Journal of Symbolic Computation 65 (2014) 49–78

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

# A symbolic transformation language and its application to a multiscale method ☆

Walid Belkhir [a], Alain Giorgetti [a,b], Michel Lenczner [c]

[a] *INRIA Nancy – Grand Est, CASSIS project, 54600 Villers-lès-Nancy, France*
[b] *FEMTO-ST institute, Département d'Informatique des Systèmes Complexes, University of Franche-Comté, 16 route de Gray, 25030 Besançon Cedex, France*
[c] *FEMTO-ST institute, Département Temps-Fréquence, University of Technology of Belfort–Montbéliard, 26 chemin de l'Epitaphe, 25030 Besançon Cedex, France*

ARTICLE INFO

*Article history:*
Received 16 November 2011
Accepted 31 October 2013
Available online 30 January 2014

*Keywords:*
Symbolic transformation
Term rewriting
Strategies
Multiscale modeling

ABSTRACT

The context of this work is the design of a software, called *MEMSALab*, dedicated to the automatic derivation of multiscale models of arrays of micro- and nanosystems. In this domain a model is a partial differential equation. Multiscale methods approximate it by another partial differential equation which can be numerically simulated in a reasonable time. The challenge consists in taking into account a wide range of geometries combining thin and periodic structures with the possibility of multiple nested scales.

In this paper we present a transformation language that will make the development of MEMSALab more feasible. It is proposed as a Maple™ package for rule-based programming, rewriting strategies and their combination with standard Maple™ code. We illustrate the practical interest of this language by using it to encode two examples of multiscale derivations, namely the two-scale limit of the derivative operator and the two-scale model of the stationary heat equation.



## 1. Introduction

The context of this work is the design of microsystem array architectures, including microcantilevers, micromirrors, droplet ejectors, micromembranes, microresistors, etc., to cite only a few.

☆ This work has been partially funded by the Labex ACTION (ANR-11-LABX-01-01), the INTERREG IV project OSCAR and the PPF MIDi.

*E-mail addresses:* walid.belkhir@inria.fr (W. Belkhir), alain.giorgetti@femto-st.fr (A. Giorgetti), michel.lenczner@utbm.fr (M. Lenczner).

0747-7171/$ – see front matter 
http://dx.doi.org/10.1016/j.jsc.2014.01.004

A model for such arrays is a Partial Differential Equation (PDE). The numerical simulation of whole arrays based on classical methods like the Finite Element Method (FEM) is prohibitive for today's computers (at least in a time compatible with the time scale of a designer). The calculation of a reasonably complex cell of a three-dimensional microsystem requires at least 1000 degrees of freedom which lead to at least 10 000 000 degrees of freedom for a $100 \times 100$ array. Fortunately there is a solution consisting in approximating the model by a multiscale method. The resulting approximated model is again a PDE. It can be rigorously derived from the exact one through a sequence of mathematical transformations, but these transformations differ for each case.

We are currently developing a software, called MEMSALab, for "MEMS Arrays Laboratory", dedicated to multiscale and multiphysics modeling of arrays of micro- and nanosystems. Unlike traditional software that is based on models built once and for all, MEMSALab is a software that constructs models. The challenge consists in taking into account a wide range of geometries combining thin and periodic structures with the possibility of multiple nested scales. One should also consider PDEs representing multiphysics systems with high contrast in equation coefficients.

Simulation software available in the market offers specialized tools for large arrays of micro- and nanosystems, but the construction of new models raises many problems. Firstly the time required for a new design varies from some weeks for a specialist to several months for a beginner. Secondly the mathematical machinery is too sophisticated to be manually applied to complex systems. Finally the resulting models require specific numerical simulation methods, that have to be implemented case by case.

The software MEMSALab we design aims at addressing these problems. It is based on multiscale models, especially on those derived by asymptotic methods. Such asymptotic models are derived from a system of PDEs when taking into account that at least one parameter is very small, such as thickness for a thin structure or the small ratio of a cell size to the global size for a periodic structure. The resulting models are other systems of PDEs, obtained by taking the mathematical limits of the nominal models, in a well-suited sense, when the small parameters tend toward zero. This approach provides a reasonably good approximation. It also offers the advantages and factors of reliability to be rigorous and systematic. The resulting PDEs can be implemented in a simulation software such as the finite element based simulator COMSOL (Multiphysics Finite Element Analysis Software, official site http://www.comsol.com), and simulations turn to be fast as needed.

The literature in this field is vast and a large number of techniques have been developed for a large variety of geometric features and physical phenomena. However, none of them have been implemented in a systematical approach to render it available to engineers as a design tool. In fact, each published paper focuses on a special case regarding geometry or physics, and very few works are considering a general picture. By contrast our software will treat the problem of systematic implementation of asymptotic methods by implementing the construction of models rather than the models themselves. This approach will cover many situations from a small number of bricks. It combines mathematical and computer science tools. The mathematical tool is the two-scale transform originally introduced in Lenczner (1997), Casado-Díaz (2000), Cioranescu et al. (2002) to model periodic and thin structures, and also referred as the unfolding method. We have extended its domain of application to cover in the same time homogenization of periodic media, see for instance Bensoussan et al. (1978), and methods of asymptotic analysis for thin domains, see Ciarlet (1988). The computer science tools include term rewriting, $\lambda$-calculus, and type systems (Cirstea and Kirchner, 2001; Marin and Piroi, 2004; Cirstea et al., 2001; Geuvers, 2009). The software is written in the symbolic computation language Maple™.

Compared to other techniques, our multiscale method requires more modular calculations, avoids any non-constructive proof and intensively relies on equational reasoning. The classical way to automate equational reasoning is to consider mathematical equalities as rewrite rules. The rewrite rule $t \to u$ orients the equality $t = u$ from left to right and states that every occurrence of an instance of $t$ can be replaced with the corresponding instance of $u$. Consequently symbolic computation with equalities is reduced to a series of term rewritings. Algebraic computation and term rewriting are two research domains with strong similarities. Both are separately well-studied but there are only few works about the combination of algebraic computation and term rewriting (Fèvre and Wang, 1998; Bündgen, 1995). Term rewriting provides a theoretical and computational framework which is very