A multi-scale model of a Micro-Mirror Array and an automatic model derivation tool

Duy Duc Nguyen[1], Walid Belkhir[1,2], Nicolas Ratier[3], Bin Yang[1], Michel Lenczner [4], Frédéric Zamkotsian[5] and Horatiu Cirstea[6]

[1]FEMTO-ST, Time and Frequency Department, University of Franche-Comté,
26 chemin de l'Epitaphe, 25030 Besançon Cedex, France
[2]INRIA Nancy - Grand Est, CASSIS project, 54600 Villers-lès-Nancy,France.
[3]FEMTO-ST, Time and Frequency Department, Technical University of Belfort-Monbéliard,
90010 Belfort Cedex, France.
[4]FEMTO-ST, Time and Frequency Department, Ecole Nationale Supérieure de Mécanique et de Microtechniques,
25030 Besançon Cedex, France
[5]LAM-CNRS, 38 rue Frédéric Joliot-Curie, 13388
Marseille Cedex 13, France.
[6] University of Lorraine - LORIA
Campus scientifique - BP 239 - 54506 Vandœuvre-lès-Nancy, France.

Email: {duyduc.nguyen, nicolas.ratier, bin.yang}@femto-st.fr, walid.belkhir@inria.fr, michel.lenczner@utbm.fr,
frederic.zamkotsian@lam.fr, Horatiu.Cirstea@loria.fr

Abstract

This paper reports recent advances in the development of a symbolic asymptotic modeling software package MEMSALab which will be used for automatic generation of asymptotic models for arrays of micro and nanosystems. First, an asymptotic model for the stationary heat equation in a Micro-Mirror Array developed for astrophysics is presented together with some key elements of its derivation. This illustrates the mathematical operations that need to be implemented in MEMSALab. The principle of operation of this software is to construct models incrementally so that model features can be included step by step. This idea conceptualized under the name "*by extension-combination*" is presented for the first time after having recalled the general functioning principles. A friendly user language recently introduced is also shortly discussed.

Keywords. *Homogenization, micro-mirror array, symbolic computation, computer-aided derivation of asymptotic models*

1. Introduction

Many systems encountered in micro or nano-technologies are governed by differential or partial differential equations (PDEs) that are too complex to be directly simulated with general software. In a number of cases, the complexity is due to a combination of several factors as several space or time scales, large coefficient heterogeneity or large aspect ratios. Many methods have been developed to overcome these difficulties, and in particular the asymptotic methods, also called perturbation techniques, constitute an active field of research in all fields of physics and mathematics for more than a century. Their application is based on a case-by-case approach so they are implemented only in specialized software. We adopt an alternate approach by developing a software package called `MEMSALab` (for MEMS Array Lab) whose aim is to derive families of asymptotic models that can be directly exploited in simulation software. Potentially it has a broad range of applications, however we focus on modeling arrays of micro or nano-systems.

Our approach of the software development is two-fold. On one hand we develop computer science concepts and tools allowing the software implementation and on the other hand we derive and implement asymptotic models to anticipate the introduction of related modeling concepts in the software library. This paper is written in this spirit, it reports on an asymptotic model of a Micro-Mirror Array (MMA) and reports our last advances in the development of the kernel of `MEMSALab`.

The MMA, see Figure 2, presented in [1], is dedicated to applications in astrophysics. It is used for instance as a field selector for multi-object spectroscopy (MOS) since it allows individual selection of objects by preventing overlapping of spectra and remove spoiling sources and background emission. The full modeling of the MMA should cover mechanical, electrical and thermal effects, however this paper focuses on the heat diffusion only. Evidently, a direct simulation of the heat equation in such MMA by a Finite Element Method (FEM) turns out to be impractical, since the MMA has a complex geometry and a large number in cells. So, we apply the multi-scale modeling method, based on the framework of periodic homogenization as in [2], that applies to model arrays of micro-systems with possibly large variations of temperature in a cell and between neighboring cells. The resulting model is implemented in COMSOL, that can be run in a reasonable time so that to be used in other postprocessing tasks as a multi-objective optimization procedure.

The technique of model derivation relies on an asymptotic method taking into account the small ratio between the sizes of a cell and of the whole array. It is not detailed since it is relatively long and technical, however we present some key facts giving an idea of the features playing a role in the derivation.

Unlike traditional software packages aimed at numerical simulations by using pre-built models, the purpose of `MEMSALab` is to derive asymptotic models for input equations by taking into account their own features e.g. the scalar valued or vector valued solution, different estimates on the solutions and sources, thin structures, periodic structures, multiple nested scales etc. The architecture of `MEMSALab` is shown in Figure 1. Its expected functioning can be schematized as follows. Firstly, the *FEM*

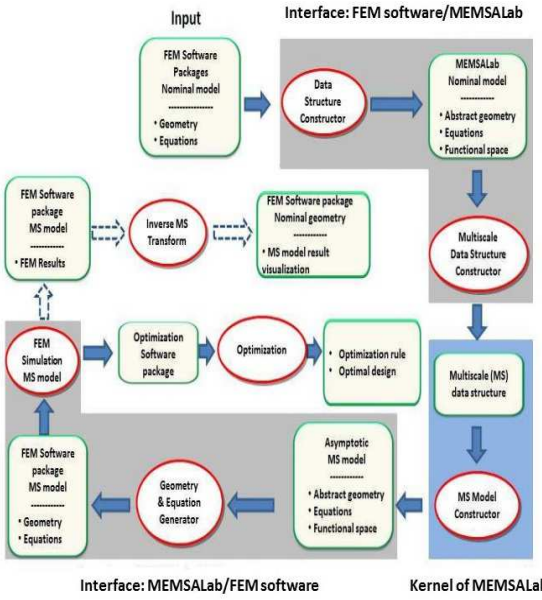

Fig. 1: The MEMSALab software envisioned architecture.

software/MEMSALab interface translates a system of PDEs from a FEM software package to the grammar used in the kernel of MEMSALab. Secondly, the multi-scale model (MSM) is automatically derived and the result is saved. Finally, the *MEMSALab/FEM Software interface*, translates back the resulting model to the format of the FEM software package and its simulation is launched. The advances in the development of this tool have been reported in [3, 4, 5]. After recalling the main ideas behind MEMSALab, this paper reports on a recent progress: the so called *"by extension-composition"* mechanism i.e. the MSM constructor depicted in the right-bottom of Figure 1. This method yields an incremental model construction so that the wanted features can be included step by step. We also comment the recent introduction of a dedicated user language.

2. Description of the Micro-Mirror Array

The structure of this MMA is detailed in [6]. Figure 3 shows the components of its elementary cell which is divided into two parts: the mirror part and the electrode part. The mirror part is composed of the mirror, two stopper beams with two landing beams on their tips and a suspending beam. The electrode part is composed of an electrode, two landing pads and two pillars.

Each cell can be addressable by applying different voltages on its line and column and then tilted due to the generation of electrostatic force on its mirror's surface, [1] and [6]. At rest, when no voltage is applied, the micro-mirror is held in a flat position by the suspended beams. When a voltage difference is applied between the micro-mirror and the electrode, an electrostatic force is generated, resulting in the attraction of the micro-mirror toward the fixed electrode, and leading to tilting and provides a restoring force For voltages below the pull-in voltage, the micro-mirror is operated in an analogue mode, allowing the angle to be set to a few degrees as a function of the applied voltage. At the pull-in voltage, the force increases and the micro-mirror snaps toward the electrode. During this motion, the micro-mirror touches its stopper beam and its landing pads. After pull-in, the micro-mirror is fixed at a precise tilt angle, due to contact with the stopper beam and landing pads. When the voltage is reduced, the micro-mirror angle remains constant until the mirror detaches from its stopper beam and increased its tilt angle. Finally, when the spring force of the suspended beams overcome the electrostatic force, the landing beams detach from the landing pads and the mirror returns to its rest position.

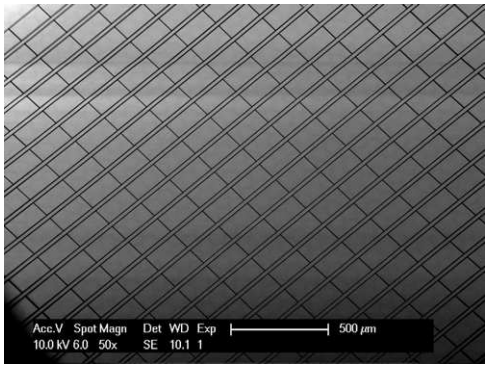

Fig. 2: A top-view of the Micro-Mirror Array micro-mirror of size: 100 x 200 square micrometers)

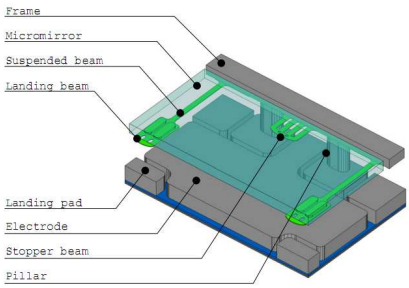

Fig. 3: Architecture of a MMA's cell

The actuation phenomena is based on an attraction through an electrostatic force generated by a difference of potential between the electrode and the mirror, and the spring force of the suspended beams becoming active when the voltage decreases.

A complete modeling should take into account the electric voltages between the micro-mirrors and the electrodes; the thermo-elasticity problem in beams; the heat diffusion in all parts, and the linear frictionless contact problem between stopper beam and landing pads. Here, we only consider the heat transfer in the whole array. Regarding the heat sources, one is coming from the environment such as black bodies at the given temperature or mechanical parts around the component gives a significance contribution to warm up the MMA, while the other coming from the stars and galaxies is very small and probably negligible. To dissipate this heat, the MMA is attached to a heat sink.

3. Homogenized model of the heat equation for the MMA

Starting from the mathematical statement of the heat equation in the MMA, we describe the assumptions taken

into account in the asymptotic model derivation, the two-scale transform which is the key mathematical tool, the a priori estimates of the solution, the asymptotic model itself, and the simulation results. These are the operations expected to be done by MEMSALab linked to a FEM software package.

3.1 Mathematical model

We consider a $N \times 1$ array of MMs as shown in Figure 4 which cell represented in Figure 5 is comprised of a micro-mirror including two stopping beams, a part of the frame, two pillars and an electrode made of silicon. After rescaling the array size to a unit length, we denote by ε the order of magnitude of the cell dimensions. This parameter decreases when the number N of cells of the array increases, and we determine, in a sense explained hereafter, an approximation of the temperature field for small values of ε. The region Ω^ε occupied by the device which is split into Ω_m^ε, Ω_f^ε, $\Omega_p^\varepsilon \subset \mathbb{R}^3$ the subregions occupied by the mirrors including the stopping beams, the frame and the pillars respectively. The body heat source r is present in the frame and the mirror, and the thermal conductivity may be anisotropic with matrices $\mathbf{a}^\varepsilon$, $\mathbf{a}^{m,\varepsilon}$, $\mathbf{a}^{f,\varepsilon}$ and $\mathbf{a}^{p,\varepsilon}$ in Ω^ε, Ω_m^ε, Ω_f^ε and Ω_p^ε. The electrodes act as a sink with an imposed ambient temperature, so a Dirichlet boundary condition is imposed on the bottom surfaces $\Gamma^{0,p}$ of the pillars, cf Figure 5. The field θ of the difference of temperature to the ambient temperature is solution to the stationary equation

$$\begin{cases} -\operatorname{div}(\mathbf{a}^\varepsilon \nabla\theta) = r & \text{in } \Omega^\varepsilon \\ \theta = 0 & \text{on } \Gamma^{0,p} \\ (\mathbf{a}^\varepsilon \nabla\theta)\cdot \mathbf{n} = 0 & \text{on } \partial\Omega^\varepsilon - \Gamma^{0,p}, \end{cases}$$

$\partial\Omega^\varepsilon$ representing the boundary of Ω^ε. The corresponding weak formulation is

$$\begin{aligned} &\oint_{\Omega_m^\varepsilon} \mathbf{a}^{m,\varepsilon}\nabla\theta\nabla v\,\mathrm{d}x + R_m^f \oint_{\Omega_f^\varepsilon} \mathbf{a}^{f,\varepsilon}\nabla\theta\nabla v\,\mathrm{d}x \\ &+R_m^p \oint_{\Omega_p^\varepsilon} \mathbf{a}^{p,\varepsilon}\nabla\theta\nabla v\,\mathrm{d}x = \oint_{\Omega_m^\varepsilon} rv\,\mathrm{d}x \\ &+R_m^f \oint_{\Omega_f^\varepsilon} rv\,\mathrm{d}x, \end{aligned} \tag{1}$$

using $\oint_\Omega F\,\mathrm{d}\Omega = \frac{1}{|\Omega|}\int_\Omega F\,\mathrm{d}\Omega$ the mean value of an integral and the volume ratios $R_m^f = \left|\Omega_f^\varepsilon\right| / |\Omega_m^\varepsilon|$ and $R_m^p = \left|\Omega_p^\varepsilon\right| / |\Omega_m^\varepsilon|$.

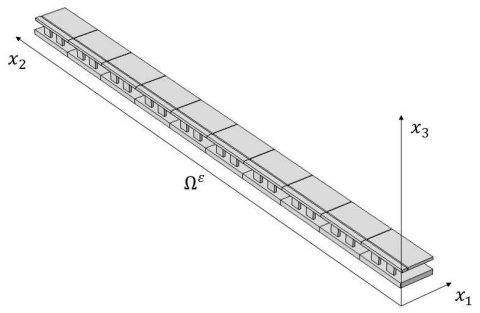

Fig. 4: A 10 x 1 MMA

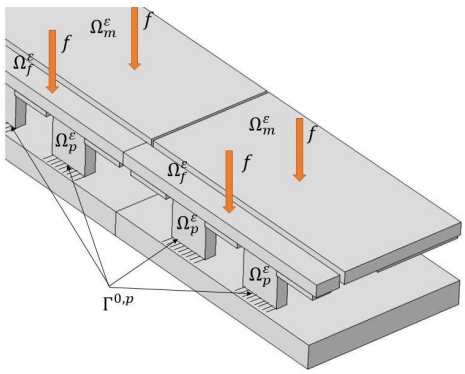

Fig. 5: Zoom on a cell of the 10x1 MMA

3.2 A priori estimates

Once the equations and the geometry of the problem have been set, as it is done in FEM software, an important step for an asymptotic analysis is to specify the behavior of the sources with respect to ε. With this purpose, we denote by $L^2(\Omega)$ the set of square integrable functions F over a domain Ω, i.e. such that $\int_\Omega |F|^2\,\mathrm{d}x < \infty$. In this problem, the volume heat source in Ω_m^ε and Ω_f^ε has to be large enough to heat the parts that is mathematically written as $\exists C > 0$ so that $||\varepsilon r||^2_{L^2(\Omega_m^\varepsilon)} / |\Omega_m^\varepsilon| \leq C$ and $||r||^2_{L^2(\Omega_f^\varepsilon)} / |\Omega_f^\varepsilon| \leq C$.

Then, *a priori* estimates satisfied by the temperature field are derived by taking into account the special *features* of the problem: ie the characteristics of the geometry and of the coefficients with respect to the small parameter ε. Here, we do not report the mathematical derivation, but simply express the physical assumptions and their mathematical consequences in terms of estimates with their meaning. The connections of the thin micromirrors to the frame are through thin and narrow beams which effect is equivalent to a connection by low conductivity components. It results in a possibly large temperature variation in the mirrors and suspending beams of the range of ε^{-1} expressed as $\|\varepsilon\nabla\theta\|^2_{L^2(\Omega_m^\varepsilon)} / |\Omega_m^\varepsilon| < C$. In the frame which is a continuous body, the simple estimates $\|\nabla_x\theta\|^2_{L^2(\Omega_f^\varepsilon)} / |\Omega_f^\varepsilon| < C$ holds. Regarding the pillars, their small section compared to the surface of the mirrors and the electrodes yields a difference of temperature of the order $O(1)$ between their bottom and top ends which requires a temperature variation in the pillar direction to be in the range of ε^{-1}. This is the meaning of the estimate $\|\varepsilon\nabla\theta\|^2_{L^2(\Omega_p^\varepsilon)} / |\Omega_p^\varepsilon| < C$.

3.3 Two-scale convergence

Following [2, 7, 8], the two-scale transform operator T maps the physical periodic domain $\Omega^\varepsilon = \Omega_m^\varepsilon \cup \Omega_f^\varepsilon \cup \Omega_p^\varepsilon$ into a two-scale domain $\omega \times \Omega^1$, see Figure 6. The microscopic cell $\Omega^1 = \Omega_m^1 \cup \Omega_f^1 \cup \Omega_p^1 \subset \mathbb{R}^3$ is deduced from any cell Ω_i^ε of the array centered at x_i^c by a translation and a dilation: $\Omega^1 = \{x^1 = (x - x_i^c)/\varepsilon \mid x \in \Omega_i^\varepsilon\}$. The macroscopic domain $\omega \subset \mathbb{R}$ is a segment in the direction x_2 having the length of the array and passing through the centers x_i^c. It is used for refereing to the cells. Precisely, the transformation T is applied to any function w defined on Ω^ε by $(Tw)(x^0, x^1) = w(x_i^c + \varepsilon x^1)$ for any x^0 in a cell

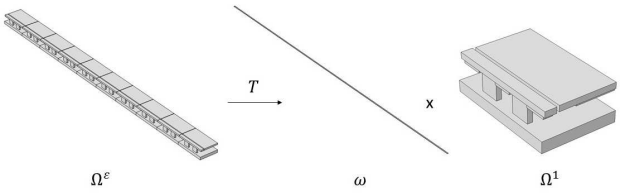

Fig. 6: Two-scale transform of Ω^ε into $\omega \times \Omega^1$

$\omega \cap \Omega_i^\varepsilon$ and any $x^1 \in \Omega^1$. We also define the operator B mapping functions defined on $\omega \times \Omega^1$ to functions defined on Ω^ε by $Bv(x) = v\left(x_2, (x - x_i^c)/\varepsilon\right)$ for any $x \in \Omega$.

We assume that there exists a main temperature field θ^0 and its corrector θ^1 such that,

$$\oint_{\Omega^\varepsilon} \theta B(v)\, \mathrm{d}x = \oint_{\omega\times\Omega^1} (\theta^0 + \varepsilon\theta^1) v\, \mathrm{d}x^0 \mathrm{d}x^1 + \varepsilon O(\varepsilon).$$

From a priori estimates we prove the approximations and equalities: $T\left(\varepsilon\partial_{x_\alpha}\theta\right) = \partial_{x_\alpha^1}\theta^0 + O_w(\varepsilon)$, $T(\partial_{x_3}\theta) = \partial_{x_3^1}\theta^1 + O_w(\varepsilon)$, $\partial_{x_3^1}\theta^0 = 0$ in $\omega \times \Omega_m^1$, $T(\partial_{x_1}\theta) = \partial_{x_1^1}\theta^1 + O_w(\varepsilon)$, $T(\partial_{x_2}\theta) = \partial_{x_2^0}\theta^0 + \partial_{x_2^1}\theta^1 + O_w(\varepsilon)$, $T(\partial_{x_3}\theta) = \partial_{x_3^1}\theta^1 + O_w(\varepsilon)$, $\nabla_{x^1}\theta^0 = 0$, θ^1 is Ω_f^1-periodic in x_2^1 in $\omega \times \Omega_f^1$ and $T(\partial_{x_\alpha}\theta) = \partial_{x_\alpha^1}\theta^1 + O_w(\varepsilon)$, $T(\varepsilon\partial_{x_3}\theta) = \partial_{x_3^1}\theta + O_w(\varepsilon)$, $\partial_{x_1^1}\theta^0 = \partial_{x_2^1}\theta^0 = 0$ in $\omega \times \Omega_p^1$, and $\alpha \in \{1, 2\}$. The notation $O_w(\varepsilon)$ refers to any weakly vanishing function in the L^2-norm.

3.4 The Homogenized model

The previous approximations and equalities are plugged in the weak formulation (1) which yields, after some steps, the two-scale model of the MMA. Since the matrix of diffusion is Ω^ε-periodic it has the form $\mathbf{a}^\varepsilon = T(\mathbf{a})$ where $\mathbf{a}(x^1)$ is the matrix of diffusion defined in the reference cell Ω^1. The temperature in the frame $\overline{\theta}^0(x^0) = \theta^0$ in $\omega \times \Omega_f^1$ is extended to the whole array $\omega \times \Omega^1$. The differences $\theta_m^0 = \theta^0 - \overline{\theta}^0$ in $\omega \times \Omega_m^1$ and $\theta_p^0 = \theta^0 - \overline{\theta}^0$ in $\omega \times \Omega_p^1$ satisfy the boundary conditions $\theta_m^0 = 0$ on $\partial\Omega_m^1 \cap \partial\Omega_f^1$ and $\theta_p^0 = 0$ on $\partial\Omega_p^1 \cap \partial\Omega_f^1$. The other equations satisfied by θ_m^0 in each mirror are discoupled from the other parts,

$$\begin{cases} -div_{\overline{x}^1}(\mathbf{a}^m \nabla_{\overline{x}^1}\theta_m^0) = r^{m,0} \text{ in } \omega \times \Omega_m^1, \\ (\mathbf{a}^m \nabla_{\overline{x}^1}\theta_m^0)^T \overline{n} = 0 \text{ on } \partial\Omega_m^1 / \partial\Omega_f^1, \end{cases} \tag{2}$$

where $\overline{x}^1 = (x_1^1, x_2^1)$, $\overline{n} = (n_1, n_2)$, $a_{\alpha\beta}^m = a_{\alpha\beta} - a_{3\beta}a_{\alpha 3}/a_{33}$ is the effective thermal conductivity of the mirror and $r^{m,0}$ is the effective internal heat source. In the pillars, $\theta_p^0 = \overline{\theta}^0 \theta_p'$ where the auxiliary function θ_p' is solution to the one-dimensional boundary value problem,

$$\begin{cases} -\partial_{x_3^1}(a^p \partial_{x_3^1}\theta_p') = 0 \text{ in } \Omega_p^1, \\ \theta_p' = 0 \text{ on } \partial\Omega_p^1 \cap \partial\Omega_f^1, \\ \theta_p' = -1 \text{ on } \Gamma^{0,p}, \end{cases} \tag{3}$$

where $a^p = \sum_{i,j=1}^3 L_i^p a_{ij} L_j^p$ and the vector $L^p = -\begin{pmatrix} a_{11} & a_{21} & 0 \\ a_{12} & a_{22} & 0 \\ 0 & 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} a_{31} \\ a_{32} \\ 1 \end{pmatrix}$. The temperature $\overline{\theta}^0$ in the frame solves

$$\begin{cases} -\partial_{x^0}(a^f \partial_{x^0}\overline{\theta}^0) = r^{f,0}, \\ -R_m^f \oint_{\partial\Omega_m^1} (\mathbf{a}^m \nabla_{x^1}\theta_m^0)^T n\, dx^1, \\ -R_m^p \overline{\theta}^0 \oint_{\partial\Omega_p^1} a^p \partial_{x_3^1}\theta_p'\, dx^1 \text{ in } \omega, \\ \overline{\theta}^0 = 0 \text{ on } \partial\omega, \end{cases} \tag{4}$$

with the effective thermal coefficient $a^f = \sum_{i,j=1}^3 (\delta_{i,2} + L_i^f) a_{ij} (\delta_{j,2} + L_j^f)$, $\delta_{i,j}$ the Kronecker delta symbol, and the vector $L^f = \nabla_{x^1} u$. The auxiliary function u is solution to $-\operatorname{div}_{x^1}(\mathbf{a}\nabla_{x^1} u) = -\operatorname{div}(e_2\mathbf{a})$ and Ω_f^1-periodicity conditions where $e_2 = (0, 1, 0)$.

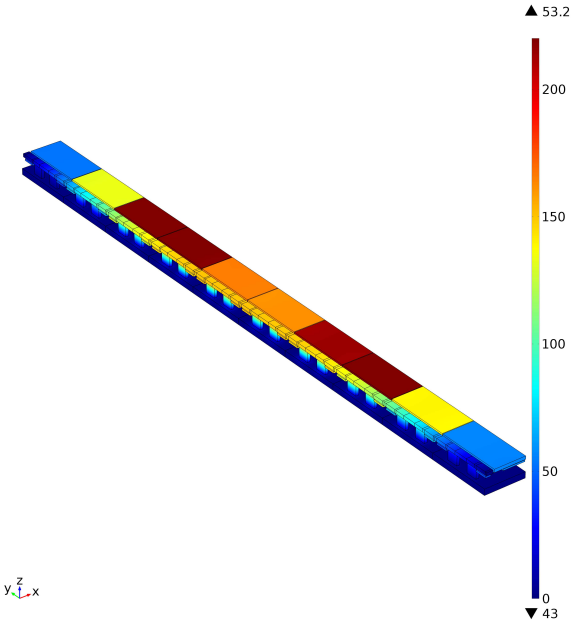

Fig. 7: Result of the two-scale model of heat transfer in a 1x10 MMA simulated in COMSOL.

In the implementation, the temperature θ_m^0 and θ_p' in the mirror and in the pillars are computed by solving Equations (2) and (3). Then, the heat fluxes $\mathbf{a}^m \nabla_{x^1}\theta_m^0$ and $a^p \partial_{x_3^1}\theta_p'$ are used as sources in Equation (4) of the temperature $\overline{\theta}^0$ in the frame. With this method, the microscopic problems are solved cell by cell which reduces dramatically the memory requirement. Further reduction are possible e.g. by solving cell problems for a small family of source terms and operate by interpolation to deduce all the other solutions.

The homogenized model has been implemented for a 1×10 mirror array with an heat source oscillating along the x_2-direction as a sine function. The distribution of temperature is reported in Figures 7 and 8. In terms of performances, the estimate of the gain is not yet precise, but it is more than 20 twenty times for this case and increases fast when the array size increases.

4. `MEMSALab`

`MEMSALab` design methodology consists of three aspects. The first one is to establish a general mathematical theoretical framework for the multi-scale model derivations

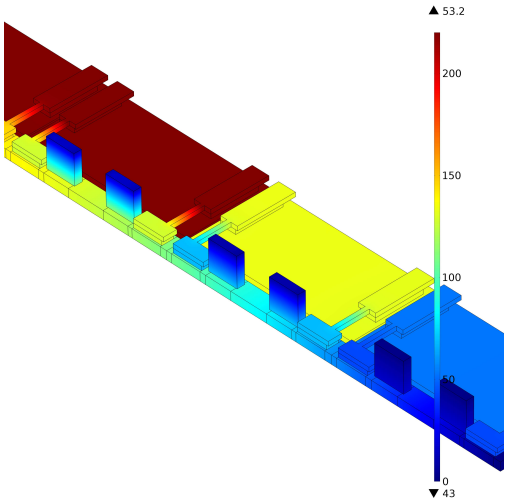

Fig. 8: Zoom on the result of the two-scale model of heat transfer in a 1x10 MMA simulated in COMSOL. The continuity of the temperature from the frame to the mirror is through the suspending beams.

so that to be able to implement the derivation of two-scale models as the model above. In this unified framework, the derivations – in a setting of different physical features and geometries – could be different in details, but the *skeleton* of the derivations remains "the same". The second aspect is the design and the implementation of a symbolic transformation tool to implement the multi-scale model derivations. The designer formulates the mathematical properties as well the elementary derivations (i.e. the skeleton proofs) with this tool. Since only the elementary derivations of the general framework are implemented in MEMSALab, the third aspect of our design methodology consists in developing an *extension mechanism* allowing the combination of the already implemented derivations. This is a systematic way to build complex models by reusing and combining already existing proofs.

4.1 Grammar of MEMSALab

In this section we describe the format in which the problem is written in MEMSALab, as well as the translation between the conventional mathematical format and the MEMSALab format. The format of a mathematical expression in MEMSALab is given by means of the formal *grammar*,

$$
\begin{aligned}
\mathcal{E} ::= {} & \texttt{Plus}(\mathcal{E},\mathcal{E}) \mid \texttt{Mult}(\mathcal{E},\mathcal{E}) \mid \\
& \texttt{Minus}(\mathcal{E}) \mid \texttt{Inverse}(\mathcal{E}) \mid \texttt{Power}(\mathcal{E},\mathcal{E}) \mid \mathcal{F} \\
\mathcal{F} ::= {} & \texttt{Fun}(f,[\mathcal{I};\ldots;\mathcal{I}],[\mathcal{V};\ldots;\mathcal{V}],[(\mathcal{R},\mathcal{E});\ldots;(\mathcal{R},\mathcal{E})],K) \mid \\
& \texttt{Oper}(A,[\mathcal{I};\ldots;\mathcal{I}],[\mathcal{E};\ldots;\mathcal{E}],[\mathcal{V};\ldots;\mathcal{V}],[\mathcal{V};\ldots;\mathcal{V}], \\
& \qquad [d;\ldots;d]) \mid \\
& \mathcal{V} \mid \texttt{MathCst}(d) \mid \texttt{Nil}, \\
\mathcal{V} ::= {} & \texttt{MathVar}(x,[\mathcal{I};\ldots;\mathcal{I}],\mathcal{R}), \\
\mathcal{R} ::= {} & \texttt{Reg}(\Omega,[\mathcal{I};\ldots;\mathcal{I}],[d,\ldots,d],[\mathcal{R};\ldots;\mathcal{R}],\mathcal{R},\mathcal{E}) \mid \texttt{Nil}, \\
\mathcal{I} ::= {} & \texttt{Ind}(i,[d,\ldots,d])
\end{aligned}
$$

It describes mathematical expressions built up by the arithmetic operations "+" (`Plus`), "·" (`Prod`), etc as well as the mathematical function constructor `Fun` and the operator constructor `Oper`. The latter allows one to build expressions for mathematical operators such as the integration operator $\int$, the derivative operator ∂, the summation operator $\sum$, the multi-scale operators T, B, etc. Besides, a mathematical expression can contain mathematical variables (`MathVar`), regions (`Reg`) and discrete variables (`Ind`).

For the sake of readability, instead of writing the full expression that follows the MEMSALab grammar above, we shall sometimes omit some of its subexpressions and use shortcut-like expressions. For instance, we shall simply write x_i instead of $\texttt{MathVar}(x,[\texttt{Ind}(i,[n])],\texttt{Reg}(\Omega,\ldots))$ and omit the domain $\texttt{Reg}(\Omega,\ldots)$ and the dimension n. We shall also write $\partial_x v(x)$ instead of $\texttt{Oper}(\texttt{Partial},\texttt{Fun}(v,\ldots),\ldots)$. Besides, these expressions can contain *rewriting variables*. A rewriting variable, denoted by $x\_, y\_$, etc, is a particular term that can match any expression. For instance, the shortcut expression $\Omega\_$, which abbreviates the expression $\texttt{Reg}(\Omega\_,\ldots)$, stands for any domain. However, it is worth mentioning that the notion of mathematical variable (e.g. x) should not be confused with the one of rewriting variable (e.g. $x\_$). We shall sometimes write and depict lists in the prefix notation using the constructor `list` and `nil` (empty list). For instance, if e_1 and e_2 are two expressions, we shall write $\texttt{list}(e_1,\texttt{list}(e_2,\texttt{nil}))$ instead of $[e_1;e_2]$. The symbol `Nil` in the grammar above represents an "empty expression".

4.2 The user specification language

We designed a specification language in which the user formulates the proofs including the proof blocks, the lemmas and the mathematical expressions. The following example shows a formula and its related specification in the user language.

Example 1 *We give an example of the Green rule*

$$
\int_{\Omega\_} u\_\frac{dv\_}{dx\_}\,\mathrm{d}x\_ \rightarrow \int_{\Gamma\_} tr\,(u\_)\,tr\,(v\_)\,n\_\,\mathrm{d}s\,(x\_) - \int_{\Omega\_} v\_\frac{du\_}{dx\_}\,\mathrm{d}x\_ \tag{5}
$$

written in the specification language as,

```
Model "Green formula for one-dimensional domain
and scalar functions" :
Variable
 x_ : [] [Omega_]
Function
 u_ : [] [xx_] [] K_
 v_ : [] [xx_] [] K_
 n : [] [XG_] [] K_
Operator
 Trace      : [] [Fun_] [X_] [XG_] []
Rule
 "Green" : ∫v_•∂u_/∂x_ dx_
 →∫Trace(u_)•Trace(v_)•n dXG_
 -∫u_∂v_/∂x_ dx_
```

This specification contains four declaration blocks for the declaration of variables, functions, operators and transforming rules, where "· → ·" denotes a transformation rule, "[ ]" represents the empty parameter, Trace is the trace operator that restricts a function to the boundary of its domain and n is the normal vector. The line "`u_: [] [xx_] [] K_`*" is a declaration of a mathematical function of arbitrary name, and whose first and third argument*

are empty lists while its second and fourth argument are arbitrary. Notice also that the derivative operator ∂ is already predefined, and the user does not need to declare it.

4.3 Principle of model derivation by extension-combination in MEMSALab.

In order to carry on a systematic approach for the derivation of multi-scale models that allows to cover a variety of physical features and geometries, we develop the by extension-combination method. Figure 9 illustrates the idea behind it through an example.

Before that, and in order to understand the transformations that operate on the proofs, it is helpful to see the mathematical objects (proofs, blocks, lemmas, mathematical expressions, etc) as *trees*.

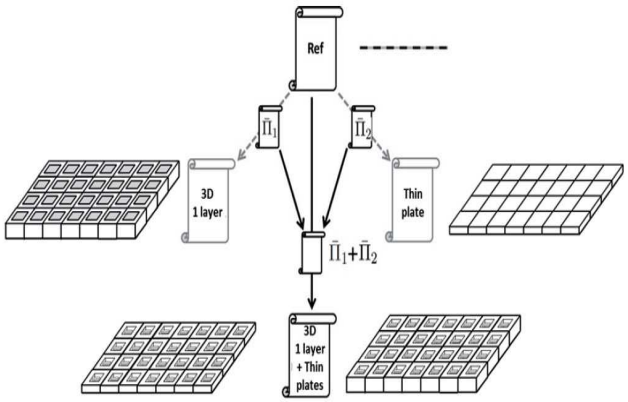

Fig. 9: By-extension-combination principle illustrated on a reference proof: an extension of the reference proof (top) to the 3-dimensional setting (left) and to the thinness setting (right). The combination of these two extensions is depicted on the bottom.

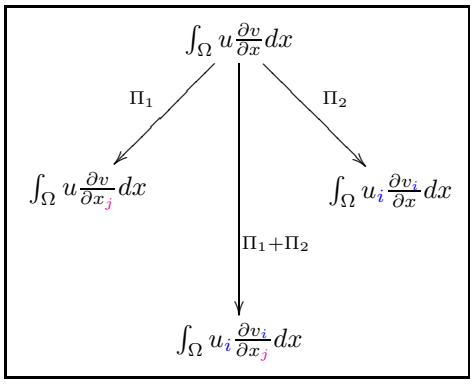

Fig. 10: An example of the by-extension-combination method applied to the mathematical expression $\int_\Omega u\frac{\partial v}{\partial x}dx$ that corresponds to the left hand side of Green formula 5, where Π_1 stands for the extension operator of the *multi-dimension setting*, Π_2 stands for the extension operator to the *vector-valued setting*, and $\Pi_1+\Pi_2$ stands for the combination of Π_1 and Π_2.

The by extension-combination relies on three key principles.

Firstly, we introduce *reference model*, also called *skeleton model*, together with its derivation. This derivation is called the *reference proof*, it is depicted on the top of Fig. 9. The reference model is the periodic homogenization model of a scalar second-order elliptic equation posed in a one-dimensional domain with Dirichlet boundary conditions. Its derivation is based on the derivation approach of [11]. Although the reference model covers a very simple case, its proof is expressed in a sufficiently general way. A number of basic algebraic *properties* are formulated as *transformation rules*, they are considered as the building blocks of the proofs. The full derivation of the model is formulated as a sequence of applications of these rules. The proof of some properties is also performed by a sequence of applications of mathematical rules when the others are admitted e.g. the Green rule.

Then, an *elementary extension* is obtained by an application of an elementary transformation, called also an *extension operator*, to the reference proof. Each extension operator covers a particular feature. In Fig. 9 the extension operators are $\overline{\Pi}_1$ and $\overline{\Pi}_2$. They respectively cover the extension to the 3-D setting and the thinness setting. More generally, many extension operators can be applied simultaneously to the reference proof, where each operator covers a distinct feature. We notice that, in practice, when a single feature is taken into account, only a small change occurs in a relatively long proof. In other words, while considering an elementary extension, most of the existing rules could be reused by operating a small change on them, and, on the other hand, only a small number of new rules has to be manually introduced. From this empirical observation, it follows that the extension of the existing proofs to cover a new feature can be generated almost automatically.

Finally, we make possible the combination of two extension operators to produce a new extension operator that takes into account the features covered by each initial extension operator. In the example of Fig. 9, the combination of the extension operators $\overline{\Pi}_1$ and $\overline{\Pi}_2$ is the extension operator $\overline{\Pi}_1+\overline{\Pi}_2$.

By iterating this process, many extension operators can be combined together giving rise to complex extensions that cover many features. Fig. 10 shows how the extension

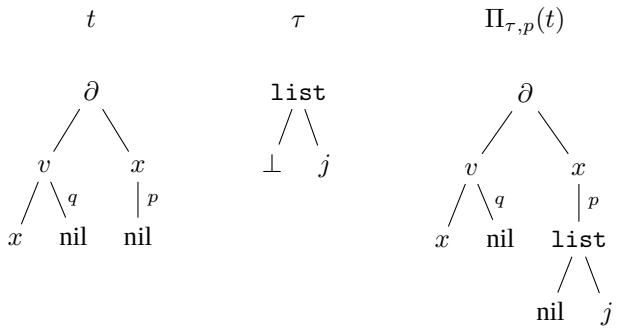

Fig. 11: Application of the extension operator $\Pi_{\tau,p}$ (with the extension constructor τ) to the term $t=\partial_x v(x)$ at the position p, yielding the term $\partial_{x_j} v(x)$.

operators and their combination operate on the mathematical expression $\int_\Omega u\frac{\partial v}{\partial x}dx$, which is the left hand side of Green formula 5.

Technically speaking, the by-extension-combination relies on three ideas:

Firstly, an extension is built up on elementary constructors called *extension constructors*. An extension constructor describes the extra term that one wants to add to a given term at a given position. For instance, the extension constructor $\tau = list(\perp, j)$ depicted in Fig. 11 captures the idea that our extension would add a discrete variable to an expression.

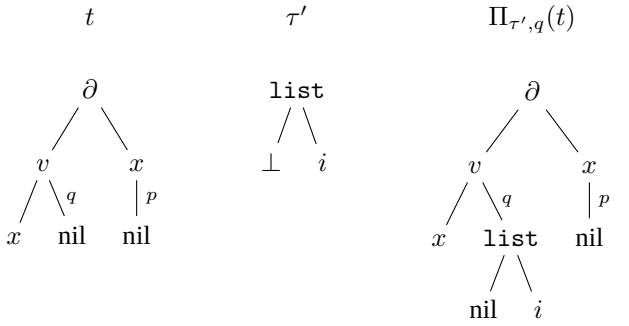

Fig. 12: Application of the extension operator $\Pi_{\tau',q}$ (with the extension constructor τ') to the term $t = \partial_x v(x)$ at the position q, yielding the term $\partial_x v_i(x)$.

The related extension operator is $\Pi_{\tau,p}$, its application to the term $t = \partial_x v(x)$ at the variable x (the parameter of the derivative operator ∂) yields the term $\Pi_{\tau,p}(t) = \partial_{x_j} v(x)$. Similarly, Fig. 12 illustrates the extension operator $\Pi_{\tau',q}$ and its application to the term $t = \partial_x v(x)$ at the function v which yields the term $\Pi_{\tau',q}(t) = \partial_x v_i(x)$

The general schema of the application of an extension is depicted in Fig. 13.

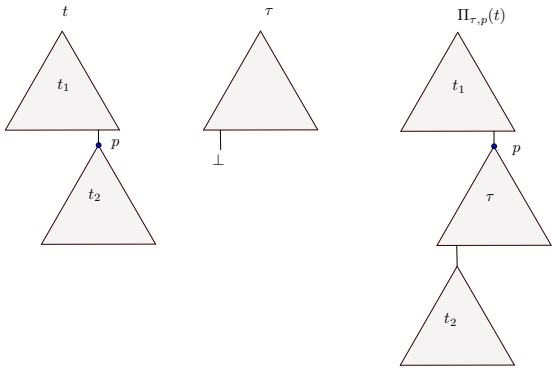

Fig. 13: Schematic diagram of the application of an extension operator $\Pi_{\tau,p}$ (with an extension constructor τ) to a term t at the positon p.

Secondly, the extension operators can be combined. Fig. 14 shows the combination of the two extension constructors $\Pi_{\tau,p}$ and $\Pi_{\tau',q}$.

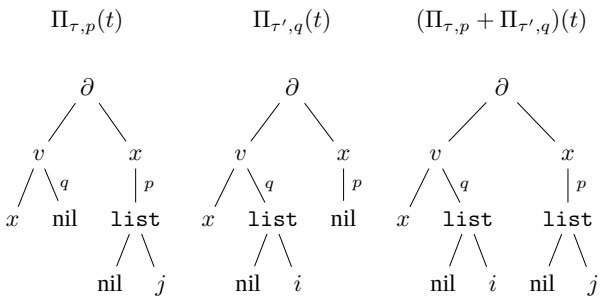

Fig. 14: The extension operator $\Pi_{\tau,q} + \Pi_{\tau',q}$ which is the combination of the two extension operators $\Pi_{\tau,p}$ and $\Pi_{\tau',q}$, and its application to the term t.

We emphasize that the structure of an extension operator is more complex, it may use both positions and nested searching *patterns* which are expressions with rewriting variables. A pattern allows one to locate the subexpression on which the extension constructor is applied.

Even if we do not define precisely our concept of extensions and combination, the following statement provides an idea of the main result.

Proposition 2 *The class of extension operators are closed under the combination operation $+$, that is, if Π_1 and Π_2 are extension operators, then their combination $\Pi_1 + \Pi_2$ is an extension operator too.*

The use of the mechanism of the combination of several existing elementary extensions instead of the development of new extension transformations has the advantage of reducing the development effort by avoiding doing complex changes manually. Thus, the by extension-combination method is a reasonable one since it facilitates the implementation of the two-scale methods. Besides, the by extension-combination method goes beyond the limitations of the methods of combination of extensions by sequential composition of [4]. More precisely, the combination of two extension operators Π_1 and Π_2 by the sequential composition of [4] consists of the application of Π_1 followed by the application of Π_2 which does not work if there are *conflicts* between Π_1 and Π_2. Such conflicts can be avoided by the new method of by extension-combination. More precisely, if the extension operator Π_2 uses a searching pattern $\mathcal{U}_2$, then the application of Π_1 to an input expression may change it so that the application of Π_2 to the resulting expression is no longer possible since the searching pattern $\mathcal{U}_2$ was intended to be applied to the input expression and not to the resulting one. That is why we avoid the sequential composition method and employ the by-extension-combination method which morally tries to consider the combination of two extension operators simultaneously and to solve the possible conflicts.

4.4 Implementation

Unlike [4] where the implementation of the extensions as sequential compositions was done in Maple, now the implementation of the extensions and their combination is done in Ocaml. In fact, the advantages of Ocaml are many: it is a free open source language, it allows fast prototyping, it supports high order functions, and it is equipped with an

advanced type system which reduce dramatically the programming errors. As an elementary example, we discuss the extension of the Green rule in Eq. (5) to both the multi-dimensional setting (6) and to the multi-valued setting (7) and the combination of these two extensions that yields (8). Precisely, the formula (5) is trivially extended for any domain $\Omega \subset \mathbb{R}^m$ with $m \in \mathbb{N}^*$,

$$\int_\Omega u \frac{\partial v}{\partial x_j} \, \mathrm{d}\mathbf{x} = \int_\Gamma \mathrm{tr}\,(u)\,\mathrm{tr}\,(v)\, n_j \, \mathrm{d}s - \int_\Omega v \frac{\partial u}{\partial x_j} \, \mathrm{d}\mathbf{x} \quad (6)$$

for any $j \in \{1, ..., m\}$. Another trivial extension is for any vector valued functions $\mathbf{u} = (u_1, ..., u_n)$, $\mathbf{v} = (v_1, ..., v_n)$ defined on Ω with $n \in \mathbb{N}$,

$$\int_\Omega u_i \frac{\partial v_i}{\partial x} \, \mathrm{d}\mathbf{x} = \int_\Gamma \mathrm{tr}\,(u_i)\,\mathrm{tr}\,(v_i)\, n_j \, \mathrm{d}s - \int_\Omega v_i \frac{\partial u_i}{\partial x} \, \mathrm{d}x \quad (7)$$

for any $i \in \{1, ..., n\}$. The combination of these two extensions states for $\Omega \subset \mathbb{R}^m$ with $m \in \mathbb{N}^*$ and $\mathbf{u} = (u_1, ..., u_n)$, $\mathbf{v} = (v_1, ..., v_n)$ defined on Ω with $n \in \mathbb{N}$

$$\int_\Omega u_i \frac{\partial v_i}{\partial x_j} \, \mathrm{d}\mathbf{x} = \int_\Gamma \mathrm{tr}\,(u_i)\,\mathrm{tr}\,(v_i)\, n_j \, \mathrm{d}s - \int_\Omega v_i \frac{\partial u_i}{\partial x_j} \, \mathrm{d}\mathbf{x} \quad (8)$$

for any $i \in \{1, ..., n\}$ and $j \in \{1, ..., m\}$. The definition of the extensions in Ocaml follows those expressed in the above mathematical formulae. To extend Equation (5) into Equation (6), we change the dimension of the domain from 1 into m and then we add index j to Equation (5)'s variable. Similarly, we add index i to the functions of Equation (6) to get Equation (7). The two operators of extensions can be expressed in the user language,

```
Extension "multi-dimensional domains" :
Index
 j : [1,JD_] ∀
Statement
 "S1": ∂u__/∂{Var x__ [] []} ⇉ ∂u__/∂{x__ [j] []}
 "S2": {Fun n [] [] []} ⇉ {Fun n [j] [] []}
```

and

```
Extension "vector valued functions" :
Index
 i : [1,ID_] ∀
Statement
 "S1": {Fun : u__ [][][]}∂{Fun : v__ [][][]}/∂x__ ⇉
       {u__ [i] [] []}∂{v__ [i] [] []}/∂x__
```

The first extension specification introduces the discrete index j and adds it to any variable $x\_$ which is the derivative variable of the operator ∂. The second extension specification introduces the discrete index i and adds it to the set of the discrete variables of the function v. This is expressed by the second-order rewriting rules $S1$ and $S2$, where second-order rules are morally like first-order ones except that they transform first-order rules and expressions while first-order rules transform only expressions. Besides, we make a distinct notation for first-order variables (e.g. $x\_$) and second-order variables (e.g. $x\_\_$).

Conclusions This paper presents recent advances in the development of `MEMSALab` including: the key points of the construction and implementation of an asymptotic model of the stationary heat equation in a micro-mirror array, the extension-combination method and a user language for the specification of proofs, extension operators and their combination. The next stages of development are through the integration of these three aspects and the extension of the modelling to take into account electrostatics and mechanical strains. The major expected advance is in the development of a library of extensions and combinations for generating a family of asymptotic models of MOEMS arrays for astrophysics. They will be validated with existing devices and afterwards used to optimize their designs. It remains also to implement an interpreter that automatically generates the extension operators, written as OCaml types, out of user extension specifications.

References

1. M.D. Canonica, F. Zamkotsian, P. Lanzoni, W. Noell, and N. De Rooij. The two-dimensional array of 2048 tilting micromirrors for astronomical spectroscopy. *Journal of Micromechanics and Microengineering*, 23(5):055009, 2013.
2. M. Lenczner and R.C. Smith. A two-scale model for an array of afm's cantilever in the static case. *Mathematical and computer modelling*, 46(5):776–805, 2007.
3. W. Belkhir, A. Giorgetti, and M. Lenczner. A symbolic transformation language and its application to a multiscale method. *Journal of Symbolic Computation*, 65:49–78, 2014.
4. B. Yang, W. Belkhir, and M. Lenczner. Computer-aided derivation of multiscale models: A rewriting framework. *International Journal for Multiscale Computational Engineering*, 12(2), 2014.
5. B. Yang, W. Belkhir, M. Lenczner, and N. Ratier. A multiscale model derivation and simulation tool for mems arrays. In *Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems (EuroSimE), 2013 14th International Conference on*, pages 1–8. IEEE, 2013.
6. M.D. Canonica. *Large Micromirror Array Based on a Scalable Technology for Astronomical Instrumentation*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2012.
7. D. Cioranescu, A. Damlamian, and G. Griso. Periodic unfolding and homogenization. *C. R. Math. Acad. Sci. Paris*, 335(1):99–104, 2002.
8. J. Casado-Díaz. Two-scale convergence for nonlinear Dirichlet problems in perforated domains. *Proc. Roy. Soc. Edinburgh Sect. A*, 130(2):249–276, 2000.
9. Todd Arbogast, Jim Douglas, Jr., and Ulrich Hornung. Derivation of the double porosity model of

single phase flow via homogenization theory. *SIAM J. Math. Anal.*, 21:823–836, May 1990.

10. Alain Bourgeat, Stephan Luckhaus, and Andro Mikelic. Convergence of the homogenization process for a double-porosity model of immiscible two-phase flow. *SIAM J. Math. Anal.*, 27:1520–1543, November 1996.

11. M. Lenczner and R. C. Smith. A two-scale model for an array of AFM's cantilever in the static case. *Mathematical and Computer Modelling*, 46(5-6):776–805, 2007.