}} \sigma_{ij}^1 \left(\mathbf{u}_{ine}^{1,3} - \mathbf{u}_{ine}^{1,2} + \widetilde{\mathbf{u}}_i^{1+} \right) n_j^{1,3} w_i ds(x^1) - \int_{\Gamma_{ine,1,interf,3}^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,3} \left(u_{ine,i}^{1,3} - u_{ine,i}^{1,2} + \widetilde{u}_i^{1+} \right) ds(x^1) \\ + \int_{\Gamma_{ine,1,interf,4}^{1,mec}} \sigma_{ij}^1 \left(\mathbf{u}_{ine}^{1,3} - \mathbf{u}_{ine}^{1,4} + \widetilde{\mathbf{u}}_i^{2+} \right) n_j^{1,3} w_i ds(x^1) - \int_{\Gamma_{ine,1,interf,4}^{1,mec}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,3} \left(u_{ine,i}^{1,3} - u_{ine,i}^{1,4} + \widetilde{u}_i^{2+} \right) ds(x^1) \\ - \sum_{k=1}^4 \int_{\Gamma_{ine,1,0}^{1,mec,k}} \sigma_{ij}^1(\mathbf{w}) n_j^{1,k} u_{ine,i}^{1,k} ds(x^1) = \sum_{k=1}^4 \int_{\Omega_{ine,1}^{1,mec,k}} f_{ine,i}^{1,k} w_i dx^1 + \sum_{k=1}^4 \int_{\Gamma_{ine,1,1}^{1,mec,k}} g_{ine,i}^{1,k} w_i ds(x^1).$$

The rest of the proof runs similarly as the proofs of previous parts. \square

ASYMPTOTIC MODEL BASED SIMULATION OF MIRA

3.1/ INTRODUCTION

In this chapter, we present numerical simulation results of the asymptotic models of the MIRA micro-mirror array derived in the previous chapter. We also present an analysis of the pull-in phenomenon.

The presentation of model results for the electrical behavior is done for the solution of one case of the periodic model, the inner and outer edge models, and two cases for the lateral boundary layer and interface models. Moreover, results of a periodic model are combined with results of boundary layer correctors to approximate the solution at the vicinity of an outer edge. In addition, one simulation result for the periodic model is presented for the electromechanical coupling problem. We postulate that the formula for the electrostatic force is the same as in the physical model, which is supported by the analysis conducted in [50]. As mentioned in the previous chapter, the solutions of the boundary layer models decrease exponentially away from the origin boundary. Our results even indicate that they are almost zero beyond the first cell. The boundary layer models related to the electrostatic coupling have not been implemented due to the fact that the expressions of the electrostatic forces have not yet been obtained but also due to limitations of the COMSOL software. When the analysis of their convergence will have been carried out, it will be necessary to consider that the implementation of the models should be done using an alternative software such as FEniCS, an open-source platform of simulation of PDE by the finite element method.

In the pull-in analysis, we first calculate the pull-in voltages by solving the inverse problem in COMSOL that finds an imposed voltage for the mirror to reach a given position. The pull-in voltage is the one that causes the mirror to move to the position corresponding to the one-third gap between the mirror and the electrode when no voltage is applied. This

problem is non-linear because the electrostatic force depends non-linearly on the voltage difference between the two conductors. Newton's solver is used to treat this problem. It requires a careful choice of the initial position of the mirror as well as the displacement increments in order to guarantee the convergence of the scheme. The parameterization of the simulation is very time consuming, not only because of the large number of degrees of freedom, since on average more than 30,000 triangular elements are required, but also because of the need to choose an appropriate mesh. As many different scales exist in MIRA, e.g. the beam thickness is about 20 times thinner than the distance between the mirrors, so a suitable mesh strongly determines the convergence of the solution. To succeed in this problem, the sweeping meshing method is employed to create proper meshes. We stress that the processes of seeking the good meshes by this method are done entirely by manual effort and are highly time-consuming.

The minimization for the pull-in voltage was also investigated by solving the optimization problem of the pull-in voltage with respect to the two most influential variables: the thickness of the beam and the length of the suspended beam. The optimization problem is solved using the homemade software SIMBAD with the metamodel based on twenty-five values of the pull-in voltage corresponding to a selection of beam thicknesses and lengths.

3.2/ DESCRIPTION OF MIRA DESIGN

Let us now briefly review the structure of one cell of MIRA that illustrated in Figure 3.1. It is composed of two components: the mirror part shown in Figures 3.2-3.3 and the electrode part described in Figures 3.8-3.9. In the mirror part, there is one micromirror supported with two flexible beams, which are demonstrated in Figures 3.6-3.7. These beams are attached to a frame enabling a displacement of the mirror when the voltage is applied. One stopper beam visualized in Figures 3.4-3.5 is situated under the frame to guarantee that a tilt angle satisfies a given constraint value after actuation. Two landing beams are established under the tilting edge of the micromirror to avoid the generation of a short-circuit between the mirror and the electrode throughout the actuation. The electrode part includes the electrode base where the electrostatic force is applied to attract the mirror; landing pads are where the landing beams contact; two pillars separate the mirror and electrode parts defining an electrostatic gap. All parameters of these components and their material can be found in Table 3.1, Table 3.2 and Table 3.3.

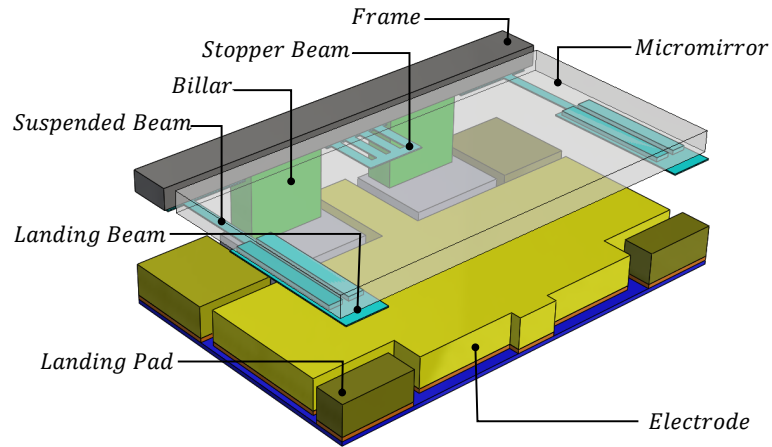


Figure 3.1: Overview of the components of a MIRA cell.

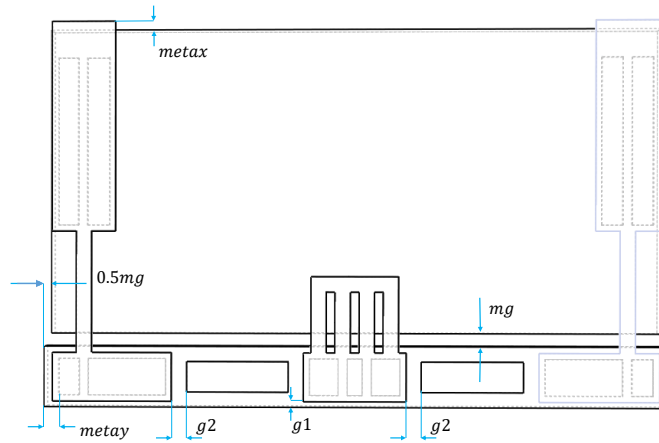


Figure 3.2: Two-dimensional representation of the mirror component.

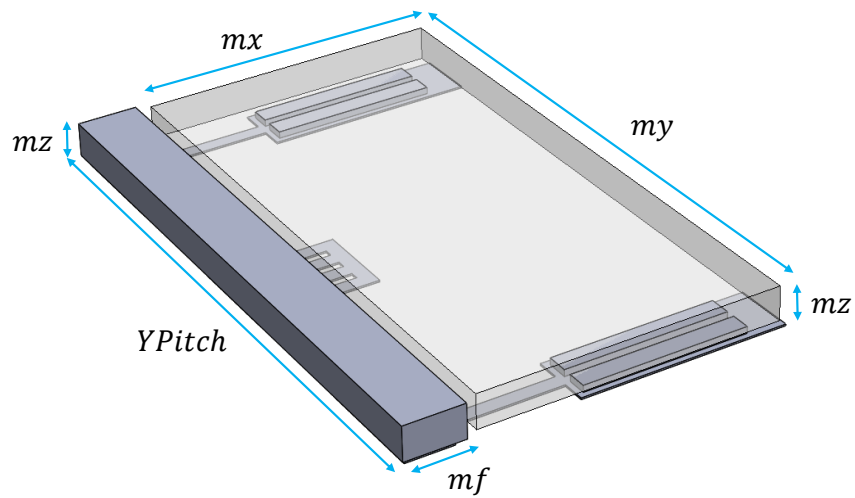


Figure 3.3: Three-dimensional representation of the mirror component.

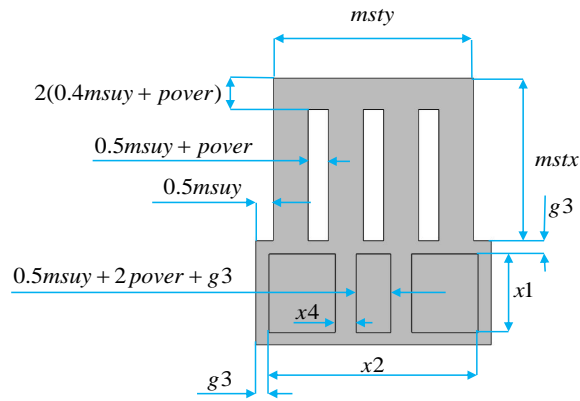


Figure 3.4: Two-dimensional representation of the stopper beam.

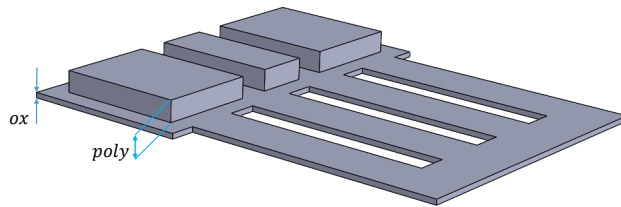


Figure 3.5: Three-dimensional representation of the stopper beam.

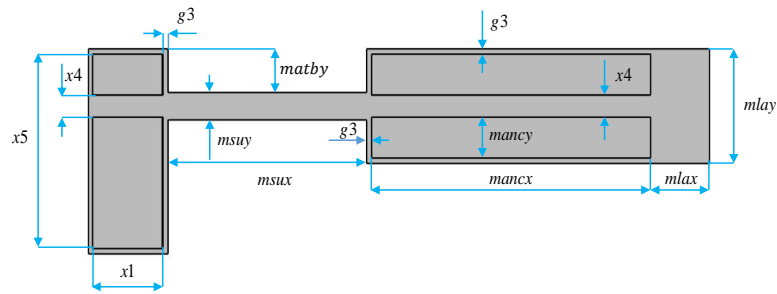


Figure 3.6: Two-dimensional representation of the right suspended beam.

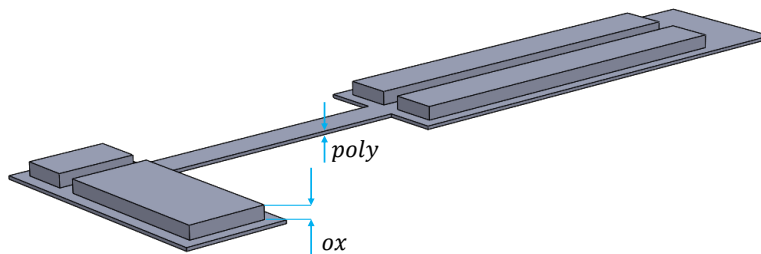


Figure 3.7: Three-dimensional representation of the right suspended beam.

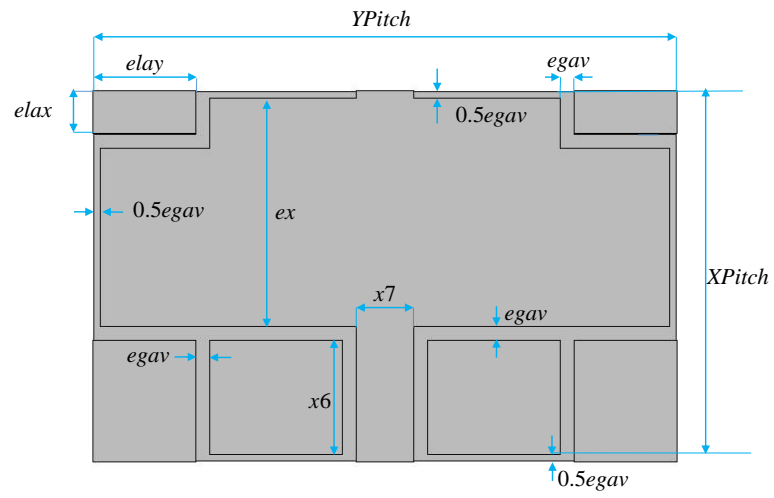


Figure 3.8: Two-dimensional representation of the electrode base.

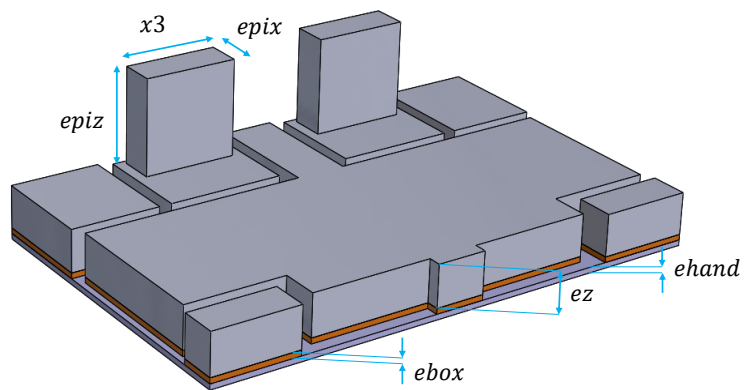


Figure 3.9: Three-dimensional representation of the electrode base.

Parameter	Value	Value Range	Description
$g1$	$2\ \mu m$		
$g2$	$6\ \mu m$		
$g3$	$2\ \mu m$		Distance between an attach and its corresponding anchor
$mpover$	$0.2\ \mu m$		The photolithography mask of the polysilicon layer to compensate the over-etching occurring during the patterning of the beam
gap	$35\ \mu m$		Distance between the micromirror and the electrode
The Handle Layer of the Electrode Wafer: Silicon			
$XPitch$	$mx + mg + mf + egav$		Width of the micromirror cell
$YPitch$	$my + egav$		Length of the micromirror cell
$ehand$	$2\ \mu m$		Thickness of the handle layer of the electrode wafer
The BOX Layer of the Electrode Wafer: Silicon Dioxide			
$ebox$	$2\ \mu m$		Thickness of the BOX Layer of the Electrode Wafer
Electrode : Silicon			
ex	$80\ \mu m$	$[40, 90]\ \mu m$	Length of the electrode
ez	$15\ \mu m$		Thickness of the electrode
$egap$	$5\ \mu m$		Width of the gap between two parts at different voltages
$elax$	$15\ \mu m$		Width of the landing pads
$elay$	$36\ \mu m$		Length of the landing pads
Pillar : Silicon			
$epix$	$14\ \mu m$		Width of the pillar of the electrode part
$epiz$	$gap - gold$		Height of the pillar
$x3$	$[YPitch - (2g3 + x2)]/2 - (metay + x5 + g3 + 2g2)$		Length of the pillar

Table 3.1: List of component parameters of MIRA's cell and their material.

Parameter	Value	Value Range	Description
Assembly: Gold			
<i>gold</i>	600 nm		Thickness of the gold layer required for the eutectic bonding
<i>goldx</i>	$epix - 2g1$		Length of the gold layer
<i>goldy</i>	$x3 - 2g3$		Width of the gold layer
Frame: Silicon			
<i>mz</i>	10 μm		Thickness of the frame
<i>mf</i>	20 μm		Width of the frame
Micromirror: Silicon			
<i>mx</i>	100 μm		Width of the micromirror
<i>my</i>	200 μm		Length of the micromirror
<i>mz</i>	10 μm		Thickness of the micromirror
<i>mf</i>	20 μm		Width of the micromirror frame
<i>mg</i>	5 μm		Width of the gap around the micromirror
<i>mcoat</i>	60 nm		The metal coating of the mirror
Anchor of the Suspended Beam Fixed under the Frame: Polycrystalline Silicon			
<i>x1</i>	$mf - 2(g1 + g3)$		Width of the anchor of the suspended beam fixed under the frame
<i>x5</i>	$msuy + mmtsy + mstsy + 4msuy + mpover$		Length of the anchor of the suspended beam fixed under the frame
<i>x4</i>	$msux - g3$		Gap between two sub-anchors
Anchor of the beam fixed under the Micromirror: Polycrystalline Silicon			
<i>mancx</i>	$mbeamx - (x1 + 2g3 + msux + g3 + mlax)$		Length of the anchor of the beam fixed under the micromirror
<i>mancy</i>	$(mlay - 2g3 - x4)/2$		Width of the anchor of the beam fixed under the micromirror

Table 3.2: Continuation of the the list of component parameters of MIRA's cell and their material.

Parameter	Value	Value Range	Description
Suspended Beams: Polycrystalline Silicon			
<i>msux</i>	$40 \mu m$	$[30, 50] \mu m$	Length of the suspended beam
<i>msuy</i>	$5 \mu m$	$[3, 6] \mu m$	Width of the suspended beam
<i>metay</i>	$5 \mu m$		Distance from the edge of the frame to the anchor of the suspended beam
<i>matby</i>	$(mlay - msuy)/2$		Distance from the attach of anchor in the frame to edge of the suspended beam
<i>poly</i>	$500 nm$	$[0.3, 0.7] \mu m$	Thickness of the beams
<i>ox</i>	$2 \mu m$	$[0.5, 2.5] \mu m$	Thickness of the sacrificial layer under the beams
Stopper Beam: Polycrystalline Silicon			
<i>mstx</i>	$25 \mu m$	$[10, 40] \mu m$	Length of the stopper beam
<i>msty</i>	$4msuy + 3(0.5msuy + 2mpover)$		Width of the stopper beam
<i>mstsy</i>	$5 \mu m$		Distance between the suspended and stopper beams for L1/LC1 model
Landing Beams: Polycrystalline Silicon			
<i>metax</i>	$10 \mu m$		Distance from the anchor tip of the landing beam to the tip of micromirror
<i>mlax</i>	$12 \mu m$		Length of the landing beam

Table 3.3: Continuation of the the list of component parameters of MIRA's cell and their material.

3.3/ ASYMPTOTIC MODEL IMPLEMENTATIONS

We now present the simulation results of the asymptotic MIRA models derived in the previous chapter. As we assumed, there are two zones in the array, the inner and outer zones driven by two different voltages $V_1 = 60V$ and $V_2 = 80V$. The periodic model, and the boundary layer models at the external boundary, the interface, the outer edge and the inner edge are implemented by the electrostatics package of the AC/DC module of COMSOL software. The results are presented in Figures 3.10, 3.11, 3.12, 3.13.

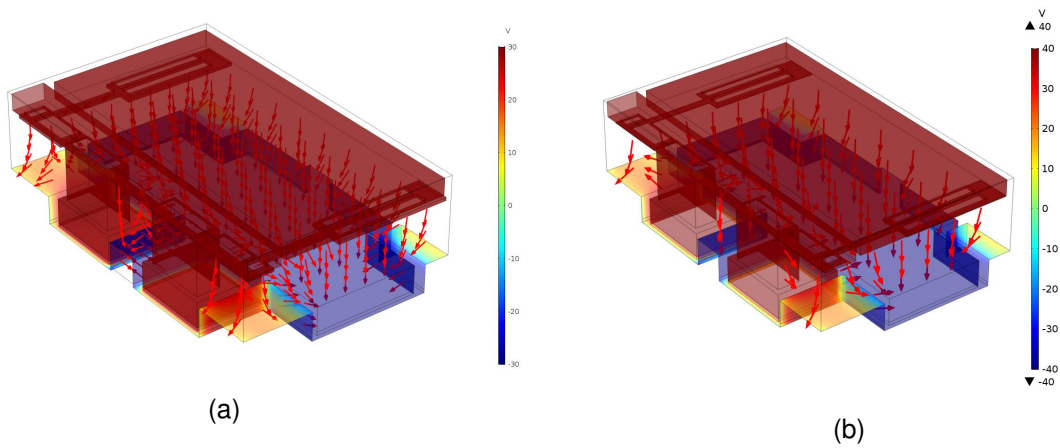


Figure 3.10: Electrical potential ϕ^0 and the electrostatic field $\nabla\phi^0$ in the internal zone (a) and in the external zone (b).

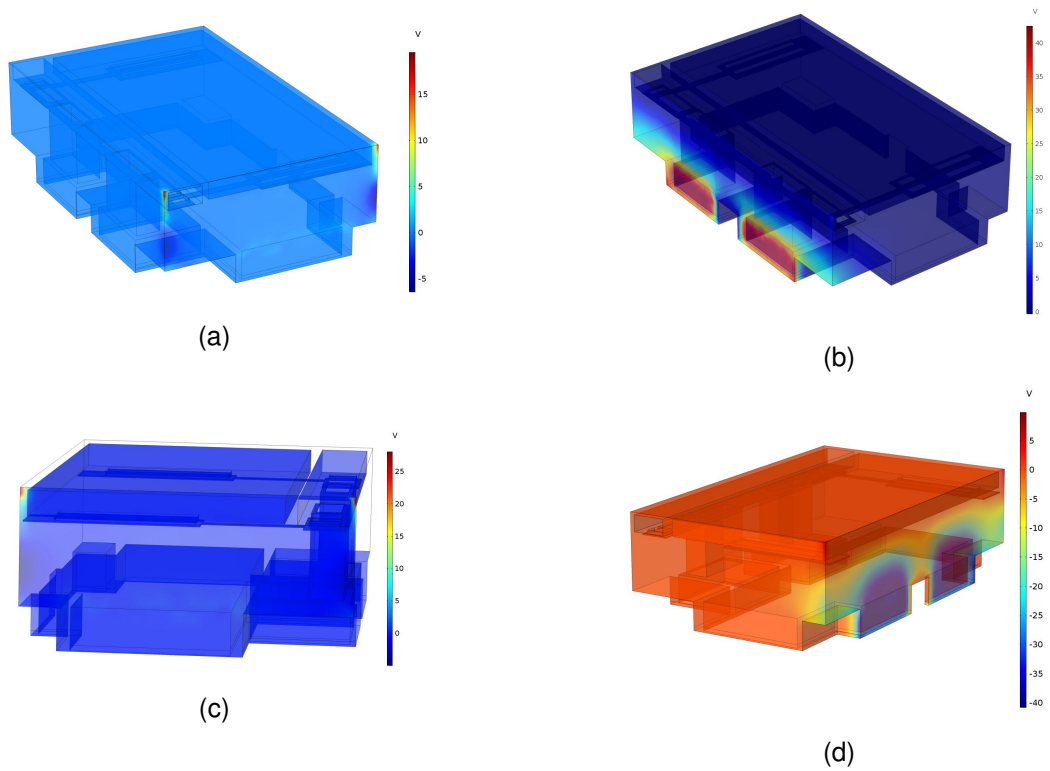


Figure 3.11: The electric potentials (a) ϕ_{bl}^1 of the first lateral boundary model, (b) ϕ_{bl}^2 of the second one, (c) ϕ_{bl}^3 of the third one, and (d) ϕ_{bl}^4 of the fourth one. The boundary layer effect is concentrated in one cell.

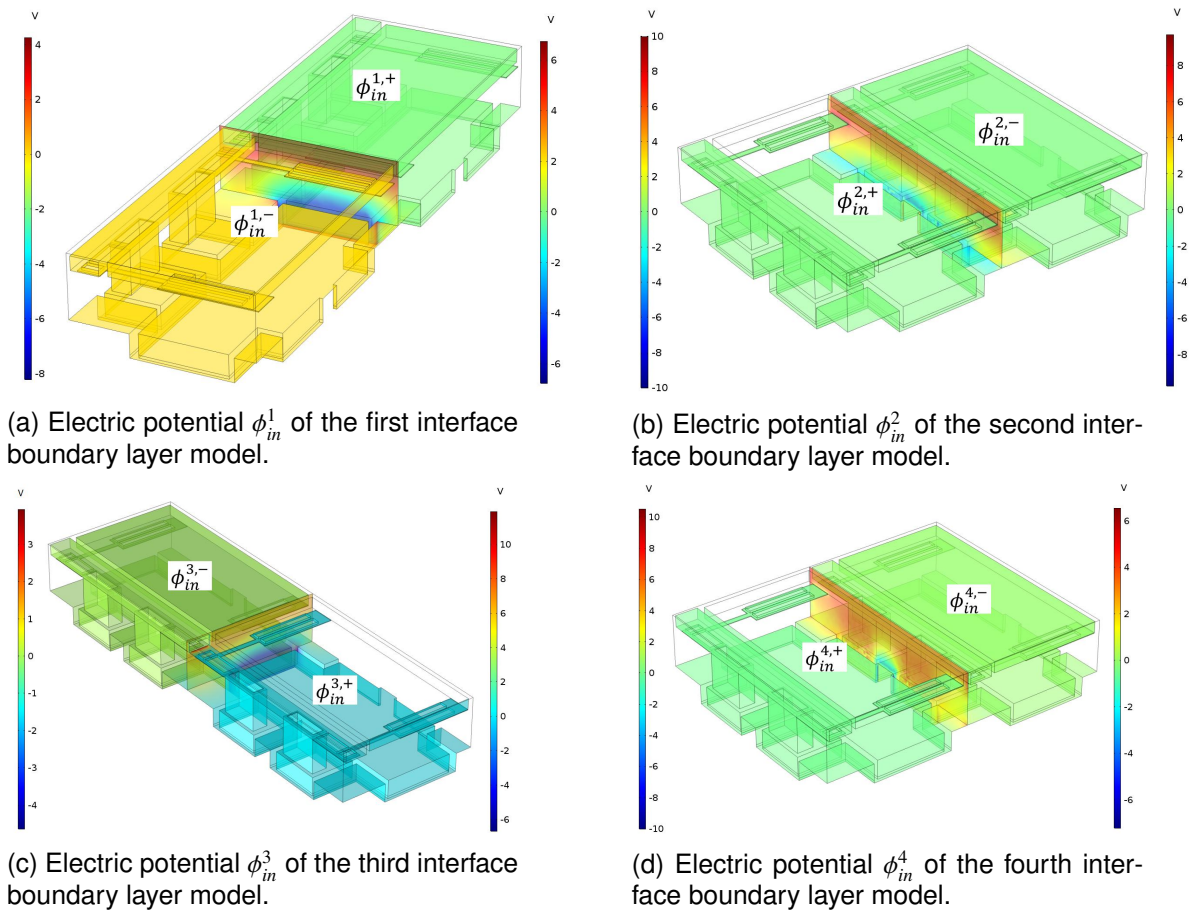


Figure 3.12: Electric potentials ϕ_{in}^i with $i = 1, 2, 3, 4$ of four interface models. Each of ϕ_{in}^i includes two parts $\phi_{in}^{i,+}$ and $\phi_{in}^{i,-}$. The mirrors are hidden to facilitate visualization of the potential values at the interfaces. The contributions are significant at the interfaces and rapidly evanescent.

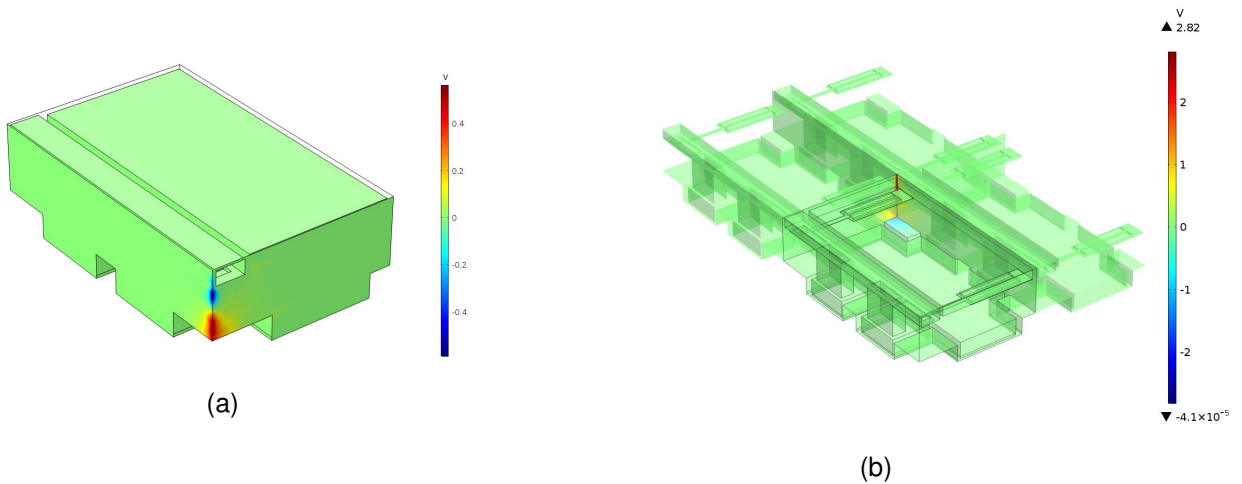


Figure 3.13: Electric potentials (a) ϕ_{exe}^1 and (b) ϕ_{ine}^1 of the outer and inner edge models. In both cases the fields are very localized and their value very small compared to the imposed voltage. It is likely that they are negligible in some cases.

Obviously, the contributions of the lateral boundary and the interface correctors are significant, while that of the remainder correctors are very slight, less than 2% in comparison with the magnitude of the applied voltage. Table 3.4 and Table 3.5 shows the comparison results of the value range of all lateral boundary and interface correctors respectively with the applied voltages.

Lateral Boundary Layer	Value Range (V)	Magnitude	Contribution (%)
First	19.6	24.2	30.25
	-4.6		
Second	42.5	42.78	53.48
	-0.28		
Third	28.1	33.06	41.33
	-4.96		
Fourth	9.88	50.58	63.23
	-40.7		

Table 3.4: Comparison of contribution of all lateral boundary correctors with the imposed voltage 80V where the magnitude is computed by the absolute value of the difference between the maximum and the minimum of the electric potential, the contribution is given as a percentage of the magnitude to the imposed voltage.

Name	Part	Value Range (V)	Magnitude	Applied Voltage(V)	Contribution (%)
First Interface	$\phi_{in}^{1,-}$	4.27	12.48	80	15.6
		-8.21			
	$\phi_{in}^{1,+}$	6.75	13.51	60	22.52
		-6.76			
Second Interface	$\phi_{in}^{2,-}$	10	16.22	80	20.28
		-6.22			
	$\phi_{in}^{2,+}$	9.74	17.01	60	28.35
		-7.27			
Third Interface	$\phi_{in}^{3,-}$	3.94	8.57	80	10.71
		-4.63			
	$\phi_{in}^{3,+}$	11.9	18.55	60	30.92
		-6.65			
Fourth Interface	$\phi_{in}^{4,-}$	10.5	20.5	80	25.63
		-10			
	$\phi_{in}^{4,+}$	6.56	13.77	60	22.95
		-7.21			

Table 3.5: Comparison of contribution of all interface correctors with the imposed voltages, the internal one 60 V and the external one 80 V.

Thanks to these results, the electrical potential in a cell near a lateral boundary, an interface, an internal and external edge can be approximated by the periodic solutions and corresponding boundary correctors. For example, Figures 3.14, 3.16, 3.15 reports solutions computed in the vicinity of lateral boundaries, of interfaces and of the first outer

edge, respectively. In the case of the internal edge, the solution was also built by the same method, but the contribution of the edge corrector being very small, it is not reported. The instruction of the COMSOL implementation of the first electrostatic interface model and the assembly of the solution near the first interface is described in Appendix I.

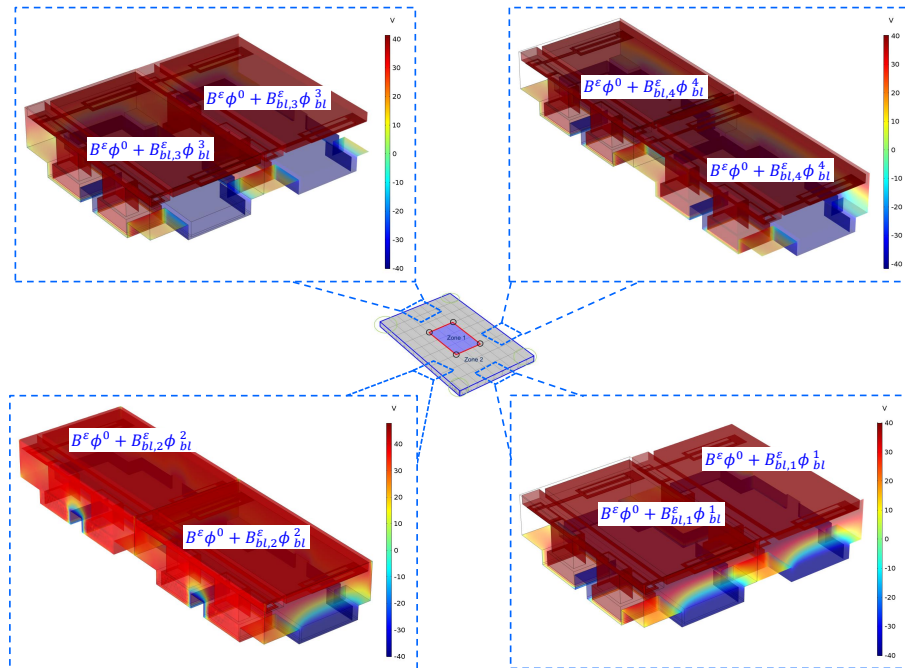


Figure 3.14: The approximate solution near lateral boundaries given by the assembly of the external periodic solution ϕ^0 and the contribution of lateral boundary correctors ϕ_{bl}^i , $i = 1, 2, 3, 4$.

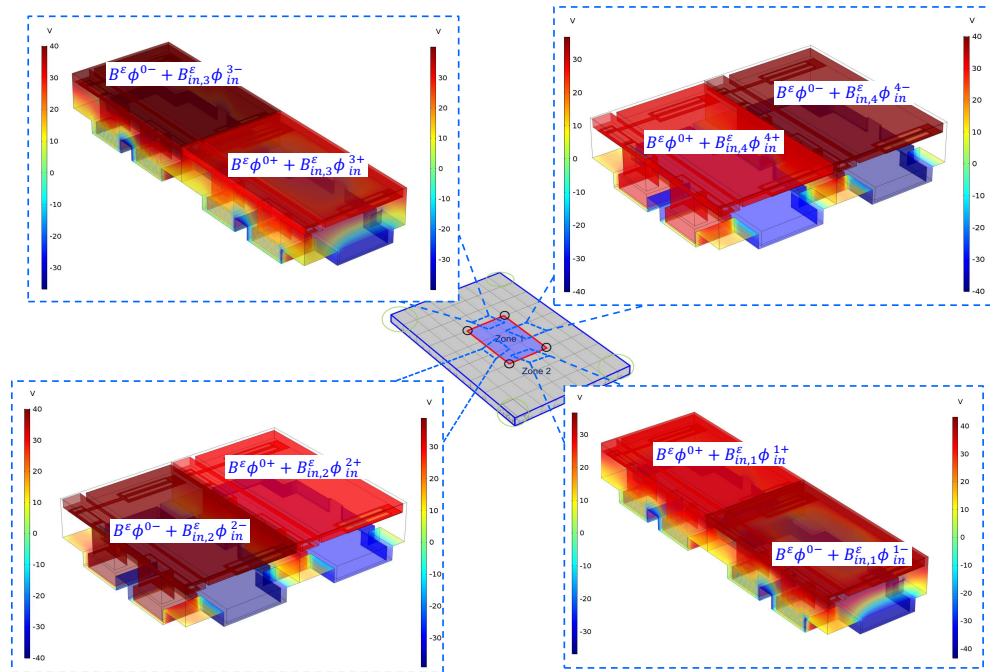


Figure 3.15: The approximate solution near the interfaces given by the assembly of the periodic solution ϕ^0 and the contribution of interface correctors ϕ_{in}^i , where ϕ^{0+} and ϕ^{0-} represents the internal and external periodic solution, ϕ_{in}^{i+} and ϕ_{in}^{i-} depicts two parts of the interface corrector ϕ_{in}^i with $i = 1, 2, 3, 4$.

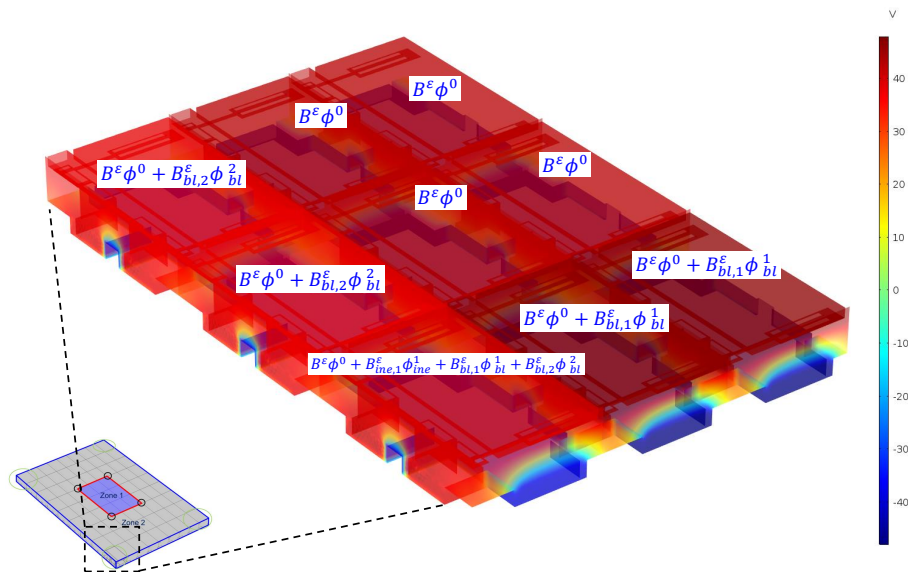


Figure 3.16: The model solution in the vicinity of the first outer edge. It is built with the external periodic solution ϕ^0 , the first and second boundary layer correctors ϕ_{bl}^1 and ϕ_{bl}^2 , and the first outer edge corrector ϕ_{exe}^1 .

Regarding the electromechanical problems, they can be solved by the *Structural Mechanical Module* in COMSOL. For instance, the periodic model of the electromechanical

coupling problem is solved by the *electromechanical interface* package of this module. We assume that the electrostatic force is $\mathbf{g}^0 = -\frac{1}{2}\epsilon_0|\phi^0|^2\mathbf{n}^1$. A result of the mechanical displacement is shown in Figure 3.17.

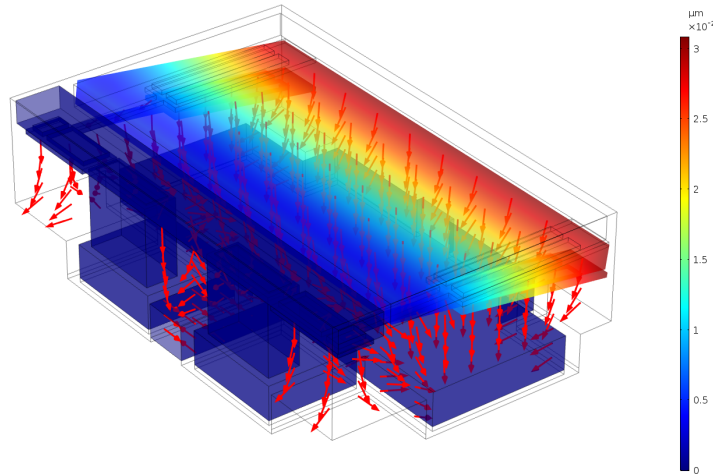


Figure 3.17: Mechanical displacement \mathbf{u}^0 of the periodic model in the internal zone with the imposed voltage $60V$. The red narrows represent the electrostatic field.

3.4/ PULL-IN ANALYSIS

3.4.1/ DESCRIPTION OF THE PULL-IN PHENOMENON

The pull-in phenomenon is one of the prevalent phenomena to be considered in MMA with electrostatic actuation, see [84, 90] for details. We briefly introduce this phenomenon by considering electrostatic actuation between two parallel plates, see Figure 3.18. A voltage source imposed between the plates induces an electrostatic force of attraction that causes the moving plate to move toward the fixed plate. The system is stable if the gap between the plates is greater than $g_0/3$, that is, if there is an equilibrium position such that the moduli of the electrostatic force and the spring force are equal. When the applied voltage is large enough that the gap between the two plates is less than $g_0/3$ then the system becomes unstable, i.e. there is no equilibrium point between the electrostatic force and the spring force, the suspended plate then moves towards the fixed plate. This phenomenon is generally named the pull-in phenomenon and the voltage to reach the position $g_0/3$ is named the pull-in voltage. In the next section, we present the method of calculating the pull-in voltage using the MEMS module of COMSOL.

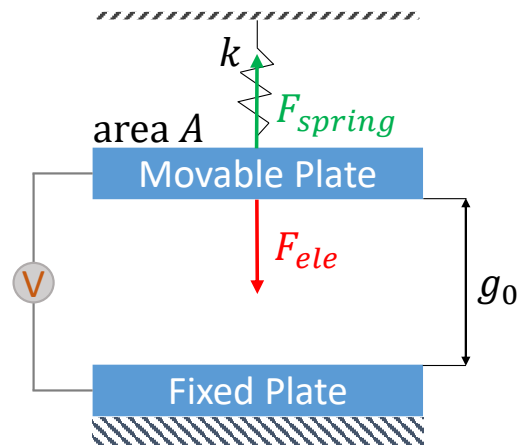


Figure 3.18: An electrode actuation with two parallel plates: the movable plate supported by the spring with the spring constant k and the fixed plate. The distance between them is g_0 and the areas of each plate is A .

3.4.2/ PULL-IN VOLTAGE COMPUTATION

The computation of the pull-in voltage is accomplished by solving an inverse problem in COMSOL. The task of COMSOL is to find a value V_{ESP} of the imposed voltage so that the gap reaches a prescribed value in a stable or unstable equilibrium state, see Figure 3.19b. In this problem, the position of the beam head is denoted as `point`. The smaller displacement corresponds with the higher position `point` and requires the higher voltage. Conversely, the larger displacement or the lower position `point` requires the smaller voltage. This pull-in voltage computation setting follows the approach of the simple example "Pull-In Voltage for a 3D Biased Resonator", see [44].

Let us take a case with the thickness of beam $poly = 400 \text{ nm}$, the length of the suspended beam $msux = 40 \mu\text{m}$, and the normal gap $g_0 = 33 \mu\text{m}$. We briefly outline the inverse problem setting. The problem statement combines the Electromechanics Interface of the Structure Mechanic Module with the Global Equations in COMSOL. In the Electromechanics Interface's setting, we set VM the applied voltage on the mirror to 1 V as in Figure 3.19b and VE that on the electrode to V_{ESP} as shown in Figure 3.19a. We note that usually V_{ESP} would be given, but here it is an unknown of the equation described in the Global Equation node. This is also the reason to refer to this problem as an inverse one.

In the Global Equation setting, we input expressions as in Table 3.6. Solving this equation is amount to the same thing as solving the inverse problem mentioned above, seeking a value of the applied voltage V_{ESP} so that the gap `intop1(point)` reaches a given value `pointZ`, `intop1(point)-pointZ=0`.

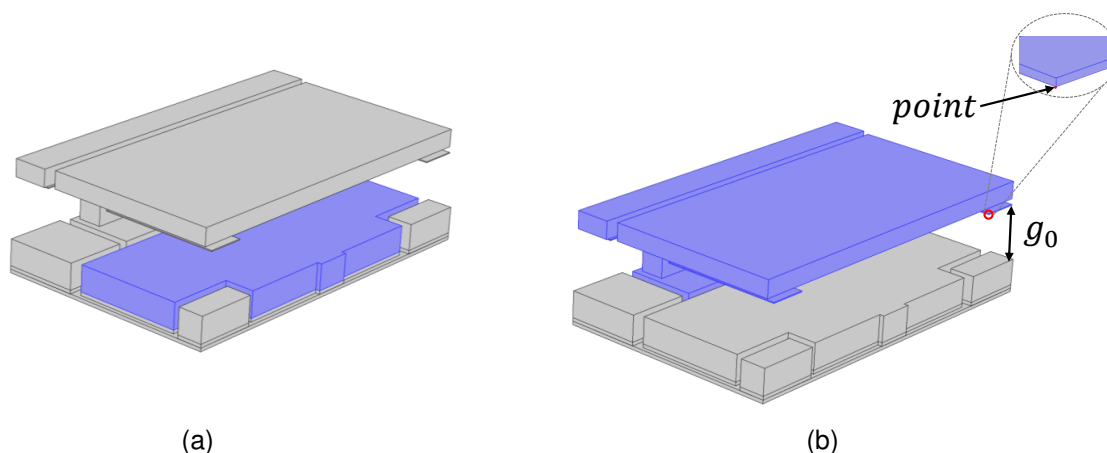


Figure 3.19: Imposed voltage locations: (a) the voltage VE is applied on the bottom electrode part and (b) the voltage VM is applied to the mirror part, which includes the mirror itself, the frames, the two golden pads, and the two pillars. The gap between the mirror and the electrode is denoted by g_0 and the position $point$ is at the end of the landing beam shown in the zoomed figure.

Name	$f(u,ut,utt,t)$	Initial value (u_0)	Initial value	Description
VESP	$intop1(point) - pointZ$			

Table 3.6: The Global Equation setting. VESP is an unknown of the equation $intop1(point) - pointZ = 0$, $intop1(point)$ being the z -coordinate position of $point$ and $pointZ$ the given position.

Obviously, this problem is non-linear, so the choice of the first value and the displacement increment of $pointZ$ must be done very carefully to guarantee the convergence of the Newtown's method. So, we set the initial value of $pointZ$ to $51.5\mu m$ with the displacement increment of $-1\mu m$ and a final value of $33\mu m$ in *Auxiliary sweep* note of Solve's setting, see Table 3.7.

Parameter Name	Parameter value list	Parameter unit
$pointZ$	$range(51.5e-6, -1e-6, 33e-6)$	

Table 3.7: The Auxiliary sweep's setting for the parameter $pointZ$.

In addition, due to the significant difference of the magnitude between the voltage and the displacement, we scale the voltage by 100 and the displacement by 10^{-5} in the solver's setting.

Moreover, as there are many different scales in the MIRA components, we use the sweeping method to generate a suitable mesh to ensure the convergence of the Newton's method. We stress that this process is entirely done by manual work and is time-consuming not only since the mesh parameters must be adjusted to accommodate each

geometry but also because the verification of whether the mesh provides the convergence of the solution is not done upfront in the problem-solving process. It is only examined for the suitability in the last values of `pointZ`.

As a result of solving the problem, we will obtain a graph with the horizontal being the value of the position `point` in the range $[51.5 : -1 : 33] \mu\text{m}$ and the vertical being the voltage `VESP` to reach this position shown in Figure 3.20. The pull-in voltage is $\max(|VESP - 1|) = 87 \text{ V}$.

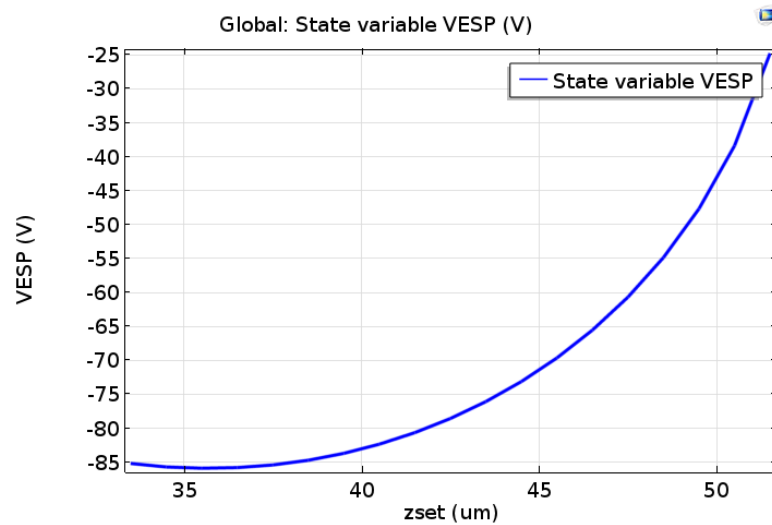


Figure 3.20: Graph of the imposed voltage with respect to the beam displacement at equilibrium for supporting beams with thickness 400 nm and length $40 \mu\text{m}$.

Pull-in voltages corresponding to a selection of lengths and thicknesses of the suspended beams are reported in Table 3.8.

<i>Poly/msux</i>	$40 \mu\text{m}$	$60 \mu\text{m}$	$80 \mu\text{m}$
400 nm	87 V	81 V	82 V
500 nm	120 V	115 V	115 V
600 nm	157 V	150 V	154 V

Table 3.8: Pull-in voltage values with respect to some lengths and thicknesses of the suspended beams.

3.4.3/ OPTIMIZATION OF THE PULL-IN VOLTAGE

Design optimization is a critical step prior to the manufacturing phase. In our scope, we consider only one issue to be optimized, which is the pull-in voltage V_{PI} with the goal of reducing the electrical operating cost. It can be recognized that the value of imposed voltage to actuate the array is dependent on two significant parameters, the thickness of

beams $poly$ and the length of suspended beams $msux$ because both of them significantly affect the beam's restoring force. Besides, reducing the actuation cost requires the pull-in voltage of each cell to be smaller than $140V$. Based on the above considerations, we address the constraint optimization problem with the single objective V_{PI} , two variables $poly$ and $msux$, and the constraint $V_{PI} \leq 140$.

Solving an optimization problem is typically an iterative process. Each step of the process requires a modification of the variables and therefore demands a new value for the objective function. In fact, in our problem, the process of updating the objective function value, the pull-in voltage, corresponding to the various variables, the thickness and the length of beams, is hugely time-consuming and almost impossible. The reason is that the pull-in calculation mentioned above is a non-linear invert problem, and the mesh generation is done manually. The average time to calculate one pull-in voltage value can vary from 8 hours to 24 hours. To avoid this difficulty, it is necessary to construct an approximate response with an acceptable error where the input is the two variables and the output is the pull-in voltage. After that, we will perform the optimization for the approximation response. With this approach, we utilize the surface response model [12] as the members of the metamodels or surrogate models, see [73, 86, 57] more details.

The surface response model for the pull-in voltage is built based on twenty-five samples corresponding to the five value variations of $poly \in \{400, 450, 500, 550, 600\}$ and of $msux \in \{40, 50, 60, 70, 80\}$. The result of response surface is shown in Figure 3.21 with the mean square error 1.8%. Then, the constrained optimization process is performed for this model with the single objective V_{PI} , the two variables $poly$, $msux$, the constraint $V_{PI} \leq 140V$, and the initial values $poly = 500nm$, $msux = 60\mu m$, $V_{PI} = 115.2V$. We find that the minimal value of $74.4V$ of the pull-in voltage V_{PI} is reached for $poly = 400nm$ and $msux = 70.16\mu m$. The full processing is accomplished with the home-made software SIMBAD.

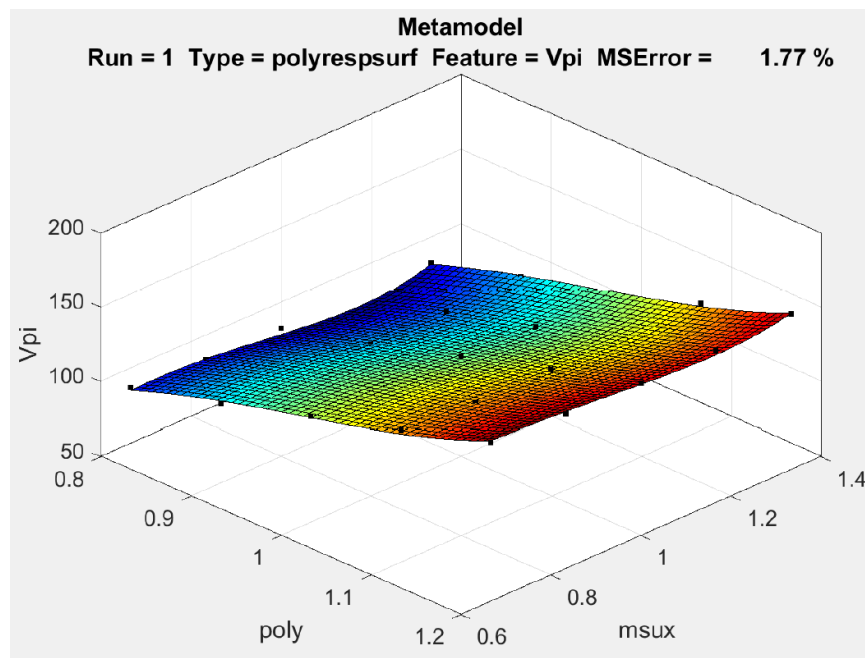


Figure 3.21: The surface response model for the pull-in voltage V_{PI} as a function of the suspended beam thickness and length variables $poly$ and $msux$. The main slope is in the direction of $poly$ while the surface is flat in the direction of $msux$, so the thickness of the beam is the variable that dominates this effect.

3.5/ EFFECT OF A NEIGHBOUR IMPOSED VOLTAGE

As mentioned in the introduction of the MIRA project, the array is applied for Multi-Object Spectroscopy (MOS), more precisely, it plays a role as the reflective field selector in the telescope. For this goal, the line-column addressing algorithm will be determined to operate the actuation of the MIRA. Each micromirror line and each electrode line is connected to separate voltages. One of the natural questions is whether there is an effect of the electrostatic field of an active cell on an inactive neighbour? In this section, we study the cross-talk effect i.e how an imposed voltage of one cell affects the displacement of a mirror of adjacent cells.

To investigate this effect, the interface electromechanical coupling model should be implemented. This leads to a difficulty in model implementation in the software COMSOL, which is not able to couple models in the electromechanical interface package. One alternative approach to overcome this difficulty is to integrate manually two modules, the electrostatics module and the solid mechanics one. In detail, the former is deployed to estimate the electrostatic force on a particular area of the mirror of the inactive cell caused by the electrostatic field of the active cell. Then the latter is employed to derive the mechanical displacement of the inactive cell yielded by this force as the boundary load on the specific surface of its mirror.

We now only consider the case where the effect of the electrostatic field of an active cell on an inactive one is the most significant, the other cases could be also investigated by the same method. Obviously, the numerical simulation results of the boundary models in Section 3.3 shows that the strongest effect is in the case where two cells are located on the same electrode line. Specifically, the right cell is assumed to be activated with an applied voltage 80 V while the left one is not.

Figure 3.22 presents the impact of the electrostatic field of the right cell on the left one. It is clear that the influence region of the electrostatic field is local and mainly concentrates in the vicinity of the interface. In addition, Figure 3.23 demonstrates that the affected area on the mirror of the inactive cell is not larger than $3 \times m_y (\mu\text{m}^2)$ where $m_y = 200\mu\text{m}$.

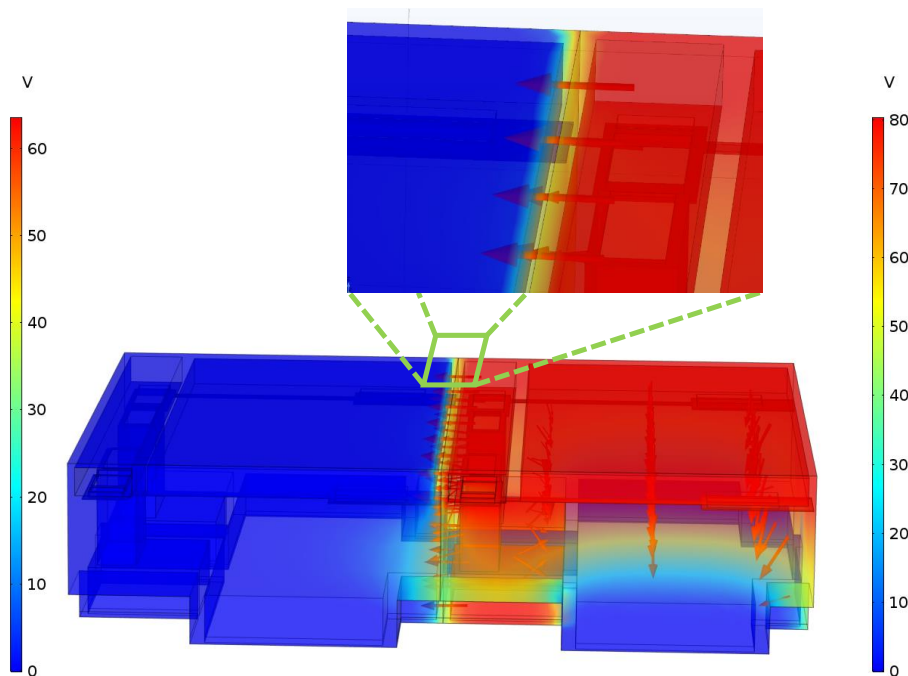


Figure 3.22: Effect of the electrostatic field of the right cell on the neighbour cell. The zoom represents the electrostatic field around the frame of the active cell with the applied voltage 80 V and the head of the mirror of the inactive one.

The average value along the edge of the underside of the mirror of the electrostatic force is estimated. This is applied in the solid mechanics model as a boundary load on the same surface of the mirror's underside, see Figure 3.23. We emphasize that the force is computed on the mirror area where the electrostatic field has a significant value. On the rest, its value is ignored since the effect on these surfaces is insignificant. As a result, the mirror is tilted about 2° corresponding to the mechanical displacement depicted in Figure 3.24.

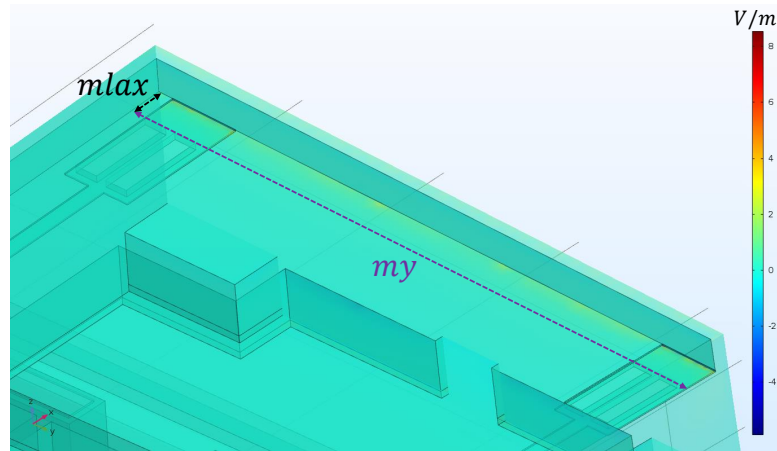


Figure 3.23: Affected location from the neighbour electrostatic field on the underside of the mirror of the inactive cell with an estimate area of $0.25 \times mlax \times my = 600 (\mu m^2)$ where my represents the length of the mirror and $mlax$ expresses that of the landing beam.

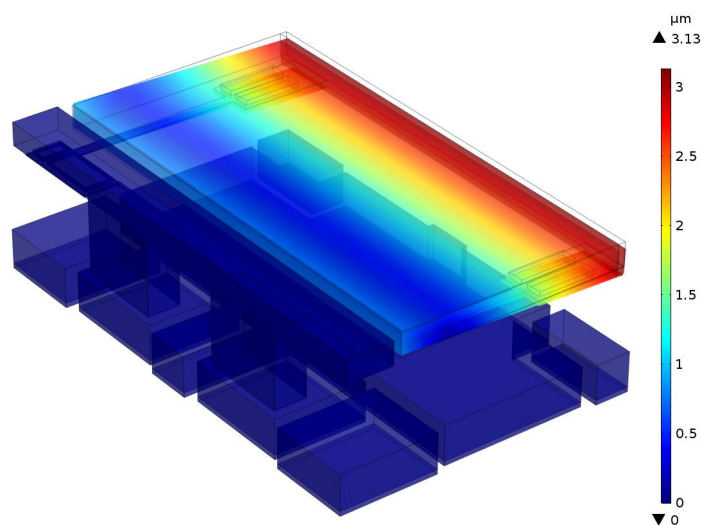


Figure 3.24: Mechanical displacement in the inactive cell with an order in the range of 10% of the gap between the electrode and the mirror.

DATA STRUCTURE AND REWRITING STRATEGIES FOR MULTISCALE MODELS CONSTRUCTION

The two previous chapters were devoted to the construction of multiscale models and their implementation in general-purpose simulation software. We have deliberately chosen to consider relatively complex models on complex geometries. The rest of this thesis is devoted to the redesign of some components of the MEMSALab software for which weaknesses have been identified and to the design of new ones that should meet newly identified needs. It remains in the same spirit as the beginning with the objective that MEMSALab can perform model building in complex configurations. At this stage we distinguish two types of difficulties to be treated for the construction of models, one inherent to the complexity of the problems and the other concerning the steps of reasoning. An experience has already been acquired on the latter during the development of the preliminary version of MEMSALab. Thus, it was decided to devote the rest of the work to the question of representation of general and complex models and by limiting ourselves to their construction by the direct method described in the introduction of the thesis.

4.1/ INTRODUCTION

During the first year of the thesis, the model building algorithm used in the proofs was implemented for relatively simple configurations, using strategies and extensions, cf [21]. As the software evolved, we realized that their modification entails a modification of the parser, of several functions of the kernel including the display functions, as well as of some unitary tests. Moreover, we realized that the strategies operating on the constructors are necessarily specific to them.

This way of proceeding proved to be too constraining for a complex software development

phase for which there are still many uncertainties. Moreover, it appeared that it requires a significant number of strategy definitions which does not seem justified.

On the other hand, the PDEs were built following the top-down hierarchy: `pde`, `equation`, `equality`, `operator`, `function`, `variable`, `domain`, and `boundary`. This approach proved to be well suited for the application of the mathematical properties used in the model proving algorithm. Nevertheless, with such a hierarchy, the definition of a PDE is too heavy a task to be entrusted to a novice.

These observations led to the following requirements for MEMSALab.

- (i) Regarding the data structure, it should be suitable for uniform processing on expressions and strategies and should be easily adaptable when performing minor changes.
- (ii) Strategies should be small in number and general enough while still being able to perform complex operations.
- (iii) The definition of a PDE and its components by a programmer should not require expertise i.e. it must be a simple operation.
- (iv) Multiscale models should be represented by a recursive structure with an arbitrary number of nested or non-nested scales. The top-down hierarchical structure of PDEs should be retained. The data structure should be able to host both the models themselves and the features used for their construction. Finally, it should be possible to express complex models on complex domains.

The needs (i-ii) for data structure and strategies led to the creation of data structures in the form of abstract syntax trees (ASTs) usable by the MEMSALab programmer, whereas ASTs already existed but were reserved to the rewriting kernel in OCAML. For reasons that it would take too long to explain here, the ASTs of the kernel are not usable in the specific MEMSALab language. The new ASTs constitute a simple and general data structure that fits with the requirements. In addition, for the simplicity of programming, each type of node in the AST is associated with a function that acts as a shortcut. The latter is optimally parameterized according to the needs. Such functions constitute what we will call concrete syntax trees. The two syntaxes are detailed in Section 4.2.

The fact that the strategies operate on ASTs allows to reduce their number and to define only a small number of strategies that are both general and powerful. These strategies are detailed in Section 4.3.

For an elegant way (iii) to define PDEs, we combine functional programming principles available in OCAML with rewriting strategies. This aspect is not discussed further in the thesis.

Regarding (iv), multiscale models and their characteristics are represented using two main nodes "subModel" and "subScale" from which recursion is built allowing an arbitrary number of nested scales as well as a variety of models related to the same scale such as the four types of boundary layer models in Chapter 2.

The representation of multiscale models and the features that allow their construction are illustrated also in Section 4.4 for the five models in Chapter 2. The construction of models using their features is described as meta-algorithms.

Finally, it is important to mention that the choice of expressing data in the form of ASTs greatly facilitates the construction of the concurrent rewriting and description logic system (RDL) introduced in Chapter 5.

4.2/ NODES OF ABSTRACT SYNTAX TREE FOR MULTISCALE MODEL REPRESENTATION

This part is devoted to the presentation of the abstract tree nodes `node(.)` of the data structures (also called expressions) used to represent PDEs and their multiscale models and `nodeS(.)` of the strategies that operate on them. The grammar of expressions `e` and strategies `s` useful in this part is detailed below. The complete grammar is detailed in Section 5.3 of Chapter 5.

$$\begin{aligned} e &:= n \mid c \mid x \mid \text{node}(e, [e, \dots]) \\ e' &:= e \mid \text{globalVar}(e) \\ s &:= \text{nodeS}(e, [e', \dots, s, \dots]) \end{aligned}$$

with

- `n` a number,
- `c` a string,
- `x` a rewriting variable,
- `globalVar(e)` the content of a memory (also called global variable) whose name is `e`.

As described in the above grammar, an expression `e` is either a terminal expression or a labeled node `node(name, list)` composed of a name and a list of nodes. We assume that the root of an expression starts with a node and not with a terminal expression and that the lists are homogeneous in the sense that they do not mix terminal and non-terminal nodes.

The main nodes that describe PDEs are `index`, `boundary`, `region`, `variable`, `function`, `operator`, `equality`, `equation`, `bc` (boundary condition), `ic` (initial condition), `pde` (partial

differential equation) and `model`. Table 4.1 summarizes them together with their subnodes. Note that the empty node, a node with name "emptyNode" and with an empty list, is noted as \emptyset and replaces any subnode that is not necessary.

nodes	Subnodes
<code>emptyNode</code>	<code>[]</code>
<code>index</code>	<code>[name, range, quantifier]</code>
<code>ident</code>	<code>[name, type, scale, level]</code>
<code>paramReg</code>	<code>[axisList, sizeList, periodicity, interfaceList]</code>
<code>region</code>	<code>[ident, paramReg, subRegionList, boundary]</code>
<code>variable</code>	<code>[name, index, region]</code>
<code>function</code>	<code>[name, indexList, varList, type]</code>
<code>operator</code>	<code>[name, indexList, mathExprList, inputVarList, outputVarList, paramOper]</code>
<code>equality</code>	<code>[name, leftMathExpr, rightMathExpr]</code>
<code>equation</code>	<code>[name, equality, varList]</code>
<code>bc</code>	<code>[name, equality, bVarList, dVarList]</code>
<code>ic</code>	<code>[name, equality, bVarList, dVarList]</code>
<code>pde</code>	<code>[name, equationList, bcList, icList]</code>
<code>model</code>	<code>[name, pdeList]</code>

Table 4.1: The main nodes of the grammar that represents PDEs. The names of nodes ending in `List` are nodes containing lists of the same record (possibly with a truncated name) with an unspecified number of elements. For example, `varList` is a node containing a list of nodes of type variable.

Name	Type	Default Value
name	constant	
range	constant	
quantifier	constant	
type	constant	GlobalVar("type")
scale	constant	GlobalVar("scale")
level	constant	GlobalVar("level")
axis	constant	GlobalVar("axis")
sizeList	constant	GlobalVar("sizeList")
indent	index	
boundary	region	
subBoundaryList	region	
domain	region	
subDomainList	region	
normal	function	
indexList	index	
reg	region	
varList	variable	
type	constant	
mathExpr	expression	
inputVarList	variable	
outputVarList	variable	
paramOper	constant	
leftMathExpr	expression	
rightMathExpr	expression	
equalityList	equality	
equationList	equation	
bcList	bc	
icList	ic	
pdeList	pde	

Table 4.2: The subnodes of the main nodes of Table 4.1, their types and their default value if they have one. The type constant means a terminal node.

Rather than the detailed writing of a node `node(name,list)` we often prefer a shorter writing in the form `name(list)` of a function whose name is the name of the node. For instance, an unknown function $u(x)$ of an equation is represented by `node("function",[name, index, variable, type])`, which one will write more sim-

ply in the form `function([name, index, variable, type])`. Its parameters are the nodes `name("u")`, `variable(name("x"), index, region)` where the index and region nodes are not explained here, and `type("unknown")`. As can be seen in these examples, when the list of a node contains only one element, the square brackets are often omitted.

It is common to use lists from the same node. To do this, each node `f` is associated with a node for the list of these nodes called `fList`. Thus, the node for index lists is `indexList`, the node for pde lists is `pdeList` and so on. An exception is made for the variable lists which is named `varList` for shortness.

It is sometimes convenient to parametrize identifiers rather than create new names. This is done by using OCaml's association lists, and keys that are lists are appended to the identifier name within brackets. For example an identifier $f(x)$ of a function depending on a function and parametrized by the parameters p, q will be noted $f\{[p, q]\}(x)$.

It is noteworthy to mention that elementary mathematical operations such as addition, multiplication, inverse and special functions such as *cosine*, *sine*, etc. are also nodes, but are not discussed here. Even if these mathematical expressions are not completely described by the presented node system, nevertheless they are referred to as "mathematical expressions" in the description of the operator and equality nodes. As their writing as nodes is usually rather complicated, we have introduced the possibility to write them in a form close to mathematics and their node form is automatically generated. For example, $\sum_i \frac{\partial^2 u}{\partial x_i^2}(x)$ is the Laplace operator applied to a function u and can be written `$\sum _i \partial(u(x([]))/\partial x(i))/\partial x(i)$` in a program.

The hierarchical structure in nodes is uniform and general. It is therefore very convenient for the use of general strategies, however, it remains too cumbersome to handle the description of models. An intermediate layer is defined in order to couple the nodes with a judicious choice of useful parameters of their subnodes for the direct construction of models. For example the function $u(x)$ will be defined by `function00("u", [], x, "unknown")` where `x` is a node corresponding to the variable x , which could be defined by `variable00("x", ind, nameD)` with `ind` an index parameter, see Section 4.2.1, and `nameD` the region name where x is defined. We call "Concrete Syntax Trees" this representation. Table 4.3 summarizes the parameters of the functions of this grammar which are those directly used in the construction of the models.

nodes	Parameters
index00	name0,rangeList,quantifier0
ident00	name0,type0,scale0,level0
paramReg00	axisList0,sizeList0,periodicityDirection0,periodicityScale0,interfaceList0
paramB00	axisList0,sizeList0
paramD00	axisList0,sizeList0,periodicityDirection0,periodicityScale0,interfaceList0
boundary00	nameB,axisList0,sizeList0,subBoundaryList0
domain00	nameD,axisList0,sizeList0,periodicityDirection0,periodicityScale0,interfaceList0,subDomainList0,boundary0
variable00	nameV,ind,nameD
function00	nameF,indList,varList0,type0
operator00	nameOp,indList,mathExprList0,inputVarList0,outputVarList0,operParam0
equality00	nameE,leftMathExpr0,rightMathExpr0
equation00	nameEq,equality0,varList0
bc00	nameBc,equality0,bVarList0,dVarList0
ic00	nameIc,equality0,bVarList0,dVarList0
pde00	pdeName,equationList0,bcList0,icList0
model00	nameModel,pdeList0

Table 4.3: List of parameters used for the concrete syntax trees.

In what follows, we describe the nodes of the abstract syntax trees summarized in Table 4.1 as well as the function of the concrete syntax trees.

4.2.1/ INDEX

The index node

```
node("index", [name, range, quantifier])
```

is used to describe vectors of variables, functions and operators, but also space dimensions, and so on.

1. name(name0) a node for the index name name0 which is a string.
2. range(range0) a node for the index range range0 that is either a single integer or a list of integers.
3. quantifier(quantifier0) a node for the index quantifier quantifier0 that is one of the strings "Given", "For all", "There exist".

Example: An index i varying in $\{1,2\}$ is defined by

```
i : node("index",[name("i"),range([1,2]),quantifier("Given")])
```

For simplicity, we define the function `index1(ind)` that reproduces the multiple manner to use an index. It returns different forms of the `index` node depending on the value of `ind`,

- "noIndex" returns an empty node,
- an integer returns `name0 = ""` and `range0 = ind`,
- a string returns `name0 = ind` and `range = GlobalVar(axisNumbers)`
- an index node returns the same node.

This construction requires that the global variable `axisNumbers` has been previously defined, see Section 4.2.3.

The function of the concrete syntax corresponding to the node `index` is defined as follows.

```
Expression index00(name0,rangeList,quantifier0) : node("index",
[name(name0),range(rangeList),quantifier(quantifier0)]).
```

Example: We redefine the index i defined above.

```
i : index00("i",[1,2],"Given")
```

4.2.2/ IDENT

The `ident` node

```
node("ident",[name,type,scale,level])
```

is for identification of the regions, the submodels and their subscale part.

1. `name(name0)` a node for the `ident` name `name0` which is a string.
2. `type(type0)` a node for the type `type0` of the region that takes the value "omega" for a domain and "gamma" for a boundary. The subnode `type` is left as an empty node in the nodes `subModel` and `subScale`.

3. $\text{scale}(\text{scale0})$ a node for the scale scale0 in a multiscale model which is a non negative integer. The physical domain is at $\text{scale0}=0$. In a multiscale model, the first scale is $\text{scale0}=1$ and then scale0 increases with the depth, see Figure 4.1.
4. $\text{level}(\text{level0})$ a node for the level level0 which is one of the three strings "micro" or "macro". For each $\text{scale0} \geq 1$ the distinction between a micro or macro region is with the parameter level0 , see Figure 4.1.

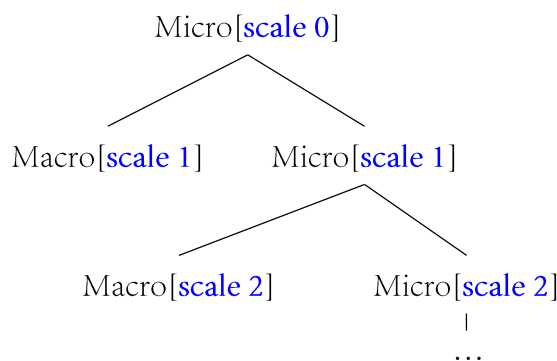


Figure 4.1: The system of representation of multiscale geometries using the parameters scale and level . multiscale models. The scales are written in blue while the levels are in black.

Example: The domain $\Omega_{\text{main}}^{0,\text{micro}}$ and its boundary in Figure 4.5 have the following identifiers $\text{identD}\{["\text{main}"]\}$ and $\text{identB}\{["\text{main}"]\}$.

```

identD{["main"]} : node("ident",[
    name("main"),type("omega"),scale("0"),level("micro")]
identB{["main"]} : node("ident",[
    name("main"),type("gamma"),scale("0"),level("micro")]
identB{["int"]} : node("ident",[
    name("int"),type("gamma"),scale("0"),level("micro")]
identB{["ext"]} : node("ident",[
    name("ext"),type("gamma"),scale("0"),level("micro")]
  
```

The function of the concrete syntax corresponding to the node ident is defined as follows.

```

ident00(name0,type0,scale0,level0) : node("ident",
    [name(name0),type(type0),scale(scale0),level(level0)]).
  
```

We observe that the values of scale0 and level0 are not in the parameters of ident00 . They are assumed to be declared and used as environment variables in a way not detailed here.

We now define the identifier `idSC`, as a special case of `ident`, used in the nodes `subModel` and `subScale` of Sections 4.2.13 and 4.2.14: `idSC(name0,scale0,level0)` : `ident00(name0,0,scale0,level0)`.

4.2.3/ PARAMREG

The `paramReg` node

```
node("paramReg",[axisList,sizeList,periodicity,interfaceList])
```

gathers geometry characteristics of a region such as its size, dimension, periodicity and also its interfaces with other regions.

1. `axisList(axisList0)` a node for the list `axisList0 = [axis(axis0),...]` of vector coordinates generating the subspace where the region is defined. For instance, `[axis([1,0,0])]` is for a one dimensional region in the O_x direction, `[axis([1,0,0]), axis([0,1,0])]` for a two-dimensional region in the (O_x, O_y) -plane and `[axis([1,0,0]), axis([0,1,0]), axis([0,0,1])]` for a three-dimensional region.
2. `sizeList(sizeList0)` a node for the list `sizeList0 = [size0,...]` of orders of magnitude of the size of the region in the directions of the axes in `axisList`. These orders of magnitude are expressed in terms of powers of asymptotic symbolic parameters as ε .
3. `periodicity([periodicityDirection,periodicityScale])` a node to describe the region periodicity properties in the following nodes. This node is for domains only.
 1. `periodicityDirection(periodicityDirection0)` a node for the list `periodicityDirection0` of the size of `axisList0`, each element takes one of the value "P" for periodicity, "T" for thin and "N" for none.
 2. `periodicityScale(periodicityScale0)` a node for the list `periodicityScale0` of the period or thickness scales in each direction of `axisList0`. Each scale is an expression of asymptotic parameters as ε .
4. `interfaceList(interfaceList0)` a node for the list `interfaceList0 = [interface,...]` of interfaces with other regions where the node `interface([boundaryNameList, neighborDomainName])` is defined with the following nodes. This node is for domains only.

1. `boundaryNameList(nameBList)` a node for the list `nameBList = [name(nameB), ...]` of interface names between the current domain and its neighbor.
2. `neighborDomainName(nameD)` a node for the name `nameD` of the neighbor domain.

Example: The following code represents the interfaces $\Gamma_{interface,1,2}^{0,micro}$ and $\Gamma_{interface,1,3}^{0,micro}$ between the domain $\Omega_1^{0,micro}$ and its neighbors $\Omega_2^{0,micro}$ and $\Omega_3^{0,micro}$ of Figure 4.2.

```
interfaceList1 : interfaceList([
  interface([boundaryNameList("interface,1,2"), neighborDomainName("2")]),
  interface([boundaryNameList("interface,1,3"), neighborDomainName("3")])])
```

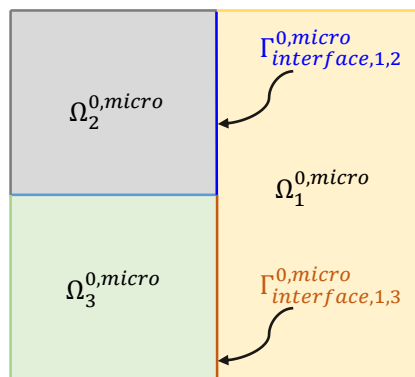


Figure 4.2: Interface between three domains.

In order to construct the `index1` function used in Section 4.2.1, the environment variable `axisNumbers` is constructed as a list `[1,2,...]` of the axis numbers declared in `axisList`.

Example: The definition of the nodes `paramReg` for the domain $\Omega_{main}^{0,micro}$ and its boundary $\Gamma_{main}^{0,micro}$ of Figure 4.5 are as follows.

```
paramD{"main"} : node("paramReg", [
  axisList([axis([1,0,0]),axis([0,1,0])]),
  sizeList([eps,eps,0]),
  periodicity(""),
  interfaceList(")])
paramB{"main"} : node("paramReg", [
  axisList([axis([1,0,0]),axis([0,1,0])]),
  sizeList([eps,eps,0]),"", "")
paramB{"int"} : node("paramReg", [
```

```

axisList([axis([1,0,0]),axis([0,1,0])]),
sizeList([eps,eps,0]),"", "")
paramB{"ext"} : node("paramReg",[
axisList([axis([1,0,0]),axis([0,1,0])]),
sizeList([eps,eps,0]),"", "")

```

The function corresponding to the node `paramReg` in the concrete syntax is defined as follows.

```

paramReg00(axisList0, sizeList0, periodicityDirection0, periodicityScale0,
interfaceList0) : node("paramReg",[axisList(axisList0), sizeList(sizeList0),
periodicity([
periodicityDirection(periodicityDirection0),
periodicityScale(periodicityScale0)]),
interfaceList(interfaceList0)])

```

The function corresponding to `interface` is the following.

```

interface00(nameBList,nameD): node("interface",[boundaryNameList(nameBList),
neighborDomainName(nameD)])

```

The definitions of `paramD00` and `paramB00` are subcases of this of `paramReg00`. They are introduced because, as seen in the above examples, the parametrization of boundaries requires less parameters than this of domains.

```

paramD00(axisList0, sizeList0, periodicityDirection0,
periodicityScale0, interfaceList0) : node("paramReg",[
axisList(axisList0), sizeList(sizeList0), periodicity([
periodicityDirection(periodicityDirection0),
periodicityScale(periodicityScale0)]),
interfaceList(interfaceList0)])
paramB00(axisList0, sizeList0) : node("paramReg",
[ axisList(axisList0), sizeList(sizeList0), "", "")]

```

4.2.4/ REGION

The region node

```

node("region",[ident,paramReg,subRegionList,boundary])

```

represents a region, that can be a domain or a boundary, with the following parameters, see also Figure 4.4.

1. `ident` a node for the region identifier.
2. `paramReg` a node for the region parameters.
3. `subRegionList(subRegionList0)` a node for the list `subRegionList0 = [region, ...]` of subregions. Only the main domain (i.e. the full domain) is allowed to have subdomains.
4. `boundary(region0)` a node that stores a boundary of a region where `region0` is the node for the boundary.

The region node is used to represent both domains and boundaries. The following two examples illustrate this.

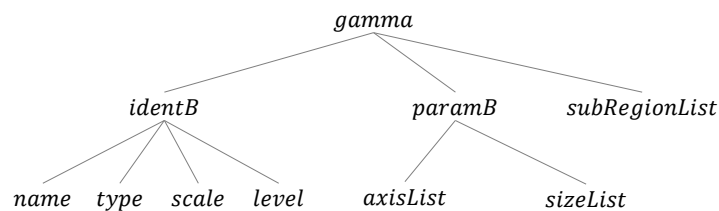


Figure 4.3: Tree representation of a region node for a boundary with name `gamma`.

Example: The following code defines the main boundary $\Gamma_{main}^{0,micro} = \Gamma_{ext}^{0,micro} \cup \Gamma_{int}^{0,micro}$ of $\Omega_{main}^{0,micro}$ of Figure 4.5. The identifiers `identB{[...]}` and `paramB{[...]}` are defined in Sections 4.2.2 and 4.2.3.

```

boundary{["ext"]} : node("region",[
  identB{["ext"]},
  paramB{["ext"]},
  subRegionList([], "", "")])
boundary{["int"]} : node("region",[
  identB{["int"]},
  paramB{["int"]},
  subRegionList([], "", "")])
boundary{["main"]} : node("region",[
  identB{["main"]},
  paramB{["main"]},
  subRegionList([boundary{["int"]}, boundary{["ext"]}]), "", "")])

```

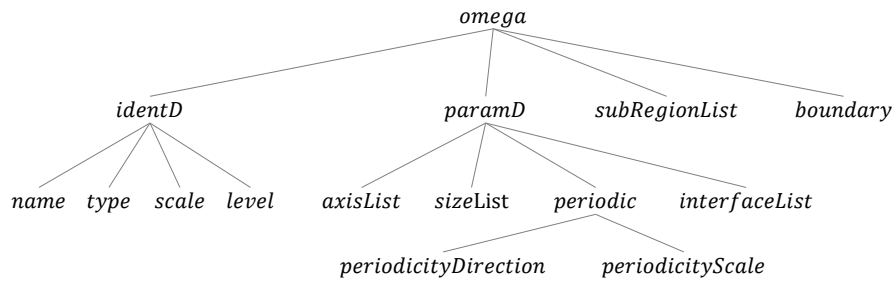


Figure 4.4: Tree representation of a region node for a domain with name ω .

Example The declaration of the physical domain $\Omega_{main}^{0,micro}$ of Figure 4.4 which boundary $\Gamma_{main}^{0,micro}$ has two sub-boundaries $\Gamma_{int}^{0,micro}$ and $\Gamma_{ext}^{0,micro}$ already defined above is as follows.

```

domain{"main"} : node("region",[
  identD{"main"},
  paramReg{"main"},
  subRegionList([],),
  boundary{"main"}])
  
```

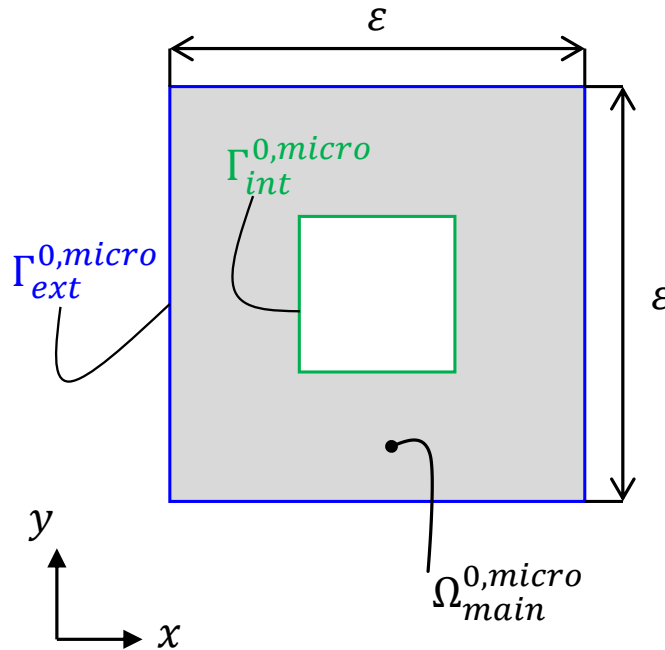


Figure 4.5: A physical domain whose boundary has two parts, one internal and one external.

Most of the time the function `region00` is not used in the concrete syntax because the distinction between domains and boundaries is generally more relevant since their num-

ber of variables differs. The functions `domain00` and `boundary00` are therefore defined in the following.

```

domain00(nameD,axisList0,sizeList0, periodicityDirection0, periodicityScale0,
  interfaceList0, subDomainList0, boundary0) : node("region",[
  ident00(nameD,"omega",scale0,level0),
  index1([]),
  paramD00(axisList0,sizeList0,
    periodicity(periodicityDirection0(periodicityDirection0)),
    periodicityScale(periodicityScale0)),
  interfaceList0),
  subRegionList(subDomainList0)
  boundary(boundary0)])
boundary00(nameB,axisList0,sizeList0,subBoundaryList0) : node("boundary",[
  ident00(nameB,"gamma",scale0,level0),
  index1([]),
  paramB00(axisList0,sizeList0),
  subRegionList(subBoundaryList0)])

```

We observe that the values of `scale0` and `level0` are not in the parameters of `domain00` and `boundary00`. They are assumed to be declared and used as environment variables in a way not detailed here.

The parameter `interfaceList0` is built thanks to the following node `interface00`.

```

interface00(nameBList0,nameD) : node("interface",[
  boundaryNameList(nameBList0),
  neighborDomainName(nameD)])

```

Example: Let us consider the physical domain in Figure 4.6 which includes two subdomains $\Omega_{int}^{0,micro}$ and $\Omega_{ext}^{0,micro}$ with boundaries $\Gamma_{int}^{0,micro}$ and $\Gamma_{ext}^{0,micro}$. The main boundary $\Gamma_{ext}^{0,micro}$ is split into $\Gamma_{ext,lat}^{0,micro}$ the external boundary and $\Gamma_{int}^{0,micro}$ the interface with $\Omega_{int}^{0,micro}$. The following code defines first the geometry parameters, then the boundaries and finally the domain.

```

scale0 : 0
level0 : "micro"
axisList0 : [axis([1,0,0]),axis([0,1,0])]
sizeList0 : [eps,eps,0]

```

The definitions of the two boundaries $\Gamma_{int}^{0,micro}$, $\Gamma_{ext,lat}^{0,micro}$, of the boundary $\Gamma_{ext}^{0,micro}$ of $\Omega_{ext}^{0,micro}$ and of the boundary $\Gamma_{main}^{0,micro}$ of $\Omega_{main}^{0,micro} = \Omega_{int}^{0,micro} \cup \Omega_{ext}^{0,micro}$ the main domain are the following.

```

boundary{"int"} : boundary00("int",axisList0,sizeList0,[])
boundary{"ext,lat"} : boundary00("ext,lat",axisList0,sizeList0,[])
boundary{"ext"} : boundary00("ext",axisList0,sizeList0,
    [boundary{"int"},boundary{"ext,lat"}])
boundary{"main"} : boundary00("main",axisList0,sizeList0,
    boundary{"ext,lat"})

```

Then, the three domains $\Omega_{int}^{0,micro}$, $\Omega_{ext}^{0,micro}$ and $\Omega_{main}^{0,micro}$ are constructed as follows.

```

domain{"int"} : domain00("int",axisList0,sizeList0, ["P","T"],
    [eps,eps], [interface00("int","ext")], [], boundary{"int"})
domain{"ext"} : domain00("ext",axisList0,sizeList0, ["P","T"],
    [eps,eps], [interface00("int","int")], [], boundary{"ext"})
domain{"main"} : domain00("main",axisList0,sizeList0, [],
    [], [], [], boundary{"main"})

```

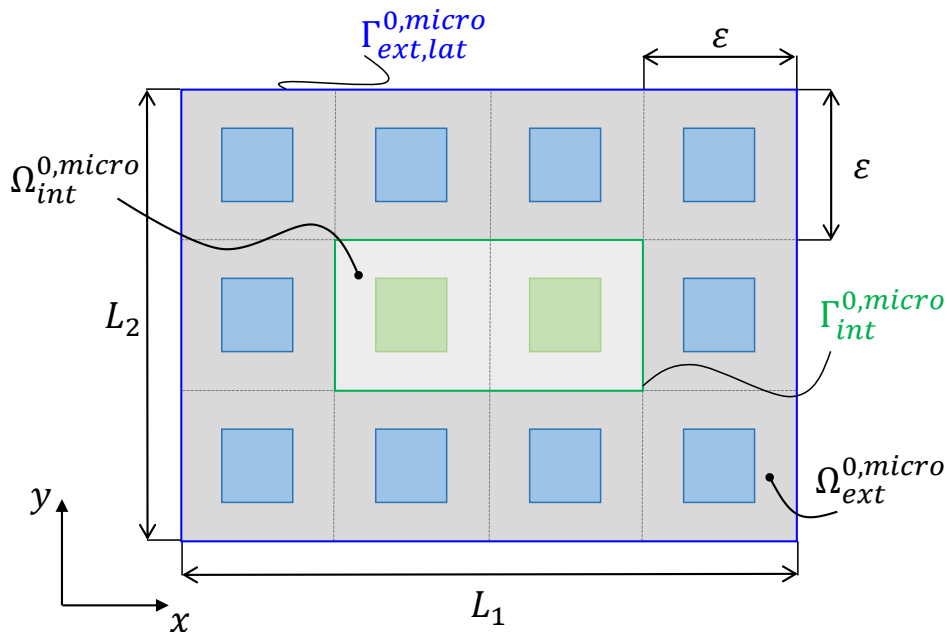


Figure 4.6: A physical domain with two subdomains. The periodicity is in the x_1 -direction. In the x_2 -direction, the domain is assumed to be thin.

4.2.5/ VARIABLE

The variable node

```
node("variable", [name, index, region])
```

represents a mathematical variable defined over a region.

1. name a node that represents the name of the variable. Common names are "x" for domains and "xg" for boundaries.
2. index a node for the index of the variable.
3. region is a node of a region that holds either for a boundary node or a domain node. It is noteworthy that a variable is defined on a single region and not on the Cartesian product of several regions.

Example: Let $x = (x_1, x_2)$ be a variable defined in $\Omega_{main}^{0,micro} \subset \mathbb{R}^2$ and xg be a variable defined in $\Gamma_{ext}^{0,micro}$, they are declared as follows.

```
x{"main"} : node("variable", [name("x"), i, domain{"main"}])
xg{"ext"} : node("variable", [name("xg"), i, boundary{"ext"}])
```

where i is the index defined in Section 4.2.1.

Variables are defined separately on boundaries and domains. To reduce the complexity of manipulations, it is assumed that the domains and boundaries are already defined and stored in the association lists `domain{[nameD]}` and `boundary{[nameB]}`. The index of the variable is defined using the function `index1` introduced in Section 4.2.1.

The function associated to the variable node in the concrete syntax is then defined as follows.

```
variable00{"domain"}(nameV, ind, nameD) : node("variable",
  [name(nameV), index1(ind), domain{[nameD]}])
variable00{"boundary"}(nameV, ind, nameB) : node("variable",
  [name(nameV), index1(ind), boundary{[nameB]}])
```

Example: Let us define a variable in the domain $\Omega_{int}^{0,micro}$ represented as `domain{"int"}` and using the index i .

```
x{"int"} : variable00{"domain"}("x", i, "int")
```

4.2.6/ FUNCTION

The function node

```
node("function", [ident, name, indexList, varList, type])
```

represents mathematical functions.

1. `ident` a node of the identifier of subScale where the function is defined.

2. name a node that represents the name of the function.
3. `indexList(indexList0)` a node for a list `indexList0 = [index,...]` of index nodes of the function.
4. `varList(varList0)` a node for a list `varList0 = [variable,...]` of variable nodes of the function.
5. `type(type0)` a node for the type `type0` of the function that can take one of the three following values.
 1. "Given" a source or a coefficient.
 2. "Unknown" for unknowns of a PDE.
 3. "Test" for test functions in PDE under weak (or variational) form.

Example: A function f of x with index i is defined as

```
f(i,x) : node("function",[ident(""), name("f"), indexList(i),
                        varList(x), type("Given")])
```

Example The unit outward normal vector n to the boundary of a domain is not directly defined on the boundary. We consider one of its extensions to the domain,

```
n(x) : function(name("n"),indexList(i),varList(x),type("Given"))
```

for a variable node x defined over the domain and i an index whose range is related to the domain. Its use as is then through the trace operator `opTrace`, see Section 4.2.7.

The function of the concrete syntax corresponding to the node `function` is defined as follows.

```
function00(nameF,indList,varList0,type0) : node("function",
        name(nameF),indexList(map(index1,indList)),variableList(varList0),type(type0))
```

where `indList = [ind1,...]` is a list of parameters of the function `index1` and `map(index1,indList)` returns the list `[index1(ind1),...]`.

Example: Let us construct a vector valued function f of a variable x and indexed by an index i .

```
f(i,x) : function00("f",i,x,"Given")
```

4.2.7/ OPERATOR

The node operator

```
node("operator", [name, indexList, mathExprList, inputVarList,
outputVarList, paramOper])
```

represents a mathematical operator.

1. name a node that represents the name of the operator.
2. indexList(indexList0) a node for a list indexList0 = [index,...] of index nodes of the operator.
3. mathExprList(mathExprList0) a node for a list mathExprList0 = [mathExpr,...] of mathematical expression nodes mathExpr(mathExpr0) where mathExpr0 is a mathematical expression.
4. inputVarList(inputVarList0) and outputVarList(outputVarList0) nodes for the input variable list inputVarList0 = [variable,...] and the output variable list outputVarList0 = [variable,...] of the operator.
5. paramOper(operParam0) a nodes for a list operParam0 = [expression,...] of parameters that can be any expressions.

Example: The following code is the definition of the trace operator of an expression expr that is assumed to be already defined. Its application could use the variables x and xg defined in Section 4.2.5.

```
opTrace(expr, x, xg) : node("operator", [
  name("opTrace"),
  indexList([]),
  mathExpr(expr),
  inputVarList(x),
  outputVarList(xg),
  paramOper([])])
```

The function of the concrete syntax corresponding to the node operator is defined as follows.

```
operator00(nameOp, indList, mathExprList0, inputVarList0,
  outputVarList0, operParam0) : node("operator", [
  name(nameOp), indexList(map(index1, indList)), mathExprList(mathExprList0),
```

```
inputVariableList(inputVarList0), outputVariableList(outputVarList0),
parameterList(paramList0)])
```

We define a mapping $\text{mapping}\{rName1, rName2\} : x(rName2)("") \rightarrow x(rName1)("")$ of a change of variable as an operator. The two variables are defined on two domains named $rName1$ and $rName2$. Precisely, defining

```
x(rName)(i) = variable("x", i, domain1(rName))
```

the mapping operates on any expression $expr$ and is defined as

```
mapping{rName1, rName2}(expr)(i) :
operator("mapping", [i], s[expr], xg(rName2)(""), xg(rName1)(""), [])
```

with the strategy s defined as

```
s:goto([boundary1(rName2)], [], rule(x_, boundary1(rName1)))
```

and where $s[expr]$ stands for the application of the strategy s to the expression $expr$.

4.2.8/ EQUALITY

The equality node

```
node("equality", [name, leftMathExpr, rightMathExpr])
```

is to express to a mathematics equality, for instance $f = g$ or $f = 5$ where f, g are functions and 5 a constant.

1. name a node that represents the name of the equality.
2. $\text{leftMathExpr}(\text{leftMathExpr0})$ and $\text{rightMathExpr}(\text{rightMathExpr0})$ the nodes for leftMathExpr0 and rightMathExpr0 the mathematical expressions of the left-hand-side and the right-hand-side of the equality.

Example: Let us consider the equality $\Delta u = \partial_{x_1 x_1}^2 u + \partial_{x_2 x_2}^2 u = 0$ named as "laplace", where u is a scalar function depending on the variable $x = (x_1, x_2)$ defined on \mathbb{R}^2 . Now, the lapLace equality can be described by the following code.

Expression

```
laplaceOp :  $\sum_i \partial(u(x([\ ]))/\partial x(i))/\partial x(i)$ 
```



```
equality{"lapLace"} : node("equality",["lapLace",
  leftMathExpr(laplaceOp),rightMathExpr(0)])
```

The function of the concrete syntax corresponding to the node equality is defined as follows.

```
equality00(nameE,leftMathExpr0,rightMathExpr0) : node("equality",[
  name(nameE),leftMathExpr(leftMathExpr0),rightMathExpr(rightMathExpr0)])
```

Example: The above laplace equality and the equalities of Section 4.2.10 homDirichlet and nonHomDirichlet are redefined as follows.

```
equality{"lapLace"} : equality00("lapLace",laplaceOp,0)
equality{"homDirichlet"} : equality00("homDirichlet",opTrace(u(x),x,xg),0)
equality{"nonHomDirichlet"} : equality00("nonHomDirichlet",
  opTrace(u(x),x,xg),opTrace(f(x),x,xg))
```

4.2.9/ EQUATION

The node equation

```
node("equation",[name,equality,varList])
```

describes a mathematical equation consisting of an equality and a list of variables on which it is defined.

1. name a node that represents the name of the equation. For PDE it may be usual equation names as lapLace, poisson, wave etc.
2. equality a node that represents the equality of the equation.
3. varList(varList0) a node for a list varList0 = [variable,...] of variable nodes of the equation. We point out that the indication of a specific domain for the equation is due to variable data structure containing the entire structure of the domain where the variable is located.

Example: Let's consider the Laplace equation defined in the domain $\Omega_{main}^{0,micro}$ where the equality of the Laplace equation is assumed to be already defined as well as the variable `x{"main"}`, see Section 4.2.5.

```
equation{["lapLace"]} : node("equation",[
  name("lapLace"),
  equality{["lapLace"]},
  varList(x{["main"]})])
```

The function of the concrete syntax corresponding to the node `equation` is defined as follows.

```
equation00(nameEq,equality0,varList0) : node("equation",[
  name(nameEq),equalityList(equality0),varList(varList0)])
```

Example: The laplace equation in $\Omega_{main}^{0,micro}$ can be reconstructed as follows.

```
equation{["laplace"]} : equation00("laplace",
  equality{["laplace"]},x{["main"]})
```

4.2.10/ BC AND IC

The `bc` node

```
node("bc",[name,equality,bVarList,dVarList])
```

is to describe boundary conditions for PDEs.

1. `name` a node that represents the name of the boundary condition. Common names are `dirichlet`, `neumann`, `mixed`, etc.
2. `equality` a node that represents the equality of the boundary condition.
3. `bVarList(bVarList0)` and `dVarList(dVarList0)` nodes for a list `bVarList0 = [variable,...]` and a list `dVarList0 = [variable,...]` of variable nodes for the boundary where the boundary condition holds and for the related domains. The reference to domains is necessary for interface conditions where the same boundary occurs in two domains.

Example: The following code defines a homogeneous and a non-homogeneous Dirichlet boundary conditions $tr(u)(xg) = 0$ and $tr(u)(xg) = f(xg)$ at all xg , provided that the functions from `variable` to function $x \rightarrow u(x)$ and $xg \rightarrow f(xg)$ have already been defined as well as x and xg two variable nodes, xg being a variable defined on a part of the boundary of the domain of x .

```

bc{"homDirichlet"} : node("bc",[
  name("homDirichlet"),
  node("equality",[
    name("homDirichlet"),
    leftMathExpr(opTrace(u(x),x,xg)),
    rightMathExpr(0)]),
  varList(xg{"ext"})])

bc{"nonHomDirichlet"} : node("bc",[
  name("nonHomDirichlet"),
  node("equality",[
    name("nonHomDirichlet"),
    leftMathExpr(opTrace(u(x),x,xg)),
    rightMathExpr(opTrace(f(x),x,xg))]),
  varList(xg{"int"})])

```

The ic node

```
node("ic",[name,equality,bVarList,dVarList])
```

is to describe initial conditions for PDEs and ODEs. Its definition is similar to this of bc.

The function of the concrete syntax corresponding to the nodes bc and ic are defined as follows.

```

bc00(nameBc,equality0,bVarList0,dVarList0) : node("bc",[
  name(nameBc), equality(equality0),
  bVarList(bVarList0), dVarList(dVarList0)])
ic00(nameIc,equality0,bVarList0,dVarList0) : node("ic",[
  name(nameIc), equality(equality0),
  bVarList(bVarList0), dVarList(dVarList0)])

```

Example: The above homogeneous and non homogeneous Dirichlet condition can be rewritten as follows.

```

bc{"homDirichlet"} : bc00("dirichlet",equality{"homDirichlet"},
  xg{"ext"}, x{"main"})
bc{"nonHomDirichlet"} : bc00("nonHomDirichlet",equality{"nonHomDirichlet"},
  xg{"int"}, x{"main"})

```

4.2.11/ PDE

The `pde` node

```
node("pde",[name,equationList,bcList,icList])
```

is to represent partial differential equations.

1. `name` a node that represents the name of the PDE.
2. `equationList(equationList0)` a node for a list `equationList0 = [equation,...]` of equation nodes of the PDE.
3. `bcList(bcList0)` and `icList(icList0)` nodes for a list `bcList0 = [bc,...]` and a list `icList0 = [ic,...]` of bc and ic nodes.

The function of the concrete syntax corresponding to the node `pde` is defined as follows.

```
pde00(pdeName,equationList0,bcList0,icList0) : node("pde",[
  name(pdeName),equationList(equationList0),bcList(bcList0),
  icList(icList0)])
```

Example: The following code defines a PDE made with the equation `equation{["laplace"]}` defined in Section 4.2.9 and the boundary conditions `bc{["homDirichlet"]}` and `bc{["nonHomDirichlet"]}` defined in Section 4.2.10.

```
pde{["laplace"]}: pde00("laplace",
  equation{["laplace"]},[bc{["homDirichlet"]},bc{["nonHomDirichlet"]}],[""])
```

4.2.12/ MODEL

The `model` node

```
node("model",[name,pdeList])
```

represents a mathematical model with possibly several PDEs.

1. `name` a node that represents the name of the model.
2. `pdeList(pdeList0)` a node for a list `pdeList0 = [pde,...]` of pde nodes of the model.

The function of the concrete syntax corresponding to the node `model` is defined as follows.

```
model00(nameModel,pdeList0) : node("model",[
name(nameModel), pdeList(pdeList0)])
```

4.2.13/ SUBMODEL

The `subModel` node

```
node("subModel", [ident,regionName,pdeName,macroRegion,macroModel,dilation,
subScale])
```

represents a submodel used in a `subScale` node. A `subModel` node contains on the one hand the assumptions and parameters allowing the construction of its macroscale and microscale models by inference and in the other hand the inferred models. The first one is located in the `macroModel` node while the second one for its macro part and in the `subScale` node for its micro part.

Together, the `subModel` and `subScale` nodes allow to generate recursive structures.

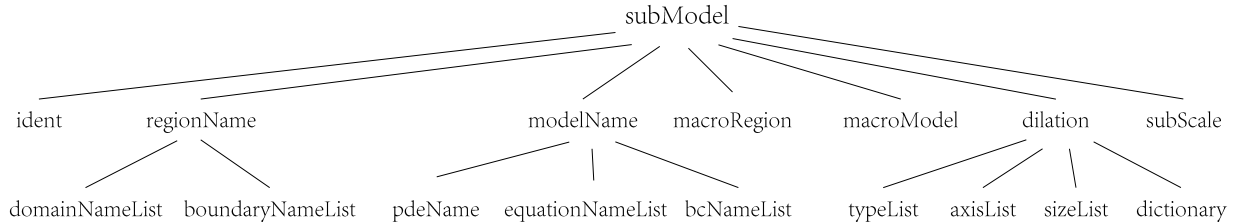


Figure 4.7: Tree representation of the `subModel` node.

1. `ident` a node for the identification of the `subModel`, see Section 4.2.2.
2. `regionName` a node represents the geometrical inclusion relationship between domains and boundaries used in a `subModel` with domains and boundaries used in a `subScale` at the upper scale.
 1. `domainNameList(domainNameList0)` a node for a list `domainNameList0 = [(name1,name0), ...]` whose elements are pairs of the domain names `name1` and `name0` and defines an inclusion relation. The name `name0` is that of a domain Ω_{name0} used in the PDE of the parent `subScale` node of the `subModel`. The name `name1` is that of the node Ω_{name1} of the region or one of its sub-regions of the `subScale` node descending from the `subModel`. The inclusion relation is between Ω_{name1} and the transformation (unfolding) of Ω_{name0} . It is

used to determine the equations used in Ω_{name0} that need to be imported into Ω_{name1} for processing.

2. `boundaryNameList(boundaryNameList0)` a node for a list `boundaryNameList0 = [(name0.1,name0),name0,...]` whose elements can be (i) pairs of a boundary name `name0` and its subboundary name `name0.1` where the boundary `name0` and `name0.1` are respectively the subboundary of the domain used in the current `subScale` and the `subModel` at upper scale; (ii) only boundary names `name0` if `name0` and `name0.1` are identical, in the sense that the boundaries used in the `subScale` are fully employed in the `subModel`.
3. `modelName` a node represents the selection of equations and boundary conditions of pdes used in a `subModel` from the pde of `subScale` at the upper scale. The names refer to pde, equation, boundary condition names in the previous scale and they must be listed in the corresponding index with the domain names and the boundary names that are given in the `domainNameList` node and `boundaryNameList` node of the current `subModel`.
 1. `pdeNameList(pdeNameList0)` a node for a list `pdeNameList0 = [name0,...]` of pde names `name0` which are strings.
 2. `equationNameList(equationNameList0)` a node for a list `equationNameList0 = [name0,...]` of equation names `name0` which are strings.
 3. `bcNameList(bcNameList0)` a node for a list `bcNameList0 = [name0,...]` whose elements can be (i) `name0` boundary condition names applied on the corresponding boundaries whose names are given in the `boundaryNameList`; (ii) be empty " " in the case of no boundary conditions applied on these boundaries.
4. `macroRegion` a region node for the submodel macroscopic region given as data for an algorithm.
5. `macroModel` a model node for the submodel macroscopic model built by an algorithm.
6. `dilation` a node defined below for the data useful to algorithms that transform the model at the previous scale into the microscopic model of the `subScale` node.
7. `subScale` a node defined in Section 4.2.14 for the microscopic model of the subnode.

The dilation node

```
node("dilation",[axisList,typeList, sizeList,dictionary))
```

Type	Kind of dilation
P	Periodicity
BL	Boundary layer at a boundary
BLI	Boundary layer at an interface
BL_Edge	Boundary layer at an edge
BLI_Edge	Boundary layer at an interfacial edge
T	Thin structure
N	No dilation

Table 4.4: Names of the types of dilation.

used in a `subModel` node for data required by algorithms to transform the model at the previous scale into the microscopic model of the subscale.

1. `typeList(typeList0)` a node for a list of type nodes, `typeList0 = [type(type0),...]` where each type node characterizes the kind of dilation in a direction of the `axisList` node. The parameter `type0` takes one of the values of Table 4.4. Note that the type node is also used in Section 4.2.3 for regions.
2. `axisList` a node for scaling directions, see its syntax in Section 4.2.3.
3. `sizeList` a node for the dilation scalings, see its syntax in Section 4.2.3.
4. `dictionary(dictionary0)` a node for data specific to each algorithm. Each entry of the list `dictionary0=[entry,...]` is a node `node(key,[description,listNode])` with `key` a string, `description` a name node defined as `node("description",[string0])` that describes the entry by strings `string0`, and `listNode=node("list",list)` where `list` is a list of any expressions that can also be other entries. The subnode `description` for data to be interpreted by an external agent (user or program) provides the expected value for algorithms. In this sense, it plays the role of an interface between an external agent and algorithms. The function to construct the description node is defined as follows.

```
descript00(string0) : node("description",[string0])
```

The functions of the concrete syntax for `subModel` and `dilation` are defined as follows.

```
subModel00(name0,ident,regionName0,pdeName0,macroRegion0,
macroModel0,typeList0,dirList0,scaleList0,dilation,subScale) : node("subModel",[
name(name0),ident,regionName,pdeName,macroRegion,macroModel,dilation,subScale])
dilation00(typeList0,dirList0,scaleList0,dictionary) : node("dilation",[
type(typeList0),direction(dirList0),scaling(scaleList0),dictionary])
```

4.2.14/ SUBSCALE

The subScale node

```
node("subScale",ident,region,model,subModelList)
```

describes either the physical problem or a microscopic model (i.e. a new subscale) of a subModel node. It is a part of the pair of nodes (subModel,subScale) that allows to produce recursive multiscale models.

1. `ident` a node defined in Section 4.2.2 for the identification of the subscale.
2. `region` a node defined in Section 4.2.4 for either the physical domain or the microscopic domain of a submodel. In the second case, it is used for the construction of the submodel by an algorithm.
3. `model` a node defined in Section 4.2.12 either for the physical model or for the microscopic model of a submodel. In the second case, it stores the result of an algorithm construction.
4. `subModelList(subModelList0)` a node for a list `subModelList0 = [subModel, ...]` of subModel nodes defined in Section 4.2.13.

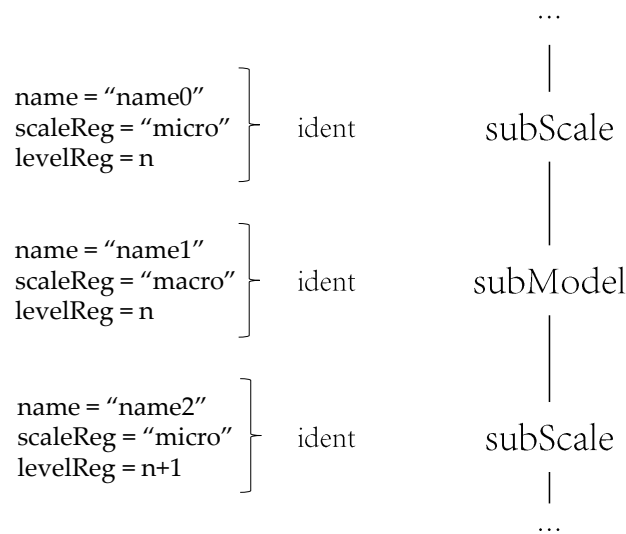


Figure 4.8: Identifiers of the recursive multiscale models built on the pair (subModel, subScale).

The function of the concrete syntax for subScale is defined as follows.

```
subScale00(name0,region,model,modelList0) : node("subScale",[
  name(name0),region,model,subModelList(modelList0)])
```


4.3/ STRATEGIES NODES

As seen in the grammar, the strategies are also structured and organized as abstract terms using the node `nodeS(e, [e, ..., s, ...])`. The nodes with their parameters are listed in Table 4.5 using the notations `e`, `e'` and `s` of the grammar. They are built on the elementary strategies recalled in Section 5.3, but we do not detail their construction.

nodeSs	Subnodes	SubnodeSs
insert	e'	-
remove	-	-
replace	e'	-
rename	e	-
store	e	-
rule	e1, e2', e3	-
goto	path, limit	s
LeftChoice	-	s, ...
compose	-	s, ...

Table 4.5: List of abstract strategies with their names and their parameters `e` or `e'` of type expression and `s` of type strategy defined in the grammar. The parameters `path` and `limit` are lists of nodes `[e, ...]`.

The following is devoted to the definitions of these strategies.

4.3.1/ GOTO

`nodeS("goto", [path, limit, s])` or `goto(path, limit, s)` is a strategy node that navigates in a tree of nodes starting from the root, searching successively in all subtrees the patterns of the list `path` as long as one of the pattern of the list `limit` is not met. In each subtree, if one of the patterns of `limit` is found the strategy returns without doing any operation. If the last pattern of `path` is met then `s` is applied. The overall strategy succeeds if at least one of the application of `s` succeeds.

Example: Let us consider the tree in Figure 4.9 whose nodes are uppercase letters. The application of the strategy `goto([B, H], C, s)`, leads to the node B then to the node H which belongs to the root subtree B. During the navigation, if C is encountered, the navigation does not continue in the subtree. Once the node H is found, the strategy `s` is applied. See the detail of the navigation path in Figure 4.10.

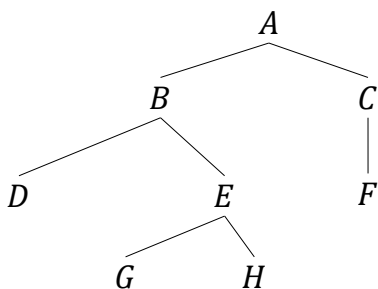


Figure 4.9: A simple tree.

One of the things to be noticed is that `goto([B,H],C,s)` and `goto([B,E,H],C,s)` in this case are equivalent, but in some other cases may produce different results. It noteworthy to mention that the more precise the address in the path and the more severe the limit in the limit, the faster the search will be.

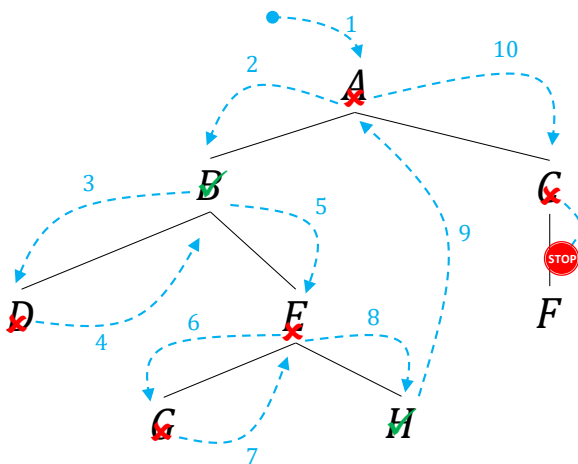


Figure 4.10: The processing of the application `goto([B,H],C,s)` to the tree of 4.9 to find successively B and H with a limit C.

4.3.2/ INSERT

`nodeS("insert",[e'])` or `insert(e')` is a strategy node that inserts the term `e'` to the right of the node list of the current node, i.e. it transforms a node list `node(name,list)` into `node(name,[list,e'])`. The list `[list,e']` is then transformed as a single list. The strategy always succeeds except if the current term is not a node (it may be a leaf).

This strategy is typically used to insert another domain in a list of domains, or another index in a list of indices etc.

Example: The operations made by the applications of the strategy `goto([B,E,H],C,insert(K))` on the tree of Figure 4.9 are illustrated on Figure

4.11.

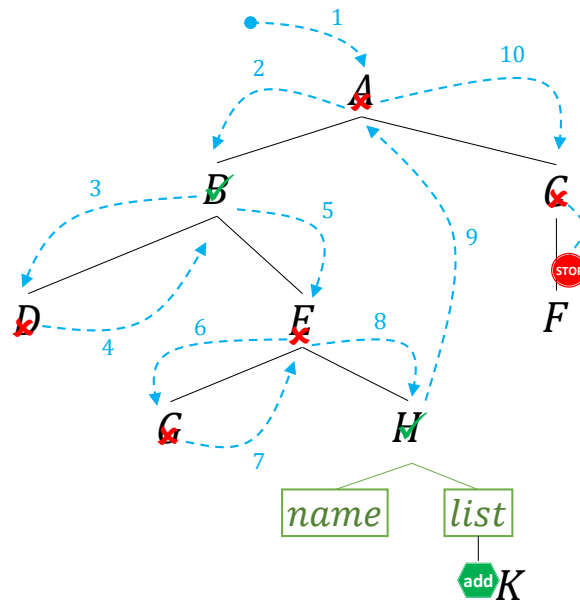


Figure 4.11: Illustration of the operations made when applying the strategy `goto([B,E,H],C,insert(K))` to the tree of Figure 4.9.

4.3.3/ REMOVE

`nodeS("remove",[])` or `remove` is a strategy node that applies to a node of a node list and removes it from the list. This strategy fails if and only if the current term is not a node or if the list of the node is empty. Otherwise it succeeds.

Example: After application of `goto([B,E,H],C,insert(K))` to the tree of Figure 4.9 as described in Figure 4.11. Now we consider the strategy `goto([B,E,H,K],C,remove)` that removes the node `K` so that the tree will be turn to its original state.

4.3.4/ REPLACE

`nodeS("replace",[e'])` or `replace(e')` is a strategy node that replaces the current node by the node `e'`. This strategy always succeeds.

Example: The strategy `textttgoto([B,E],C,replace(F))` applied to the tree of Figure 4.9 replaces the node `E` by the node `K`.

4.3.5/ RULE

`nodeS("rule",[left,right,condition])` or `rule(left,right,condition)` is a strategy node that is a usual rewriting rule that checks if the condition `condition` is satisfied.

If its evaluation returns `True` then the matching between `left` and the current term is checked. If it is a success then the strategy replaces the current term by `right` where its rewriting variables have been substituted by the solutions of the substitution computed during the matching phase.

Example: The strategy

```
goto(index(x_), [], rule(index00(x_, y_, z_), rule(index00([x_, [y_, 3], z_-
])), True)
```

search all nodes `index` and adds 3 to their range. The strategy `rule` can be replaced by the strategy `insert`.

4.3.6/ RENAME

`nodeS("rename", [e])` or `rename(e)` is a strategy node that renames the current node by the a new name `e`. This strategy fails only if the current term is not a node (it may be a leaf).

Example: The strategy `goto(index00("i", x_, y _), [], rename("j"))` applied to the index `i` defined as

Expression

```
i : index00("i", [1, 2], "Given")
```

returns `index00("j", [1, 2], "Given")`. The same result can be achieved by

```
goto([index(x_), name("i")], [], replace(name("i"), name("j")))
```

or

```
goto([index(x_), name("i")], [], rule(name("i"), name("j"), True)).
```

4.3.7/ STORE

`nodeS("store", [e])` or `store(e)` is a strategy node that stores the current node in a memory with name `e`. This strategy always succeeds.

Example: Let the index `i` defined in Section 4.2.1, it can be searched and stored by the strategy `goto([index(x _), name("i")], [], store())`.

4.3.8/ LEFTCHOICE

`nodeS("leftChoice", [s1, ...])` or `leftChoice([s1, ...])` is a strategy node that applies to a list of strategies. It applies the first one of the list, if it fails then it applies the

second one. This is repeated until the last strategy of the list. The `leftChoice` strategy fails if all strategies of the list fail otherwise it succeeds.

Example: The strategy

```
leftChoice([goto([B,E,H],[],remove),goto([B,E],[],remove)])
```

applied to the tree of Figure 4.9 removes H. Actually, its first strategy `goto([B,E,H],[],remove)` succeeds thus the second one is not applied.

4.3.9/ COMPOSE

`nodeS("compose",[s1,...])` or `compose([s1,...])` is a strategy node that applies to a list of strategies. It applies the first one of the list, then the second one and so on until the last strategy of the list. The `compose` strategy fails if at least one of the strategies of the list fails otherwise it succeeds.

Example: The strategy

```
compose([goto([C,F],[],remove),goto([B,E,G],[],remove)])
```

applied to the tree of Figure 4.9 removes F and G.

4.4/ CONSTRUCTION OF A FAMILY OF MULTISCALE ELECTROSTATIC MODELS OF MIRA

Some of the asymptotic models developed in the first part of this thesis are used to illustrate the data structures of Section 4.2 and the construction of algorithms based on the strategies of Section 4.3. The algorithms are given in the form of metaprograms, which we do not consider useful to specify the details in this document. Five classes of models are considered: periodic models, boundary layer models on boundaries, boundary layer models at interfaces between two sub-domains, boundary layer models at outer edges, and boundary layer models at interface edges. Only the equation of electrostatics is considered here because, our main objective being the illustration of the data structure and the related algorithms, taking into account more complex equations would complicate the presentation without highlighting many more concepts.

The data structure includes at its root the `subScale` node of the physical problem and all asymptotic models are placed in its list of `subModel` nodes. Here we assume that the `model` node of the physical system is already built on the principles of the examples described in Section 4.2. It is also assumed that the specific data necessary for the application of the algorithms are already present in the `subModel` nodes of each model. These data will be precisely described. All these available data should eventually be

constructed by automatic transfer from a PDE simulation software to the data tree.

These algorithms are presented here not as complete representations of theorems, but as validations of the data structures introduced above. The aim is to ensure that the latter are able to represent both a physical model and a relatively complex family of asymptotic models. The effort is made to specify the content of the data structures and the transformation operations to be performed. On the other hand, the mechanism for inferring conclusions from hypotheses is not discussed here.

In Section 4.4.1, we will detail the content of the `subScale` node at the root that contains the physical problem. Then for each model, a section will be dedicated to the description of the required data and another one to the model construction algorithm. The first section of the description of the required data will be detailed while for the other ones only the data will be provided without much comment.

4.4.1/ DOMAIN DEFINITIONS AND PDE SETTINGS FOR THE PHYSICAL PROBLEM

As mentioned above, the root of the data structure is the `subModel` node that describes the physical problem. In this section, we describe the datastructure content without precise details on the node structure since this was done in Section 4.2.

Figure 4.12 represents the `subScale` node of the physical problem. Its identifier `idSC.Mira` is defined by `idSC("mira", 0, "micro")`. The `region` node consists of $\Omega_{\text{air.main}}^{0,\text{micro}}$ whose subregions, boundaries, and subboundaries are represented in Figure 4.13. The PDE of the physical model represented by `pdeMira` with a name "pde.Mira" is not reproduced, it is considered to be on the model of the electrostatic equation of Section 4.2.11.

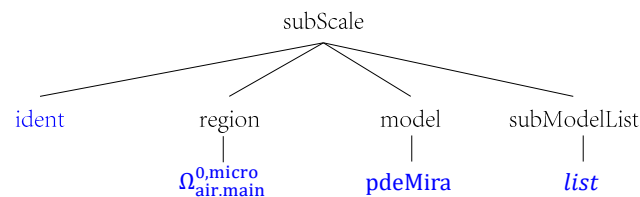


Figure 4.12: The tree representation for the top `subScale` node of the physical model of electrostatic equations posed in the MIRA architecture. The domain $\Omega_{\text{air.main}}^{0,\text{micro}}$ and the electrostatic equations are defined in details in the first part of the thesis. The names of the submodels in the list `subModelList` are "periodic.int", "periodic.ext", "boundaryLayer.i", "boundaryLayer.edge.i", "boundaryLayer.interface.i", and "boundaryLayer.interface.edge.i" for $i = 1, 2, 3, 4$.

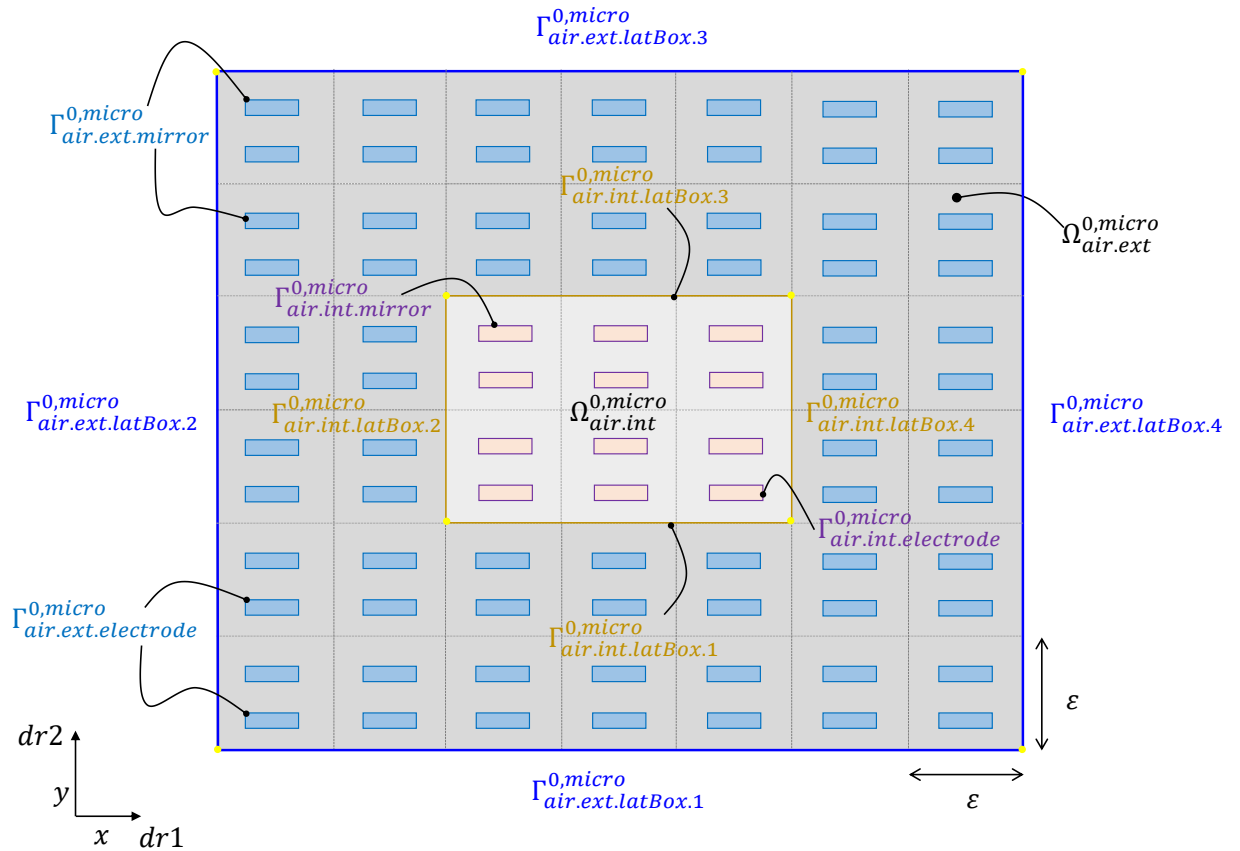


Figure 4.13: The physical air domain of the MIRA model, the subdomains, their boundaries and subboundaries.

The geometry $\Omega_{\text{air.main}}^{0,\text{micro}}$ of the MIRA system has been described in detail in the first part of this thesis. For more clarity, Figure 4.13 gives a two-dimensional representation of the geometry. It distinguishes the two sub-domains of "air.main" associated with zones of different imposed voltages, namely "air.ext" the external air zone and "air.int" the internal air zone. Table 4.6 summarizes the names of Figure 4.13. As mentioned above, we do not detail the PDE that governs the electrostatic phenomenon, but the names of the voltage sources, the electrostatic field in the two regions, the equations and the boundary conditions are listed in Tables 4.7, 4.8 and 4.9.

Domain Name	“air.int”	“air.ext”
Boundary Name	“air.int”	“air.ext”
SubBoundary Name	[“air.int.mirror”, “air.int.electrode”, “air.int.latBox.1”, “air.int.latBox.2”, “air.int.latBox.3”, “air.int.latBox.4”, “air.int.top”]	[“ air.ext.mirror”, “air.ext.electrode”, “air.ext.latBox.1”, “air.ext.latBox.2”, “air.ext.latBox.3”, “air.ext.latBox.4”, “ air.ext.top” “air.int.latBox.1”, “air.int.latBox.2”, “air.int.latBox.3”, “air.int.latBox.4”]

Table 4.6: Summary of the names of the two air domains, of their boundaries and of the subboundaries.

Name	Function
“volMInt”	Imposed voltage on the mirrors of the internal zone
“volEInt”	Imposed voltage on the electrodes of the internal zone
“volMExt”	Imposed voltage on the mirrors of the external zone
“volEExt”	Imposed voltage on the mirrors of the electrode zone
“phi.Int”	Unknown potential in the internal zone
“phi.Ext”	Unknown potential in the external zone

Table 4.7: Names of voltage sources and of the unknown electrical potential in the two regions of the physical PDE.

Bc Name	Equality Name	Unknown	Given	Bd Name List
“dirichlet.int.mirror”	“dirichlet”	“phi.Int”	“volMInt”	“air.int.mirror”
“dirichlet.int.electrode”	“dirichlet”	“phi.Int”	“volEInt”	“air.int.electrode”
“neumann.int.top”	“neumann”	“phi.Int”	“zeroF”	“air.int.top”
“continuity.potential”	“continuity”	“phi.Int”	“phi.Ext”	“air.int.latBox”
“continuity.flux”	“continuityFlux”	“phi.Int”	“phi.Ext”	“air.int.latBox.i”
“dirichlet.ext.mirror”	“dirichlet”	“phi.Ext”	“volMExt”	“air.ext.mirror”
“dirichlet.ext.electrode”	“dirichlet”	“phi.Ext”	“volEExt”	“air.ext.electrode”
“neumann.ext.top”	“neumann”	“phi.Ext”	“zeroF”	“air.ext.top”
“neumann.ext.latBox”	“neumann”	“phi.Ext”	“zeroF”	“air.ext.latBox.i”

Table 4.8: Names of the boundary condition in *the physical PDE pdeMira* where *i* represents “1,2,3,4”.

Then, we introduce equations applying two-zones

Equation Name	Equality Name	Unknown	Given	Region Name
"laplace.int"	"lapLace"	"phi.Int"	"zeroF"	"air.int"
"laplace.ext"	"lapLace"	"phi.Ext"	"zeroF"	"air.ext"

Table 4.9: Names of the equations of the physical PDE.

4.4.2/ PERIODIC MODELS

4.4.2.1/ DATA STRUCTURE FOR A PERIODIC MODEL

The data related to a periodic model posed in a subdomain of the main domain is illustrated through the case of the sub-domain "air.ext". The case of the subdomain "air.int" is similar.

We assume that the microscopic domain $\Omega_{air.ext.per}^{1,micro}$ is already built in the subScale node. We recall that its inclusion relationship with $\Omega_{air.ext}^{0,micro}$ described in Section 4.2.13 is expressed in the node `domainNameList`.

The data is stored in the subModel node shown in Figure 4.14. Its identifier `idSM.Per.Ext` defined as `idSC("periodic.ext",0, "micro")` is illustrated in Figure 4.15. The names of the domain and its subboundaries in the `regionName` node are relative to the "air.ext" part. It is the same for the equations and boundary conditions of the `pdeName` node. For this model there is no macroregion so the `macroRegion` node is empty.

The scale transformation parameters of the dilation node are visible on Figure 4.16. In the list `axisList` the directions are `dr1=axis([1,0,0])`, `dr2=axis([0,1,0])` and `dr3=axis([0,0,1])`. The list `typeList = ["P", "P", "T"]` of transformation types indicates transformations related to periodic structures (periodic unfolding) in the first two directions of `axisList`, while in the third direction it is a thin structure type transformation (simple dilation). The `scaleList` node indicates the three dilations with the parameter ε .

The dictionary node holds names of boundaries stored in the keys `physBdName` and `microBdName`, unknown and given functions retained in the keys `unknownFunction` and `givenFunction`, which are data relating to a two-scale transformation characterized by the dilation node. The entries `physBdName` and `microBdName` are names of boundaries related by the unfolding operation from the physical domain to the microscopic domain.

The `physBdName` key

```
node("physBdName",[descript00("lateral physical boundary"),listNode0])
```

with `listNode0` particular keys that retain all physical boundary names . In the periodic case, there is one key `latBdP` `node("latBdP",[descript00("lateral physical boundary"),listNode0])` with `listNode0` a name node storing all lateral physical boundary names "air.ext.latBox.1", "air.ext.latBox.2", "air.ext.latBox.3", "air.ext.latBox.4".

The `microBdName` key

```
node("microBdName",[descript00("microscopic boundary"),listNode0])
```

with the `listNode0`, particular keys corresponding to keys in the `physBdName` retain all microscopic boundary names. In the periodic model, this `listNode0` holds one key `perBd` with a key "perBd", a string description "periodic boundary", and a `listNode` a name node of the periodic boundary `air.ext.periodic` of the microscopic domain.

Next, we introduce two keys, `unknownFunction` and `givenFunction`, to store respectively the data of the unknowns and given functions of the PDE of the physical model for the external periodic problem.

The key `unknownFunction`

```
node("unknownFunction",[descript00("unknown of pde"),
    [physUnknown, microUnknown]])
```

includes (i) a key `physUnknown` with the name "physUnknown", the description "physical unknown", the list node a node storing an unknown name "phi.Ext" used in a pde of the external physical model and its identifier `idSC.Mira`; (ii) a key `microUnknown` with the name "microUnknown", the description "micro unknown", and the list node a node storing an unknown name "phi.Per" used in a pde of the external microscopic model and its identifier `idSC.Per.Ext`.

The key `givenFunction`

```
node("ginvenFunction",[descript00("given function"),nodeList0])
```

contains all the given source names and their identifiers of the new pde of the microscopic model which are stored as nodes in `nodeList0`. In this periodic case, `nodeList0` is an empty list.

The `subScale` node shown in Figure 4.18 includes the identifier `idSC.Per.Ext` defined by `idSC("periodic.ext", 1, "micro")` which is shown in Figure 4.15 and the microscopic domain $\Omega_{air.ext.per}^{1,micro}$ shown in Figure 4.17. The `model` node is left empty at this step, it will be filled by Algorithm of Section 4.4.2.2.

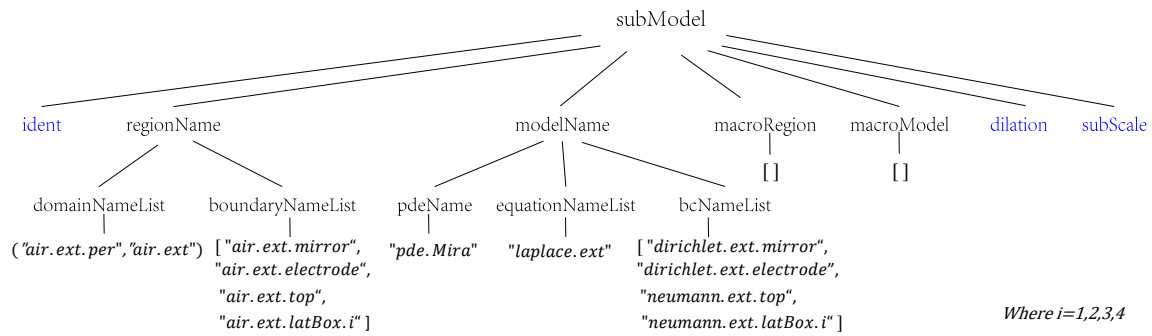


Figure 4.14: The submodel node of the external periodic model.

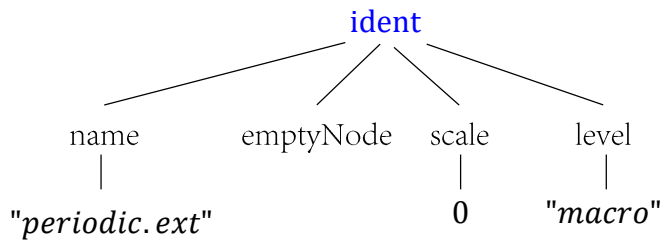


Figure 4.15: The ident node of the external periodic model.

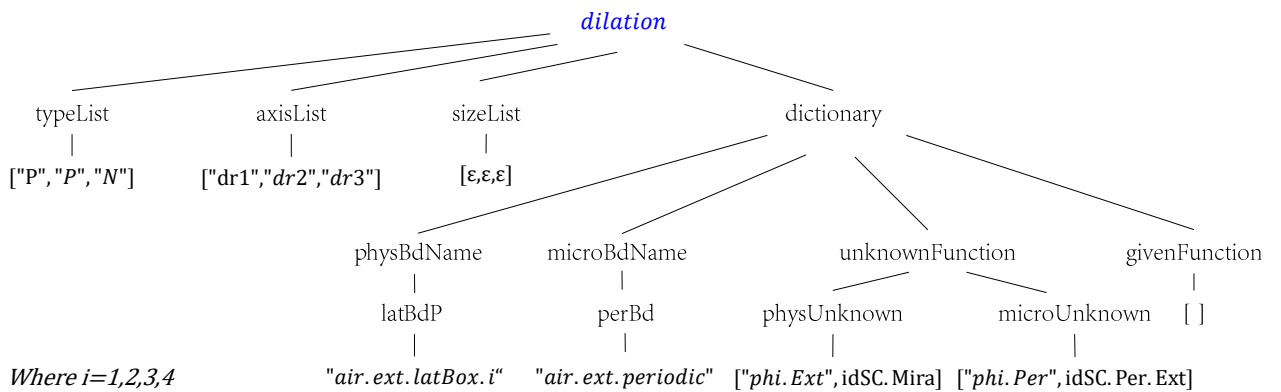


Figure 4.16: The dilation node of the external periodic model.

Domain Name	“air.int.per”	“air.ext.per”
Boundary Name	“air.int.per”	“air.ext.per”
SubBoundary Name	[“air.int.mirror”, “air.int.electrode”, “air.int.top”, “air.int.per”]	[“air.ext.mirror”, “air.ext.electrode”, “air.ext.top”, “air.ext.per”]

Table 4.10: The names of the microscopic domains, boundaries and subboundaries for the models periodic.int and periodic.ext.

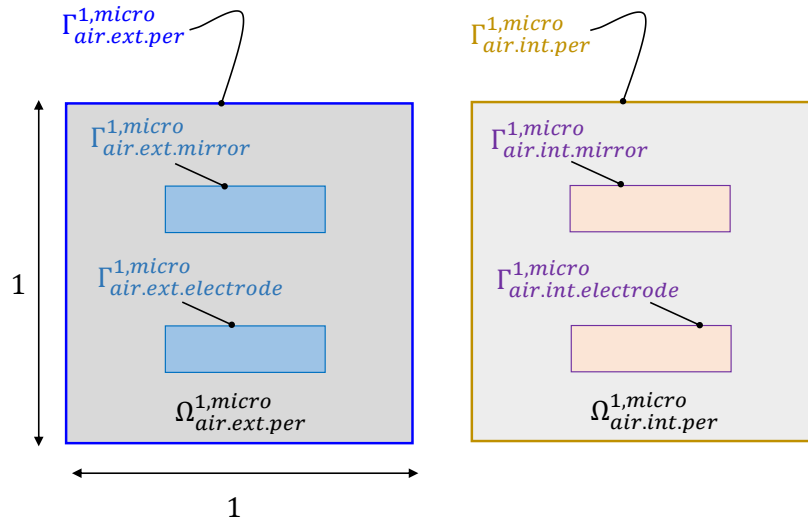


Figure 4.17: The microscopic domains in the subScale node of the periodic.ext and periodic.int models.

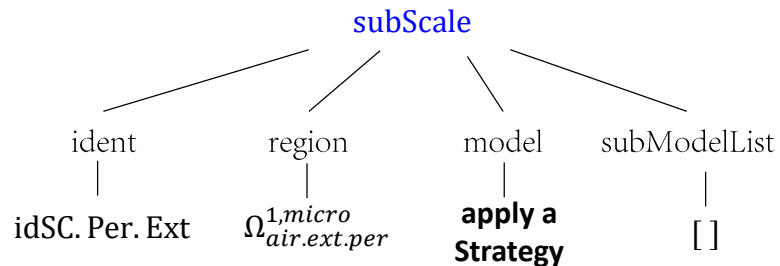


Figure 4.18: The subScale node of the external periodic model.

4.4.2.2/ STRATEGY FOR A PERIODIC MODEL

The following algorithm consists in building the model node of the subScale node of a periodic model. It uses data from the physical model and from the subModel node of the periodic model described in the sections 4.4.1 and 4.4.2.1. It consists in copying the useful part of the physical PDE into the PDE of the microscopic problem. Then it performs changes of names of unknown, deletion of boundary conditions on the outer boundary, replaces everywhere the physical domain by the microscopic domain, sets scale:=1 in the region identifiers, adds periodicity boundary conditions and finally changes the name of the PDE to "pde.micro".

Algorithm 1 Derivation of a periodic model from the physical model.

1: In model of the top subScale node, store all equations in equationList and all boundary conditions in bcList whose names are in pdeName/equationNameList and bcNameList of the model "periodic.X" .

== In the following the operations are in the subModel node with name "periodic.X" ==

2: In the subScale/model node insert the copied data.

3: In subScale/model replace the name given in physUnknown of the unknown function by this in microUnknown. This function is noted ϕ in the following.

4: remove the bc node in subScale/model/pde/bcList operating on the boundary with name in the dilation dictionary at physBdName/latBdP.

5: replace in subScale/model the domain with name in regionName/domainName by the microscopic domain in subScale/region.

6: rewrite the identifier of each boundary in subScale/model to update their scale to 1.

7: in subScale/model/bcList node insert bc nodes for periodic condition on ϕ and anti-periodic condition on $\nabla\phi \cdot n$ on the boundary whose name is in microBdName/perBd.

8: rename the pde name by "pde.micro".

4.4.3/ BOUNDARY LAYER MODELS

4.4.3.1/ DATA STRUCTURE FOR A BOUNDARY LAYER MODEL

The boundary layer model is illustrated with a single case. The selection of its domain in the physical domain and its microscopic domain are in Figure 4.19. Table 4.11 summarizes the region names of the microscopic domain $\Omega_{air.ext.lat.1}^{1,micro}$. The subModel data are in Table 4.12 and Table 4.13 is for the dictionary data.

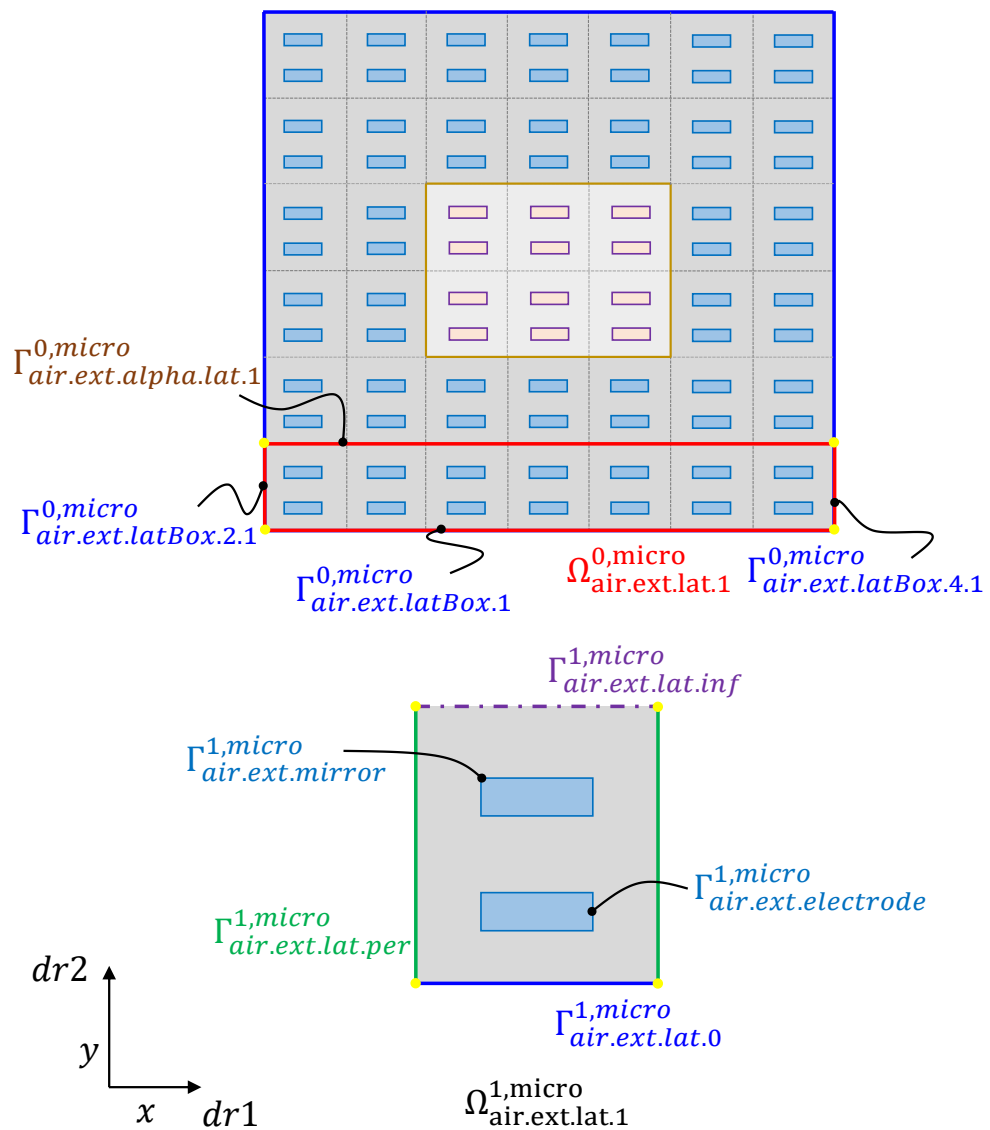


Figure 4.19: Selection of the domain of a boundary layer model in the physical domain and the related microscopic domain.

Domain Name	"air.ext.lat.1"
Boundary Name	"air.ext.lat.1"
Subboundary Name	["air.ext.mirror", "air.ext.electrode", "air.ext.top", "air.ext.lat.periodic" "air.ext.lat.0", "air.ext.lat.inf"]

Table 4.11: The names of the regions of the microscopic domain $\Omega_{air.ext.lat}^{1,micro}$ for the boundary layer model.

name	"boundaryLayer.1"			
regionName	domainNameList	("air.ext.lat.1", "air.ext")		
	boundaryNameList	["air.ext.mirror", "air.ext.electrode", "air.ext.top", "air.ext.latBox.1", ("air.ext.latBox.2.1", "air.ext.latBox.2"), ("air.ext.latBox.4.1", "air.ext.latBox.4"), "air.ext.alpha.lat.1"]		
modelName	pdeName	"pde.Mira",		
	equationNameList	"laplace.ext"		
	bcNameList	["dirichlet.ext.mirror", "dirichlet.ext.electrode", "neumann.ext.top", "neumann.ext.latBox", "neumann.ext.latBox", "neumann.ext.latBox", ""]		
dilation	typeList	["P", "BL", "N"]		
	axisList	[dr1, dr2, dr3]		
	scaleList	["eps", "eps", ""]		
	dictionary	physBdName	[latBdP, latOriBdP, intBdP]	
		microBdName	[perBd, latOriBd, latInfBd]	
		unknownFunction	[physUnknown, microUnknown]	
givenFunction		[givenFunction[1]]		
subScale	name	"lateral.1"		
	region	$\Omega_{air.ext.lat.1}^{1,micro}$		

Table 4.12: The submodel nodes of the boundary layer model.

Key	Description	Data list
latBdP	"lateral physical boundary"	["air.ext.latBox.1.1", "air.ext.latBox.4.1"]
latOriBdP	"original lateral physical boundary"	"air.ext.latBox.2"
intBdP	"interior physical boundary"	"air.ext.alpha.lat.1"
perBd	"periodic boundary"	"air.ext.periodic"
latOriBd	"original lateral boundary"	"air.ext.lat.0"
latInfBd	"infinity lateral boundary"	"air.ext.lat.inf"
physUnknown	"physical unknown"	["phi.Ext", idSC.Mira]
microUnknown	"microscopic unknown"	["phi.Lat", idSC.BL.1]
givenFunction[1]	"external periodic solution"	["phi.Ext", idSC.Per.Ext]

Table 4.13: Dictionary of the dilation node for the boundary layer model. Each entry of the dictionary includes a key, a description and a list of data. The data list of the unknowns have two elements, the name of the unknown and the identifier of their submodel.

4.4.3.2/ STRATEGY FOR A BOUNDARY LAYER MODEL

Algorithm 2 Derivation of a boundary layer model from the physical model.

1: In model of the top subScale node, store all equations in equationList and all boundary conditions in bcList whose names are in pdeName/equationNameList and bcNameList of the model "lateral.X" .

== In the following the operations are in the subModel node with name "lateral.X" ==

2: goto a pde node of the subscale at a lower level, insert all copied data.

3: In subScale/model replace the name given in physUnknown of the unknown function by this in microUnknown. This function is noted ϕ in the following.

4. goto in pde/bc and then in dirichlet equalities, replace the right-hand side by 0.

5: goto in pde/bc and then in a neumann equality applied on the boundary given in latOriBdP, replace the right-hand side by $\text{mapping}\{\text{rName1},\text{rName2}\}(-\nabla\phi_1)("") \cdot n$, where ϕ_1 is given in givenFunction[1], rName1 is a microscopic domain name in subScale/region of this boundary layer model, rName2 is a microscopic domain name of the periodic model given in the subScale/region represented by its identifier idSC.Per.Ext. We note that the function mapping defined in Section 4.2.6 plays a role of a change of variable.

6: replace everywhere the name of the boundary with name in latOriBdP by the name in latOriBd.

7: replace in subScale/model the domain with name in regionName/domainName by the microscopic domain in subScale/region.

8: goto bcList node, insert a "neumann" bc for ϕ with a right-hand side 0 on a boundary with name in the node latInfBd.

9: goto bcList node, insert a periodic bc and an anti-periodic bc for ϕ and $\nabla\phi \cdot n$, respectively, on a boundary which name is given in the node perBd.

10: rename the pde name by "pde.micro".

4.4.4/ EDGE BOUNDARY LAYER MODELS

4.4.4.1/ DATA STRUCTURE FOR AN EDGE BOUNDARY LAYER MODEL

The edge boundary layer model is illustrated with a single case shown in Figure 4.20. The region names of the microscopic domain $\Omega_{\text{air.ext.edge.1}}^{\text{1.micro}}$ are summarized in Table 4.14. The subModel data are in Table 4.15 and Table 4.16 is for the dictionary data.

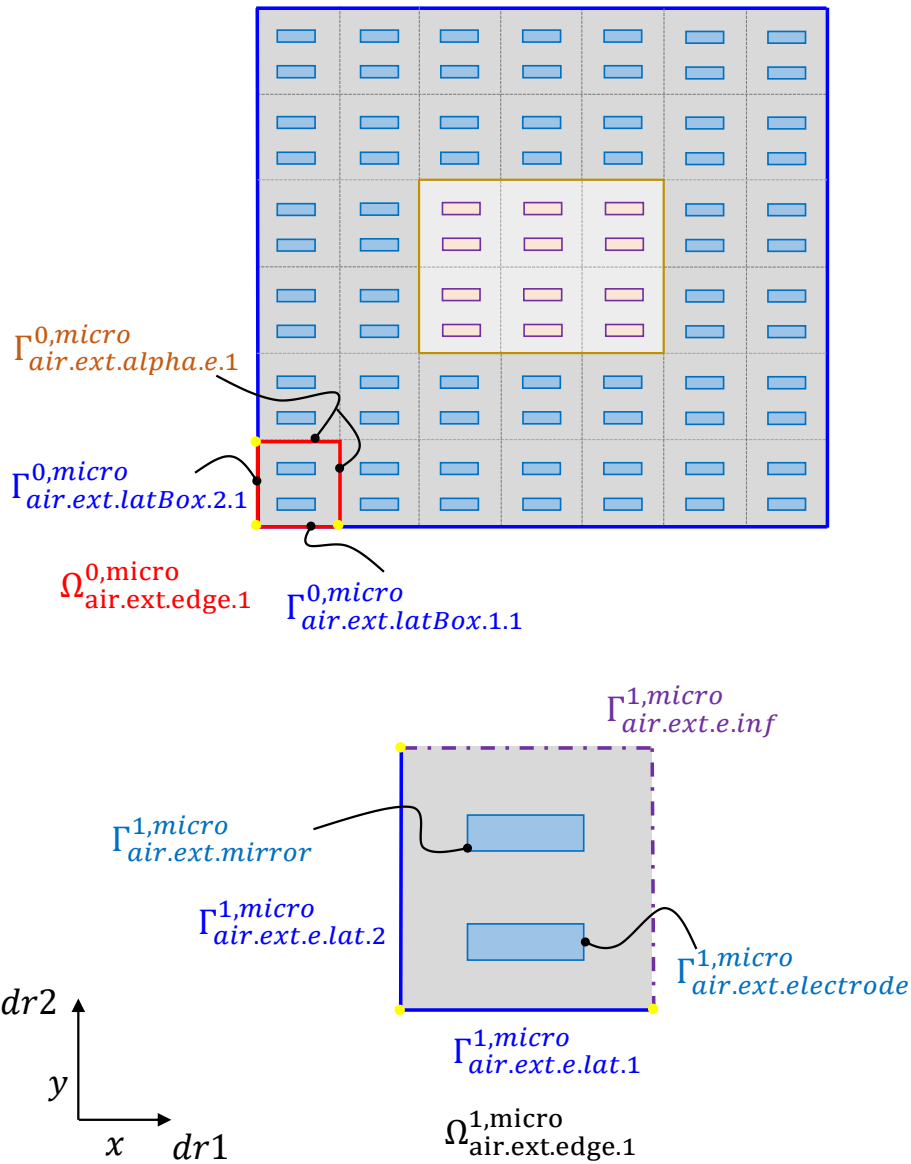


Figure 4.20: Selection of the domain of an edge boundary layer model in the physical domain and the related microscopic domain.

Domain Name	"air.ext.edge.1"
Boundary Name	"air.ext.edge.1"
Subboundary Name	["air.ext.mirror", "air.ext.electrode", "air.ext.top", "air.ext.e.lat.1", "air.ext.e.lat.2", "air.ext.e.inf"]

Table 4.14: The names of the regions of the microscopic domain $\Omega_{air.ext.edge.1}^{1,micro}$ for the edge boundary layer model.

name	"boundaryLayer.edge1"		
regionName	domainNameList	("air.ext.edge.1", "air.ext")	
	boundaryNameList	["air.ext.mirror", "air.ext.electrode", "air.ext.top", ("air.ext.latBox.1.1", "air.ext.latBox.1") ("air.ext.latBox.2.1", "air.ext.latBox.2"), "air.ext.e.alpha"]	
modelName	pdeName	"pde.Mira"	
	equationNameList	"laplace.ext"	
	bcNameList	["dirichlet.ext.mirror", "dirichlet.ext.electrode" "neumann.ext.top", "neumann.ext.latBox", "neumann.ext.latBox", ""]	
dilation	typeList	["BL_Edge", "BL_Edge", "T"]	
	axisList	[dr1, dr2, dr3]	
	scaleList	["eps", "eps", "eps"]	
	dictionary	physBdName	[latOriBdP, intBdP]
		microBdName	[latOriBd, latInfBd]
		unknownFunction	[physUnknown, microUnknown]
givenFunction		[givenFunction[1], givenFunction[2]]	
subScale	name	"externalEdge.1"	
	region	$\Omega_{air.ext.edge.1}^{1,micro}$	

Table 4.15: The submodel of the external edge model.

Key	Description	Data list
latOriBdP[1]	"first original lateral physical boundary"	"air.ext.latBox.1.1"
latOriBdP[2]	"second original lateral physical boundary"	"air.ext.latBox.2.1"
intBdP	"interior physical boundary"	"air.ext.alpha.e.1"
latOriBd[1]	"first original lateral boundary"	"air.ext.e.lat.1"
latOriBd[2]	"second original lateral boundary"	"air.ext.e.lat.2"
latInfBd	"infinity lateral boundary"	"air.ext.e.inf"
physUnknown	"physical unknown"	["phi.Ext", IdSC.Mira]
microUnknown	"microscopic unknown"	["phi.Ext.Edge", IdSC.BL.E.1]
givenFunction[1]	"first lateral boundary solution"	["phi.Lat", IdSC.BL.1]
givenFunction[2]	"second lateral boundary solution"	["phi.Lat", IdSC.BL.2]

Table 4.16: Dictionary of the dilation node for the edge boundary layer model.

4.4.4.2/ STRATEGY FOR AN EDGE BOUNDARY LAYER MODEL

Algorithm 3 Derivation of an edge boundary layer model from the physical model.

1: Same as Algorithm 2 but for a model "externalEdge.X".

= In the following the operations are in the subModel node with name "externalEdge.X" =

2-4: Same as Algorithm 2.

5: goto in pde/bc and then in a neumann equality applied on the boundary given in latOriBdP[2], replace the right-hand side by $\text{mapping}\{\text{rName1}, \text{rName2}\}(-\nabla\phi_1)("") \cdot n$, where ϕ_1 is given in givenFunction[1], rName1 is a microscopic domain name in subScale/region of this edge boundary model, rName2 is a microscopic domain name of the first boundary layer model given in the subScale/region represented by its identifier idSC.BL.1.

6: goto in pde/bc and then in a neumann equality applied on the boundary given in latOriBdP[1], replace the right-hand side by $\text{mapping}\{\text{rName1}, \text{rName3}\}(-\nabla\phi_2)("") \cdot n$, where ϕ_2 is given in givenFunction[2], rName2 is a microscopic domain name of the second boundary layer model that is given in the subScale/region represented by its identifier idSC.BL.2.

7: replace everywhere the name of the boundary with name in latOriBdP[1] and latOriBdP[2] by the name in latOriBd[1] and latOriBd[2] respectively.

8: replace in subScale/model the domain with name in regionName/domainName by the microscopic domain in subScale/region.

9-10: Same as Step 8 and 10 of Algorithm 2.

4.4.5/ INTERFACE BOUNDARY LAYER MODELS

4.4.5.1/ DATA STRUCTURE FOR AN INTERFACE BOUNDARY LAYER MODEL

The interface boundary layer model is illustrated with a single case shown in Figure 4.21. The region names of the microscopic domain $\Omega_{\text{air.interface.1}}^{\text{1,micro}}$ are summarized in Table 4.17. The subModel data are in Table 4.18 and Table 4.19 is for the dictionary data.

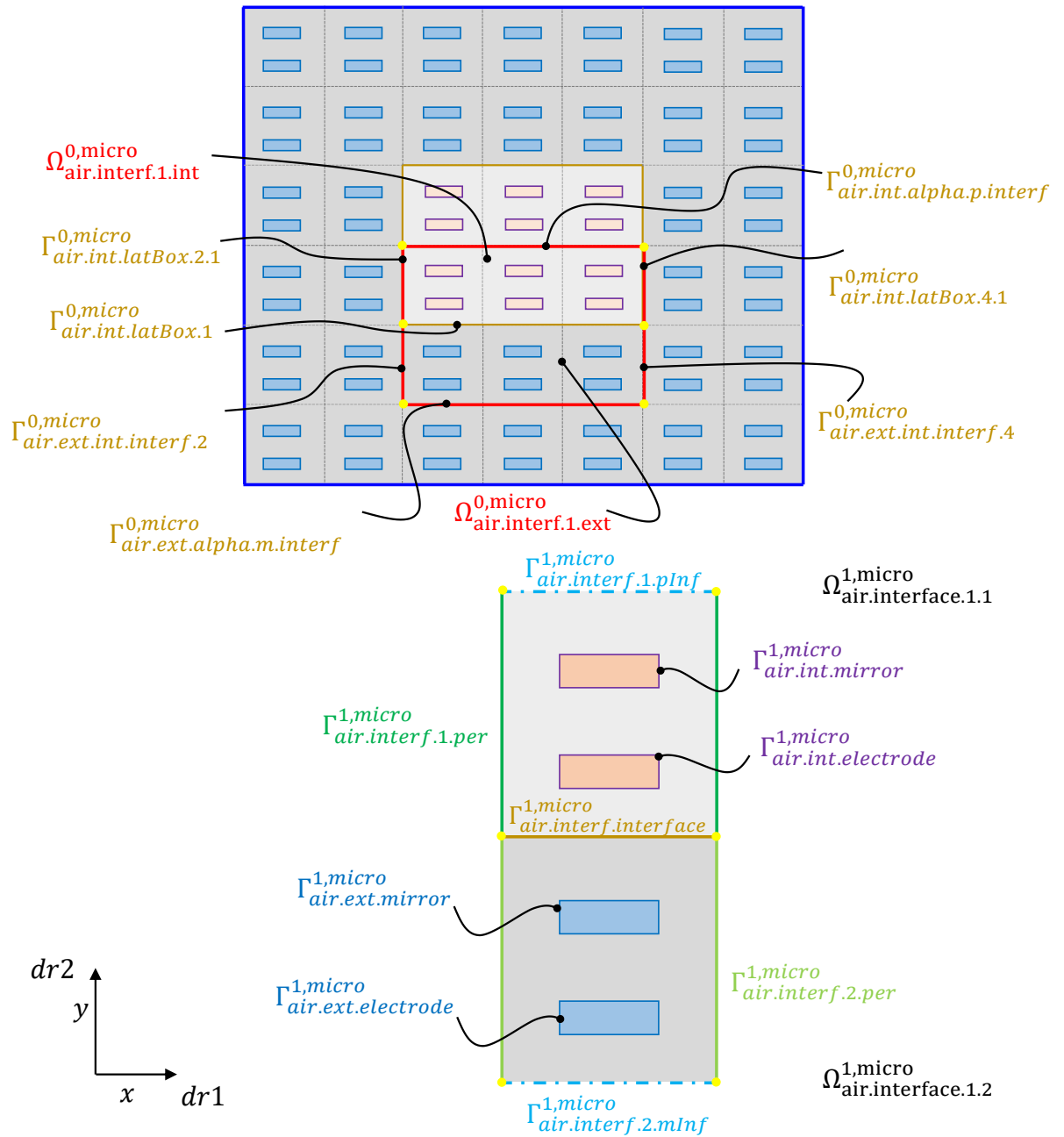


Figure 4.21: The domain of an interface boundary layer model in the physical domain and the related microscopic domain. The physical domain $\Omega^{0,micro}_{air.interface.1}$ includes two non-overlapping subdomains $\Omega^{1,micro}_{air.interface.1.ext}$ and $\Omega^{1,micro}_{air.interface.1.int}$ with the interface $\Gamma^{0,micro}_{air.int.latBox.1}$, the corresponding microscopic domain $\Omega^{1,micro}_{air.interface.1}$ has also two non-overlapping subdomains $\Omega^{1,micro}_{air.interface.1.1}$ and $\Omega^{1,micro}_{air.interface.1.2}$ with the interface $\Gamma^{1,micro}_{air.interf.interface}$.

Domain Name	“air.interface.1.1”	“air.interface.1.2”
Boundary Name	“air.interface.1.1”	“air.interface.1.2”
Subboundary Name	[“air.int.mirror”, “air.int.electrode”, “air.int.top”, “air.interf.1.plnf”, “air.interf.1.periodic”, “air.interf.interface”]	[“air.ext.mirror”, “air.ext.electrode”, “air.ext.top”, “air.interf.2.mlnf”, “air.interf.2.periodic”, “air.interf.interface”]

Table 4.17: Summary of the two subregion names $\Omega_{\text{air.interface.1.1}}^{1,\text{micro}}$ and $\Omega_{\text{air.interface.1.2}}^{1,\text{micro}}$ of the microscopic domain $\Omega_{\text{air.interface.1}}^{1,\text{micro}}$ for the interface boundary layer model.

Next, we describe a submodel for the first interface model.

name	“boundaryLayer.interface.i”		
regionName	domainNameList	[“air.interf.ext”, “air.ext”), (“air.interf.int”, “air.int”)]	
	boundaryNameList	[“air.ext.mirror”, “air.ext.electrode”, “air.ext.top”, “air.ext.int.interf.2”, “air.ext.int.interf.4”, “air.ext.alpha.m.interf”, “air.int.mirror”, “air.int.electrode”, “air.int.top”, (“air.int.latBox.2.1”, “air.int.latBox.2”), “air.int.latBox.1” (“air.int.latBox.4.1”, “air.int.latBox.4”), “air.int.alpha.p.interf”]	
modelName	equationNameList	[“laplace.ext”, “laplace.int”]	
	bcNameList	[“dirichlet.ext.mirror”, “dirichlet.ext.electrode”, “neumann.ext.top”, “”, “”, “”, “dirichlet.int.mirror”, “dirichlet.int.electrode”, “neumann.int.top”, “”, (“continuity.potential”, “continuity.flux”), “”, “”]	
dilation	typeList	[“P”, “BLI”, “N”]	
	axisList	[dr1, dr2, dr3]	
	scaleList	[“esp”, “esp”, “esp”]	
	dictionary	physBdName	[latIntBdP, intBdP, interfaceP]
		microBdName	[perBd, latInfBd, interface]
		unknownFunction	[physUnknown, microUnknown]
givenFunction	[givenFunction[1], givenFunction[2]]		
subScale	name	“interface.1”	
	region	$\Omega_{\text{air.interface.1}}^{1,\text{micro}}$	

Table 4.18: The subModel node of the interface boundary layer model.

Key	Description	Data list
latIntBdP[1]	“first interior lateral boundary”	[“air.ext.int.interf.2”, “air.ext.int.interf.4”]
latIntBdP[2]	“second interior lateral boundary”	[“air.int.latBox.2.1”, “air.int.latBox.4.1”]
intBdP[1]	“first interior boundary”	“air.int.alpha.p.interf”
intBdP[2]	“second interior boundary”	“air.ext.alpha.m.interf”
interfaceP	“physical interface”	“air.int.latBox.1”
perBd[1]	“first periodic boundary”	“air.interf.1.periodic”
perBd[2]	“second periodic boundary”	“air.interf.2.periodic”
latInfBd[1]	“first infinity lateral boundary”	“air.interf.1.pInf”
latInfBd[2]	“second infinity lateral boundary”	“air.interf.2.mInf”
interface	“microscopic interface”	“air.interf.interface”
physUnknown[1]	“internal physical unknown”	[“phi.Int”, idSC.Mira]
physUnknown[2]	“external physical unknown”	[“phi.Ext”, idSC.Mira]
microUnknown[1]	“first microscopic unknown”	[“phi.IntF.L”, idSC.BL.I.1]
microUnknown[2]	“second microscopic unknown”	[“phi.IntF.R”, idSC.BL.I.1]
givenFunction[1]	“internal periodic solution”	[“phi.Per”, idSC.Per.Int]
givenFunction[2]	“external periodic solution”	[“phi.Per”, idSC.Per.Ext]

Table 4.19: Dictionary of the dilation node for the interface boundary layer model.

4.4.5.2/ STRATEGY FOR AN INTERFACE BOUNDARY LAYER MODEL

Algorithm 4 Derivation of an interface boundary layer model from the physical model.

1: Same as Algorithm 2 but for a model "interface.X".

== In the following the operations are in the subModel node with name "interface.X" ==

2: Same as Algorithm 2.

3: In subScale/model replace the name given in physUnknown[1] and physUnknown[2] of the unknown function by this in microUnknown[1] and microUnknown[2] respectively. These functions are noted ϕ_L and ϕ_R in the following.

4: Same as Algorithm 2.

5: goto in pde/bc and then in the continuity equality, replace the right-hand side by $-(\text{mapping}\{rName1, rName2\}(\phi_1)("") - \text{mapping}\{rName3, rName4\}(\phi_2)(""))$, where ϕ_1 and ϕ_2 are respectively given in givenFunction[1] and givenFunction[2], rName1 and rName3 are subdomain names of the microscopic domain in subScale/region of this interface model, rName2 and rName4 are microscopic domain names of the internal and external periodic model given in the subScale/region represented by their identifiers idSC.Per.Int and idSC.Per.Ext.

6: goto in pde/bc and then in the continuityFlux equality, replace the right-hand side by $-(\text{mapping}\{rName1, rName2\}(\nabla\phi_1)("") \cdot n - \text{mapping}\{rName3, rName4\}(\nabla\phi_2)("") \cdot n)$.

7: replace the name of the boundary with name in interfaceP by the name in interface node.

8: replace in subScale/model the domain with name in regionName/domainName by the microscopic domain in subScale/region.

9: goto bcList node, insert homogeneous neumann bc for the unknowns ϕ_L and ϕ_R on the boundary with names in latInfBd[1] and latInfBd[2] respectively.

10: goto bcList node, insert a periodic bc and an anti-periodic bc for ϕ_L and $\nabla\phi_L \cdot n$, respectively, on a boundary which name is given in the node perBd[1].

11: goto bcList node, insert a periodic bc and an anti-periodic bc for ϕ_R and $\nabla\phi_R \cdot n$, respectively, on a boundary which name is given in the node perBd[2].

12: rename the pde name by "pde.micro".

4.4.6/ INTERFACE EDGE BOUNDARY LAYER MODELS

4.4.6.1/ DATA STRUCTURE FOR AN INTERFACE EDGE BOUNDARY LAYER MODEL

The interface edge boundary layer model is illustrated with a single case shown in Figure 4.22. The region names of the microscopic domain $\Omega_{\text{air.int.edge.1}}^{1,\text{micro}}$ are summarized in Table 4.20. The subModel1 data are in Table 4.21 and Table 4.22 is for the dictionary data.

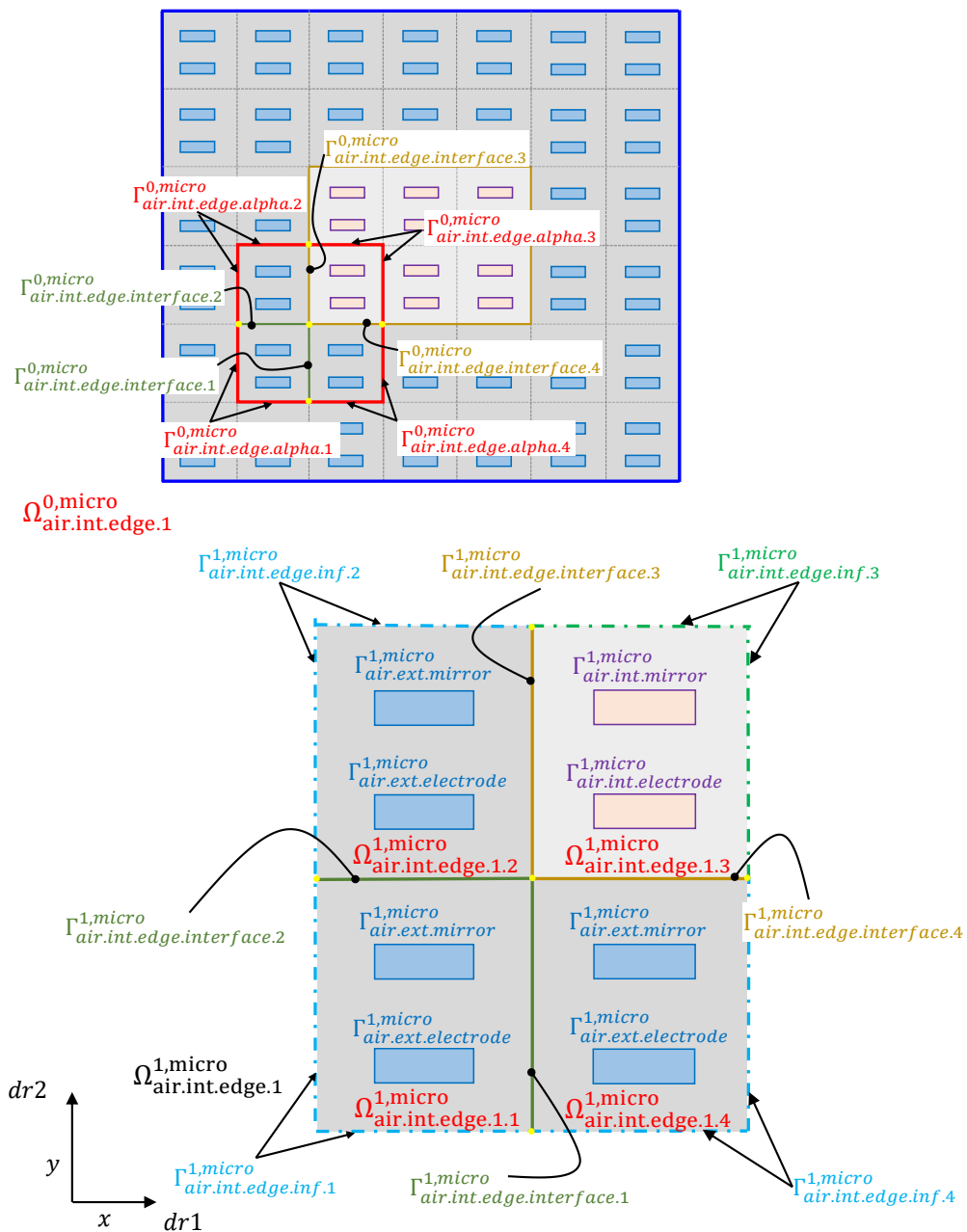


Figure 4.22: Selection of the domain of an interface edge boundary layer model in the physical domain and the related microscopic domain.

Domain Name	“air.int.edge.1.1”	“air.int.edge.1.2”
Boundary Name	“air.int.edge.1.1”	“air.int.edge.1.2”
Subboundary Name	[“air.ext.mirror”, “air.ext.electrode”, “air.ext.top”, “air.ext.edge.inf.1”, “air.int.edge.interface.1” “air.int.edge.interface.2”	[“air.ext.mirror”, “air.ext.electrode”, “air.ext.top”, “air.ext.edge.inf.2”, “air.int.edge.interface.2” “air.int.edge.interface.3”
Domain Name	“air.int.edge.1.3”	“air.int.edge.1.4”
Boundary Name	“air.int.edge.1.3”	“air.int.edge.1.4”
Subboundary Name	[“air.int.mirror”, “air.int.electrode”, “air.int.top”, “air.ext.edge.inf.3”, “air.int.edge.interface.3” “air.int.edge.interface.4”	[“air.ext.mirror”, “air.ext.electrode”, “air.ext.top”, “air.ext.edge.inf.4”, “air.int.edge.interface.4” “air.int.edge.interface.1”

Table 4.20: Summary of the region names of the microscopic domain $\Omega_{\text{air.int.edge.1}}^{1,\text{micro}}$ for the interface edge boundary layer model.

name	“boundary.interface.edge.1”		
regionName	domainNameList	[(“air.int.edge.1”, “air.ext”), (“air.int.edge.2”, “air.ext”), (“air.int.edge.3”, “air.int”), (“air.int.edge.4”, “air.ext”)]	
	boundaryNameList	[“air.ext.mirror”, “air.ext.electrode”, “air.ext.top” “air.int.edge.alpha.1”, “air.int.edge.interface.1” “air.int.edge.interface.2”, “air.int.edge.alpha.2”, (“air.int.edge.interface.3”, “air.int.latBox.2”), (“air.int.edge.interface.4”, “air.int.latBox.1”), “air.int.mirror”, “air.int.electrode”, “air.int.top” “air.int.edge.alpha.3”, “air.int.edge.alpha.4”]	
modelName	equationNameList	[“laplace.ext”, “laplace.ext”, “laplace.ext”, “laplace.int”]	
	bcNameList	[“dirichlet.ext.mirror”, “dirichlet.ext.electrode”, “neumann.ext.top”, “”, “”, “”, “”, (“continuity.potential”, “continuity.flux”), (“continuity.potential”, “continuity.flux”), “dirichlet.int.mirror”, “dirichlet.int.electrode”, “neumann.int.top”, “”, “”]	
dilation	typeList	[“BLI_Edge”, “BLI_Edge”, “N”]	
	axisList	[dr1, dr2, dr3]	
	scaleList	[“eps”, “eps”, “”]	
	dictionary	physBdName	[intBdP, interfaceP]
		microBdName	[latInfBd, interface]
unknownFunction		[physUnknown, microUnknown]	
	givenFunction	[givenFunction]	
subScale	name	“internalEdge.1”	
	region	$\Omega_{\text{air.int.edge.1}}^{1,\text{micro}}$	

Table 4.21: The subModel node of the interface edge boundary layer.

Key	Description	Data list
intBdP[1]	“first interior lateral boundary”	“air.int.edge.alpha.1”
intBdP[2]	“second interior lateral boundary”	“air.int.edge.alpha.2”
intBdP[3]	“third interior boundary”	“air.int.edge.alpha.3”
intBdP[4]	“fourth interior boundary”	“air.int.edge.alpha.4”
interfaceP[1]	“first physical interface”	“air.int.edge.interface.1”
interfaceP[2]	“second physical interface”	“air.int.edge.interface.2”
interfaceP[3]	“third physical interface”	[“air.int.edge.interface.3”, “air.int.latBox.2”]
interfaceP[4]	“fourth physical interface”	[“air.int.edge.interface.4”, “air.int.latBox.1”]
interface[1]	“first microscopic interface”	“air.int.edge.interface.1”
interface[2]	“second microscopic interface”	“air.int.edge.interface.2”
interface[3]	“third microscopic interface”	“air.int.edge.interface.3”
interface[4]	“fourth microscopic interface”	“air.int.edge.interface.4”
latInfBd[1]	“first infinity lateral boundary”	“air.ext.edge.inf.1”
latInfBd[2]	“second infinity lateral boundary”	“air.ext.edge.inf.2”
latInfBd[3]	“third infinity lateral boundary”	“air.ext.edge.inf.3”
latInfBd[4]	“fourth infinity lateral boundary”	“air.ext.edge.inf.4”
physUnknown[1]	“internal physical unknown”	[“phi.Int”, IdSC.Mira]
physUnknown[2]	“internal physical unknown”	[“phi.Int”, IdSC.Mira]
physUnknown[3]	“internal physical unknown”	[“phi.Int”, IdSC.Mira]
physUnknown[4]	“external physical unknown”	[“phi.Ext”, IdSC.Mira]
microUnknown[1]	“first microscopic unknown”	[“phi.IntE.1”, IdSC.BL.E.1]
microUnknown[2]	“second microscopic unknown”	[“phi.IntE.2”, IdSC.BL.E.1]
microUnknown[3]	“third microscopic unknown”	[“phi.IntE.3”, IdSC.BL.E.1]
microUnknown[4]	“fourth microscopic unknown”	[“phi.IntE.4”, IdSC.BL.E.1]
givenFunction[1]	“internal periodic solution”	[“phi.Per.Int”, IdSC.Per.Int]
givenFunction[2]	“external periodic solution”	[“phi.Per.Ext”, IdSC.Per.Ext]
givenFunction[3]	“left first interface solution”	[“phi.IntF.L”, IdSC.BL.I.1]
givenFunction[4]	“right first interface solution”	[“phi.IntF.R”, IdSC.BL.I.1]
givenFunction[5]	“left second interface solution”	[“phi.IntF.L”, IdSC.BL.I.2]
givenFunction[6]	“right second interface solution”	[“phi.IntF.R”, IdSC.BL.I.2]

Table 4.22: Dictionary of the dilation node for the interface edge boundary layer model.

4.4.6.2/ STRATEGY FOR AN INTERFACE EDGE BOUNDARY LAYER MODEL

Algorithm 5 Derivation of an interface edge boundary layer model from the physical model.

1: Same as Algorithm 2 but for a model "intEdge.X".

== In the following the operations are in the subModel node with name "intEdge.X" ==

2: Same as Algorithm 2.

3: In subScale/model replace the name given in physUnknown of the unknown function by this in microUnknown. This function is noted respectively ϕ_{ine}^1 , ϕ_{ine}^2 , ϕ_{ine}^3 and ϕ_{ine}^4 in the following.

4: Same as Algorithm 2.

5: goto in pde/bc and then in the continuity equality applying the boundary given in interfaceP[3], replace the right-hand side by $- \text{mapping}\{\text{rName1}, \text{rName2}\}(\phi_{in,R}^1)("")$, where $\phi_{in,R}^1$ is respectively in givenFunction[4], rName1 is the third microscopic subdomain name in subScale/region of this interface edge model, rName2 is the second microscopic subdomain name of the first interface model given in subScale/region represented by its identifier idSC.BL.I.1.

6: goto in pde/bc and then in the continuityFlux equality applying the boundary given in interfaceP[3], replace the right-hand side by $- \text{mapping}\{\text{rName1}, \text{rName2}\}(\nabla\phi_{in,R}^1)("") \cdot n$.

7: goto in pde/bc and then in the continuity equality applying the boundary given in interfaceP[4], replace the right-hand side by $- \text{mapping}\{\text{rName1}, \text{rName3}\}(\phi_{in,R}^2)("")$, where $\phi_{in,R}^2$ is respectively in givenFunction[6], rName3 is the second microscopic subdomain name of the second interface model given in subScale/region represented by its identifier idSC.BL.I.2.

8: goto in pde/bc and then in the continuityFlux equality applying the boundary given in interfaceP[4], replace the right-hand side by $- \text{mapping}\{\text{rName1}, \text{rName3}\}(\nabla\phi_{in,R}^2)("") \cdot n$.

9: replace the name of the boundary with name in interfaceP by the name in interface node.

10: replace in subScale/model the domain with name in regionName/domainName by the microscopic domain in subScale/region.

11: goto bcList node, insert an continuity bc on the interface given in interface[1] with the left-hand side of bc $\phi_{ine}^1 - \phi_{ine}^4$ and the right-hand side $\text{mapping}\{\text{rName4}, \text{rName5}\}(\phi_{in,L}^1)("")$, where $\phi_{in,L}^1$ is respectively in givenFunction[3], rName4 is the fourth microscopic subdomain name in subScale/region of this interface edge model, rName5 is the first microscopic subdomain name of the first interface model given in subScale/region represented by its identifier idSC.BL.I.1.

Algorithm 5 Continuous of the Algorithm 5.

12: goto bcList node, insert an continuityFlux bc on the interface given in interface[1] with the left-hand side of bc $\nabla\phi_{ine}^1 \cdot n - \nabla\phi_{ine}^4 \cdot n$ and the right-hand side mapping{rName4,rName5}($\nabla\phi_{in,L}^1 \cdot n$)("").

13: goto bcList node, insert an continuity bc on the interface given in interface[2] with the left-hand side of bc $\phi_{ine}^1 - \phi_{ine}^2$ and the right-hand side mapping{rName6,rName7}($\phi_{in,L}^2$)(""), where $\phi_{in,L}^2$ is respectively in givenFunction[5], rName6 is the second microscopic subdomain name in subScale/region of this interface edge model, rName7 is the first microscopic subdomain name of the second interface model given in subScale/region represented by its identifier idSC.BL.I.2.

14: goto bcList node, insert an continuityFlux bc on the interface given in interface[5] with the left-hand side of bc $\nabla\phi_{ine}^1 \cdot n - \nabla\phi_{ine}^2 \cdot n$ and the right-hand side mapping{rName6,rName7}($\nabla\phi_{in,L}^2 \cdot n$)("").

15: goto bcList node, insert homogeneous neumann bcs for respectively $\phi_{ine}^1, \phi_{ine}^2, \phi_{ine}^3, \phi_{ine}^4$ on boundaries with names in the corresponding nodes latInfBd[1], latInfBd[2], latInfBd[3], latInfBd[4].

16: goto a pde name node, rename a current name by "pde.micro".

A REWRITING AND DESCRIPTION LOGIC CONCURRENT SYSTEM

5.1/ INTRODUCTION

Our previous works have led on the one hand to a data structure in the form of a tree that represents multi-scale models made up of partial differential equations and on the other hand to a rewriting system allowing to operate on it. The interest of a rewriting system is that it allows fast and simple navigation in a tree of data that can contain complex structures. It allows application of rules to verify hypotheses or perform transformations. It also allows to perform chaining of rule applications for a reasoning.

However, a rewriting system is not enough to organize and manage a large number of hypotheses and conclusions associated for instance for a large number of theorem statements. A natural choice to overcome this limitation is to use elements of logic and related inferences. Rather than developing an inference engine embedded in the rewriting engine (which would be ideal), we propose to build a concurrent system made with our rewriting engine and an existing inference engine based on a Description Logic (DL). This approach must take into account the complementarity and limitations of the two engines as well as their interaction.

Description logic is used for ontology construction and inference operations. It allows data structuring in the form of graphs, with labeled branches, which makes it possible to describe any relation between any entities. Expressions can be built, using constructors detailed hereafter, on these relations yielding possible inferences and therefore queries. Description logic techniques have strongly developed during the last forty years, see e.g. the reference books [15, 14, 13, 16], and today are at the heart of the knowledge graph concept used by the major actors of the web, see the comprehensive review [58]. Among others, some of their practical advantages are their relative simplicity of use compared to other logical systems, their support by the W3C community with the language *OWL*, a

number of open inference engines as *Hermit* or *Pellet*, and the *Protégé* editor. These language and tools are widely used.

The motivations for a concurrent RDL system of Rewriting and Description Logic are on the one hand the simplicity of hypothesis testing on very complex (mathematical) expressions and their transformation inherited from the rewriting system and on the other hand a relatively simple manner to represent a large variety of relations between the subtree of a data tree used for rewriting and to build complex expressions on these relations. It also allows to take into account other data, not present in the data tree, to infer on DL expressions and finally to formulate queries. Ideally, all these operations must be integrated.

This work presents an attempt to integrate our rewriting system with a description logic. It elaborates a representation of the elementary principles of rewriting (terms, patterns, strategy and strategy application to a term) in description logic, and describes mechanisms of operation of the concurrent system. It has been realized in a relatively short time with the objective to study the feasibility of this approach and to sketch the way it could be used for the multi-scale models developed in the thesis. We are aware that a number of points remain to be clarified and deepened.

The approach presented in this work is to be distinguished from many works associating rewriting and logic. Although not experts on the subject, it seems to us that rewriting logic such as that deployed in Maude, ELAN or CafeOBJ software, see references [72, 68] for the foundations of Maude or the more recent review [61], consists in enriching or reinforcing a rewriting system with logical operations. In comparison, the logic layer deployed in the RDL system focuses on the possibility of representing a large amount of knowledge. In this context, logic is used to categorize knowledge and inference is used to draw conclusions from this knowledge.

From the ontology point of view, we believe that the RDL approach could bring a new possibility to operate on complex mathematical expressions commonly encountered in engineering and science. The limitation of ontologies for this use is moreover an accepted fact. We however notice that the development of graph navigation tools is a highly developed topic at least for queries, see the review [9].

The rest of the chapter starts with a short introduction of the description logic formalism $SHOI^{(D)}$ common to the OWL2 language ($SROIQ^{(D)}$) and the editor Protégé ($SHOIN^{(D)}$). Then, the rewriting concepts useful for this work and the structure of our rewriting language are presented in a synthetic way. This is followed by a description of the RDL system and finally an application to the representation of the algorithms associated with the multi-scale models of the previous chapter. For lack of time, only the rewriting strategies including queries to the ontology are reported. The rest of the ontology presenting the link between the different algorithms remains to be developed. Moreover, we believe that the representation adopted, in the algorithms, the strategies and

the ontology is very preliminary and will evolve significantly towards a simpler and more structured design in order to represent a larger number of models.

5.2/ DESCRIPTION LOGIC

We recall the principles of description logic, in a manner strongly inspired from the french Wikipedia page.

5.2.1/ SEMANTICS

Let $C = \{c_1, c_2, \dots\}$ be a finite set of atomic concepts, $R = \{r_1, r_2, \dots\}$ a finite set of atomic roles and $I = \{i_1, i_2, \dots\}$ a finite set of individuals. If C, R, I are two by two disjoint, $S = \langle C, R, I \rangle$ is a signature. Once a signature S is fixed, an interpretation \mathcal{I} of S is a couple $\mathcal{I} = \langle \Delta^{\mathcal{I}}, \cdot^{\mathcal{I}} \rangle$, where :

- $\Delta^{\mathcal{I}}$ is a non-empty set, the domain of interpretation.
- $\cdot^{\mathcal{I}}$ is a function affecting :
 - an element $i^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ to each individual $i \in I$;
 - a subset $c^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each atomic concept $c \in C$;
 - and a relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ to each atomic role $r \in R$.

5.2.2/ KNOWLEDGE BASE

Given a descriptive language \mathcal{L} and a signature S , a knowledge base Σ in \mathcal{L} is a pair $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ such that :

- \mathcal{T} is the \mathcal{T} (erminology)-Box, a finite set, which can be empty, of expressions called GCI (General Concept Inclusion) of the form $c_1 \sqsubseteq c_2$ where c_1 and c_2 are unrestricted concepts. $c_1 \dot{\sqsubseteq} c_2$ is a notation for $c_1 \sqsubseteq c_2$ and $c_2 \sqsubseteq c_1$. The formulae of \mathcal{T} are called "terminological axioms".
- \mathcal{A} is the \mathcal{A} (ssertion)-Box, a finite set, which can be empty, of expressions of the form $i : c$ or $(i, j) : r$, where c is an unrestricted concept, r is a role which is not necessarily atomic, and i, j belong to I . The formulae \mathcal{A} are called "assertions".

In our work, we view an ontology knowledge base as a graphs, with edges labelled by roles and nodes labelled by sets of concept names. More precisely, in such a graph

- the nodes are the elements of the interpretation and they are labelled with all the concept names to which this element belongs in the interpretation;

- an edge with label r between two nodes says that the corresponding two elements of the interpretation are related by the role r .

5.2.3/ DIFFERENT DESCRIPTION LOGICS

The description logics have a common base enriched with different extensions (see Table 5.1 below). It is therefore possible to have complex concepts composed of atomic concepts, and the same applies to roles.

Letter	Constructor	Syntax	Semantics
\mathcal{AL}	concept name	c	c^I
\mathcal{AL}	role name	r	r^I
\mathcal{AL}	top	\top	Δ^I
\mathcal{AL}	conjunction	$c_1 \sqcap c_2$	$c_1^I \cap c_2^I$
\mathcal{AL}	universal quantifier	$\forall r.c$	$\{d_1 \in \Delta^I \mid \forall d_2 \in \Delta^I.(r^I(d_1, d_2) \rightarrow d_2 \in c^I)\}$
\mathcal{AL}	limited existential quantifier	$\exists r.\top$	$\{d_1 \in \Delta^I \mid \exists d_2 \in \Delta^I.(r^I(d_1, d_2) \wedge d_2 \in \top^I)\}$
C	negation of concepts that are not necessarily primitive	$\neg c$	$\Delta^I \setminus c^I$
\mathcal{U}	disjunction	$c_1 \sqcup c_2$	$c_1^I \cup c_2^I$
\mathcal{E}	typed existential quantifier	$\exists r.c$	$\{d_1 \in \Delta^I \mid \exists d_2 \in \Delta^I.(r^I(d_1, d_2) \wedge d_2 \in c^I)\}$
\mathcal{N}	cardinality restriction	$(\geq n r)$ or $(\leq n r)$	$\{d_1 \in \Delta^I \mid \{d_2 \mid r^I(d_1, d_2)\} \geq n\}$ or $\{d_1 \in \Delta^I \mid \{d_2 \mid r^I(d_1, d_2)\} \leq n\}$
Q	qualified cardinality restriction	$(\geq n r.c)$ or $(\leq n r.c)$	$\{d_1 \in \Delta^I \mid \{d_2 \mid r^I(d_1, d_2), d_2 \in c^I\} \geq n\}$ or $\{d_1 \in \Delta^I \mid \{d_2 \mid r^I(d_1, d_2), d_2 \in c^I\} \leq n\}$
O	one of	$\{a_1, \dots, a_n\}$	$\{d \in \Delta^I \mid d = i^I \text{ for an } i\}$
\mathcal{B}	role filler	$\exists r.\{i\}$	$\{d \in \Delta^I \mid r^I(d, i^I)\}$
\mathcal{R}	conjunction of roles	$r_1 \sqcap r_2$	$r_1^I \cap r_2^I$
\mathcal{I}	inverse roles	r^{-1}	$\{(d_1, d_2) \in \Delta^I \times \Delta^I \mid r^I(d_2, d_1)\}$
\mathcal{H}	hierarchy of roles	$r_1 \sqsubseteq r_2$	$r_1^I \subseteq r_2^I$

Table 5.1: Constructors of Description Logic.

The description logics that exist are combinations of the different elements in the table above. For example, if we add the complete negation C to the logic \mathcal{AL} , we obtain the logic \mathcal{ALC} . Moreover, \mathcal{S} is an abbreviation for \mathcal{ALC} with transitive roles and $^{(D)}$ means use of datatype properties, data values or data types.

The Protégé ontology editor supports $\mathcal{SHOIN}^{(D)}$. OWL 2 provides the expressiveness of $\mathcal{SROIQ}^{(D)}$, OWL-DL is based on $\mathcal{SHOIN}^{(D)}$, and for OWL-Lite it is $\mathcal{SHIF}^{(D)}$.

5.2.4/ INFERENCE

In DL, the concept of inference is described as follows. Let \mathcal{I} an interpretation and φ a terminological axiom or assertion. Thus \mathcal{I} modelizes φ (notation $\mathcal{I} \models \varphi$) if :

- $\varphi = c_1 \sqsubseteq c_2$ and $c_1^I \subseteq c_2^I$, or
- $\varphi = i : c$ and $i^I \in c^I$, or

- $\varphi = (i, j) : r$ and $(i^{\mathcal{I}}, j^{\mathcal{I}}) \in r^{\mathcal{I}}$.

Let $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ a knowledge base and \mathcal{I} an interpretation, then \mathcal{I} is a model of Σ (notation, $\mathcal{I} \models \Sigma$) so for all $\varphi \in \mathcal{T} \cup \mathcal{A}$, $\mathcal{I} \models \varphi$. We say in this case that \mathcal{I} is a model of the knowledge base Σ . Given a knowledge base Σ and a terminological axiom or assertion φ , $\Sigma \models \varphi$ if for any model \mathcal{I} of Σ we have $\mathcal{I} \models \varphi$.

5.2.5/ REASONING TASKS

In DL, the expression "reasoning on the \mathcal{T} -Box" refers to the ability to make inferences from a knowledge base. $\Sigma = \langle \mathcal{T}, \mathcal{A} \rangle$ \mathcal{T} is non-empty, and in a similar way, "A-Box reasoning" is the implication for a non-empty A-Box.

Let Σ a knowledge base, $c_1, c_2 \in C, r \in R$ and $i, j \in I$, we define the following basic deduction tasks.

- **Subsumption:** $\Sigma \models c_1 \sqsubseteq c_2$ Checks whether for all interpretations \mathcal{I} such that $\mathcal{I} \models \Sigma$, we have $c_1^{\mathcal{I}} \subseteq c_2^{\mathcal{I}}$.
- **Instance Verification:** $\Sigma \models i : c$ Checks whether for all interpretations \mathcal{I} such that $\mathcal{I} \models \Sigma$, we have $i^{\mathcal{I}} \in c^{\mathcal{I}}$.
- **Relationships check:** $\Sigma \models (i, j) : r$ Checks whether for all interpretations \mathcal{I} such that $\mathcal{I} \models \Sigma$, we have $(i^{\mathcal{I}}, j^{\mathcal{I}}) \in r^{\mathcal{I}}$.
- **Consistency of concept:** $\Sigma \not\models c \sqsubseteq \perp$ Checks whether for all interpretations \mathcal{I} such that $\mathcal{I} \models \Sigma$, we have $c^{\mathcal{I}} \neq \{\}$.
- **Consistency of the knowledge base:** $\Sigma \not\models \perp$ Checks whether there exists \mathcal{I} such that, $\mathcal{I} \models \Sigma$.

Basic deduction tasks can be used to define more complex tasks.

- **Search:** Given a concept, find the individuals mentioned in the knowledge base who are instances of that concept.
- **Fulfilment:** given an individual mentioned in the knowledge base, find the most specific concept, in accordance with the relations of subsumption, of which the individual is an instance.

5.3/ A SHORT OVERVIEW OF THE MEMSALAB REWRITING SYSTEM

There are many rewriting languages and libraries in the literature, such as *Tom* [3], *ρ -log* [43], *Elan* [1] and *Maude* [2]. *Tom* is a rewriting language built on java, that is, a language

that extends Java with high level rewriting constructors. ρ -log is a rule-based programming system implemented in the symbolic mathematical computation system Mathematica. *Elan* is a closed and functional language used to express non-deterministic computations via rewriting rules and strategies. *Maude* is a closed system that implements the rewriting logic.

The MEMSALab rewriting system that has been developed in our laboratory is an *OCaml* library. Its design was inspired by the open aspects of *Tom*, since it extends *OCaml* with high level rewriting constructors. It shares with *Elan* its functional aspect.

In the following, the notions of rewriting are reformulated in view of expressing their relationship with the logic of description. We start with the notions of node, pattern and tree related to Abstract Syntax Trees (AST). We continue with the notion of strategies before to present a simplified view of the grammar of strategies of MEMSALab.

5.3.1/ NODE, PATTERN AND TREE

The set of functions considered in our approach is restricted to the single form $\text{node}(\text{id}, \text{list})$ for nodes made with the an identifier id and a list list . The constants Q are considered separately. A constant has a simple type among boolean, string or number. A tree is defined by induction from nodes and constants according to the grammar

$$t := Q \mid \text{node}(\text{id}, [t, \dots])$$

If the list size of a node is a constant, one often prefer to manipulate the corresponding function whose name is the identifier id and whose variables are the list elements. For instance the function $\text{equality}(\text{name}, \text{left}, \text{right})$ represents the node $\text{node}(\text{"equality"}, [\text{name}, \text{left}, \text{right}])$.

The sets of node names and of constants are noted N and D . The signature of the grammar is $V = N \cup D$ and the set of trees t (or terms) built on the signature V is noted $T(V)$.

Let X be a set of rewriting variables, we note $T(V, X)$ the set of trees built on the signature $V \cup X$, we call them patterns. To simplify, we note P the set of patterns. For two patterns p_1, p_2 of P we recall that p_1 matches p_2 that we write $p_1 \ll p_2$, if there is a substitution σ such that $\sigma(p_1) = p_2$. The relation \ll defines a partial order on P .

5.3.2/ STRATEGY

For the purpose of strategy definition, the set $T(V)$ is completed by Fail : $T^*(V) = T(V) \cup \text{Fail}$ also noted T^* . A strategy s is a relation of $T^* \times T^*$, $s : t_1 \rightarrow t_2$, which we also

note $t_2 = s[t_1]$ to distinguish between this relation of 'strategy applied to a term' with the parenthesis used for strategy parametrization. For any strategy s , $s[\text{Fail}] = \text{Fail}$.

As for trees, strategies are built from strategy nodes $\text{nodeS}(\text{id}, \text{list})$. The grammar of strategies is

$$s := r \mid \text{nodeS}(\text{id}, [\text{p}, \dots, \text{s}, \dots])$$

where r represents terminal strategies and the variables of a strategy node can be patterns p or other strategy nodes s . The terminal strategies, i.e. strategies with no strategy as variable, are the rewriting rules $\text{nodeS}(\text{"rule"}, [\text{left}, \text{right}, \text{condition}])$ or $\text{rule}(\text{left}, \text{right}, \text{condition})$. A strategy consists in navigation and transformation operations.

The elementary navigation is the OuterMost strategy $\text{om}(p, s)$ that navigates as the outermost strategy with pattern p and then applies s . The two elementary operations on pairs of strategies are

- $\text{compose}(s_1, s_2)$ is the composition of two strategies, also noted $s_1; s_2$. When applied to a term t it applies s_1 to t and if the result t_1 is Fail returns Fail otherwise applies s_2 to t_1 and returns the result.

- $\text{leftChoice}(s_1, s_2)$ is the left-choice of two strategies, also noted $s_1|s_2$. When applied to a term t it applies s_1 to t and if the result t_1 is Fail applies s_2 to t and returns the result otherwise returns t_1 .

Any of the above strategies s can be repeated until a fixed point is reached with the $\text{repeat}(s)$ strategy.

Let us describe a few number of other strategy nodes defined from rewriting rules and from the OuterMost strategy.

- $\text{remove} := \text{rule}(x_, [])$ removes the current term of a list of nodes.
- $\text{replace}(t) := \text{rule}(x_, t)$ replaces the current term by t .
- $\text{insert}(t) := \text{rule}(\text{node}(x_, y_), \text{node}(x_, [y_, t]))$ inserts the term t to the right of the node list of the current node.
- $\text{goto}(p, p_1, s)$ check if the pattern p matches the current position and then applies $\text{om}(p_1, s)$.
- $\text{goto}(p, \pi, s)$ generalizes $\text{goto}(p, p_1, s)$, for $\pi = [p_1, p_2, \dots]$ a list of patterns, by induction $\text{goto}(p, p_1, \text{goto}(p_1, p_2, \dots))$.

Remark 7: Different forms of goto

In the following, different forms of the `goto` strategy, adapted to the need, are used. In Section 5.6.2.2 the first pattern p is omitted meaning that p is a rewriting variable matching any term. An aspect of the `goto` strategy that is hidden and that appears only in Section 5.6.2.2 is that in the actual implementation of `goto` there is a limit pattern a i.e. a pattern which is used to stop the strategy when it is found. Precisely, the strategy `goto(π , a , s)` navigates along the path π before to apply the strategy s except if a subterm that matches a is reached during navigation that causes the stop and success of the strategy.

Remark 8:

Three other usual elementary navigation strategies are available, namely `bottomUp`, `topDown`, and `innerMost`. They are not presented here since they have not yet been taken into account in the RDL system construction.

5.3.3/ EXTENDED GRAMMARS OF EXPRESSIONS AND STRATEGIES

In principle the above principles are sufficient to define a rewriting system. In practice, more functionalities are required for a more efficient programming.

We call Expressions the generalization of patterns that takes into account the following constructors.

- `globalVar(e)` the content of the memory (also called global variable) with name $e \in T(V)$, such memory is set by the strategy `store(e)` defined below,
- `dEval(e)` delays the evaluation of e until the execution of its hosting function (e.g. a rewriting rule),
- `eval(e)` evaluates an expression in case where e includes a delayed evaluation or a `globalVar(.)` out of a strategy,
- `evalCond(e)` evaluates an expression that is a condition and returns `True` or `False`,
- `s[e]` the term in $T(V)$ returned by the application of a strategy s to an expression e .

Accordingly, the grammar of strategies is completed with the constructor `store(e)` that stores the current term in the memory with name e .

Expression can also include any functions $f(e, \dots, s, \dots)$ parametrized by expressions e, \dots and strategies s, \dots and returning an expression. Among many possibilities, such a function can simply assemble tree nodes and return a tree as $t(x, y) : f(g(x), h(y))$ where f, g and h are three nodes, or be an external function. The same possibility holds for strategies, but in this case the external functions return strategies.

The extended grammar of expressions e and strategies s is detailed below. The sets of expressions and strategies are noted \mathcal{E} and \mathcal{S} .

$e := n \mid c \mid x \mid \ell \mid \text{node}(e, [e', \dots])$

$e' := e \mid \text{globalVar}(e) \mid \text{dEval}(e') \mid \text{eval}(e') \mid \text{evalCond}(e') \mid s[e'] \mid f(e', \dots, s, \dots)$

$s := \text{rule}(e, e', e) \mid \text{store}(e) \mid \text{om}(e, s) \mid s; s \mid s|s \mid \text{repeat}(s) \mid g(e', \dots, s, \dots)$

with

- n a number,
- c a string,
- x a rewriting variable,
- ℓ the logical True or False,
- $\text{rule}(l, r, c)$ a conditional rewriting rule with left hand side l , right hand side r and condition c where the left-hand side l and the condition c cannot be external functions.

Finally, it is noteworthy that expressions and strategies can be stored in identifiers (characterized by its name and its possible parameters) while only expressions can be stored memory accessible via `globalVar()`.

5.4/ REPRESENTATION OF REWRITING NOTIONS IN DESCRIPTION LOGIC

Now that the systems of rewriting and of logic of description have been recalled, it is a matter of establishing how they operate together in a RDL system. In this section, we propose a representation of certain rewriting system notions in a DL system. Keeping in mind that a rewriting system allows mainly to apply a strategy to a tree, the main requirement for such a construction are summarized as follows. The DL system must represent

- trees, which are simply made with tree nodes having subnodes or leafs,
- strategies, which are made with strategy nodes, some being navigation strategies from patterns to patterns and the other being action strategies,
- operations between strategies (`repeat`, `;` and `|`),
- and strategy applications to subtrees.

The mapping between rewriting notions and DL notions is formalized through the relation $\Phi_t \subset I \times T$ between the set of individuals and the set of trees, the relation $\Phi_p \subset C \times P$ between the set of concepts and the set of patterns, and the relation $\Phi_s \subset C \times S$ between

the set of concepts and the set S of strategies.

In the following, the expressions of the rewriting system are written in the typewriter font while those of the description logic system use the standard Latex mathematical font. For example, a pattern of the rewriting system will be noted p while its representation in ontology concepts will be noted p . The same is true for an expression t and the associated individual i , an expression node n and the associated concept n or for a strategy node s and the associated concept s .

5.4.1/ TREES REPRESENTED BY INDIVIDUALS

Each subtree t' of a tree t can be represented by an individual of the $\mathcal{A}Box$. The name of the individual is made with four fields $[type, name, position, instance]$ that we call an *address*. The *type* is either "identifier" or "global variable" depending whether the tree t is stored in an identifier or in a global variable. The *name* is the name of the identifier or of the memory. The *position* is ε if the subtree t' is the full tree t and is the position of the root of t' in t otherwise. The field *instance* is the instance number either of a global variable changed by the strategy store or of an identifier changed by application of a rewriting rule. Here are two examples.

- If the first instance of the identifier t is the tree $f(g(h(a)), b)$ then t and its subtree $g(h(a))$ are represented by the individuals $i := [identifier, t, \varepsilon, 1]$ and $i_1 := [identifier, t, 1, 1]$.
- An expression allocated to the global variable "t" by the command `store("t")` called for the second time in the main strategy is represented by the individuals $i := [global\ variable, t, \varepsilon, 2]$.

A leaf of a tree is represented as any other subtree by an address individual but also by another individual whose name is the data of the leaf which is always represented as a string. For example, for the identifier $t : node("name", "laplace")$ the leaf's data "laplace" is described by two individuals $[identifier, t, 1, 1]$ for its address and *laplace* for its data.

A subtree can be simply represented by a single individual, but it is generally represented, partially or completely, by the roles described in Section 5.4.2.

The correspondance of an individual i to a subtree t is written $(i, t) \in \Phi_t$. Note that each individual is associated with a subtree but the converse statement is generally not true since not all subtrees need to be expressed in the ontology.

5.4.2/ PATTERNS REPRESENTED BY CONCEPTS AND ROLES

A pattern $p \in P$ can be represented in Σ by a concept $p \in C$ or by a concept expression p . As seen below in Remark 10, some concept expressions correspond to patterns. These relations define the relation $(p, p) \in \Phi_p$.

The assumption below defines the most used patterns in this work.

Assumption 11: Node Pattern

We adopt the convention that each node $\text{node}(\text{id}, [\text{n1}, \dots])$, which is not of kind list (as bcList , variableList etc...), is associated to a pattern where all its direct subnodes are replaced by distinct rewriting variables. This pattern is then associated to a concept of the $\mathcal{T}\text{Box}$ with name the node name. The super concept covering all node concepts is *node*.

For instance, the pattern $\text{name}(a_)$ is associated to the atomic node $\text{node}(\text{"name"}, [\text{name0}])$ or $\text{name}(\text{name0})$ and therefore to the $\mathcal{T}\text{Box}$ atomic concept *name*. This is expressed as $(\text{name}, \text{name}) \in \Phi_p$ which is an abuse of notation since we identify the node name *name* with the associated pattern. This notation abuse will be constantly done in the following.

The partial order relation \ll of matching between patterns is expressed by the relation \sqsubseteq between associated concepts. For $(p_1, p_1) \in \Phi_p$ and $(p_2, p_2) \in \Phi_p$, $p_1 \ll p_2$ if and only if $p_2 \sqsubseteq p_1$. In case where $p_2 \in T(V)$, it may be considered as a tree t and as such it can be represented as an individual of the $\mathcal{A}\text{Box}$. The relation $p_1 \ll t$ is then interpreted as the belonging $t \in p_1$ of an individual to a concept.

Given two nodes p_1, p_2 , the role $\text{hasDirectSubTree}(p_1, p_2)$ has the concepts p_1 and p_2 as domain and image. The role $\text{hasDirectSubTree}(p_1, p_2)$ is to express direct parent-child relationship. For instance, for three nodes f, g, h and two constants a, b if $t_1 := f(g(h(a)), b)$, $t_2 := g(h(a))$, $t_3 := h(a)$, and $(i_1, t_1), (i_2, t_2)$ and $(i_3, t_3) \in \Phi_t$ then one can set $(i_1, i_2) \in \text{hasDirectSubTree}(f, g)$ and $(i_2, i_3) \in \text{hasDirectSubTree}(g, h)$. Moreover, by inference $i_1 \in \exists \text{hasDirectSubTree}(f, g).g$ and $i_2 \in \exists \text{hasDirectSubTree}(g, h).h$.

We observe that in the previous example, there is no ambiguity about the position of the nodes $g(x)$ and $h(y)$ of f in the concept intersection representation, which is a commutative operation, since this information is in the individual's position. We recall that the order of the subnodes in the list of a node is fixed.

Remark 9: Role Generalization to Any Patterns

The role $\text{hasDirectSubTree}(p_1, p_2)$ can be extended to any patterns $p_1, p_2 \in P$. However, this definition and its consequences are not explored in this work.

For the leaves, we use the concept *leaf* that contains all leaf's address individuals and the data property *hasData* between the concept *leaf* and the corresponding data. For

instance, the term $t : \text{name}(\text{"laplace"})$ is associated with an address individual $i \in \text{name}$ and the subterm $t_1 : \text{"laplace"}$ is associated with an address individual i_1 in the concept *leaf* together with a data "laplace" through *hasData*. The individuals i and i_1 are related with the role $\text{isDirectSubTree}(\text{name}, \text{leaf})$, i.e. $(i, i_1) \in \text{isDirectSubTree}(\text{name}, \text{leaf})$ and the individuals i_1 and "laplace" are related with the data property *hasData*, i.e. $(i_1, \text{"laplace"}) \in \text{hasData}$.

Each complex pattern i.e. a pattern made with more than one node is uniquely represented by a concept expression using existential quantifications, conjunctions and compositions of the role *hasDirectSubTree*. For instance the pattern $f(g(x), y)$ is uniquely represented by the concept expression $\exists \text{hasDirectSubTree}(f, g).g$. A pattern with several subnodes as $f(g(x), h(y))$ is represented by a conjunction of existential quantification as $\exists \text{hasDirectSubTree}(f, g).g \sqcap \exists \text{hasDirectSubTree}(f, h).h$. A pattern whose tree has several levels is defined by role composition as the pattern $f(g(h(x)), y)$ which is represented by the concept expression $\exists \text{hasDirectSubTree}(f, g) \circ \text{hasDirectSubTree}(g, h).h$ or $\exists \text{hasDirectSubTree}(f, g). \exists \text{hasDirectSubTree}(g, h).h$, where \circ denotes the role composition. As an illustration $\exists \text{hasDirectSubTree}(\text{equation}, \text{name}) \circ \text{hasDirectSubTree}(\text{name}, \text{leaf}) \circ \text{hasData}.\text{"laplace"}$ is the concept of addresses of equations with name "laplace".

The role $\text{hasDirectSubTree}(p_1, p_2)$ is generalized by $\text{hasDirectSubTree}(\text{list})$ where *list* is a list of node patterns. For a two-element list, $\text{hasDirectSubTree}([p_1, p_2])$ is equal to $\text{hasDirectSubTree}(p_1, p_2)$. For a one-element list, $(i, j) \in \text{hasDirectSubTree}([p_1])$ if and only if there exists a node pattern concept p_2 such that $(i, j) \in \text{hasDirectSubTree}([p_1, p_2])$. For an empty list, $(i, j) \in \text{hasDirectSubTree}([])$ if and only if there exists a node pattern p_1 such that $(i, j) \in \text{hasDirectSubTree}([p_1])$. For a list with $n > 2$ elements, the definition is based on the recursion and the role composition $\text{hasDirectSubTree}([p_1, \dots, p_n]) = \text{hasDirectSubTree}([p_1, \dots, p_{n-1}]) \circ \text{hasDirectSubTree}(p_{n-1}, p_n)$.

The next property follows from what has been said and expresses that patterns can be associated to concept formulas that are built from roles defined by nodes.

Remark 10: Patterns Defined by Concept Formulas

In what has been done above, nodes are used to parameterize $\text{hasDirectSubTree}(\text{list})$ and associated roles. Concept formulas defined by existential quantifications on these roles and by intersections are associated with equivalence classes of patterns defined modulo commutativity in node lists.

Remark 11: Same Rewriting Variable at Different Positions

In the above representation of patterns repetition of a same rewriting variable at different positions has not been taken into consideration. This is left for further investigation.

We conclude this section with roles derived from *hasDirectSubTree*. They can be constructed by role composition and in this case their pairs of individuals are obtained by inference. Otherwise their pairs of individuals are constructed by applying assemblies of strategies and ontology declarations not yet designed and therefore not detailed here.

Given two patterns p_1 and p_2 , the role *hasSubTree*(p_1, p_2) is defined as $(i_1, i_2) \in \text{hasSubTree}(p_1, p_2)$ if there exists a composition $r = \text{hasDirectSubTree}(p_1, q_1) \circ \text{hasDirectSubTree}(q_1, q_2) \circ \dots \circ \text{hasDirectSubTree}(q_n, p_2)$ such that $(i_1, i_2) \in r$. Its extension *hasSubTree*(*list*) to lists of node patterns is done similarly to this of *hasDirectSubTree*.

The inverse roles of *hasSubTree*(*list*) and *hasDirectSubTree*(*list*) are *isSubTreeOf*(*list'*) and *isDirectSubTreeOf*(*list'*) where *list'* is the list *list* in reverse order.

More roles combining these roles as well as more data properties combining these roles with the data property *hasData* are defined in Section 5.6.2.1.

5.4.3/ STRATEGIES REPRESENTED BY CONCEPTS AND ROLES

A strategy is a tree of nodes `nodeS(name, [nodeS(),...])` as for instance `s : nodeS("replace", node("name", "laplace"))` or equivalently `replace(name("laplace"))`. As a strategy includes expressions, e.g. `name("laplace")`, the tree of a strategy also holds expression nodes `node(...)`. Like expression trees, each tree of a substrategy s' has an address, generically noted iS , in a strategy s . The addressing system is the same as for expression tree nodes, without the type, since the memory (`globalVar`) is used for expressions only. Thus, the parameter *name* always refers to an identifier. The parameter *position* still refers to the position in the strategy tree. The parameter *instance* is initialized at 1 and is incremented in case the substrategy is repeated by the strategy `repeat`. The address of the node `replace` in s is $[s, \varepsilon, 1]$. In the other illustrative examples of this section, the instance number of the addresses is usually omitted for simplicity.

In an ontology Σ , a strategy tree is represented using

- atomic concepts of atomic strategy nodes `nodeS` as `insert`, `remove`, `replace`, `goto`, `compose`, `leftChoice` etc,
- the role *hasDirectSubTreeS*(n, k) defined below,
- individuals corresponding to the addresses iS of the nodes in the tree,
- and also concepts, roles and individuals of expression trees.

The role name *hasDirectSubTreeS*(n, k) is defined by a strategy node name n and a position k in the node list. For example, let iS the address of a node `nodeS("compose", [s1, s2])` and iS_2 the address of `s2`, then $(iS, iS_2) \in \text{hasDirectSubTreeS}(\text{compose}, 2)$ since the strat-

egy node s_2 is at the second position in the subnode list. Another example is the strategy $s : \text{compose}(\text{remove}, \text{remove})$ that is represented by the memberships to concepts of strategy nodes $iS_0 = [s, \varepsilon] \in \text{compose}$, $iS_1 = [s, 1] \in \text{remove}$, $iS_2 = [s, 2] \in \text{remove}$ and by the role memberships $(iS_0, iS_1) \in \text{hasDirectSubTreeS}(\text{compose}, 1)$ and $(iS_0, iS_2) \in \text{hasDirectSubTreeS}(\text{compose}, 2)$.

The branch of a strategy tree between a strategy node and an expression node is defined by the role $\text{hasDirectSubTree}(n, p)$ where n is the strategy node concept and p the expression node concept. For example, the strategy $s : \text{replace}(\text{name}(\text{"laplace"}))$ is expressed using the role membership $(iS_1, iS_2) \in \text{hasDirectSubTree}(\text{replace}, \text{name})$ with $iS_1 = [s, \varepsilon] \in \text{replace}$ and $iS_2 = [s, 1] \in \text{name}$. Therefore role $\text{hasDirectSubTree}(p_1, p_2)$ is extended to the concepts p_1 of strategy nodes that have parameters of type expression.

According to these principles, any substrategy s can be represented by a strategy node concept s , and conversely, any concept s of strategy nodes can be represented by a substrategy s . The related substrategies and concepts are represented by the membership $(s, s) \in \Phi_S$.

The representation of the success of a strategy node applied to an expression node is represented by the role hasSuccess , thus $(iS, i) \in \text{hasSuccess}$ represents the success of the substrategy node at the address iS on the expression node with address i . If a same (i.e. same address) substrategy is repeated on a same expression subtree (same position in the same tree), at each repetition a new individual associated to the subtree is created with the same address excepted its instance number which is incremented by one. It results that a substrategy individual has at most one image through the role hasSuccess .

5.4.4/ SUMMARY OF THE CONVERSION BETWEEN REWRITING AND DESCRIPTION LOGIC

Table 5.2 summarizes the conversion operations that are made from the rewriting system to the DL system.

5.5/ THE CONCURRENT REWRITING AND DESCRIPTION LOGIC SYSTEM

While the modes of conversion between the notions of rewriting and description logic have been established, in this section we describe the operations necessary for the functioning of an RDL system (τ_d, S, Σ) . In the context of multiscale model construction, τ_d is an expression including all the data regarding the system to analyze i.e. its PDE and the related assumptions, S is a strategy to be applied to τ_d that must return the expected multiscale models in $S[\tau_d]$, and Σ is the ontology gathering the knowledge, i.e. the theorems

Rewriting	Concept	Individual	Role
$p : \text{node}(n, [x, \dots])$	$n \sqsubseteq \text{node}$		
$t : \text{node}(n, [\dots])$		$i : [id, t, \varepsilon] \in n$	
$\text{store}(\text{"mem"})[\text{node}(n, [\dots])]$		$i' : [gV, \text{mem}, \varepsilon] \in n$	
$p \ll t$		$i \in n$	
$p_1 \ll p_2$	$p_2 \sqsubseteq p_1$		
$t : \text{node}(n, [\text{node}(n_1, \dots), \text{node}(n_2, \dots), \dots])$	n_1, n_2	$i_1 : [id, t, 1] \in n_1, i_2 : [id, t, 2] \in n_2$	$(i, i_1) \in \text{hasDST}(n, n_1), (i, i_2) \in \text{hasDST}(n, n_2)$
$t : \text{node}(n, d)$	leaf, data	$i : [id, t, \varepsilon] \in n, i_1 : [id, t, 1] \in \text{leaf}, i_2 : d \in \text{data}$	$(i, i_1) \in \text{hasDST}(n, \text{leaf}), (i, i_2) \in \text{hasData}$
$s : \text{nodeS}(nS, [\dots])$	$nS \sqsubseteq \text{strategy}$	$iS : [s, \varepsilon] \in nS$	
$s : \text{nodeS}(nS, [\text{nodeS}(nS_1, \dots), \text{nodeS}(nS_2, \dots)])$	nS_1, nS_2	$iS_1 : [s, 1] \in nS_1, iS_2 : [s, 2] \in nS_2$	$(iS, iS_1) \in \text{hasDSTS}(nS, nS_1), (iS, iS_2) \in \text{hasDSTS}(nS, nS_2)$
$s[t]$ successful			$(iS, i) \in \text{hasSuccess}$

Table 5.2: Summary of Rewriting System conversion into Description Logic in terms of individuals, concepts and roles. Rewriting expressions are in column 1 and the other columns summarize the related objects. Objects already defined in a previous line are not recalled. Identifiers of trees, patterns, strategies and individual start with t , p , s and i respectively. A data used in a tree or pattern leaf is noted as d . A node name starts with n . For shortness, in individual addresses the instance number is omitted. The role shortcuts $\text{hasDST}(\dots)$ and $\text{hasDSTS}(\dots)$ are used for $\text{hasDirectSubTree}(\dots)$ and $\text{hasDirectSubTreeS}(\dots)$, id for "identifier" and gV for "globalVar".

in the form of strategy individuals, invoked in the strategy S . In this work we assume that the data tree t_d does not contain any reference to the ontology Σ .

5.5.1/ THE COMPONENTS OF A RDL SYSTEM

Next, we detail the distinction between three parts of the RDL system: a part represented in both systems as described in Section 5.4, a part used only by the rewriting system and a part used only by the ontology.

- The common part is made with a forests of expressions and strategies in the rewriting system together with their representation in the ontology. They are useful to the knowledge description of the ontology in the third part. For an efficient use of the hasDirectSubTree roles and their variants or by-products, we will always consider that there is a partial order relation on the set of nodes taken into account in the common part. This aspect is not further discussed here.
- The part represented only in the rewriting system includes
 - a forest of expression trees that contain details of equations of t_d and of their transformation in $S[t_d]$ for whose an ontology representation can be avoided,
 - a forest of strategy trees involving details of S not useful in the ontology,

- and the master strategy S that governs the overall operations not handled by an ontology.
- The part represented only in the ontology is on the one hand relationships between tree nodes of expressions or strategies, that are not represented in τ_d , S and the other identifiers of the rewriting system. On the other hand it includes higher level individuals, concepts and roles suitable for the overall knowledge, i.e. the theorems, to be represented.

Now we focus on the composition of the ontology Σ in which we distinguish four parts Σ_0 , Σ_t , Σ_d and Σ_s that are characterized as follows.

- Σ_0 is a fragment made with the concepts, roles and data properties used for the representation of atomic nodes used in the data tree τ_d , the strategy S and their dependencies stored in identifiers.
- Σ_t is a fragment comprising individuals associated with nodes of the data tree τ_d and their membership relations to concept nodes, roles or data properties defined in Σ_0 .
- Σ_d is a fragment that enriches the ontology with data (assignment of individuals to concepts or roles) not included in the data tree τ_d but useful to the knowledge representation. This includes strategy representation as individuals.
- Σ_s is the fragment made with individuals and their membership associated with strategy application.

The operations done on the Σ_0 , Σ_t , Σ_d and Σ_s parts of the ontology are performed by strategies or functions. They concern construction, querying and strategy execution. The description and the use of the strategy O of declaration of components of the ontology are in Section 5.5.2, those of the function Q of interrogation of a concept are in Section 5.5.3 and those of the strategy S of execution of a strategy associated with an individual of strategy concept are in Section 5.5.4.

Remark 12:

Applying a strategy s to a tree t of data returns a tree of data t' which may require a modification of the ontology fragments Σ_t , Σ_d and Σ_s to be made at the same time as the modification of t . The related algorithms have not yet been investigated and therefore are not discussed further in the thesis.

5.5.2/ STRATEGY 0 OF ONTOLOGY CONSTRUCTION

Any part of an ontology is declared by the strategy 0 writing commands in a file (OWL2 commands for example) used to generate an ontology Σ on which a DL reasoner operates. For instance the concepts *name* and *equation* are declared with the strategy 0(["Declaration(Class(: name))", "Declaration(Class(: equation))"). The fragment Σ_0 is built with 0 only.

The construction of the ontology fragments Σ_t and Σ_d corresponding to trees of expressions or of strategies is done by navigation and by generation of the individuals associated with the encountered nodes as well as their membership to node concepts and to the roles *hasDirectSubTree*, *hasData* and *hasDirectSubTreeS*. For instance the declaration of the substrategy *replace(name("laplace"))* in an ontology is made, assuming that the concepts *replace* and *name* have been declared, by 0(["ClassAssertion(: replace : iS)", "ClassAssertion(: name : i)", "ClassAssertion(: isDirectSubTree (replace,name) : iS : i)"]) meaning that $iS \in \text{replace}$, $i \in \text{name}$ and $(iS, i) \in \text{isDirectSubTree (replace,name)}$, where *iS* and *i* are the addresses of the nodes *replace* and *name*.

As already said, only a part of the trees of expressions is represented in Σ . The selection is done by using a set \mathcal{N} of the node names taken into account in Σ . When a node name not belonging to \mathcal{N} is encountered it is skipped. Then, nodes that are not directly related are declared with the roles *hasSubTree(.,.)* instead of *hasDirectSubTree(.,.)*.

5.5.3/ COMMANDS Q AND QData OF ONTOLOGY INTERROGATION

Ontology queries are done either with the query command Q operating on concept expressions or with the query command QData operating on data properties.

The evaluation of a query Q returns an expression, thus it can be used in place of any expression in a strategy. Precisely, for a concept expression *c* written in a query language (e.g. DLQuery or SPARQL), $Q(c)$ returns the list $[\Phi_t(i_1), \dots]$ of the subtrees associated with the individuals i_1, \dots of the concept *c*. For instance $Q([\text{domain}])$ returns the list of all nodes *domain* declared in the ontology. The class expression based query $Q(\exists \text{hasDirectSubTree}(\text{equation}, \text{name}) \circ \text{hasDirectSubTree}(\text{name}, \text{leaf}) \circ \text{hasData.} \text{"laplace"})$ returns the list of all nodes *equation* with name "laplace" declared in the ontology.

The other kinds of interrogations available in reasoners are not taken into account in this text, for instance regarding subconcepts, superconcepts etc. Moreover, it would also be possible to define an intermediate query command that could return a list of addresses. Thus, it could be composed with the application of a strategy defined by an address as introduced in Section 5.5.4.

The QData command operates like the Q command except that it queries a concept

expression that corresponds to a node without subnodes but whose children are only data. It returns the list of all the data (strings) of the children of the nodes corresponding to individuals in the concept. For instance $QData(name)$ returns the list of all the names of all the nodes associated to individuals in the concept $name$. Similarly, $QData(\exists hasDirectSubTree(equation, name))$ returns the list of all the names of equations.

5.5.4/ APPLICATION OF A STRATEGY DEFINED IN AN ONTOLOGY

The application of the strategy represented by an individual iS to the term represented by an individual i is noted as $iS[i]$. It is semantically equivalent to the application $\Phi_s(iS)[\Phi_t(i)]$ of the strategy $\Phi_s(iS)$ to the term $\Phi_t(i)$. Its success or failure are declared by the membership declarations $(iS, i) \in hasSuccess$ or $(iS, i) \in hasFailure$.

5.6/ RDL REPRESENTATION OF A FAMILY OF MODELS OF MMAS

The model building algorithms presented in Chapter 4 are expressed in the RDL formalism, except for the internal edge model. Table 5.3 summarizes the operations performed in the four models with the numbers of the steps where they are performed.

Operation	Per	BL	BLI	BL Edge
Duplicate PDE	1,2	1,2	1,2	1,2
Rename unknowns	3	3	3	3
Remove sources in Dirichlet bc	-	4	4	4
Change bc at the origin	-	5	5,6	5,6
Rename boundaries (origin)	-	6	7	7
Remove some bc	4	-	-	-
Replace domains	5	7	8	8
Increment boundary scale	6	-	-	-
Insert bc at infinity	-	8	9	9
Insert periodicity bc	7	9	10,11	-
Rename PDE	8	10	12	10

Table 5.3: Roles in the ontology of domains and boundaries.

It is noteworthy that the ontology and the strategies described in this section are an illustration of the RDL system but do not presage any final form. A serious study of the organization of families of models has yet to be carried out.

5.6.1/ ROLES BETWEEN SUBMODELS, MODEL PROPERTIES AND STRATEGIES

In the approach of Chapter 4, each submodel corresponds to an algorithm made of a composition of strategies. In order to factorize the use of strategies, we introduce an intermediate layer, that of properties. Thus, to each submodel correspond properties

and to each property correspond strategies. Thus, several models can share the same property and several properties can involve the same strategy.

Table 5.4 gives the correspondences between the operations done in the algorithms, the strategy names and the properties.

Operation	Strategy	Property
Duplicate PDE	duplicatePDE	-
Rename unknowns	renameUnknown	-
Remove sources in Dirichlet bc	removeDirichletBcSource	correctionOtherSubModel
Replace bc rhs at the origin	replaceBcRhsOrigin	boundaryLayer
Rename boundaries (origin)	renameBoundaryOrigin	boundaryLayer
Remove some bc	removeBc	periodicity
Replace domains	replaceDomain	-
Increment boundary scale	incrementBoundaryScale	-
Insert bc at infinity	insertBcInfinity	boundaryLayer
Insert periodicity bc	insertBcPeriodicity	periodicity
Rename PDE	renamePDE	-

Table 5.4: Correspondences between the operations done in the algorithms, the strategy names and the properties. The strategies involved in all submodels are marked by a "-".

Model properties are defined as individuals of the concept *modelProperties*, for example *periodicity* \in *modelProperties*. Each particular *subModel* has a name which is an individual of the concept *subModel*. For example the model *BL* \in *subModel*. To each *subModel* name is associated properties by the role *modelHasProperty* whose inverse is *isPropertyOfModel*. For example $(BL, periodicity) \in modelHasProperty$. To each model property are associated, by the role *propertyHasStrategy* of inverse *isStrategyOfProperty*, strategies to be applied. For example, to the property *periodicity* is associated the strategy *insertBcPeriodicity* which is an individual of the concept *strategy* (and also of the concept of its root node). Thus, $(periodicity, insertBcPeriodicity) \in propertyHasStrategy$. Thus, the properties of the *BL* model are the individuals of the concept $\exists isPropertyOfModel.BL$, the strategies to be applied for the periodicity property are the individuals of the concept $\exists isStrategyOfProperty.periodicity$ and the strategies to be applied for the *BL* model are in the concept defined by $\exists isStrategyOfProperty.\exists isPropertyOfModel.BL$ or by the composition $\exists isStrategyOfProperty \circ isPropertyOfModel.BL$.

The selected strategies are to be executed by composition, in an order yet to be specified. To do this, we use the *isBefore* role or its inverse *isAfter* which establishes an order between strategies. For example, the strategy *insertBcPeriodicity* is necessarily executed after the replacement of the domain by the strategy *replaceDomain*, which is expressed by $(replaceDomain, insertBcPeriodicity) \in isBefore$. These relations are transitive but not reflexive, they define strict partial orders in the sets of strategies.

This part of the system concerning the properties is not further detailed in this work.

5.6.2/ STRATEGIES IN RDL

The strategies summarized in Table 5.4 of properties are written in Section 5.6.2.2 using a RDL meta-language mixing rewriting strategies and queries of the ontology. We start in Section 5.6.2.1 with the description of useful concepts, data and roles of the ontology. Some of these definitions are parameterized by strings (data) that can be used both in the rewriting system and in the DL system. This allows to compose these functions if their result is also a string. All the roles defined below are given without detailing the way of building explicitly their definition and their individuals which can be as combinations of strategies with ontology declarations or inferences. The reader should be aware that a number of programming details are omitted for simplicity.

5.6.2.1/ CONCEPT, DATA AND ROLE FORMULAS

For the sake of brevity, it is convenient to start by introducing few other general roles *isOfHas*, *isOfHasData* and *isOfHasDirectData* derived from *hasSubTree*, *hasDirectSubTree* and their inverse.

isOfHas($[p_1, \dots, p_k, \dots, p_n]$) is the role defined for concepts p_1, \dots, p_n of the data tree as the (assumed) unique composition $isSubTreeOf([p_1, \dots, p_k]) \circ hasSubTree([p_k, \dots, p_n])$ where the concept p_k is the only one that allows this composition without leaving a subModel. If such concept p_k does not exist or is not unique, then an error is triggered.

isOfHasData($[p_1, \dots, p_k, \dots, p_n]$) is the role defined for concepts p_1, \dots, p_n of the data tree as the composition $isOfHas([p_1, \dots, p_k, \dots, p_n]) \circ hasData$.

isOfHasDirectData($[p_1, \dots, p_k, \dots, p_n]$) is as *isOfHasData* but with direct relations between the corresponding nodes of the tree.

The following concepts, data and roles are defined thanks to the above roles and are specific to the data structure of Chapter 4. Most of them are parametrized, as such they are defined in the RDL system but not directly in the ontology side since the latter does not directly support parametrization of concepts or roles.

◇ *subModelWithName*(*idName*) is the concept of *subModel* with name *idName*.

$$subModelWithName(idName) : \exists hasSubTreeData([subModel, ident, name]).idName$$

◇ *conceptInSubModel*(*idName*, *nodeName*) is the concept associated with a node *nodeName* in a subModel which name is *idName*.

$$conceptInSubModel(idName, nodeName) : \exists isOfHas([nodeName, subModel, ident, name]).idName$$

◇ *equationName*(*idName*) is the concept of equation names of a node *equationNameList* in a subModel with name *idName*.

$equationName(idName) : conceptInSubModel(idName, equationNameList)$

◇ $bcName(idName)$ is the concept of bc names of a node $bcNameList$ in a $subModel$ with name $idName$.

$bcName(idName) : conceptInSubModel(idName, bcNameList)$

◇ $equationPhysic$ is the concept of the equations of the physical PDE.

$equationPhysic : \exists isOfHasData([equation, subModel, ident, scale]).0$

◇ $bcPhysic$ is the concept of the bcs of the physical PDE.

$bcPhysic : \exists isOfHasData([bc, subModel, ident, scale]).0$

◇ $equationWithNameIn(eqName)$ is the concept of equations whose names are in a concept $eqName$.

$equationWithNameIn(eqName) : \exists hasDirectSubTree([equation, name]).eqName$

◇ $bcWithNameIn(bcName)$ is the concept of bcs whose names are in a concept $bcName$

$bcWithNameIn(bcName) : \exists hasDirectSubTree([name]).bcName$

◇ $variableInEquation(idName)$ is the concept of variables in equations of the microscale node $model$ of a $subModel$ which name is $idName$.

$variableInEquation(idName) : \exists isDirectOfHasData([variable, equation, subModel, ident])$
 $.idName$

◇ $equationPhysicWithName(eqName)$ is the concept of equations of the physical PDE whose names are in a concept $eqName$.

$equationPhysicWithName(eqName) : equationPhysic \cap equationWithNameIn(eqName)$

◇ $bcPhysicWithName(bcName)$ is the concept of bcs of the physical PDE whose names are in a concept $bcName$

$bcPhysicWithName(bcName) : bcPhysic \cap bcWithNameIn(bcName)$

◇ $equationPhysicWithNameInSubModelList(idName)$ is the concept of the equations of the physical problem whose names are in the node list $equationNameList$ of a $subModel$ with name $idName$.

$equationPhysicWithNameInSubModelList(idName) : equationPhysic \cap$

$\exists hasDirectSubTree([equation, name]).conceptInSubModel(idName, equationNameList)$

◇ $bcPhysicWithNameInSubModelList(idName)$ is the concept of the bcs of the physical problem whose names are in the node list $bcNameList$ of a $subModel$ with name $idName$.

$bcPhysicWithNameInSubModelList(idName) : bcPhysic \cap$

$\exists hasDirectSubTree([bc, name]).conceptInSubModel(idName, bcNameList)$

◇ $bcOfRegion(boundaryName)$ is the concept bc posed on a boundary with name $boundaryName$.

$bcOfRegion(boundaryName) : \exists hasDirectSubTreeData(bc, variable, region, name) .boundaryName$

◇ $microDomain(idName)$ is the concept of region micro of a concept $subModel$ with name $idName$.

$microDomain(idName) : \exists isOfHasDirectData([region, subScale, subModel, ident, name]) .idName$

◇ $domainName(idName)$ is the concept of domain names (micro and physical) in the node $domainNameList$ of a node $subModel$ with name $idName$.

$domainName(idName) : conceptInSubModel(idName, domainNameList)$

◇ $phiMicro(idName)$ is the concept of unknown functions of the concept of microscale model $model$ of a concept $subModel$ which name is $idName$

$phiMicro(idName) : \exists hasDirectSubTreeData(function, name).conceptInSubModel(idName, microUnknown)$

◇ $periodicBoundaryName(idName)$ is the concept of periodic boundaries of the concept of microscale model $model$ of a concept $subModel$ with name $idName$.

$periodicBoundaryName(idName) : conceptInSubModel(idName, perBd)$

◇ $infBdName(idName)$ is the concept of boundaries at infinity of boundary layer models of the microscale model of a $subModel$ with name $idName$.

$infBdName(idName) : conceptInSubModel(idName, latInfBd)$

◇ $givenFunctionName(idName)$ is the concept of names of the concept $givenFunction$ in the $subModel$ with name $idName$.

$givenFunctionName(idName) : \exists isDirectSubTreeOfData(name, givenFunction, subModel, ident, name).idName$

◇ $givenFunctionIdent(idName)$ is the concept of identifiers of the concept $givenFunction$ in the $subModel$ with name $idName$.

$givenFunctionIdent(idName) : \exists isDirectSubTreeOfData(ident, givenFunction, subModel, ident, name).idName$

◇ $isCorrectorSourceOfBoundary$ is a role mapping the concept $givenFunction$ of source in the equation of correctors to the concept (here $latOriBdPY$ or $interface$) of the boundaries where a correction is applied. These concepts are associated to nodes in the dictionary of the dilation node. The pairs of individuals of this role are built while scanning the data tree, each node $givenFunction$, that includes a name and an identifier of a

corrector, being associated to the related node of boundary name. Its inverse role is *isBoundaryOfCorrectorSource*. For instance

- for model BL:

$$(givenFunction1, latOriBdP) \in isCorrectorSourceOfBoundary$$

- for model BL Edge:

$$(givenFunction2, latOriBdP1) \quad \text{and} \quad (givenFunction1, latOriBdP2) \quad \in \\ isCorrectorSourceOfBoundary,$$

- for model BLI:

$$(givenFunction1, interface) \quad \text{and} \quad (givenFunction2, interface) \quad \in \\ isCorrectorSourceOfBoundary.$$

◇ *isSubModelDomainNameOfCorrectorSource* is a role mapping the concept made with the *name* of domains in a corrector model to the concept *givenFunction* of sources in the equation of correctors. The role is between the sources and the names of domains where they are mapped by the change of variable. The inverse role is *isCorrectorSourceOfSubModelDomainName*.

- for model BL: let denote the domain name Ω of the region node of the subModel thus

$$(\Omega, givenFunction1) \in isSubModelDomainNameOfCorrectorSource$$

- for model BL Edge: idem but

$$(\Omega, givenFunction1) \text{ and } (\Omega, givenFunction2) \in \\ isSubModelDomainNameOfCorrectorSource,$$

- for model BLI: here, the region names are those of the subregion Ω_1, Ω_2 of Ω thus

$$(\Omega_1, givenFunction1) \text{ and } (\Omega_2, givenFunction2) \in \\ isSubModelDomainNameOfCorrectorSource.$$

◇ *isSubModelDomainOfBoundary* is the composition of the two above roles.

$$isSubModelDomainOfBoundary : isSubModelDomainNameOfCorrectorSource \\ \circ isCorrectorSourceOfBoundary$$

◇ *isCorrectorSourceNameOfBoundary* is a role mapping the concept *givenFunction* of sources in corrector equations to the concept *name* of the names of boundaries where a correction is applied.

$$isCorrectorSourceNameOfBoundary : \exists isDirectSubTreeOf(name, givenFunction) \\ \circ isCorrectorSourceOfBoundary$$

◇ *isCorrectorSourceIdentOfBoundary* is a role mapping the concept *givenFunction* of

sources in corrector equations to the concept *ident* of the identifiers of boundaries where a correction is applied.

$$isCorrectorSourceIdentOfBoundary : \exists isDirectSubTreeOf(ident, givenFunction)$$

$$\circ isCorrectorSourceOfBoundary$$

◇ *correctorSourceNameOfBoundary(iBdName)* is a concept of source names (node name) associated with a correction boundary name given as a data (string) *iBdName* in a node *latOriBdP* or *interface*.

$$correctorSourceNameOfBoundary(iBdName) : \exists isCorrectorSourceNameOfBoundary$$

$$. \{iBdName\}$$

◇ *correctorSourceIdentOfBoundary(iBdName)* is a concept of source identifiers (node *ident*) associated with a correction boundary name given as an individual *iBdName* in a node *latOriBdP* or *interface*.

$$correctorSourceIdentOfBoundary(iBdName) : \exists isCorrectorSourceIdentOfBoundary$$

$$. \{iBdName\}$$

◇ *correctorSourceOfBoundary(iBdName)* is the concept of source (node function) in a corrector equation of a node *subModel* with name *idName* associated with an individual *iBdName* of a boundary name (node name). It is assumed that a single function is associated to a name and an identifier thus this class may have several individuals but all related to a same subtree.

$$correctorSourceOfBoundary(iBdName) :$$

$$\exists hasDirectSubTreeData(function, name).correctorSourceNameOfBoundary(iBdName)$$

$$\cap \exists hasDirectSubTreeData(function, ident).correctorSourceIdentOfBoundary(iBdName)$$

◇ *microRegionName(idName)* is the concept of name (node name) of the micro region associated to the concept *subModel* with name *idName*. This is used for the models BL and BL Edge.

$$microRegionName(idName) : \exists isOfHasDirectSubTree([name, region, subScale$$

$$, subModel, ident, name]). \{idName\}$$

◇ *regionNameOfCorrector(iBdName)* is the name (node name) of the micro region of a corrector *corrector(iBdName)*.

$$regionNameOfCorrector(idName, iBdName) : \exists isDirectSubTreeOf([name, region, variable])$$

$$.corrector(iBdName)$$

5.6.2.2/ STRATEGIES OF THE PROPERTY TABLE

The strategies in Table 5.4 are expressed in as meta programs of the RDL system which mixes rewriting strategies and queries of the ontology with one of the two commands `Q` and `QData` defined in Section 5.5.3. We recall that the `Q` command is a query by a concept formula that returns a list of expressions associated to the individuals belonging to the concept while the `QData` command still queries only concept expression corresponding to terminal nodes and returns a list of their data (strings). As in the correspondance between trees and individual, `iBdName` refers to the subtree related to the individual `iBdName` i.e. $(iBdName, iBdName) \in \Psi_t$. In addition, `idName` is a string in the rewriting system corresponding to the data `idName` in the ontology.

Before to describe the strategies, let us introduce few shortcuts for the navigation strategy `goto` that was defined in Section 5.3.2. First, its syntax is simplified, we use

$$\text{goto}(\pi, s)$$

instead of `goto(p, π' , s)` where π is the list gathering `p` and π' . Moreover, here π is not only a list of patterns but can involve a disjunction,

$$\text{goto}(\pi_1 \mid \pi_2, a, s) : \text{goto}(\pi_1, a, s) \mid \text{goto}(\pi_2, a, s)$$

for two lists π_1, π_2 of patterns. This construction is used recursively. We often use the most general pattern of a node

$$\text{nodeNameVar} : \text{node}(\text{"nodeName"}, [n1_, \dots])$$

where `nodeName` is the name of any atomic node and `n1_, ...` are for any different rewriting variables. For instance the node

$$\text{nameVar} : \text{node}(\text{"name"}, \text{name}_)$$

$$\text{functionVar} : \text{node}(\text{"function"}, [\text{ident}_, \text{name}_, \text{indexList}_, \text{varList}, \text{type}_]).$$

To specify some of the children of a node

$$\text{nodeNameVar}(\text{"nodeName}_{k_1} : t_1, \dots, \text{"nodeName}_{k_\ell} : t_\ell)$$

is as `nodeNameVar` except that its subnodes `nodeName1, ..., nodeName ℓ` are replaced by the expressions `t1, ..., t ℓ` . For instance the pattern for functions with name `f` is `functionVar("name" : "f")`.

An example of path which is often used is this starting from a `subModel` node and ending to its `model` node:

$$\text{pathSubModelToModel}(\text{idName}) : [\text{subModelVar}(\text{"ident"} : \text{identVar}(\text{"name"} : \text{idName})), \text{subScaleVar}, \text{modelVar}]$$

◇ `duplicatePDE(idName)` duplicates the equations and bcs of the physical problem whose names are in the lists `equationNameList` and `bcNameList` of a `subModel` of name `idName`.

The copy is in the microscale model i.e. in the subScale/model node of the subModel with name idName.

```
duplicatePDE(idName) : goto([pathSubModelToModel(idName), equationListVar]
    , insert(Q(equationPhysicWithNameInSubModelList(idName))))
; goto([pathSubModelToModel(idName), bcListVar]
    , insert(Q(bcPhysicWithNameInSubModelList(idName))))
```

◇ renameUnknown(idName) renames the unknown function whose name is in the physUnknown node by the name of the microUnknown node (a single name is assumed) of the subModel which name is idName.

```
pathFunctionName(name0) : [functionVar, name(name0)]
renameUnknown(idName) : goto([pathSubModelToModel(idName), equationListVar
    , pathFunctionName(Q(conceptInSubModel(idName, physUnknown)))]
    , replace(Q(conceptInSubModel(idName, microUnknown))))
```

◇ removeBc(idName) removes the bc operating on the boundary which name is in the latBdP node of the subModel which name is idName.

```
removeBc(idName) : goto([pathSubModelToModel(idName)
    , bcVar, variableVar, regionVar("name", Q(conceptInSubModel(idName, latBdP)))]
    , delete)
```

◇ replaceDomain(idName) replaces all occurrences of the domain in the node model of the subModel with name idName by the domain in the node region of subScale.

```
replaceDomain(idName) : goto([pathSubModelToModel(idName)
    , regionVar("name" : Q(domainName(idName)))]
    , replace(Q(microDomain(idName))))
```

◇ incrementBoundaryScale(idName) Updates the boundary identifiers with scale = 1 in model of the subModel which name is idName.

```
incrementBoundaryScale(idName) : goto([pathSubModelToModel(idName)
    , scaleVar], replace(scale(1)))
```

◇ insertPeriodicityBc(idName) inserts periodicity boundary conditions in the PDE of the model node of the subModel which name is idName.

```
insertPeriodicityBc(idName) : goto([pathSubModelToModel(idName), bcListVar]
    , insert(periodicityBc(x(microDomain(idName))))
```



```

    ,xg(QData(periodicBoundaryName( idName))),Q(phiMicro(idName)));
insert(fluxPeriodicityBc(x(microDomain(idName))
    ,xg(QData(periodicBoundaryName)), Q(phiMicro(idName)), 1,n))

```

◇ `renamePDE(idName)` renames the PDE of the model node of the subModel with name `idName`.

```

renamePDE(idName) : goto([pathSubModelToModel(idName),pdeVar]
    ,replace(pdeVar("name" : "micro.pde")))

```

◇ `removeDirichletBcSource(idName)` replaces the source in the right-hand side of Dirichlet bc by 0 in the model node of the subModel which name is `idName`.

```

removeDirichletBcSource(idName) : goto([pathSubModelToModel(idName)
    ,bcListVar,bcVar,equalityVar("name" : "dirichlet"),rightMathExpressionVar],[
    ,replace(rightMathExpression(0))]

```

◇ `normalDerivativeTrace(f, bd)` is the trace of normal derivative of a function `f` on a boundary `bd`.

◇ `pathSubModelToBc(idName, iBdName)` is the path from a subModel node to its boundary condition posed on a boundary `iBdName`.

```

pathSubModelToBc(idName,iBdName) : [pathSubModelToModel(idName),bcListVar
    ,bcVar("variableList" : [variableVar("region" : regionVar("name" : iBdName))])]

```

◇ `correctorSourceOfBoundary(iBdName)` is the list of sources (node function) of the corrector equation associated to the boundary with name (node name) `iBdName`. We recall that the class `correctorSourceOfBoundary(iBdName)` is made with one or two (in case of interfaces) subtrees.

```

correctorSourceOfBoundary(iBdName) : Q(correctorSourceOfBoundary(iBdName))

```

◇ `correctorSourceOfBoundaryMapped(iBdName, iFun)` is the list of sources `correctorSourceOfBoundary(iBdName)` of the corrector equation but which variables are mapped to the variable of the corrector equation with corrector source (node givenFunction) `iFun`.

```

correctorSourceOfBoundaryMapped(iBdName, iFun) :
    mapping{Q(isSubModelDomainNameOfCorrectorSource.iFun),
        Q(isDirectSubTree(ident, givenFunction).iFun)}
        (correctorSourceOfBoundary(iBdName))

```

◇ `correctionSource(iBdName, iFun)` is the normal derivative of a source (node

givenFunction) iFun to be added to a boundary condition at the boundary with name iBdName.

```
correctionSource(iBdName, iFun) : rightMathExpr0(
  normalDerivativeTrace(correctorSourceOfBoundaryMapped(iBdName, iFun)
    , Q( $\exists$ hasDirectSubTree(boundary, name).{iBdName}))
```

◇ `replaceBcRhsOrigin(idName)` replaces the right hand side of the boundary condition at the origin boundary by a corrector. This is done in the `model` node of the `subModel` which name is `idName`.

```
replaceBcRhsOrigin(idName) :
  for iBdName  $\in$  Q((latOriBdP  $\cup$  latOriBdP1  $\cup$  latOriBdP2  $\cup$  interface)
     $\cap$   $\exists$ isOfHasSubTree(subModel, name).idName)
  for iFun  $\in$  Q(correctorSourceOfBoundary(iBdName))
    goto([pathSubModelToBc(idName, iBdName)
      , goto([rightMathExprVar], rule(x_
        , x_ + correctionSource(iBdName, iFun)))]
    endFor iFun
  endFor iBdName
```

◇ `insertBcInfinity(idName)` inserts a Neumann bc at the boundary at infinity

```
insertBcInfinity(idName) : goto([pathSubModelToModel(idName), bcListVar]
  , insert(neumannEq(xg(QData(infTyBdName(idName)))
  , x(QData(domainName(idName))), Q(phi(idName)), n, 1, 0)))
```

◇ `renameBoundaryOrigin(idName)` rename the boundary at the origin in the `model` node of the `subModel` which name is `idName`.

```
renameBoundaryOrigin(idName) : goto([pathSubModelToModel(idName)
  , boundaryVar("name" : QData(conceptInSubModel(idName, latOriBdP))]
  , replace(QData(conceptInSubModel(idName, latOriBdP))))
```

CONCLUSIONS AND PERSPECTIVES

The goal of the MEMSALab software is to automatically build asymptotic models of MEMS Arrays using a reuse concept. Its operation is based on the combination of two techniques: asymptotic analysis from mathematics and rewriting from theoretical computer science. In a way, MEMSALab can be seen as a bridge to transfer well-developed asymptotic theories to more practical problems from research or industry in a way that users do not need to be familiar with the asymptotic analysis.

It took a number of years of contributions including those of this thesis before a stable operating principle for MEMSALab was developed. This thesis has significantly contributed to its necessarily multidisciplinary development. The maturing of the ideas required to embrace three scientific fields: that of the asymptotic modelling of the electromechanical problem posed in the MIRA array, that of the numerical simulation of the asymptotic models and that of the software development in the context of rewriting techniques.

From a numerical analysis point of view, asymptotic modes have advantages in reducing the computational cost of direct simulations of physical phenomena in MMAs comprising up to thousands of cells thanks to using the approximation models derived by the homogenization methods. The asymptotic models for the coupled electromechanical problem in the MIRA array derived by the two-scale transformation method in this thesis are considered exciting examples.

The array is regarded as consisting of two actuation regions operating with two different imposed voltages. Such a situation makes the model close to the actual problem and the relevant asymptotic models will cover all the possible cases. There are five kinds of asymptotic models comprising the periodic models, the lateral boundary and interface models, the external and internal edge models. Unlike the classical results of homogenization methods, due to the discretization in the mirror alignment, the homogenized equations are disappeared, and the periodic cell problems now play a role as the first-order approximation for the exact solutions. The external and internal edge models are new contributions in asymptotic models of MIRA as well as in homogenization methods.

The derivation for the periodic cases is as the usual process. However, seeking models for the boundary correctors is more interesting. As we have seen that the proofs of these models are performed following a reference algorithm in MEMSALab, then application of this reference algorithm for derivations of the asymptotic models for the family of MMAs will be possible. However, handwritten proofs are more sophisticated because they require determining the suitable transform operators and an enormous amount of notations that are impossible to do manually. Therefore, these are more appropriate for management by the symbolic computation.

Besides, thanks to the numerical implementation results of these asymptotic models for the electrostatic part of the MIRA in COMSOL, the electrostatic potential and its relevant physical issues at any point in the array can be computed approximately with the reduction of computation cost. Further, the optimization for the pull-in voltage economizes on actuation costs.

The new data structure is uniform and flexible for the creation of expressions. This data structure can define all the mathematical expressions that describe the models. In addition, the recursive structure permits the characteristics of multiscale problems to be thoroughly described. These broaden the ability to represent the kinds of asymptotic models for MMAs in MEMSALab. Further, the strategies are small in number but have a sizeable operational capability to deal with complex tasks. Especially, the strategies are entirely independent of the data structure, which leads to more convenience and more flexibility in their creations, management and developments.

Furthermore, the connection between the rewriting technique and the description logic is a significant achievement to enrich MEMSALab in the knowledge presentation aspect. With the inherent advantages offered by the rewriting technique in the tree-like data structure, which can navigate fast and manageable in a tree and can apply rules to verify hypotheses or perform transformations, MEMSALab is now equipped with the potential of description logic for ontology representation. In fact, the ontology plays a role as a processing centre to give conclusions from the enormous hypotheses provided by the rewriting technique, which is then in charge of carrying out these conclusions. This is the starting step towards the construction of mathematical knowledge representations of the asymptotic method aspect in MEMSALab.

As mentioned above, MEMSALab is currently close to the stable phase of the fundamental operations, and there are obviously various problems to be developed and studied. Then, we propose future work in the following.

From the mathematical theory point of view, the priori estimates of the physical solutions [1](#), [6](#) and of their correctors e.g [2](#), [7](#) should be proven as the classical problems in homogenization methods. In addition, the exponential decay to 0 at infinity of correctors and their gradients should be proven by the methods in [[6](#), [89](#)]. Furthermore, the convergence

of electrostatic forces should be studied based on the method presented in [50]. Then, numerical implementations of the asymptotic models of the linear elasticity part coupling with the electrostatic part through the electrostatic force should be performed. This is significant for making the asymptotic models for MIRA close to practical applications and for preparation of the linking of MEMSALab with finite element software. The enrichment of new asymptotic models for a family of MMAs should be studied to expand the library of asymptotic models in MEMSALab.

From the software development point of view, the reference proof should be implemented with the new data structure. An investigation of the reduction of extension size in the combination process should be studied to improve the extension-combination method. Applying the new reference proof to the coupled electromechanical problem in the MIRA array to derive the asymptotic models will be the first impressive examples. In addition, the implementation of the study of linking rewriting technique and description logic in OWL2 Language should be done. Further exploration of two fields should be carried out, e.g. the knowledge representation of the asymptotic methods for the MIRA array in ontology, a connection between the description logic and the extension in MEMSALab. It promises a meaningful improvement of the proof method. Plus, the geometrical relation package should be investigated and built as a preparation step for the connection step of MEMSALab with FEM software.

BIBLIOGRAPHY

- [1] Elan. <https://elan.loria.fr/>.
- [2] Maude. http://maude.cs.illinois.edu/w/index.php/The_Maude_System.
- [3] Tom. <https://github.com/rewriting/tom>.
- [4] ALLAIRE, G. Homogenization and two-scale convergence. *SIAM Journal on Mathematical Analysis* 23, 6 (1992), 1482–1518.
- [5] ALLAIRE, G. *Shape optimization by the homogenization method*, vol. 146. Springer Science & Business Media, 2012.
- [6] ALLAIRE, G., AND AMAR, M. Boundary layer tails in periodic homogenization. *ESAIM: Control, Optimisation and Calculus of Variations* 4 (1999), 209–243.
- [7] ALLAIRE, G., DAMLAMIAN, A., ET AL. Two-scale convergence on periodic surfaces and applications. In *In Mathematical Modelling of Flow through Porous Media, Bourgeat AP, Carasso C, Luckhaus S, Mikelić A (eds)*. World Scientific (1995), Citeseer.
- [8] AMIRAT, Y., CHECHKIN, G. A., AND GADYL'SHIN, R. R. Asymptotics of simple eigenvalues and eigenfunctions for the laplace operator in a domain with an oscillating boundary. *Computational Mathematics and Mathematical Physics* 46, 1 (2006), 97–110.
- [9] ANGLES, R., ARENAS, M., BARCELÓ, P., HOGAN, A., REUTTER, J., AND VRGOČ, D. Foundations of modern query languages for graph databases. *ACM Computing Surveys (CSUR)* 50, 5 (2017), 1–40.
- [10] ARBOGAST, T., DOUGLAS, JR, J., AND HORNUNG, U. Derivation of the double porosity model of single phase flow via homogenization theory. *SIAM Journal on Mathematical Analysis* 21, 4 (1990), 823–836.
- [11] A/S, I. P. Ibsen photonics a/s. <https://ibsen.com/technology/spectrometer-tutorial/dmd-spectrometers>, May 2021.
- [12] AUDET, C., DENNI, J., MOORE, D., BOOKER, A., AND FRANK, P. A surrogate-model-based method for constrained optimization. In *8th symposium on multidisciplinary analysis and optimization* (2000), p. 4891.

- [13] BAADER, F., CALVANESE, D., MCGUINNESS, D., PATEL-SCHNEIDER, P., NARDI, D., ET AL. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.
- [14] BAADER, F., HORROCKS, I., LUTZ, C., AND SATTLER, U. *Introduction to description logic*. Cambridge University Press, 2017.
- [15] BAADER, F., HORROCKS, I., AND SATTLER, U. Description logics. In *Handbook on ontologies*. Springer, 2004, pp. 3–28.
- [16] BAADER, F., AND NUTT, W. Basic description logics. In *The description logic handbook: theory, implementation, and applications*. 2003, pp. 43–95.
- [17] BAKHVALOV, N. S., AND PANASENKO, G. *Homogenisation: averaging processes in periodic media: mathematical problems in the mechanics of composite materials*, vol. 36. Springer Science & Business Media, 2012.
- [18] BELKHIR, W., GIORGETTI, A., AND LENCZNER, M. A symbolic transformation language and its application to a multiscale method. *Journal of Symbolic Computation* 65 (2014), 49–78.
- [19] BELKHIR, W., RATIER, N., NGUYEN, D. D., AND LENCZNER, M. Unification and combination of a class of traversal strategies made with pattern matching and fixed-points. *submitted*.
- [20] BELKHIR, W., RATIER, N., NGUYEN, D. D., AND LENCZNER, M. Unification and combination of iterative insertion strategies with one-step traversals. *arXiv preprint arXiv:1904.07668* (2019).
- [21] BELKHIR, W., RATIER, N., NGUYEN, D. D., NGUYEN, N. B. T., LENCZNER, M., AND ZAMKOTSIAN, F. A tool for aided multi-scale model derivation and its application to the simulation of a micro mirror array. In *2017 18th International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE)* (2017), IEEE, pp. 1–8.
- [22] BELKHIR, W., RATIER, N., NGUYEN, D. D., YANG, B., LENCZNER, M., ZAMKOTSIAN, F., AND CIRSTEAN, H. Towards an automatic tool for multi-scale model derivation illustrated with a micro-mirror array. In *2015 17th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)* (2015), IEEE, pp. 47–54.
- [23] BENSOUSSAN, A., LIONS, J.-L., AND PAPANICOLAOU, G. *Asymptotic analysis for periodic structures*, vol. 374. American Mathematical Soc., 2011.

- [24] BENSOUSSAN, A., LIONS, J. L., AND PAPANICOLAOU, G. C. Boundary layers and homogenization of transport processes. *Publications of the Research Institute for Mathematical Sciences* 15, 1 (1979), 53–157.
- [25] BENSOUSSAN, ALAIN; LIONS, J.-L. P. G. *Asymptotic analysis for periodic structures*. Elsevier, 1978.
- [26] BERLYAND, L., AND RYBALKO, V. *Getting Acquainted with Homogenization and Multiscale*. Springer, 2018.
- [27] BRAIDES, A., AND DEFRANCESCHI, A. *Homogenization of multiple integrals*. No. 12. Oxford University Press, 1998.
- [28] BRAUN, S., OBERHAMMER, J., AND STEMME, G. Row/column addressing scheme for large electrostatic actuator mems switch arrays and optimization of the operational reliability by statistical analysis. *Journal of microelectromechanical systems* 17, 5 (2008), 1104–1113.
- [29] CANON, E., AND LENCZNER, M. Models of elastic plates with piezoelectric inclusions part i: Models without homogenization. *Mathematical and Computer Modelling* 26, 5 (1997), 79–106.
- [30] CANONICA, M., ZAMKOTSIAN, F., LANZONI, P., AND NOELL, W. Large micromirror array for multi-object spectroscopy in space. *International Conference on Space Optics 2012, Ajaccio, Corsica, France 10564* (2012).
- [31] CANONICA, M. D. *Large Micromirror Array Based on a Scalable Technology for Astronomical Instrumentation*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2012.
- [32] CASADO-DÍAZ, J. Two-scale convergence for nonlinear dirichlet problems in perforated domains. *Proceedings of the Royal Society of Edinburgh Section A: Mathematics* 130, 2 (2000), 249–276.
- [33] CASADO-DÍAZ, J., LUNA-LAYNEZ, M., AND MARTÍN, J. D. An adaptation of the multi-scale methods for the analysis of very thin reticulated structures. *Comptes Rendus de l'Académie des Sciences-Series I-Mathematics* 332, 3 (2001), 223–228.
- [34] CHO, I.-J., AND YOON, E. A low-voltage three-axis electromagnetically actuated micromirror for fine alignment among optical devices. *Journal of Micromechanics and Microengineering* 19, 8 (2009), 085007.
- [35] CIARLET, P. G. *Mathematical elasticity. Vol. I, volume 20 of Studies in Mathematics and its Applications*. North-Holland Publishing Co., Amsterdam, 1988.

- [36] CIORANESCU, D., DAMLAMIAN, A., AND GRISO, G. Periodic unfolding and homogenization. *Comptes Rendus Mathematique* 335, 1 (2002), 99–104.
- [37] CIORANESCU, D., DAMLAMIAN, A., AND GRISO, G. The periodic unfolding method in homogenization. *SIAM Journal on Mathematical Analysis* 40, 4 (2008), 1585–1620.
- [38] CIORANESCU, D., DAMLAMIAN, A., AND GRISO, G. The periodic unfolding method. *Series in Contemporary Mathematics* 3 (2018).
- [39] CIORANESCU, D., AND DONATO, P. *An introduction to homogenization*, vol. 17. Oxford University Press Oxford, 1999.
- [40] CIORANESCU, D., AND DONATO, P. *Introduction to homogenization*. Oxford Lecture Series in Mathematics and Applications 17, 2000.
- [41] CIORANESCU, D., DONATO, P., AND ZAKI, R. Periodic unfolding and robin problems in perforated domains. *Comptes Rendus Mathematique* 342, 7 (2006), 469–474.
- [42] CIORANESCU, D., AND PAULIN, J. S. J. *Homogenization of reticulated structures*, vol. 136. Springer Science & Business Media, 2012.
- [43] CIRSTEANU, H., AND KIRCHNER, C. The rewriting calculus — Part I and II. *Logic Journal of the Interest Group in Pure and Applied Logics* 9, 3 (May 2001), 427–498.
- [44] COMSOL. *MEMS Application Library Manual*, 2016.
- [45] CONANT, R. A., NEE, J. T., LAU, K. Y., AND MULLER, R. S. A flat high-frequency scanning micromirror. In *Proc. Solid-State Sensor and Actuator Workshop* (2000), pp. 6–9.
- [46] CORNELISSEN, S. A., BIERDEN, P. A., BIFANO, T. G., AND LAM, C. V. 4096-element continuous face-sheet mems deformable mirror for high-contrast imaging. *Journal of Micro/Nanolithography, MEMS, and MOEMS* 8, 3 (2009), 031308–031308.
- [47] COUSTEIX, J., AND MAUSS, J. *Analyse asymptotique et couche limite*, vol. 56. Springer Science & Business Media, 2006.
- [48] ECKHAUS, W. *Matched asymptotic expansions and singular perturbations*, vol. 6. Elsevier, 2011.
- [49] GAUDIELLO, A., GUIBÉ, O., AND MURAT, F. Homogenization of the brush problem with a source term in I 1. *Archive for Rational Mechanics and Analysis* 225, 1 (2017), 1–64.
- [50] GAUDIELLO, A., AND LENZNER, M. A two-dimensional electrostatic model of interdigitated comb drive in longitudinal mode. *SIAM Journal on Applied Mathematics* 80, 2 (2020), 792–813.

- [51] GÉRARD-VARET, D., AND MASMOUDI, N. Homogenization in polygonal domains. *J. Eur. Math. Soc.(JEMS)* 13, 5 (2011), 1477–1503.
- [52] GÉRARD-VARET, D., AND MASMOUDI, N. Homogenization and boundary layers. *Acta mathematica* 209, 1 (2012), 133–178.
- [53] GERARD-VARET, D., AND MASMOUDI, N. Recent progress in the theory of homogenization with oscillating dirichlet data. *arXiv preprint arXiv:1301.7229* (2013).
- [54] GEYMONAT, G., LICHT, C., AND WELLER, T. Plates made of piezoelectric materials: When are they really piezoelectric? *Applied mathematical modelling* 35, 1 (2011), 165–173.
- [55] GRIFFITHS, D. J., AND COLLEGER, R. Introduction to electrodynamics, prentice hall upper saddle river. *New Jersey* 7458 (1999).
- [56] GRISO, G. Error estimates in periodic homogenization with a non-homogeneous dirichlet condition. *Asymptotic Analysis* 87, 1-2 (2014), 91–121.
- [57] HELLER, B. Statistics for experimenters, an introduction to design, data analysis, and model building: Gep box, wg hunter and js hunter, john wiley and sons, new york, ny. 1978., 1986.
- [58] HOGAN, A., BLOMQUIST, E., COCHEZ, M., D'AMATO, C., DE MELO, G., GUTIERREZ, C., GAYO, J. E. L., KIRrane, S., NEUMAIER, S., POLLERES, A., ET AL. Knowledge graphs. *arXiv preprint arXiv:2003.02320* (2020).
- [59] HOPKINS, J., PANAS, R., AND SONG, Y. Categorization and review of existing micro-mirror array technologies. Tech. rep., Lawrence Livermore National Laboratory (LLNL), Livermore, CA, 2015.
- [60] JAIN, A., QU, H., TODD, S., AND XIE, H. A thermal bimorph micromirror with large bi-directional and vertical actuation. *Sensors and Actuators A: Physical* 122, 1 (2005), 9–15.
- [61] KIRCHNER, C., AND KIRCHNER, H. Equational logic and rewriting. In *Computational Logic* (2014), pp. 255–282.
- [62] KOVETZ, A. *Electromagnetic theory*, vol. 975. Oxford University Press Oxford, 2000.
- [63] LENCZNER, M. Homogénéisation d'un circuit électrique. *Comptes Rendus de l'Académie des Sciences-Series IIB-Mechanics-Physics-Chemistry-Astronomy* 324, 9 (1997), 537–542.
- [64] LENCZNER, M. Multiscale model for atomic force microscope array mechanical behavior. *Applied Physics Letters* 90, 9 (2007), 091908.

- [65] LENCZNER, M., AND SMITH, R. A two-scale model for an array of afm's cantilever in the static case. *Mathematical and computer modelling* 46, 5 (2007), 776–805.
- [66] MADUREIRA, A. L. Hierarchical modeling of piezoelectric plates. *Applied Mathematical Modelling* 36, 8 (2012), 3555–3569.
- [67] MARCHENKO, V. A., AND KHRUSLOV, E. Y. *Homogenization of partial differential equations*, vol. 46. Springer Science & Business Media, 2008.
- [68] MARTÍ-OLIET, N., AND MESEGUER, J. Rewriting logic as a logical and semantic framework. *Electronic Notes in Theoretical Computer Science* 4 (1996), 190–225.
- [69] MARUŠIĆ, S., AND MARUŠIĆ-PALOKA, E. Two-scale convergence for thin domains and its applications to some lower-dimensional models in fluid mechanics. *Asymptotic Analysis* 23, 1 (2000), 23–57.
- [70] MASE, G. T., AND MASE, G. E. *Continuum mechanics for engineers*. CRC press, 1999.
- [71] MEL'NYK, T., AND SADOVYI, D. *Multiple-scale analysis of boundary-value problems in thick multi-level junctions of type 3: 2: 2*. Springer Nature, 2020.
- [72] MESEGUER, J. Conditional rewriting logic as a unified model of concurrency. *Theoretical computer science* 96, 1 (1992), 73–155.
- [73] MYERS, R. H., MONTGOMERY, D. C., AND ANDERSON-COOK, C. M. *Response surface methodology: process and product optimization using designed experiments*. John Wiley & Sons, 2016.
- [74] NEUSS, N., NEUSS-RADU, M., AND MIKELIĆ, A. Effective laws for the poisson equation on domains with curved oscillating boundaries. *Applicable Analysis* 85, 05 (2006), 479–502.
- [75] NGUETSENG, G. A general convergence result for a functional related to the theory of homogenization. *SIAM Journal on Mathematical Analysis* 20, 3 (1989), 608–623.
- [76] NGUYEN, D., BELKHIR, W., RATIER, N., YANG, B., LENCZNER, M., ZAMKOTSIAN, F., AND CIRSTEANU, H. A multi-scale model of a micro-mirror array and an automatic model derivation tool. In *EurosimE* (2015).
- [77] NGUYEN, D. D. *Modeling a micro-mirror array and contribution to development of a simulation of micro-system arrays*. PhD thesis, University of France Comté, 2017.
- [78] OCAML. Ocaml. <https://ocaml.org/index.fr.html>.
- [79] OLEĬNIK, O. A., SHAMAEV, A., AND YOSIFIAN, G. *Mathematical problems in elasticity and homogenization*. Elsevier, 2009.

- [80] PAPANICOLAU, G., BENSOUSSAN, A., AND LIONS, J.-L. *Asymptotic analysis for periodic structures*. Elsevier, 1978.
- [81] PENGWANG, E., RABENOROSOA, K., RAKOTONDRABE, M., AND ANDREFF, N. Scanning micromirror platform based on mems technology for medical application. *Micro-machines* 7, 2 (2016), 24.
- [82] PRANGE, C. Asymptotic analysis of boundary layer correctors in periodic homogenization. *SIAM Journal on Mathematical Analysis* 45, 1 (2013), 345–387.
- [83] PTASHNYK, M. Two-scale convergence for locally periodic microstructures and homogenization of plywood structures. *Multiscale Modeling & Simulation* 11, 1 (2013), 92–117.
- [84] SENTURIA, S. D. *Microsystem design*. Springer Science & Business Media, 2007.
- [85] SHEN, Z. Boundary estimates in elliptic homogenization. *Analysis & PDE* 10, 3 (2017), 653–694.
- [86] SIMPSON, T. W., MAUERY, T. M., KORTE, J. J., AND MISTREE, F. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA journal* 39, 12 (2001), 2233–2241.
- [87] SONG, Y., PANAS, R. M., AND HOPKINS, J. B. A review of micromirror arrays. *Precision Engineering* (2018).
- [88] TARTAR, L. Quelques remarques sur l’homogénéisation. proc. of japan-france seminar 1976 “functional analysis and numerical analysis”, 469–482. *Japan Society for the Promotion of Science* (1978).
- [89] TARTAR, L. *The general theory of homogenization: a personalized introduction*, vol. 7. Springer Science & Business Media, 2009.
- [90] UTTAMCHANDANI, D. *Handbook of MEMS for wireless and mobile applications*. Elsevier, 2013.
- [91] VIAÑO, J., FIGUEIREDO, J., RIBEIRO, C., AND RODRÍGUEZ-ARÓS, Á. A model for bending and stretching of piezoelectric rods obtained by asymptotic analysis. *Zeitschrift für angewandte Mathematik und Physik* 66, 3 (2015), 1207–1232.
- [92] WALDIS, S., ZAMKOTSIAN, F., CLERC, P.-A., NOELL, W., ZICKAR, M., AND DE DE ROOIJ, N. Arrays of high tilt-angle micromirrors for multiobject spectroscopy. *IEEE Journal of Selected Topics in Quantum Electronics* 13, 2 (2007), 168–176.
- [93] WALDIS, S., ZAMKOTSIAN, F., NOELL, W., AND DE ROOIJ, N. Tiny mirrors for huge telescopes. *SPIE Newsroom* 10, 2.1200705 (2007), 0734.

- [94] YANG, B., BELKHIR, W., AND LENGZNER, M. Computer-aided derivation of multi-scale models: A rewriting framework. *International Journal for Multiscale Computational Engineering* 12, 2 (2014).
- [95] ZAMKOTSIAN, F., WALDIS, S., NOELL, W., ELHADI, K., LANZONI, P., AND DE ROOIJ, N. Micro-mirror array for multi-object spectroscopy. In *Optomechanical Technologies for Astronomy* (2006), vol. 6273, International Society for Optics and Photonics, p. 62731Q.
- [96] ZHIKOV, V. On an extension and an application of the two-scale convergence method. *Mat. Sb* 191, 7 (2000), 31–72.
- [97] ZHU, Y., LIU, W., JIA, K., LIAO, W., AND XIE, H. A piezoelectric unimorph actuator based tip-tilt-piston micromirror with high fill factor and small tilt and lateral shift. *Sensors and Actuators A: Physical* 167, 2 (2011), 495–501.

LIST OF FIGURES

1.1	The digital micro-mirror device (DMD) was invented by solid state physicist and Texas Instruments Fellow Emeritus Dr. Larry Hornbeck in 1987. This device is the heart of projection technology which uses millions of mirrors to generate a visual screen. The figure is taken from [11].	1
1.2	The electrostatic actuator group with three subgroups: (a) the surface actuators, (b) and (c) the in-plane and out-plane comb actuators.	2
1.3	Examples of the three other groups of actuators.	3
1.4	Examples of discrete and continuous mirrors.	4
1.5	Top view of the MIRA array with 100×200 cells. The zoom represents a single cell.	5
1.6	Application of the MIRA array in the the Multi-Object Spectroscopy. The figure taken from [93].	5
1.7	Illustration of a composite material. It occupies the domain $\Omega =]0, L[^2$. The structure in Ω has two components: a matrix in Ω_1 with conductivity γ_1 and inclusions in Ω_2 with conductivity γ_2 . We assume that the inclusions have a size in the same order than l and is periodically distributed with period l in each direction.	6
1.8	Description of (a) the composite material domain Ω and of (b) the unit cell Y . Y is the union of a matrix Y_1 and an inclusion Y_2 with conductivity tensor $\gamma(y)$. Obviously, Ω can be covered by cells of the form $\varepsilon Y = \varepsilon Y_1 \cup \varepsilon Y_2$	7
1.9	Expected flowchart of MEMSALab starting from an input PDE and producing an output sent in a simulation tool for PDEs as a finite element method software package (FEM). The input can also be generated by a user thanks to the User Language (UL). The <i>Selector</i> is an interface that selects the multiscale characteristics to be taken into account for building the multi-scale model in the <i>Core</i>	12

1.10 Script defining the differential equation of PDE_{Ref} in the earlier version of the User Language. Functions used in boundary conditions are also defined. The boundary conditions are omitted since their writing is similar to that of a PDE. The functions a , f and u are depending on the variable x which is defined on the domain ω . The latter has for boundary γ and for outward unit normal n . The functions h and g are functions of x_{γ_D} or x_{γ_N} two variables defined on two parts γ_D and γ_N of the boundary. 14

1.11 The script in the User Language of the reference proof $Proof_{Ref}$ that is applied to PDE_{Ref} . It is based on the proof algorithm that underlies the construction of the models developed in the thesis. The proof is made as a composition of strategies expressed by semi-columns. 14

1.12 Schematic view of the scheme of asymptotic model generation for the pair $(PDE_{Ref}, Proof_{Ref})$ and their extensions $(PDE_1, Proof_1)$ and $(PDE_2, Proof_2)$ by Ext_1 and Ext_2 15

1.13 Schematic view of the combination of Ext_1 , Ext_2 and the complementary extension Ext'_{12} built to generate a correct extension Ext_{12} 15

1.14 Schematic view of the pair $(Proof_{12}, PDE_{12})$ built by the combination Ext_{12} and the asymptotic model $Model_{12}$ obtained by application of $Proof_{12}$ to PDE_{12} 15

2.1 Zones where the asymptotic models are taken into account. The corresponding color numbers indicate the models'index. Assume that the array is actuated by two different imposed voltages in the internal and external zones, then it leads to two periodic models named the internal and the external periodic models. 25

2.2 Representation of two zones the external zone Ω_1 and the internal zone Ω_2 with different actuation voltage in the MIRA array. The zoom illustrates one cell $\Omega_{(1,1)}$ of the array with the mechanical structure in $\Omega_{(1,1)}^{mec}$ surrounded by the vacuum in $\Omega_{(1,1)}^{vac}$ 26

2.3 Illustration of the components of the cell Ω_c of the MIRA array. The mechanical part Ω_c^{mec} is made with two components, (a) the mirror part $\Omega_{ele,c}^{mec}$ and (b) the electrode part $\Omega_{mir,c}^{mec}$. Figure (c) represents the vacuum part Ω_c^{vac} . 27

2.4 The reference cell Ω^1 constructed by the mechanical body $\Omega^{1,mec}$ surrounded by the vacuum part $\Omega^{1,vac}$ occupies the block $] - 1/2, 1/2[^3$ 33

2.5 The physical domain $\Omega_{bl,1}^{\alpha\varepsilon}$ for the first lateral boundary model with two subdomains the mechanical body $\Omega_{bl,1}^{\alpha\varepsilon,mec}$ and the vacuum part $\Omega_{bl,1}^{\alpha\varepsilon,vac}$ with $\alpha = 1$. The zoom represents the internal subboundaries of the vacuum and the mechanical part between cells of the external zone. 44

2.6 The microscopic domain $\Omega_{bl,1}^1$ with two subdomains $\Omega_{bl,1}^{1,mec}$ and $\Omega_{bl,1}^{1,vac}$ with $\alpha = 1$ 45

2.7 The first exterior edge physical domain $\Omega_{exe,1}^{\alpha\varepsilon}$ including two subdomains $\Omega_{exe,1}^{\alpha\varepsilon,vac}$ and $\Omega_{exe,1}^{\alpha\varepsilon,mec}$ with $\alpha = 1$. The zoom illustrates their boundaries. 53

2.8 The first exterior edge physical domain $\Omega_{exe,1}^{\alpha\varepsilon}$ with two subdomains $\Omega_{exe,1}^{\alpha\varepsilon,vac}$ and $\Omega_{exe,1}^{\alpha\varepsilon,mec}$ 53

2.9 The first interface physical domain $\Omega_{in,1}^{\alpha\varepsilon}$ with two nonoverlapping subdomains $\Omega_{in,1}^{\alpha\varepsilon+}$ and $\Omega_{in,1}^{\alpha\varepsilon-}$, each domain $\Omega_{in,1}^{\alpha\varepsilon\pm}$ is assembled by two parts the vacuum part $\Omega_{in,1}^{\alpha\varepsilon,vac\pm}$ and the mechanical part $\Omega_{in,1}^{\alpha\varepsilon,mec\pm}$, with $\alpha = 1$ 59

2.10 The first interface microscopic domain $\Omega_{in,1}^1$ with two nonoverlapping subdomains $\Omega_{in,1}^{1,\pm}$, each of them also involves two parts, the vacuum part $\Omega_{in,1}^{1,vac\pm}$ and the mechanical part $\Omega_{in,1}^{1,mec\pm}$, in the case of $\alpha = 1$ 60

2.11 Description of the geometry of the internal edge problem. The green and maroon colors represent the zones of the first and the second interface models. The red region is the zone of the first internal edge model made with four subregions. The electrostatic potential has a different approximation in each of these subregions. 66

2.12 The first internal edge $\Omega_{me,1}^{\alpha\varepsilon}$ in the physical domain with $\alpha = 1$ 67

2.13 The first internal edge $\Omega_{me,1}^1$ in the microscopic domain with $\alpha = 1$ 68

3.1 Overview of the components of a MIRA cell. 95

3.2 Two-dimensional representation of the mirror component. 95

3.3 Three-dimensional representation of the mirror component. 95

3.4 Two-dimensional representation of the stopper beam. 96

3.5 Three-dimensional representation of the stopper beam. 96

3.6 Two-dimensional representation of the right suspended beam. 96

3.7 Three-dimensional representation of the right suspended beam. 96

3.8 Two-dimensional representation of the electrode base. 97

3.9 Three-dimensional representation of the electrode base. 97

- 3.10 Electrical potential ϕ^0 and the electrostatic field $\nabla\phi^0$ in the internal zone (a) and in the external zone (b). 101
- 3.11 The electric potentials (a) ϕ_{bl}^1 of the first lateral boundary model, (b) ϕ_{bl}^4 of the second one, (c) ϕ_{bl}^3 of the third one, and (d) ϕ_{bl}^4 of the fourth one. The boundary layer effect is concentrated in one cell. 101
- 3.12 Electric potentials ϕ_{in}^i with $i = 1, 2, 3, 4$ of four interface models. Each of ϕ_{in}^i includes two parts $\phi_{in}^{i,+}$ and $\phi_{in}^{i,-}$. The mirrors are hidden to facilitate visualization of the potential values at the interfaces. The contributions are significant at the interfaces and rapidly evanescent. 102
- 3.13 Electric potentials (a) ϕ_{exe}^1 and (b) ϕ_{ine}^1 of the outer and inner edge models. In both cases the fields are very localized and their value very small compared to the imposed voltage. It is likely that they are negligible in some cases. 102
- 3.14 The approximate solution near lateral boundaries given by the assembly of the external periodic solution ϕ^0 and the contribution of lateral boundary correctors $\phi_{bl}^i, i = 1, 2, 3, 4$ 104
- 3.15 The approximate solution near the interfaces given by the assembly of the periodic solution ϕ^0 and the contribution of interface correctors ϕ_{in}^i , where ϕ^{0+} and ϕ^{0-} represents the internal and external periodic solution, $\phi_{in}^{i,+}$ and $\phi_{in}^{i,-}$ depicts two parts of the interface corrector ϕ_{in}^i with $i = 1, 2, 3, 4$ 105
- 3.16 The model solution in the vicinity of the first outer edge. It is built with the external periodic solution ϕ^0 , the first and second boundary layer correctors ϕ_{bl}^1 and ϕ_{bl}^2 , and the first outer edge corrector ϕ_{exe}^1 105
- 3.17 Mechanical displacement \mathbf{u}^0 of the periodic model in the internal zone with the imposed voltage $60V$. The red narrows represent the electrostatic field. 106
- 3.18 An electrode actuation with two parallel plates: the movable plate supported by the spring with the spring constant k and the fixed plate. The distance between them is g_0 and the areas of each plate is A 107
- 3.19 Imposed voltage locations: (a) the voltage VE is applied on the bottom electrode part and (b) the voltage VM is applied to the mirror part, which includes the mirror itself, the frames, the two golden pads, and the two pillars. The gap between the mirror and the electrode is denoted by g_0 and the position `point` is at the end of the landing beam shown in the zoomed figure. 108
- 3.20 Graph of the imposed voltage with respect to the beam displacement at equilibrium for supporting beams with thickness $400nm$ and length $40\mu m$. . . 109

3.21	The surface response model for the pull-in voltage V_{PI} as a function of the suspended beam thickness and length variables $poly$ and $msux$. The main slope is in the direction of $poly$ while the surface is flat in the direction of $msux$, so the thickness of the beam is the variable that dominates this effect.	111
3.22	Effect of the electrostatic field of the right cell on the neighbour cell. The zoom represents the electrostatic field around the frame of the active cell with the applied voltage $80 V$ and the head of the mirror of the inactive one.	112
3.23	Affected location from the neighbour electrostatic field on the underside surface of the mirror of the inactive cell with an estimate area of $0.25 \times mlax \times my = 600 (\mu m^2)$ where my represents the length of the mirror and $mlax$ expresses that of the landing beam.	113
3.24	Mechanical displacement in the inactive cell with an order in the range of 10% of the gap between the electrode and the mirror.	113
4.1	The system of representation of multiscale geometries using the parameters <code>scale</code> and <code>level</code> . multiscale models. The scales are written in blue while the levels are in black.	123
4.2	Interface between three domains.	125
4.3	Tree representation of a <code>region</code> node for a boundary with name <code>gamma</code> .	127
4.4	Tree representation of a <code>region</code> node for a domain with name <code>omega</code> .	128
4.5	A physical domain whose boundary has two parts, one internal and one external.	128
4.6	A physical domain with two subdomains. The periodicity is in the x_1 -direction. In the x_2 -direction, the domain is assumed to be thin.	130
4.7	Tree representation of the <code>subModel</code> node.	139
4.8	Identifiers of the recursive multiscale models built on the pair (<code>subModel</code> , <code>subScale</code>).	142
4.9	A simple tree.	144
4.10	The processing of the application <code>goto([B,H],C,s)</code> to the tree of 4.9 to find successively B and H with a limit C.	144
4.11	Illustration of the operations made when applying the strategy <code>goto([B,E,H],C,insert(K))</code> to the tree of Figure 4.9.	145

4.12	The tree representation for the top subScale node of the physical model of electrostatic equations posed in the MIRA architecture. The domain $\Omega_{\text{air.main}}^{0,\text{micro}}$ and the electrostatic equations are defined in details in the first part of the thesis. The names of the submodels in the list subModelList are "periodic.int", "periodic.ext", "boundaryLayer.i" , "boundaryLayer.edge.i" , "boundaryLayer.interface.i" , and "boundaryLayer.interface.edge.i" for $i = 1,2,3,4$	148
4.13	The physical air domain of the MIRA model, the subdomains, their boundaries and subboundaries.	149
4.14	The submodel node of the external periodic model.	154
4.15	The ident node of the external periodic model.	154
4.16	The dilation node of the external periodic model.	154
4.17	The microscopic domains in the subScale node of the periodic.ext and periodic.int models.	155
4.18	The subScale node of the external periodic model.	155
4.19	Selection of the domain of a boundary layer model in the physical domain and the related microscopic domain.	157
4.20	Selection of the domain of an edge boundary layer model in the physical domain and the related microscopic domain.	160
4.21	The domain of an interface boundary layer model in the physical domain and the related microscopic domain. The physical domain $\Omega_{\text{air.interface.1}}^{0,\text{micro}}$ includes two non-overlapping subdomains $\Omega_{\text{air.interface.1.ext}}^{1,\text{micro}}$ and $\Omega_{\text{air.interface.1.int}}^{1,\text{micro}}$ with the interface $\Gamma_{\text{air.int.latBox.1}}^{0,\text{micro}}$, the corresponding microscopic domain $\Omega_{\text{air.interface.1}}^{1,\text{micro}}$ has also two non-overlapping subdomains $\Omega_{\text{air.interface.1.1}}^{1,\text{micro}}$ and $\Omega_{\text{air.interface.1.2}}^{1,\text{micro}}$ with the interface $\Gamma_{\text{air.interf.interface}}^{1,\text{micro}}$	163
4.22	Selection of the domain of an interface edge boundary layer model in the physical domain and the related microscopic domain.	167
6.1	Array Setting.	226
6.2	Vacuum Setting.	227

LIST OF TABLES

3.1	List of component parameters of MIRA's cell and their material.	98
3.2	Continuation of the the list of component parameters of MIRA's cell and their material.	99
3.3	Continuation of the the list of component parameters of MIRA's cell and their material.	100
3.4	Comparison of contribution of all lateral boundary correctors with the imposed voltage 80V where the magnitude is computed by the absolute value of the difference between the maximum and the minimum of the electric potential, the contribution is given as a percentage of the magnitude to the imposed voltage.	103
3.5	Comparison of contribution of all interface correctors with the imposed voltages, the internal one 60 V and the external one 80 V.	103
3.6	The Global Equation setting. VESP is an unknown of the equation $\text{intop1}(\text{point})-\text{pointZ}=0$, $\text{intop1}(\text{point})$ being the z -coordinate position of point and pointZ the given position.	108
3.7	The Auxiliary sweep's setting for the parameter pointZ.	108
3.8	Pull-in voltage values with respect to some lengths and thicknesses of the suspended beams.	109
4.1	The main nodes of the grammar that represents PDEs. The names of nodes ending in List are nodes containing lists of the same record (possibly with a truncated name) with an unspecified number of elements. For example, varList is a node containing a list of nodes of type variable. . .	118
4.2	The subnodes of the main nodes of Table 4.1, their types and their default value if they have one. The type constant means a terminal node.	119
4.3	List of parameters used for the concrete syntax trees.	121
4.4	Names of the types of dilation.	141

4.5	List of abstract strategies with their names and their parameters e or e' of type expression and s of type strategy defined in the grammar. The parameters <code>path</code> and <code>limit</code> are lists of nodes $[e, \dots]$	143
4.6	Summary of the names of the two air domains, of their boundaries and of the subboundaries.	150
4.7	Names of voltage sources and of the unknown electrical potential in the two regions of the physical PDE.	150
4.8	Names of the boundary condition in <i>the physical PDE</i> <code>pdeMira</code> where i represents "1,2,3,4".	150
4.9	Names of the equations of the physical PDE.	151
4.10	The names of the microscopic domains, boundaries and subboundaries for the models <code>periodic.int</code> and <code>periodic.ext</code>	154
4.11	The names of the regions of the microscopic domain $\Omega_{\text{air.ext.lat}}^{1,\text{micro}}$ for the boundary layer model.	157
4.12	The <code>submodel</code> nodes of the boundary layer model.	158
4.13	Dictionary of the <code>dilation</code> node for the boundary layer model. Each entry of the dictionary includes a key, a description and a list of data. The data list of the unknowns have two elements, the name of the unknown and the identifier of their submodel.	158
4.14	The names of the regions of the microscopic domain $\Omega_{\text{air.ext.edge.1}}^{1,\text{micro}}$ for the edge boundary layer model.	160
4.15	The submodel of the external edge model.	161
4.16	Dictionary of the <code>dilation</code> node for the edge boundary layer model.	161
4.17	Summary of the two subregion names $\Omega_{\text{air.interface.1.1}}^{1,\text{micro}}$ and $\Omega_{\text{air.interface.1.2}}^{1,\text{micro}}$ of the microscopic domain $\Omega_{\text{air.interface.1}}^{1,\text{micro}}$ for the interface boundary layer model.	164
4.18	The <code>subModel</code> node of the interface boundary layer model.	164
4.19	Dictionary of the <code>dilation</code> node for the interface boundary layer model.	165
4.20	Summary of the region names of the microscopic domain $\Omega_{\text{air.int.edge.1}}^{1,\text{micro}}$ for the interface edge boundary layer model.	168
4.21	The <code>subModel</code> node of the interface edge boundary layer.	168
4.22	Dictionary of the <code>dilation</code> node for the interface edge boundary layer model.	169
5.1	Constructors of Description Logic.	176

5.2	Summary of Rewriting System conversion into Description Logic in terms of individuals, concepts and roles. Rewriting expressions are in column 1 and the other columns summarize the related objects. Objects already defined in a previous line are not recalled. Identifiers of trees, patterns, strategies and individual start with t , p , s and i respectively. A data used in a tree or pattern leaf is noted as d . A node name starts with n . For shortness, in individual addresses the instance number is omitted. The role shortcuts <i>hasDST(...)</i> and <i>hasDSTS(...)</i> are used for <i>hasDirectSubTree(...)</i> and <i>hasDirectSubTreeS(...)</i> , <i>id</i> for "identifier" and <i>gV</i> for "globalVar".	187
5.3	Roles in the ontology of domains and boundaries.	190
5.4	Correspondences between the operations done in the algorithms, the strategy names and the properties. The strategies involved in all submodels are marked by a "-".	191
6.1	Imposed Voltages.	225
6.2	Electric displacement field jump setting, where <i>es12</i> represents the first part of the interface corrector, <i>es11</i> and <i>es10</i> expresses the internal and external periodic component.	230



APPENDIX

COMSOL IMPLEMENTATION

6.1/ THE FIRST ELETROSTATIC INTERFACE MODEL

6.1.1/ COMPONENT

- Open COMSOL Multiphysics, from the **file** choose **New**.
- In the new window, click **Model Wizard** shortcut and then choose **3D** and click **Done**. At the setting of **Component 1**, in the **Label**, replace it by **Philn1**.

6.1.2/ GLOBAL DEFINITION

Parameters:

1. In the Home toolbar, click **Parameter**.
2. Click icon - **Load from file** under the **Setting Parameters**, and locate where the file *parameters.txt* is, then choose it.
3. In the table, enter as Table 6.1.

Name	Expression	Description
VM1	40V	an applied voltage in the mirror part of external zone
VE1	-40V	an applied voltage in the electrode part of external zone
VM2	30V	an applied voltage in the mirror part of internal zone
VE	-30V	an applied voltage in the mirror part of internal zone

Table 6.1: Imposed Voltages.

6.1.3/ GEOMETRY

1. In the **Model Builder** window, under **Philn1**, right-click **Geometry1** and choose **Import**.
2. In the **Import Setting**, under the **Filename**, click **Browse** to locate the *RCell.mphbin* and click **Import**.
3. Right-Click **Geometry**, click **Array**. In the **Array Setting**, input as Figure 6.1.

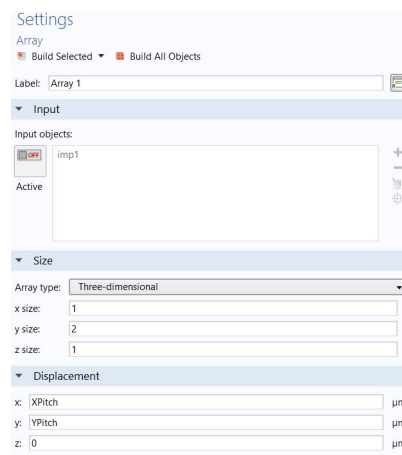


Figure 6.1: Array Setting.

6.1.4/ MATERIAL

1. In the **Model Builder**, right-click **Material** > **Add Material from Library**.
2. In the **Add Material**, under **MEMS** > **Semiconductors**, double-click **Si - Silicon (single-crystal, isotropic)**.
3. In the **settings** window for the **Material**, select domains correspond to the Mirror and the Electrode part 1, 3, 6, 8-9, 11, 14, 16-17, 19, 21-22, 24, 26, 31-34, 47-48, 58, 60, 62, 64.
4. Repeat the process and add **Si - Polycrystalline Silicon** - the beams 27-30, 35-42, 49-56, **Au - Gold** - golden part 43-46, **SiO2 - Silicon oxide** - layer 2, 5, 7, 10, 13, 15, 18, 20, 23, 25, 57, 59, 61, 63.
5. Right-Click **Material**, click **Blank Material** and rename **Vacuum**. In the **Vacuum Setting**, input as Figure 6.2.

6.1.5/ PHYSICS (INTERNAL AND EXTERNAL PERIODIC MODELS)

6.1.5.1/ EXTERNAL PERIODIC MODEL

1. Click **Physics**, add **Electrostatic** and rename it by **Phi01In1**. In its setting, at **Domain Selection** field, choose 4.
2. Right-Click **Phi01In1**, click **Electrode Potential** and rename it by **VM**. In its setting, at **Boundary Selection**, choose boundaries corresponding to the mirror part 71, 74, 78-79, 81, 83, 87-88, 90, 92, 117-119, 121-125, 136-137, 139-142, 144-145, 156-157, 160-162, 165-167, 170-172, 175, 196-197, 200-202, 205, 216-217, 220-223, 228-229, 232-234, 236-241, 252, 254-255, 257-258, 267, 269, 297, 301, 303,

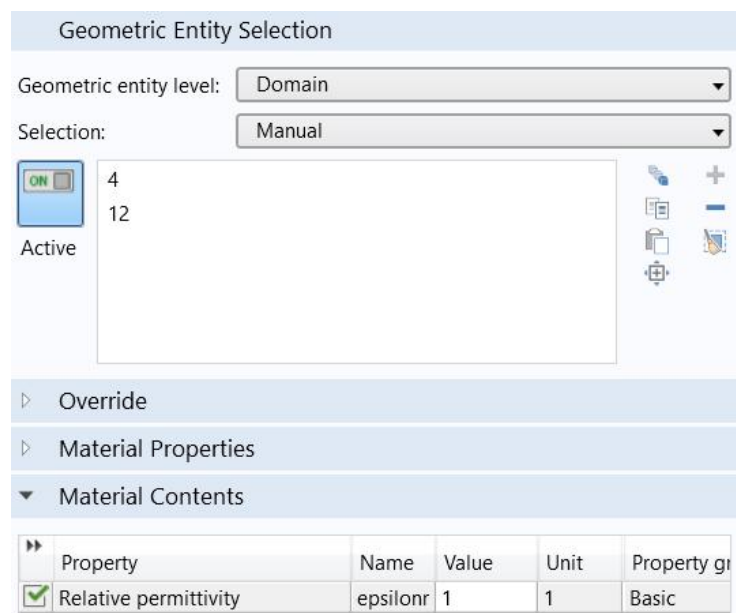


Figure 6.2: Vacuum Setting.

312-313, 316-318, 321-323, 326-327, 331, 402-405, 410, 428-429. In the **Electric Potential**, input VM1.

3. Right-Click **Phi01In1**, click **Electrode Potential** and rename it by **VE**. In its setting, at **Boundary Selection**, choose boundaries corresponding to the electrode part 21, 23, 26, 282-283, 285, 287, 353, 355, 358-359, 376, 412, 414, 417-418. In the **Electric Potential**, input VE1.
4. Right-Click **Phi01In1**, click **Zero charge**. In its setting, at **Boundary Selection** field, choose 14, 18, 24, 28, 76, 82, 85, 91, 264, 266, 268, 270, 280-281, 284, 286, 352, 354, 356-357, 368, 375, 378, 411, 413, 415-416.
5. Right-Click **Phi01In1**, click **Periodic Condition**. In its setting, at **Boundary Selection** field, choose 10, 435.
6. Right-Click **Phi01In1**, click **Periodic Condition**. In its setting, at **Boundary Selection** field, choose 11, 44.

6.1.5.2/ INTERNAL PERIODIC MODEL

1. Click **Physics**, add **Electrostatic** and rename it by **Phi02In1**. In its setting, at **Domain Selection** field, choose 12.
2. Right-Click **Phi02In1**, click **Electrode Potential** and rename it by **VM**. In its setting, at **Boundary Selection**, choose boundaries corresponding to the mirror part 93,

95-96, 100-101, 103, 105, 109-110, 112, 114, 126-135, 146-147, 149-152, 154-155, 176-177, 180-182, 185-187, 190-192, 195, 206-207, 210-212, 215, 218-219, 224-227, 230-231, 242-249, 253, 259-263, 275, 277, 304-310, 332-333, 336-338, 341-343, 347-348, 351, 406-409, 419, 430-431. In the **Electric Potential**, input VM2.

3. Right-Click **Phi02In1**, click **Electrode Potential** and rename it by **VE**. In its setting, at **Boundary Selection**, choose boundaries corresponding to the electrode part 54, 56, 59, 290-291, 293, 295, 361, 363, 366-367, 421, 423, 426-427. In the **Electric Potential**, input VE1.
4. Right-Click **Phi02In1**, click **Zero charge**. In its setting, at **Boundary Selection** field, choose 46-47, 51, 57, 61, 97-98, 104, 106, 113, 272, 274, 276, 278, 288-289, 292, 294, 360, 362, 364-365, 384, 391, 393, 420, 422, 424-425.
5. Right-Click **Phi02In1**, click **Periodic Condition**. In its setting, at **Boundary Selection** field, choose 43, 443.
6. Right-Click **Phi02In1**, click **Periodic Condition**. In its setting, **Boundary Selection** field, choose 44, 70.

6.1.6/ STUDY (COMPUTE TWO PERIODIC SOLUTIONS)

1. Click **Study**, choose **Stationary** and in **Physics interfaces in study**, click two note at **Solve** for **Phi01In1** and **Phi02In2**.
2. In **Study setting**, click **Compute**. Then, we get two periodic solutions stored in the data **sol5**.

6.1.7/ PHYSICS SET UP FOR THE FIRST INTERFACE MODEL

6.1.7.1/ FIRST PART OF THE INTERFACE CORRECTOR

1. Click **Physics**, add **Electrostatic** and rename it by **Phi1In1**. In its setting, at **Domain Selection** field, choose 4.
2. Right-Click **Phi1In1**, click **Electrode Potential** and rename it by **VM**. In its setting, at **Boundary Selection**, choose boundaries corresponding to the mirror part 71, 74, 78-79, 81, 83, 87-88, 90, 92, 117-119, 121-125, 136-137, 139-142, 144-145, 156-157, 160-162, 165-167, 170-172, 175, 196-197, 200-202, 205, 216-217, 220-223, 228-229, 232-234, 236-241, 252, 254-255, 257-258, 267, 269, 297, 301, 303, 312-313, 316-318, 321-323, 326-327, 331, 402-405, 410, 428-429. In the **Electric Potential**, input 0.

3. Right-Click **Phi1In1**, click **Electrode Potential** and rename it by **VE**. In its setting, at **Boundary Selection**, choose boundaries corresponding to the electrode part 21, 23, 26, 282-283, 285, 287, 353, 355, 358-359, 376, 412, 414, 417-418. In the **Electric Potential**, input 0.
4. Right-Click **Phi1In1**, click **Zero charge**. In its setting, at **Boundary Selection** field, choose 14, 18, 24, 28, 76, 82, 85, 91, 264, 266, 268, 270, 280-281, 284, 286, 352, 354, 356-357, 368, 375, 378, 411, 413, 415-416.
5. Right-Click **Phi1In1**, click **Periodic Condition**. In its setting, at **Boundary Selection** field, choose 10, 435.
6. Right-Click **Phi1In1**, click **Electrode Potential** and rename it by **Potential Jump**. In its setting, at **Boundary Selection**, choose boundaries corresponding to interface **44**. In the **Potential Jump**, input $\text{Phi2In1} + \text{withsol('sol5',Phi02In1 - Phi01In1)}$.

6.1.7.2/ SECOND PART OF THE INTERFACE CORRECTOR

1. Click **Physics**, add **Electrostatic** and rename it by **Phi2In1**. In its setting, at **Domain Selection** field, choose 12.
2. Right-Click **Phi2In1**, click **Electrode Potential** and rename it by **VM**. In its setting, at **Boundary Selection**, choose boundaries corresponding to the mirror part 93, 95-96, 100-101, 103, 105, 109-110, 112, 114, 126-135, 146-147, 149-152, 154-155, 176-177, 180-182, 185-187, 190-192, 195, 206-207, 210-212, 215, 218-219, 224-227, 230-231, 242-249, 253, 259-263, 275, 277, 304-310, 332-333, 336-338, 341-343, 347-348, 351, 406-409, 419, 430-431. In the **Electric Potential**, input 0.
3. Right-Click **Phi2In1**, click **Electrode Potential** and rename it by **VE**. In its setting, at **Boundary Selection**, choose boundaries corresponding to the electrode part 54, 56, 59, 290-291, 293, 295, 361, 363, 366-367, 421, 423, 426-427. In the **Electric Potential**, input 0.
4. Right-Click **Phi2In1**, click **Zero charge**. In its setting, at **Boundary Selection** field, choose 46-47, 51, 57, 61, 97-98, 104, 106, 113, 272, 274, 276, 278, 288-289, 292, 294, 360, 362, 364-365, 384, 391, 393, 420, 422, 424-425.
5. Right-Click **Phi2In1**, click **Periodic Condition**. In its setting, **Boundary Selection** field, choose 43, 443.
6. Right-Click **Phi2In1**, click **Electric Displacement Field** and rename it by **Electric Displacement Field Jump**. In its setting, at **Boundary Selection**, choose boundaries corresponding to interface **44**. In the **Electric Displacement Field Jump**, input as the following table.

D_0	$es12.Dx - withsol('sol5', es11.Dx - es10.Dx)$	x
	$es12.Dy - withsol('sol5', es11.Dy - es10.Dy)$	y
	$es12.Dz - withsol('sol5', es11.Dz - es10.Dz)$	z

Table 6.2: Electric displacement field jump setting, where $es12$ represents the first part of the interface corrector, $es11$ and $es10$ expresses the internal and external periodic component.

6.1.8/ STUDY (COMPUTE INTERFACE SOLUTIONS)

1. Click **Study**, choose **Stationary** and in **Physics interfaces in study**, click two notes at **Solve** for **Phi1In1** and **Phi2In1**.
2. In **Study** toolbar, click **Show Default Solver**.
3. At **Solver Configurations**, right-click **Stationary Solvers**, add **Fully Coupled**.
4. In **Study Setting**, click **Compute**. The solution is stored in the data named **sol8**.

6.1.9/ RESULTS (ASSEMBY OF SOLUTIONS)

1. In the **Results** toolbar, tick **Volume** and rename it by **Philn1**.
2. At the **Results** field, right-click **Philn1**, add **Surface** rename it by **PhiNear1In1**.
3. In **PhiNear1In1** setting, at **Expression** field, input $Phi1In1 + withsol('sol5', Phi01In1)$.
4. At the **Results** field, right-click **Philn2**, add **Surface** rename it by **PhiNear2In1**.
5. In **PhiNear2In1** setting, at **Expression** field, input $Phi2In1 + withsol('sol5', Phi02In1)$.
6. In **Philn1**, click **plot**.

Title: Asymptotic modeling of a micro-mirror array and software design for automatically derived multiscale models

Keywords: Micro-Mirror Array, Periodic Homogenization, Two-Scale Convergence, Boundary Layer Correctors, Numerical Simulation, Symbolic Computation, Rewriting Technique, Description Logic

Abstract:

This Ph.D. thesis is in the context of the development of mathematical tools for software whose aim is to automatically derive asymptotic models of MEMS Arrays using symbolic computation and a reuse concept. The contribution covers three aspects: asymptotic modeling, numerical simulations, and software design. The derivation of asymptotic models for the coupled electromechanical problem in the micro-mirror array MIRA is achieved by employing the two-scale transformation method. With the goal to capture a larger family of asymptotic models, we consider the case where the array is actuated by two different voltages in two zones. As a result, the model includes periodic sub-models and boundary corrector sub-models for the lateral boundary, interfaces, inner edges and outer edges. With these models, the simulation of the network is feasible despite its

large size. Numerical simulations of the electrostatic part are implemented in finite element software. In addition, the system design is optimized to minimize pull-in voltage. In terms of software design, new forms of data structure and rewriting strategies are designed using abstract syntax trees. This leads, on the one hand, to the standardization of the data structure and to make it more flexible and, on the other hand, to make the strategies more general which allows to decrease their number and thus the complexity of the software. The data structure is constructed with a recursive form that broadens the capability of description of families of asymptotic models. Finally, a concurrent system made with the rewriting system and a description logic framework is investigated. This is the first step in building an efficient knowledge representation in software for asymptotic model building.

Titre : Modélisation asymptotique d'un réseau de micro-miroirs et design de logiciel pour les modèles multi-échelles dérivés automatiquement

Mots-clés : Matrice de Micro-Miroirs, Homogénéisation Périodique, Convergence à Deux Échelles, Correcteur de Couche Limite, Simulation Numérique, Calcul Symbolique, Techniques de Réécriture, Logique de Description

Résumé :

Cette thèse de doctorat s'inscrit dans le cadre du développement d'outils mathématiques pour un logiciel dont le but est de dériver automatiquement des modèles asymptotiques de matrices de MEMS en utilisant un calcul symbolique et un concept de réutilisation. La contribution couvre trois aspects : la modélisation asymptotique, les simulations numériques et la conception du logiciel. La dérivation de modèles asymptotiques pour le problème électromécanique couplé dans le réseau de micro-miroirs MIRA est réalisée en employant la méthode de transformation à deux échelles. Dans le but de capturer une plus grande famille de modèles asymptotiques, nous considérons le cas où le réseau est actionné par deux tensions différentes dans deux zones. Par conséquent, le modèle comprend des sous-modèles périodiques et des sous-modèles de correcteurs de frontières pour la frontière latérale, les interfaces, les arêtes intérieures et les arêtes extérieures. Grâce à ces modèles, la simulation du réseau est réalisable malgré sa grande taille.

Les simulations numériques de la partie électrostatique sont mises en œuvre dans un logiciel d'éléments finis. En outre, la conception du système est optimisée pour minimiser la tension de pull-in. En ce qui concerne la conception du logiciel, de nouvelles formes de structure de données et des stratégies de réécriture est conçue en utilisant des arbres syntaxiques abstraits. Cela conduit d'une part à l'uniformisation de la structure de données et à la rendre plus flexible et d'autre part de rendre les stratégies plus générales ce qui permet de diminuer leur nombre et donc la complexité du logiciel. La structure de données est construite avec une forme récursive qui élargit la capacité de description des familles de modèles asymptotiques. Enfin, un système concurrent réalisé avec le système de réécriture et un cadre de logique de description est étudié. Il s'agit d'une première étape dans la construction d'une représentation efficace dans un logiciel des connaissances pour les construction de modèles asymptotiques.