

# Michele Scuttari

michele.scuttari@polimi.it | Mantova, IT | [Website](#) | [LinkedIn](#) | [GitHub](#)

---

## WORK EXPERIENCE

---

### PhD candidate

Nov. 2022 – Present

*Politecnico di Milano. Defending in February - April 2026*

*Milan, IT*

- Lead and main developer of the **MARCO compiler**: <https://github.com/marco-compiler/marco>
  - Designed and implemented MLIR dialects, R-Tree based index sets, compressed hyper-dimensional matrices, array-aware causalization, runtime environment with multi-core equation dispatching, and other several optimizations. All the data structures and algorithms have been developed in an array-oriented manner, giving MARCO beyond-state-of-the-art capabilities for large-scale models based on differential-algebraic equations: compilation times exhibit an  $O(1)$  scaling with respect to the size of arrays, and simulation times are up to two order of magnitude lower.
  - The project will be presented at the LLVM Developers' Meeting in October 2025.
- Languages and technologies: **C++ 17, LLVM, MLIR, Modelica, Git, Docker, Jenkins**

### LLVM Contributor

August. 2022 – Present

- Upstreamed improvements to the MLIR project.
- Accepted contributions: folding of chains of unrealized casts, addressing of the quadratic scaling of the bufferization infrastructure, symbols caching in Arith & Func To LLVM dialect conversion.
- Languages and technologies: **C++ 17, MLIR, Git**

### Teaching Assistant

Sept. 2022 – Sept. 2025

*Politecnico di Milano*

*Milan, IT*

- Teaching assistant for the “Informatics for aerospace engineers” course
  - Exercise, laboratory, and tutoring sessions
  - Topics: C language
- Teaching assistant for the “Formal Languages and Compilers” course
  - Exercise, laboratory, and tutoring sessions
  - Topics: Grammars, Automata, Parsing, Hands-on activities on the teaching-oriented compiler ACSE
- Languages and technologies: **C**

### Research Fellow

Nov. 2021 – Oct. 2022

*Politecnico di Milano*

*Milan, IT*

- Post-graduation assignee
- Development of the MARCO compiler: full embracement of the MLIR ecosystem
- Languages and technologies: **C++ 17, LLVM, MLIR, Modelica, Git**

## EDUCATION

---

### Politecnico di Milano

Oct. 2021

*MSc Computer Science and Engineering*

*Milan, IT*

- Grade: **109/110**, EQF Level 7
- Thesis: “Design and implementation of a Modelica compiler with MLIR and LLVM”

## PUBLICATIONS

---

### **“An Introduction to Languages and Tools for Modeling and Simulation of Cyber-Physical Systems”** Feb. 2025

- Submitted to ACM Computing Surveys
- Simulations play a pivotal role in modern science and engineering, enabling researchers and practitioners to design, analyze, and optimize complex systems beyond the constraints of physical experimentation. Their broad applicability reflects in the large amount of available languages and tools that have been developed throughout the last 70 years. This sheer variety of solutions poses significant challenges when choosing the possibly best tool for the domain interest, particularly for those seeking an initial understanding of the simulation environments. This paper provides a surface-level exploration of this diverse landscape, focusing on cyber-physical systems with continuous, discrete, or hybrid dynamics. We discuss the mathematical foundations underpinning the simulation of such systems, trace the evolution of some of the software solutions involved in their modeling, and analyze the most relevant abstractions that are currently in use. Additionally, we discuss recent trends in modeling languages with respect to their adaptation to modern, highly parallel hardware architectures. These advancements are increasingly critical in the context of high-performance computing, where scalability and efficiency are essential for simulating large-scale systems.

### **“MLIR Dialects for DAE-based Modeling Languages”** Jan. 2025

- Submitted to ACM Transactions on Architecture and Code Optimization
- Models based on Differential-Algebraic Equations (DAE) are ubiquitous across engineering fields. Specialized compilers and languages are key enablers to automate the numerical simulation of these models, which are experiencing a steady increase in their size. In this paper, we discuss the uniqueness of this class of compilers, we propose a modern compiler design based on MLIR (Multi-Level Intermediate Representation), and we showcase its implementation – named MARCO – targeting Base Modelica, a simple yet industrially-relevant DAE-based language. The proposed design allows us to overcome the structural limitations of other modeling compilers. Our solution reduces the compilation time and the runtime complexities. We test our approach using an arbitrarily scalable thermal model for integrated circuits implementing the finite volume spatial discretization method. On this benchmark, MARCO achieves constant-time and constant-size compilation with respect to the number of finite volumes, together with an 8× to 34× speedup in simulation time when compared with the most prominent open-source competitor.

### **“MARCO: An Experimental High-Performance Compiler for Large-Scale Modelica Models”** Dec. 2023

- <https://doi.org/10.3384/ecp204>
- This paper introduces MARCO, a research compiler aimed at the efficient generation of efficient simulation code from a large-scale Modelica model. MARCO’s design goals, requirements, and specifications are discussed in the paper, as well as the software architecture, the current development status, and a future development roadmap. The results of two test cases demonstrate MARCO’s capability to handle non-trivial Modelica models with over 10 million equations very efficiently.

### **“Clever DAE: Compiler Optimizations for Digital Twins at Scale”** Aug. 2023

- <https://doi.org/10.1145/3587135.3589945>
- Modeling and simulation are fundamental activities in engineering to facilitate prototyping, verification and maintenance. Declarative modeling languages allow to simulate physical phenomena by expressing them in terms of Differential and Algebraic Equations (DAE) systems. In this paper, we focus on the problem of generating code for performing the numerical integration of the model equations, and in particular on the overhead introduced by external numerical solver libraries. We propose a novel methodology for minimizing the amount of equations which require to be solved through an external solver library, together with the number of computations that are required to compute the Jacobian matrix of the system. Through a prototype LLVM-based compiler, we demonstrate how this approach achieves a linear speed-up in simulation time with respect to the baseline.

## “Efficient memory management for Modelica simulations”

Jun. 2022

- <https://doi.org/10.4230/OASICS.PARMA-DITAM.2022.7>
- The paper explains how to improve the memory management of Modelica simulations. One aspect regards the reduction of the heap memory usage, which is obtained by modifying functions whose resulting arrays could instead be allocated on the stack by the caller. The other aspect regards the possibility of avoiding garbage collection altogether by performing all memory lifetime tracking statically. Once implemented in a prototype Modelica compiler, these techniques show an improvement of the memory management overhead of over 10 times compared to a garbage collected solution, and an improvement of 56 times compared to the production-grade compiler OpenModelica.

## SKILLS & INTERESTS

---

- **Skills:** Software Engineering, End-To-End System Architecture
- **Interests:** Homelabs (Proxmox, TrueNAS), Linux, Automotive & Aerospace sectors
- **Languages:** native Italian, fluent English
- **Driving License:** B (Italian)