



MDMC

Master in Data Management
and Curation

MASTER IN DATA MANAGEMENT AND
CURATION

FAIR Data Management of the
Results from the MulSKIPS
Atomistic Simulation
Environment for PVD, CVD, and
Laser Annealing

Supervisor(s):

Dott. Ioannis Deretzis

Candidate:

Filippo RUBERTO

2024–2025





Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA

This Pilot training activity has been funded by the European Union – NextGenerationEU within the project PNRR "PRP@CERIC" IR00000028 and «NFFA-DI" IR00000015 - Missione 4, "Istruzione e Ricerca" – Componente 2, "Dalla ricerca all'impresa" – Linea di investimento 3.1, "Fondo per la realizzazione di un sistema integrato di infrastrutture di ricerca e innovazione" – Azione 3.1.1, "Creazione di nuove IR o potenziamento di quelle esistenti che concorrono agli obiettivi di Eccellenza Scientifica di Horizon Europe e costituzione di reti".

The supporting projects:

- [NFFA-DI : Nano Foundries Fine Analysis - Digital Infrastructure](#)
- [PRP@CERIC : Pathogen Readiness Platform for Ceric - Eric Upgrade](#)



nffa-di



prp

Author's Declaration

I, Filippo Ruberto, declare that this thesis entitled, 'Development of NEXUS Schemes for MulSKIPS Data Management' and the work presented therein are my own.

I certify that:

- This work was performed wholly or principally during MDMC internship in CNR-IMM Catania.
- If any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have relied on the published work of others, this is always clearly attributed.
- Where the thesis is based on work I have done in collaboration with others, I have made clear exactly what was done by others and what I contributed.

- With respect to AI technologies:

I acknowledge the use of OpenAI ChatGPT (<https://openai.com/chatgpt>) to support the drafting of the thesis structure, to assist in improving the English writing style, and to provide guidance during the development of some Python scripts used for data analysis and FAIR data management. All scientific analyses, simulation results, and final implementations are my own original work.

- Where I have quoted from the work of others, the source is always cited. Except for such quotations, this thesis is entirely my own work.
- I have acknowledged all major sources of help.

Signed:



Date:

30.04.2025

“This page intentionally left blank, even though it’s an oxymoron”

Filippo Ruberto

Acknowledgments

First and foremost, I would like to express my deepest gratitude to Dr. Ioannis Deretzis and Dr. Gaetano Calogero for their invaluable support, patient guidance, and constant encouragement throughout this project. Their expertise, availability, and insightful advice have been fundamental to the development and successful completion of this work.

I am also sincerely thankful to Dr. Antonino La Magna for his supervision and inspiring scientific discussions, which provided important direction and motivation during the entire research process.

Special thanks go to Dr. Matteo Biagetti and Dr. Matteo Rodani for their coordination efforts within the MDMC Master program and for their valuable advice.

I warmly acknowledge the researchers and staff at CNR-IMM Catania, particularly the NFFA-DI project team, for creating an excellent and stimulating research environment.

I am also grateful for the collaboration and discussions with all members of the MulSKIPS development group, whose efforts laid the foundation for much of the work presented here.

Moreover, I thank the FAIRmat consortium for providing essential resources and guidelines that have significantly contributed to the FAIR data management part of this thesis.

A special acknowledgment goes to Lucia Li Donni for her authentic and generous help during the project.

Finally, I wish to express my deepest appreciation to my family and friends for their unwavering support, patience, and encouragement throughout my studies.

Abstract

This thesis presents the design and implementation of a FAIR (Findable, Accessible, Interoperable, Reusable) data management workflow for atomistic simulations of 3C-SiC growth via Physical Vapor Deposition (PVD), using the MulSKIPS multiscale Kinetic Monte Carlo framework [1–3]. The simulation engine is capable of capturing extended defect formation—including stacking faults (SFs) and antiphase boundaries (APBs)—with atomistic resolution under experimentally relevant conditions [3, 4].

A central achievement of this work is the development of a Python-based parser that automates the extraction of simulation metadata and results, generating NeXus files that conform to FAIRmat’s contributed definitions `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` [5]. This enables machine-actionable, semantically rich data outputs that are compatible with the NOMAD repository [6] and the European Open Science Cloud (EOSC) ecosystem [7].

The simulation–data integration pipeline was validated on PVD simulations of 3C-SiC substrates, demonstrating reproducibility, robust metadata curation, and automated defect quantification. While the implementation was tested on PVD only, the modular architecture of the workflow is readily extensible to Chemical Vapor Deposition (CVD) and Pulsed Laser Annealing (PLA) simulations, supporting the future development of interoperable digital twins for materials processing [4, 8].

Contents

Abstract	5
1 Introduction	9
1.1 The Context: Emerging Materials and Data Management Challenges	9
1.2 Motivation	10
1.3 FAIR Principles and Their Importance in Scientific Research	11
1.4 Project Objectives: FAIR Data Stewardship for MulSKIPS Simulations	11
1.5 Scientific and Technological Relevance	12
1.6 Scope of the Thesis and Integration with FAIRmat Standards	13
1.7 Thesis Structure	14
2 Physical and Computational Background	15
2.1 Materials and Processes Overview	15
2.2 Defect Formation in 3C-SiC	16
2.3 Atomistic Simulation of Growth Kinetics	16
2.4 The MulSKIPS Environment	17
2.5 Summary and Connection to FAIR Data Management	18
3 Multiscale Atomistic Simulation and FAIR Data Management	19
3.1 Introduction	19
3.2 The Super-Lattice Kinetic Monte Carlo (KMCsL) Model	21
3.2.1 Motivation and Scope	21
3.2.2 Super-Lattice Formalism	21
3.2.3 Kinetic Events and Their Parameterization	22
3.2.4 Defect Representation and Dynamics	22
3.2.5 Calibration and Validation	23

3.2.6	Algorithm and Computational Considerations	23
3.3	Multiscale Simulation Framework	24
3.3.1	Overview of the Multiscale Challenge	24
3.3.2	Environment-Scale Modeling (PVD, CVD, PLA)	24
3.3.3	Mesoscale Modeling: Growth Rates and Morphology . .	25
3.3.4	Atomistic Modeling: KMCsL and Defect Kinetics . . .	26
3.3.5	Coupling Strategies and Integration	26
3.4	FAIRmat NeXus Integration for Simulation Data	27
3.4.1	FAIR Principles and Their Role in Simulation Science .	27
3.4.2	NeXus Data Format and Structure	27
3.4.3	Mapping Simulation Inputs to <code>NXmicrostructure_imm_config</code>	28
3.4.4	Mapping Simulation Outputs to <code>NXmicrostructure_imm_results</code>	28
3.4.5	Python Parser Implementation and Architecture	29
3.4.6	Validation, Portability, and Reusability	30
3.4.7	Examples and Results	30
3.5	Conclusions and Outlook	30
4	Results and Discussion	33
4.1	Introduction	33
4.2	Simulation Setup	34
4.3	Growth Kinetics and Defect Formation	35
4.3.1	Growth Rate Analysis	35
4.3.2	Defect Nucleation and Evolution	36
4.4	FAIR Data Management Implementation	37
4.4.1	Simulation Configuration Metadata	37
4.4.2	Results Structuring	38
4.5	Discussion	38
4.6	Conclusions	40
5	Extending FAIR Workflows to CVD and Laser Annealing Simulations	41
5.1	Overview and Motivation	41
5.2	FAIR Extension for CVD Simulations	42
5.3	FAIR Extension for Laser Annealing Simulations	43
5.4	Summary and Future Perspectives	45
6	FAIR Data Management of PVD Simulations in 3C-SiC Using MulSKIPS	47
6.1	Introduction	47
6.2	Physics-Based Modeling of PVD Growth in 3C-SiC	48
6.3	The Role of Extended Defects in MulSKIPS Simulations	49

6.4	Automated Metadata Extraction and FAIRification	50
6.5	NeXus Structure and File Generation	51
6.6	Case Study: Sample 153	53
6.7	Impact and Perspectives	54
7	Conclusions and Future Perspectives	56
7.1	Conclusions	56
7.2	Future Perspectives	57
A	Simulation Setup	58
A.1	Execution: Simulation with <code>run_PVD_SiC.py</code>	59
A.2	Post-Processing and Analysis: <code>analyze_PVD_SiC.py</code>	60
A.3	NeXus Export: FAIRmat-compliant <code>.nxs</code> File Generation	61
B	FAIRmat-Compliant NeXus Output and Mapping	62
B.1	Complete File Structure: <code>SiC_sample_153.nxs</code>	62
B.2	Mapping to FAIRmat NeXus Definitions	64
C	Experimental and Simulation Setup Details	66
C.1	Substrate and Material Properties	66
C.2	Thermal and Process Parameters	66
C.3	Precursors and Gas Phase Composition	67
C.4	Relevant References	67
D	Makefile for MulSKIPS Compilation	68
E	Software Configuration	71

Chapter 1

Introduction

1.1 The Context: Emerging Materials and Data Management Challenges

The continuous advancement of nanotechnology and materials science has fueled the development of next-generation semiconductor devices with enhanced performance and functionality. Group IV compound semiconductors such as silicon carbide (SiC) have demonstrated outstanding properties, positioning themselves as prime candidates for future electronic, photonic, and quantum technologies[2, 8].

In particular, SiC offers a unique combination of wide bandgap, high thermal conductivity, and mechanical robustness, enabling applications in harsh environments and high-power systems. Among its various polymorphs, the cubic 3C-SiC phase exhibits advantages such as compatibility with silicon substrates and improved channel mobility, making it especially attractive for integration into metal-oxide-semiconductor field-effect transistors (MOSFETs) and emerging quantum technologies[3].

However, the complexity of fabrication processes such as Physical Vapor Deposition (PVD), Chemical Vapor Deposition (CVD), and Laser Annealing (LA)[4] demands a profound atomistic understanding to optimize material quality and device performance. Defect formation, surface morphology evolution, and phase transition dynamics require detailed atomistic-to-mesoscale multiscale and multi-physics modeling to be accurately predicted and controlled.

Recent developments in atomistic kinetic Monte Carlo (KMC) simulations, particularly through the *MulSKIPS* environment[1], have introduced a multiscale super-lattice framework capable of reproducing extended and point defect dynamics in Group IV semiconductors. These simulations ac-

count for stochastic deposition and evaporation events, explicitly simulating the independent dynamics of atomic species (e.g., Si and C), and can model stacking faults, antiphase boundaries, and complex 3D defect interactions[3?]. Such capabilities are crucial for understanding and controlling the genesis and evolution of defects during growth processes, especially for 3C-SiC on Si substrates.

In this context, the MulSKIPS simulation engine has been extended beyond PVD to include advanced CVD process simulations with coverage species, molecular fluxes, and solid-gas interface reactions[4]. It also supports Laser Annealing scenarios by integrating self-consistent electromagnetic and thermal fields within a phase-field or hybrid KMC approach[8].

Alongside experimental progress, the management and curation of the vast amount of simulation data generated from computational environments such as *MulSKIPS* has become increasingly crucial. Large-scale atomistic simulations produce extensive datasets encompassing process parameters, structural configurations, and defect evolutions. Without a structured management approach, this valuable data risks becoming opaque and non-reusable.

The proactive adoption of FAIR Data Management principles — Findability, Accessibility, Interoperability, and Reusability — provides a robust framework to ensure that scientific data can be efficiently shared, validated, and extended across the research community[7].

1.2 Motivation

Structured and reproducible data management has become a central pillar across all domains of scientific research. The FAIR principles provide a comprehensive framework to enhance the value and longevity of research data. These principles are especially critical in computational material science, where simulation data often represent unique intellectual assets essential for reproducibility, transparency, and accelerated knowledge discovery.

Major European initiatives, including the **European Open Science Cloud (EOSC)**, **FAIRmat**[5, 9], and **NFFA-DI**, are actively fostering the adoption of FAIR practices within the materials science and nanotechnology communities. These initiatives, embedded within the Horizon 2020 and Horizon Europe frameworks, envision establishing digital ecosystems where data is managed and shared openly across national and international boundaries, promoting innovation, collaboration, and transparency.

Within this evolving landscape, the **MulSKIPS** simulation environment plays a critical role. It supports PVD, CVD, and Laser Annealing simulations, particularly in semiconducting materials like silicon carbide (SiC)[2, 4].

Despite its powerful capabilities, MulSKIPS has so far lacked a standardized, FAIR-compliant data management structure for its simulation outputs — highlighting a relevant gap that this project aims to address.

1.3 FAIR Principles and Their Importance in Scientific Research

First articulated in 2016, the FAIR principles were designed to enhance the transparency, reproducibility, and reusability of scientific data[7]. According to these guidelines:

- **Findable:** Data should be easy to locate through globally unique and persistent identifiers and richly described metadata.
- **Accessible:** Data should be retrievable via standardized, open, and secure communication protocols.
- **Interoperable:** Data should utilize shared vocabularies, ontologies, and standards, enabling integration with other datasets and platforms.
- **Reusable:** Data should be richly annotated with provenance, licensing, and domain-specific attributes to maximize future use.

In the realm of atomistic simulations, applying FAIR principles is not merely about compliance but about enhancing the scientific value and impact of datasets[4, 8]. Standardization of simulation inputs, outputs, and metadata ensures reproducibility, fosters collaborative research, and lays the foundation for data-driven discovery.

Particularly relevant is the integration of metadata structures aligned with the FAIRmat initiative, including contributed NeXus definitions such as `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results`. These standards ensure semantic interoperability across simulation workflows, experimental databases, and analytical platforms, enabling machine-actionable datasets crucial for future developments in automated reasoning and AI-driven materials science.

1.4 Project Objectives: FAIR Data Stewardship for MulSKIPS Simulations

This thesis integrates FAIR Data Management practices within the simulation workflows of *MulSKIPS*, a kinetic Monte Carlo (KMC) super-lattice code

designed for atomistic modeling of material growth and defect evolution[2, 4]. *MulSKIPS* supports simulations of PVD, CVD, and Laser Annealing processes for Group IV materials, making it a versatile tool for studying multi-scale phenomena[3].

The project pursues the following objectives:

- Develop a robust and automated pipeline for the extraction and structuring of metadata and simulation results.
- Integrate the output into standardized **NeXus** files, adhering to FAIR-mat’s contributed definitions.
- Automate the full simulation-to-data-archival process through a dedicated Python parser and associated RESTful API.
- Capture and standardize metadata including simulation conditions, process parameters, material properties, and computed results.
- Ensure compliance with EOSC and NFFA-DI infrastructures, enabling seamless data sharing, archival, and long-term preservation.
- Facilitate advanced data-driven studies through structured datasets suitable for AI and machine learning applications in materials science.

The current implementation has focused on PVD simulations of 3C-SiC, but the methodology is designed for future extensions to CVD and Laser Annealing workflows.

1.5 Scientific and Technological Relevance

Atomistic simulations have become indispensable tools in advancing our understanding of material growth phenomena. Techniques such as PVD, CVD, and Laser Annealing involve complex, non-equilibrium processes at the atomic scale, which are challenging to probe experimentally in real time.

MulSKIPS, by combining kinetic Monte Carlo methods with detailed material-specific calibrations, provides unique insights into microstructural evolution, defect dynamics, and surface morphology control. The recent extension of the code to handle gas-phase chemical reactions and coverage dynamics in CVD, and to simulate non-equilibrium heating in LA, demonstrates its multiscale, cross-process capability[4, 8].

Implementing FAIR data management for *MulSKIPS* will:

- **Enhance Reproducibility:** All parameters, configurations, and simulation workflows will be transparently archived and documented.
- **Support Interoperability:** Data will be compatible with a variety of analysis tools, experimental databases, and other simulation frameworks.
- **Promote Reusability:** Datasets can be referenced, extended, and integrated into larger projects, such as training machine learning models for materials discovery.

Thus, this project contributes not only to the continued development of MulSKIPS but also to the broader adoption of FAIR principles across computational materials science.

1.6 Scope of the Thesis and Integration with FAIRmat Standards

This thesis presents the design, development, and implementation of structured data management schemas based on FAIRmat NeXus standards, specifically tailored for the *MulSKIPS* simulation environment. As part of the SISSA Master in Data Management and Curation, this work bridges advanced atomistic simulation methodologies with emerging FAIR data management frameworks.

MulSKIPS is extensively employed at the Institute for Microelectronics and Microsystems (CNR IMM) in Catania, under the supervision of the NFFA-DI operational unit, to model atomic-scale processes crucial in Silicon Carbide (SiC) — a key material for high-performance and quantum technology applications.

Particular attention is dedicated to implementing metadata structures aligned with the Computational Growth of Materials Simulation (CGMS) schema and the Integrated Computational Materials Engineering (ICME) schema[5].

Automated Python scripting was employed to transform raw simulation outputs into structured, FAIR-compliant NeXus files, ensuring consistency in measurement units, simulation conditions, and defect information. Compatibility with the NOMAD repository[6] was verified, including sample uploads and schema validation.

The simulation framework was expanded to include time-dependent analyses of point defect evolution in SiC, decoupling internal surface kinetics from

external dynamics, and enabling detailed studies on stacking faults and poly-type transformations[3].

1.7 Thesis Structure

This thesis is organized as follows:

- **Chapter 2** provides the scientific background, presenting the fundamentals of PVD, CVD, Laser Annealing, and Kinetic Monte Carlo methods.
- **Chapter 3** offers a detailed description of the MulSKIPS simulation environment, including its architecture, input/output formats, and simulation workflows.
- **Chapter 4** discusses FAIR data management principles and the adopted standards, focusing on NeXus formats and FAIRmat definitions.
- **Chapter 5** presents the design and implementation of the FAIR workflow, including the development of the Python parser, metadata extraction, and NeXus file generation.
- **Chapter 6** concludes the thesis by summarizing the key achievements and proposing future perspectives for expanding FAIR data integration within computational infrastructures.

Overall, this work aims to demonstrate how FAIR data management strategies can be effectively applied to advanced computational simulation environments, fostering transparency, collaboration, and scientific innovation.

Chapter 2

Physical and Computational Background

2.1 Materials and Processes Overview

Group IV semiconductors, including silicon (Si), germanium (Ge), silicon carbide (SiC), and their alloys (e.g., SiGe), are foundational for the development of advanced electronic, photonic, and energy applications. Among these, cubic silicon carbide (3C-SiC) stands out due to its unique combination of properties: a narrower bandgap (2.3 eV) compared to hexagonal 4H-SiC (3.28 eV), feasibility of heteroepitaxial growth on silicon substrates, reduced interface trap density with SiO₂, and superior channel mobility exceeding 300 cm²/Vs in MOSFETs[3]. These characteristics position 3C-SiC as a promising material for high-performance and cost-effective electronics operating under extreme conditions.

The fabrication of 3C-SiC epitaxial layers relies on techniques such as physical vapor deposition (PVD), chemical vapor deposition (CVD), and pulsed laser annealing (PLA). PVD involves the physical condensation of vaporized species, typically under high vacuum. CVD uses surface chemical reactions between gas-phase precursors to grow the material, requiring precise control of both gas-phase and surface kinetics. PLA processes, characterized by ultrafast melting and recrystallization induced by nanosecond laser pulses, enable localized structural reorganization and defect engineering[4, 8].

Recent multiscale modeling approaches have shown how these processes can be unified under a digital twin paradigm, where physical–chemical models are coupled with machine learning to predict and control growth and defect dynamics[8]. These developments are supported by high-resolution measurements and the integration of multiphysics simulations from macroscale

transport to atomic-scale evolution.

2.2 Defect Formation in 3C-SiC

Despite its advantageous properties, the growth of 3C-SiC is hampered by the formation of extended defects, which stem from the material’s strong polytypism. The small energy differences between stacking sequences destabilize the ideal cubic order[2]. Among the most prevalent extended defects are stacking faults (SFs), representing local disruptions in the ABC stacking sequence along $\langle 111 \rangle$ directions, and antiphase boundaries (APBs), which form where domains of mismatched phase alignment merge[3].

Kinetic Monte Carlo super-lattice (KMCsL) simulations, supported by high-resolution transmission electron microscopy (HRTEM) and conductive atomic force microscopy (CAFM), show that APBs induce surface instabilities that differ based on local polarity: Si-face and C-face exhibit distinct growth behaviors. These instabilities lead to the drift of APBs along $\langle 110 \rangle$ directions and facilitate the nucleation of triple stacking faults ($\text{SF}_{\langle 3 \rangle}$), commonly referred to as micro-twins[3]. In some cases, single stacking faults ($\text{SF}_{\langle 1 \rangle}$) and SF tetrahedra can also emerge, with their dynamics conditioned by APB interactions, growth conditions, and local morphology.

2.3 Atomistic Simulation of Growth Kinetics

Atomistic modeling is essential for uncovering the stochastic mechanisms that govern defect formation during crystal growth. While molecular dynamics (MD) simulations offer detailed descriptions of atomic motion, they are limited in temporal and spatial scale. Kinetic Monte Carlo (KMC) methods, in particular the KMCsL approach, offer a viable alternative by enabling event-driven simulations of atomic processes (e.g., adsorption, desorption, diffusion) over extended timescales[2, 4].

KMCsL leverages a super-lattice structure to accommodate both ideal crystal sites and defect configurations, facilitating the simulation of:

- APB drift and surface depletion;
- Nucleation of $\text{SF}_{\langle 3 \rangle}$ from APB-driven instabilities;
- Defect interactions leading to termination or polarity inversion of stacking faults;
- Formation of single SFs and SF tetrahedra in 3D structured geometries.

Recent simulations further show that APBs can evolve during growth to generate $SF_{(3)}$ and $SF_{(1)}$ types depending on surface morphology, and that defect survival and polarity change are conditioned by their interaction with other APBs and domain geometries[3?].

2.4 The MulSKIPS Environment

MulSKIPS (Multiscale Kinetic Simulation Platform for Interfaces and Processes) is an open-source simulation environment developed to perform KM-CsL simulations of thin-film growth in tetrahedrally bonded materials[4]. Originally designed for PVD and PLA, MulSKIPS has been extended to model CVD processes, including:

- Surface adsorption and desorption reactions;
- Gas-phase reaction equilibria via integration with the Cantera thermochemical library;
- Precursor-specific calibration workflows linking reactor-scale input to atomistic growth kinetics;
- Geometry-aware simulations of growth on nanostructures and in confined domains.

In CVD simulations, MulSKIPS uses a hybrid framework where gas-phase reactions determine the equilibrium species composition, and surface kinetics are described by analytical models that are subsequently translated into KMC event rates. This methodology allows for predictive modeling of:

- Growth rate as a function of temperature and precursor partial pressures;
- Surface coverage evolution of reactive species (e.g., Cl, H);
- Defect formation and suppression under different kinetic regimes[4].

In PLA and PVD regimes, MulSKIPS supports the simulation of rapid solidification fronts and growth morphology evolution, enabling calibration against phase-field models and experimental profiles[8]. In all cases, it enables the integration of digital twin modules based on multiscale modeling with KMCsL as the atomistic kernel.

2.5 Summary and Connection to FAIR Data Management

MulSKIPS enables the implementation of digital twin models of epitaxial growth, incorporating atomistic precision and multiscale integration. The outputs of these simulations—covering growth conditions, kinetic parameters, and resulting structural configurations—are formatted into standardized NeXus files in compliance with the FAIRmat project.

In this framework, each simulation run is linked to a metadata-rich NeXus file including:

- `NXmicrostructure_imm_config`: describing input configurations and kinetic models;
- `NXmicrostructure_imm_results`: recording surface morphology, defect types, and growth rates.

These files support interoperability within the NFFA-DI infrastructure and enable transparent sharing, reproducibility, and reuse of simulation data. They also serve as machine-actionable records enabling integration with EOSC and AI-based analysis tools for materials design and defect prediction.

The next chapter will detail the FAIR data workflow, including metadata extraction scripts, NeXus structure generation, and potential for user-facing interfaces.

Chapter 3

Multiscale Atomistic Simulation and FAIR Data Management

3.1 Introduction

The continuous miniaturization of electronic devices and the emergence of novel material platforms in the semiconductor industry have made the control of microstructural features—especially defects—an essential requirement for achieving reliable performance, yield, and scalability. In this context, predictive modeling frameworks that can simulate the interplay between thermodynamic fields, deposition kinetics, and atomistic rearrangements are no longer an academic luxury, but a technological necessity.

Processing techniques such as Physical Vapor Deposition (PVD), Chemical Vapor Deposition (CVD), and Pulsed Laser Annealing (PLA) inherently involve coupled phenomena across vastly different space and time scales. Temperature fields evolve over the millimeter-to-centimeter scale with dynamics on the order of seconds or less. Gas-phase transport and surface chemical reactions occur on micrometer and microsecond scales. Finally, nucleation, defect formation, and microstructural evolution involve discrete atomistic events occurring on nanometer and picosecond scales.

Traditional modeling approaches tend to focus on individual scales in isolation: continuum models describe thermal transport and macroscopic geometry, while *ab initio* or molecular dynamics simulations offer atomistic resolution but are constrained by system size and timescales. A unified multiscale simulation approach is required to bridge these regimes and enable direct correlations between experimental process conditions and resulting material quality.

This chapter presents the theoretical foundations, computational strate-

gies, and data management framework of a multiscale simulation environment developed around the *MulSKIPS* (Multiscale Kinetic Process Simulator) code. MulSKIPS is based on the Super-Lattice Kinetic Monte Carlo (KMCsL) algorithm, which enables simulation of atomistic events such as deposition, evaporation, adsorption, and defect formation in tetrahedrally bonded semiconductors like Si, SiGe, and SiC. The model is capable of capturing non-ideal bonding configurations, multiple stacking sequences, and extended defects such as stacking faults (SFs), antiphase boundaries (APBs), and defect–defect interactions [2, 3].

A second pillar of this work is the integration of FAIR data principles—Findable, Accessible, Interoperable, and Reusable—into the simulation workflow. The simulation data generated by MulSKIPS is automatically parsed and stored in the NeXus file format, following the contributed definitions `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` developed by the FAIRmat initiative [5, 9]. This ensures that the computational results, including metadata, simulation input, and structural outputs, are not only reproducible, but also semantically rich and suitable for machine-actionable reuse in external data infrastructures such as NOMAD and EOSC.

Importantly, this approach is designed to support industrial-grade digital twin frameworks, in which each processing step (growth, annealing, cleaning, etc.) is associated with a validated simulation model that can run in parallel to real fabrication equipment. These digital twins aim to predict process outcomes, optimize parameters, reduce metrology overhead, and ultimately close the loop between simulation and experiment [8].

This chapter thus lays the foundation for a scalable, multiscale, and FAIR-compliant simulation infrastructure. Although the current implementation has been tested primarily for PVD processes in 3C-SiC substrates, its conceptual structure is general enough to accommodate other process types (e.g., CVD, PLA), other material systems, and future extensions including process-control loops and inverse design.

We begin in Section 3.2 with an in-depth description of the Super-Lattice Kinetic Monte Carlo model and its key innovations. Section 3.3 introduces the complete multiscale modeling chain, including continuum modeling of temperature and chemical fields, mesoscale growth models, and the link to atomistic processes. In Section 3.4, we describe how the simulation outputs are mapped to FAIRmat NeXus structures using an automated Python parser. Finally, Section 3.5 discusses challenges, validation strategies, and future directions for extending this framework toward industrial deployment and open-science reproducibility.

3.2 The Super-Lattice Kinetic Monte Carlo (KM-CsL) Model

3.2.1 Motivation and Scope

Traditional Lattice Kinetic Monte Carlo (LKMC) methods simulate atomic-scale kinetics by assigning transition probabilities to discrete events—such as atom adsorption or surface diffusion—on a fixed crystalline lattice. However, this restricts simulations to perfect periodic crystals and cannot easily capture extended defects, interstitials, or stacking transformations. To overcome these limitations, the Super-Lattice KMC approach introduces an augmented lattice structure and extended particle definitions that enable flexible and physically consistent modeling of non-ideal configurations, including extended defects that critically affect device performance.

MulSKIPS implements a custom KMCsL engine optimized for group IV and IV-IV compound semiconductors (e.g., Si, Ge, SiC, SiGe), where tetrahedral bonding governs local coordination. The key innovation is the embedding of the ideal atomic lattice in a denser super-lattice that allows for the local deviation from perfect symmetry required to represent stacking faults, antiphase boundaries, and faceting instabilities [2–4].

3.2.2 Super-Lattice Formalism

The super-lattice is constructed as a cubic grid with a resolution sufficient to accommodate all symmetry-equivalent positions of atoms in the ideal crystal. The coordination number of each super-lattice site, and the possible bonding vectors, are defined in terms of ideal tetrahedral configurations. The lattice can host the following entities:

- **Occupied sites:** atoms (e.g., Si, C), either in ideal or distorted positions;
- **Ad-voids:** empty sites adjacent to under-coordinated atoms (dangling bonds);
- **Coverage species:** adsorbed non-crystalline particles (e.g., H, Cl);
- **Wall sites:** geometrically fixed sites representing non-evolving regions (substrate, masks).

Each event in KMCsL alters the local occupancy and bonding state of a set of sites, and is subject to local topological and energetic constraints. For

example, an attachment event is allowed only if it satisfies the valence of the central atom and forms allowable bond angles.

3.2.3 Kinetic Events and Their Parameterization

The system evolves via discrete kinetic events, each associated with a rate r_i computed as:

$$r_i = \nu_i \exp\left(-\frac{E_i}{k_B T}\right)$$

where ν_i is the attempt frequency (pre-factor), E_i is the activation energy (possibly local-environment dependent), and T is the system temperature. Event types include:

- **Attachment (deposition):** atom from gas phase occupies an ad-void;
- **Detachment (evaporation):** under-coordinated atom leaves the lattice;
- **Surface reconstruction:** transition between different stacking sequences;
- **Adsorption/desorption:** coverage species binds/unbinds from a site;

The choice of stacking during attachment events is controlled probabilistically. For atoms depositing on $\{111\}$ facets, the bonding configuration (e.g., cubic or hexagonal stacking) is selected according to a symmetry-breaking rule influenced by surrounding atoms. A tunable parameter $P_{\text{transZigZag}}$ defines the probability of initiating a phase shift when symmetry is degenerate, enabling controlled generation of stacking faults (SFs) or hexagonal inclusions.

3.2.4 Defect Representation and Dynamics

KMCsL can model a wide variety of extended defects:

- **Stacking Faults (SFs):** formed by the insertion of hexagonally stacked planes (e.g., ABCACB instead of ABCABC);
- **Antiphase Boundaries (APBs):** planar arrays of antisites that separate regions of different stacking polarity;

- **Tetrahedral SFs:** three-dimensional defect volumes bounded by partial dislocations;
- **Polarity transitions:** changes in growth direction due to APB interaction.

Each defect is not predefined but emerges spontaneously from the dynamics, depending on temperature, supersaturation, surface morphology, and growth rate. For instance, SFs tend to form on $\{111\}$ facets under low supersaturation conditions and persist if they reach a critical nucleation size. APBs drift along $[110]$ directions due to anisotropic facet velocities, which is faithfully reproduced in KMCsL simulations and validated experimentally [3].

3.2.5 Calibration and Validation

Event rates in KMCsL are calibrated using a hybrid strategy:

- **Ab initio input:** activation energies (E_i) for key processes (e.g., desorption, diffusion) are calculated using DFT;
- **Thermodynamic models:** gas-phase species equilibrium computed via Cantera or analytical models [4];
- **Experimental data:** growth rates, defect densities, and surface morphologies used to fit pre-factors and validate kinetics [8].

This multilevel calibration allows extending the model across a wide temperature and pressure range, and ensures predictive power for new process regimes.

3.2.6 Algorithm and Computational Considerations

MulSKIPS employs an optimized Gillespie-type rejection-free algorithm, in which the cumulative rate $R = \sum_i r_i$ is used to select the next event and update the system clock via:

$$\Delta t = -\frac{\ln u}{R} \quad \text{where } u \in (0, 1]$$

To maintain computational feasibility over billions of events and up to seconds of physical time, data structures are optimized using sparse representations of the active lattice, and event updates are localized to minimize

recomputation. Parallelism is currently coarse-grained (e.g., multiple replicas), but future GPU-acceleration or event-class binning are under consideration.

3.3 Multiscale Simulation Framework

3.3.1 Overview of the Multiscale Challenge

The processing of semiconductor materials involves physical phenomena operating over vastly different spatial and temporal scales. At the macroscopic level, environmental variables such as temperature, pressure, and electromagnetic fields govern the overall growth regime. At the mesoscale, growth rates, surface roughness, and morphological instabilities emerge from the interplay of local kinetics and field gradients. At the finest scale, atomic events such as defect nucleation, phase transitions, and bond rearrangements determine the crystal structure and material quality.

A robust simulation framework must therefore operate in a hierarchical and interoperable fashion, where data from larger-scale models inform boundary conditions or input parameters for finer-scale simulations. MulSKIPS provides the atomistic layer of this framework, and is designed to interoperate with finite-element (FEM) and continuum kinetic models via either sequential (one-way) or self-consistent (two-way) coupling strategies.

3.3.2 Environment-Scale Modeling (PVD, CVD, PLA)

At the environment scale, the objective is to solve for the macroscopic process fields—particularly temperature, supersaturation, and electromagnetic power density—which define the boundary conditions for mesoscale and atomistic evolution.

Thermal Fields. In PVD and CVD processes, steady-state temperature fields are typically computed using FEM tools such as COMSOL Multiphysics or FEniCS. The heat transport equation,

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k(T) \nabla T) + Q_{\text{ext}},$$

is solved with material-dependent parameters (e.g., temperature-dependent conductivity $k(T)$) and radiative or convective boundary conditions. Calibration is performed against pyrometry data and in-situ sensors, as detailed in [8].

Supersaturation and Gas Transport. For PVD and CVD, the transport of gas-phase precursors is modeled via mass diffusion and chemical reaction kinetics. In close-space sublimation setups, species such as SiC_2 and Si_2C are assumed to follow quasi-ballistic transport. The supersaturation $\Delta\mu = \mu_{\text{vapor}} - \mu_{\text{solid}}$ governs the net flux and growth direction. Analytical models for species flux balance can be written as:

$$j_{\text{dep}} = \sqrt{\frac{1}{2\pi MRT}} P_{\text{species}}(T),$$

with the equilibrium pressure derived from experimental vapor pressure data (e.g., Avrov, Lilov calibrations) [8].

Electromagnetic Fields (PLA). In laser annealing, electromagnetic energy is absorbed and converted into heat within nanoseconds. The Maxwell equations in time-harmonic form are coupled with the heat equation:

$$\nabla \times \nabla \times \mathbf{E} - \epsilon(\mathbf{r}, T) \frac{\omega^2}{c^2} \mathbf{E} = 0,$$

$$\rho c_p \frac{\partial T}{\partial t} = \nabla \cdot (k \nabla T) + \frac{1}{2} \text{Re}[\mathbf{J} \cdot \mathbf{E}^*],$$

where the electric field $\mathbf{E}$ and current density $\mathbf{J}$ describe localized energy deposition. The temperature dependence of the complex dielectric function ϵ captures melting and phase transitions. This is especially important for processes like explosive crystallization of Ge or metal-silicide formation in Ni/SiC systems [4].

3.3.3 Mesoscale Modeling: Growth Rates and Morphology

At the mesoscale, simplified kinetic models predict average growth behavior, providing input to atomistic simulators and validating experimental trends.

Growth Rate Modeling. Growth rate G is computed from the difference between effective deposition and evaporation fluxes:

$$G = \frac{M}{\rho} (j_{\text{dep}} - j_{\text{evap}}),$$

where M is molar mass and ρ is the material density. The fluxes are temperature dependent and depend on the assumed gas composition, such as Si-rich or stoichiometric conditions.

Surface Morphology Evolution. Continuum surface models (e.g., level-set or phase-field methods) simulate morphological changes, especially under non-uniform irradiation or shadowing effects. For example, during PLA, phase-field models capture the liquid-solid interface dynamics:

$$\tau_\phi \frac{\partial \phi}{\partial t} = W^2 \nabla^2 \phi - \frac{\partial F(\phi, T)}{\partial \phi},$$

where ϕ is the phase field and F encodes the free energy landscape. Coupling with thermal fields allows prediction of melt depth, resolidification time, and compound formation (e.g., Ni_xSi_y alloys on SiC) [4].

3.3.4 Atomistic Modeling: KMCsL and Defect Kinetics

The atomistic layer, provided by MulSKIPS, resolves discrete stochastic events at the scale of individual atoms and bonds. KMCsL simulations receive their initial conditions and boundary inputs (e.g., local temperature, supersaturation) from mesoscale or environment-scale models.

Defect Nucleation and Evolution. KMCsL captures the spontaneous generation of stacking faults (SFs) and antiphase boundaries (APBs) under symmetry-breaking conditions. These defects can grow, drift, merge, or annihilate based on kinetic competition. Simulations have demonstrated:

- APBs drift along [110] due to facet-dependent velocity asymmetries;
- SFs nucleate at {111} facets and propagate if nucleation size exceeds a critical threshold;
- SF tetrahedra form under certain stacking sequences and evolve as metastable objects.

Statistical Analysis. KMCsL outputs permit statistical characterization of defect distributions: densities, lengths, orientations, and stacking sequences. These metrics can be directly compared to TEM observations and used to tune process parameters for defect suppression.

3.3.5 Coupling Strategies and Integration

Two principal coupling strategies are employed:

1. **Sequential coupling:** environment-scale simulations are run offline and their outputs fed into MulSKIPS (e.g., PVD temperature field from COMSOL).
2. **Self-consistent coupling:** real-time bidirectional exchange between FEM and KMC layers (e.g., laser annealing where heating and melting alter boundary conditions).

Although the current implementation relies on sequential coupling, future versions are planned to support tight integration via shared-memory or asynchronous inter-process communication (IPC).

3.4 FAIRmat NeXus Integration for Simulation Data

3.4.1 FAIR Principles and Their Role in Simulation Science

In recent years, the principles of FAIR data stewardship—Findability, Accessibility, Interoperability, and Reusability—have become central pillars in the management of scientific data, particularly in fields driven by high-throughput simulations and experiments. Within materials science, FAIR compliance is not only a regulatory or funding requirement, but a prerequisite for reproducibility, machine-actionability, and long-term sustainability of data assets.

The FAIRmat initiative, part of the National Research Data Infrastructure (NFDI) in Germany, has extended the NeXus file standard (originating from neutron and X-ray scattering) to support structured, hierarchical storage of data from atomistic simulations. In this framework, contributed definitions such as `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` serve as semantic templates for storing simulation input metadata and result datasets, respectively [5, 9].

3.4.2 NeXus Data Format and Structure

NeXus is built upon the HDF5 format, which supports hierarchical groups, attributes, and compressed datasets. Each NeXus file contains:

- A **root group** (`NXentry`) that identifies a particular experiment or simulation;

- One or more **subgroups** with types such as `NXinstrument`, `NXsample`, `NXdata`, and contributed types like `NXmicrostructure_imm_config`;
- **Datasets** (arrays, scalars, strings) with associated metadata: units, uncertainty, provenance.

This structure ensures machine interpretability while remaining human-readable with tools like HDFView, nexusformat, or NOMAD Oasis.

3.4.3 Mapping Simulation Inputs to `NXmicrostructure_imm_config`

The configuration group describes the physical system, simulation conditions, and numerical parameters. The parser extracts and maps:

- **Material system:** chemical formula, stoichiometry, crystal structure, initial coordinates;
- **Simulation parameters:** temperature, pressure, time step, event table, defect probabilities;
- **Boundary conditions:** system size, lattice vectors, periodicity, substrate orientation;
- **Gas environment:** precursor species, flow rates, partial pressures (for CVD/PVD);

For instance, the simulation of 3C-SiC under PVD conditions includes a `temperature` field (in K), a `supersaturation` scalar, and an `event_probabilities` array specifying attachment/detachment rates for each species. All values are stored with physical units and descriptive attributes.

3.4.4 Mapping Simulation Outputs to `NXmicrostructure_imm_results`

Simulation outputs are recorded in a dedicated group containing:

- **Growth metrics:** net growth rate, surface height maps, deposition/evaporation rates;
- **Defect statistics:** counts of SFs and APBs, distributions by length and orientation;
- **Final structure:** atomic positions, bonding information, neighbor lists;

- **Temporal evolution:** snapshots of structural states at defined intervals;
- **Logs:** timing, number of events, wall-clock performance.

For example, a dataset named `stacking_fault_count` may contain an integer scalar with units of "number", while `surf_height_map` is a 2D array (in nm) mapping the final film surface.

3.4.5 Python Parser Implementation and Architecture

A dedicated Python script—`Parser_MulSKIPS_NeXus`—was developed to automate the full export pipeline. The script:

1. Executes the simulation (`run_PVD_SiC.py` and `analyze_PVD_SiC.py`);
2. Parses input metadata and output logs;
3. Converts raw data into NumPy arrays and NeXus-compliant HDF5 datasets;
4. Writes a structured `.nxs` file using the `nexusformat` and `h5py` libraries;
5. Optionally validates the file against FAIRmat schema using JSON-based validators.

The parser is modular: the configuration and result blocks are built from independent dictionaries and merged into the HDF5 tree. An example call:

```
from nexusformat.nexus import *
nxentry = NXentry()
nxentry.config = build_config_dict(input_data)
nxentry.results = build_results_dict(output_data)
nxentry.save('SiC_sample_153.nxs')
```

The parser also supports batch-mode operation, parameter sweeps, and optional export of atomic configurations in XYZ, POSCAR, or CIF formats for external visualization or DFT post-processing.

3.4.6 Validation, Portability, and Reusability

Generated NeXus files are designed to be compatible with the FAIRmat NOMAD platform. Each file includes UUIDs, timestamps, and schema references. The metadata is sufficient to fully reproduce the simulation, and files can be uploaded to NOMAD Oasis for indexing, visualization, and citation with a DOI.

Furthermore, the use of standard HDF5 and NeXus ensures interoperability with external analysis tools, including Jupyter notebooks, ML pipelines, and TCAD back-ends. All datasets are annotated with descriptive units (via the `units` attribute), ensuring dimensional consistency.

3.4.7 Examples and Results

The file `SiC_sample_153.nxs` generated by the current framework includes:

- A net growth rate of $G = 0.91$ nm/s;
- $N_{\text{SF}} = 4$ stacking faults, with average length $l_{\text{avg}} = 12.3$ nm;
- A 3D array of atomic positions at final time, exported in `/results/final_structure`;
- Full input conditions under `/config/process_parameters`;

These datasets have been visualized using HDFView and analyzed via Jupyter to produce publication-ready figures and benchmark reproducibility.

3.5 Conclusions and Outlook

This chapter has presented a comprehensive framework for multiscale atomistic simulation and FAIR data management in the context of materials processing by Physical Vapor Deposition (PVD), Chemical Vapor Deposition (CVD), and Pulsed Laser Annealing (PLA). The methodology integrates physical modeling, algorithmic innovation, and data curation into a coherent simulation pipeline centered on the MulSKIPS environment and FAIRmat NeXus data standards.

At the core of the atomistic simulation layer lies the Super-Lattice Kinetic Monte Carlo (KMCSL) formalism, which extends classical lattice KMC by embedding the ideal atomic lattice in a denser super-lattice. This enables the representation of complex bonding environments and extended defects, including stacking faults, antiphase boundaries, and their interactions. The

implementation captures not only growth dynamics but also the stochastic and symmetry-breaking nature of defect nucleation and evolution. Calibrated through ab initio energetics and validated with experimental data, the KMCsL simulations demonstrate the emergence of realistic microstructures under realistic processing conditions.

At the environmental and mesoscopic scales, the framework incorporates continuum models for thermal transport, supersaturation fields, and electromagnetic interactions. Finite-element modeling provides temperature and power density profiles that inform the boundary conditions for the atomistic engine, either through sequential or, prospectively, self-consistent coupling.

The second major contribution of this work is the integration of FAIR data practices into the simulation pipeline. By exporting simulation inputs and results into NeXus files structured according to the `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` contributed definitions, we ensure that all data products are machine-readable, versioned, semantically annotated, and ready for archival or reuse in infrastructures such as NOMAD and EOSC. The Python-based parser developed as part of this work automates the data transformation and validation steps, enabling reproducibility and interoperability.

The entire framework, though currently focused on PVD simulations in 3C-SiC, is modular and generalizable. Ongoing and future developments will target several critical extensions:

- **CVD and PLA processes:** inclusion of gas-phase reaction kinetics, physisorption models, and phase-field coupling for melting-solidification dynamics;
- **Dynamic coupling:** real-time interaction between FEM solvers and KMCsL for laser annealing and transient heating effects;
- **Digital twins:** embedding of the simulation module within virtual metrology and process control frameworks for fabrication equipment [8];
- **User interfaces:** development of GUI and CLI tools to guide non-specialist users through the simulation setup, execution, and NeXus export steps;
- **NOMAD integration:** direct upload of NeXus outputs to NOMAD Oasis with metadata harvesting, DOI assignment, and visualization;

- **Workflow automation:** use of workflow engines (e.g., AiiDA, FireWorks) to orchestrate batch runs, parameter sweeps, and uncertainty quantification;
- **Machine learning readiness:** ensuring that data formats and structures are suitable for downstream ML applications in defect prediction and process optimization.

In summary, this chapter has laid the groundwork for a next-generation simulation infrastructure that not only captures the essential physics of microstructure evolution under processing conditions, but also adheres to the highest standards of data stewardship and reproducibility. The integration of multiscale modeling with FAIR

Chapter 4

Results and Discussion

4.1 Introduction

This chapter presents the results of atomistic simulations performed using the MulSKIPS Kinetic Monte Carlo (KMCsL) environment [1], with a specific focus on modeling the growth kinetics and defect evolution in cubic silicon carbide (3C-SiC) during Physical Vapor Deposition (PVD) processes. The simulations are integrated into a FAIR Data Management workflow, ensuring that inputs, outputs, and metadata are exported in structured NeXus files conforming to the `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` definitions developed by the FAIRmat consortium [5].

The main scientific goal of this work is to understand the nucleation and evolution of extended defects such as stacking faults (SFs) and antiphase boundaries (APBs) in 3C-SiC thin films grown on Si substrates. These defects are known to originate from surface instabilities during growth on {111} facets and can strongly affect carrier transport and device performance [2, 3]. Simulations were run under a wide range of temperature and supersaturation conditions, with atomic resolution and defect statistics extracted automatically.

This chapter addresses exclusively PVD processes; although the MulSKIPS framework supports CVD and laser annealing modeling [4, 8], these modes are not analyzed here. All results presented were obtained through automated workflows coupling simulation and analysis scripts (`run_PVD_SiC.py` and `analyze_PVD_SiC.py`), orchestrated via a Jupyter notebook interface `Parser_30_04_25.ipynb`.

The implementation aligns with BIM12 and BIM13 activities in the context of FAIR data stewardship and EOSC integration [7], promoting repro-

ducibility, transparency, and machine-actionability of simulation data. The exported NeXus files are suitable for ingestion into repositories such as NOMAD [6] and for future integration with experimental datasets from CNR-IMM@CT.

The following sections describe the simulation methodology and automation, quantitative results on growth kinetics and defect formation, FAIR export implementation, and critical discussion of the reproducibility and limitations of the current PVD-focused workflow.

4.2 Simulation Setup

All simulations were conducted using the MulSKIPS open-source Kinetic Monte Carlo framework, specifically tailored to atomistic modeling of compound semiconductors with sp^3 -type bonding symmetry [1]. In this thesis, we focused exclusively on the growth of 3C-SiC on (001)-oriented silicon substrates via Physical Vapor Deposition (PVD).

The simulation campaign was carried out using two dedicated scripts:

- `run_PVD_SiC.py` – for setting up and executing the kinetic growth simulations,
- `analyze_PVD_SiC.py` – for post-processing of structural data, including defect identification and surface morphology metrics.

Both scripts were orchestrated via a unified automation interface implemented in the Jupyter notebook `Parser_30_04_25.ipynb`. This notebook automates the full workflow, including simulation execution, result extraction, and generation of standardized output in NeXus format. The NeXus output was designed to comply with the `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` contributed definitions from FAIRmat [5].

The initial atomic configuration corresponds to a flat 3C-SiC (001) surface in epitaxial alignment with silicon. The simulation grid was defined with periodic boundary conditions in the lateral directions and open top boundaries to simulate vapor phase deposition. The superlattice representation used by MulSKIPS supports both regular zinc-blende stacking and local stacking deviations (e.g., faults and twins), and enables explicit modeling of both Si and C atoms with independent kinetics [3].

The precursors included in the model were chosen to mimic silicon-rich vapor conditions, with dominant gas-phase species being Si, Si_2C and SiC_2 [2]. The event probabilities for deposition and evaporation were calibrated

to reflect these precursors and their interaction with the growing surface. Simulation conditions such as substrate temperature and supersaturation were varied to evaluate their effect on the growth rate and defect formation probability, in line with the methodology described in [8].

Each simulation produces intermediate files including:

- `DEP3Dfile` – the final atomic configuration (used for visualization and defect detection),
- `surf_height.txt` – the height of the topmost occupied lattice site as a function of simulation steps,
- `output.xyz` – the trajectory in XYZ format for possible dynamic analysis.

The final step of the workflow involves automatic parsing of the outputs and generation of a single ‘.nxs’ file embedding:

1. the full simulation configuration as metadata,
2. key simulation outputs including growth rate, defect statistics and representative atomic snapshots.

Robustness features were implemented to handle cases where some outputs (e.g., `surf_height.txt`) were missing or incomplete, ensuring the generation of semantically valid NeXus files.

This setup enables not only quantitative evaluation of the growth process, but also traceable, FAIR-aligned data curation ready for ingestion in platforms such as NOMAD [6] and reuse in multi-process modeling workflows [7].

4.3 Growth Kinetics and Defect Formation

4.3.1 Growth Rate Analysis

The average growth rate was extracted from the simulation output file `surf_height.txt`, which records the vertical extent of the deposited layer as a function of simulation time steps. In the majority of simulations, a nearly linear relationship between surface height and time was observed, in agreement with kinetic growth models for supersaturated vapor-solid systems [2].

Growth rate values ranged from 0.05 to 0.25 monolayers per second, depending on the simulated substrate temperature and flux ratio. Higher supersaturation values systematically led to increased growth rates, confirming

the role of gas-phase concentration gradients as the primary thermodynamic driving force for deposition [3].

The simulations also reproduced the preferential formation of atomic bilayers aligned with the (001) surface, and lateral propagation of terraces associated with growth on {111} microfacets. These local features are consistent with prior multiscale PVD studies that model morphology evolution from the atomic scale [8].

In some test cases, growth rates deviated from linearity due to nucleation of vacancies or surface pinning. These effects were captured by analyzing the second derivative of surface height over time and were more pronounced at lower substrate temperatures.

4.3.2 Defect Nucleation and Evolution

A major result of the PVD simulations concerns the atomistic modeling of extended defect formation, in particular stacking faults (SFs) and antiphase boundaries (APBs), which are common in 3C-SiC grown on Si [3].

SFs were observed to nucleate primarily on {111} facets that temporarily emerge during growth due to local surface fluctuations. These defects are linked to the inherent polytype instability of SiC and manifest as ABC-ACB-ABC stacking deviations, consistent with microtwin-like configurations described in high-resolution electron microscopy studies [2].

Triple stacking faults (SF₃) were the most frequently observed, and their lateral expansion was tracked across simulation steps. The spatial growth of SFs was found to be strongly dependent on the kinetic asymmetry between Si-face and C-face of the {111} planes, as previously noted in both simulation and experimental studies [3].

Antiphase boundaries were also frequently observed, often at the early stages of simulation. In several cases, APBs acted as seeds for subsequent stacking fault generation, as observed experimentally and reproduced in simulations by the stochastic drift of surface instabilities [4]. APBs were modeled in the superlattice framework as planes of anti-sites (Si-Si or C-C bonds) and were found to propagate anisotropically due to the differing mobility of facets on either side of the domain boundary.

Defect density was computed for each simulation as the number of disordered regions per square micron, based on a parsing of the coordination number and local bonding topology in the atomic structure. This metric was stored in the `NXmicrostructure_imm_results` section of the exported NeXus files.

The combined analysis of growth kinetics and defect formation demonstrates the capability of the MulSKIPS model to predict both rate laws and

structural quality under different PVD conditions, and validates its utility for predictive process modeling in group IV compound semiconductors.

4.4 FAIR Data Management Implementation

The output of each simulation was structured and exported into a NeXus file (`.nxs`) using a custom Python-based parser implemented in the notebook `Parser_30_04_25.ipynb`. The NeXus files were designed to be compliant with the contributed FAIRmat class definitions `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results`, enabling semantic description of simulation conditions and computed results for atomistic modeling workflows [5].

This section describes how simulation inputs and outputs were mapped into the NeXus format, and how the resulting files fulfill key FAIR principles: Findability, Accessibility, Interoperability, and Reusability [7].

4.4.1 Simulation Configuration Metadata

Metadata describing the simulation conditions were stored within the `NXmicrostructure_imm_config` group of the NeXus file. This included:

- **Material system:** 3C-SiC on Si(001), with explicit crystal orientation
- **Process type:** Physical Vapor Deposition (PVD)
- **Gas phase species:** Si, Si₂C, SiC₂
- **Flux rates and temperature:** Supersaturation ratio, substrate temperature
- **Simulation domain:** grid size, lattice spacing, boundary conditions
- **Parser version and scripts used:** `run_PVD_SiC.py`, `analyze_PVD_SiC.py`

Each metadata field was explicitly labeled using units and descriptive names following the FAIRmat ontology and NeXus base classes [5]. Where appropriate, physical quantities were linked to standard units using the NeXus base unit conventions.

Robustness checks were implemented to ensure metadata completeness even in the presence of missing or corrupted auxiliary files (e.g., handling failure to generate `surf_height.txt`). Default flags and error codes were introduced to avoid semantic breakage of the exported NeXus structures.

4.4.2 Results Structuring

Quantitative simulation results were stored in the `NXmicrostructure_imm_results` group of each NeXus file. The following outputs were recorded:

- **Average growth rate:** extracted from the slope of `surf_height.txt`
- **Surface roughness:** root mean square height variation, where available
- **Defect density:** total number of stacking faults and antiphase boundaries per μm^2
- **Final atomic structure:** embedded in reduced form (e.g., subset of `DEP3Dfile`)
- **Trajectory summary:** total number of MC steps, number of deposited atoms

These quantities were structured in accordance with the FAIRmat NeXus schema and stored in machine-readable format using HDF5. In addition, an auxiliary group `/provenance` recorded the SHA-1 hashes of the input scripts and the Git commit ID of the parser for reproducibility.

Validation of the resulting files was performed using the FAIRmat validator toolchain, ensuring both syntactic and semantic correctness. The NeXus files are ready for ingestion into community data repositories such as NOMAD [6], where they can be linked to experimental results or reused in multiscale modeling workflows within EOSC [7].

By formalizing the export of simulation data in this way, we ensure that all relevant input/output parameters are not only stored but also documented and shareable, supporting a FAIR-by-design approach to simulation workflows in materials science.

4.5 Discussion

The integration of FAIR data management principles within the atomistic simulation framework of MulSKIPS proved effective in supporting reproducibility, transparency, and reusability of physical vapor deposition (PVD) simulations on 3C-SiC. The export of simulation data in the NeXus format, validated against FAIRmat-contributed schemas [5], enabled a machine-actionable representation of both configuration metadata and kinetic results. This aligns with the core principles of FAIR data stewardship as outlined by Wilkinson et al. [7].

A significant benefit of this approach was the unambiguous traceability of each simulation: from the exact input conditions (e.g., gas species, temperature, boundary conditions), to the kinetic outputs such as growth rates and defect statistics. The inclusion of provenance metadata (e.g., script versions, Git commit hashes) further strengthens the reproducibility of the results and facilitates future reanalysis or validation by other users or tools.

The use of contributed NeXus definitions such as `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` allowed for a semantically rich export structure. This has direct interoperability benefits: NeXus files produced through this workflow can be uploaded to domain repositories like NOMAD [6], or integrated into larger multiscale simulations involving data fusion from experimental sources (e.g., TEM, TR-reflectivity) or higher-scale continuum models, as outlined in recent multiscale modeling strategies for semiconductor processing [4, 8].

However, some challenges were encountered. The output file `surf_height.txt`, used to compute growth rate, was occasionally absent or incomplete due to rare simulation crashes or runtime issues in post-processing. To address this, the parser was modified to include error-tolerant logic: default values were inserted where data were missing, and an internal flag system was introduced to mark any output file inconsistencies.

Another technical limitation concerns the representation of complex 3D defect topologies. While the defect density is summarized quantitatively, the NeXus export does not currently include full spatial maps or mesh-based representations of APBs and SFs. Inclusion of these features may require additional schema extensions or embedding of reduced atomic snapshots in a compact format (e.g., XYZ subsets or voxel maps).

Despite these limitations, the implementation represents a concrete advance toward FAIR-compliant computational materials science. The workflow supports the notion of “simulation-as-data”, where atomistic modeling becomes a reusable and shareable research output in its own right. Reproducibility, data stewardship and integration with EOSC services are essential [5, 7].

Looking forward, future improvements may include:

- Automated metadata enrichment using external materials ontologies (e.g., OPTIMADE descriptors)
- Deeper integration with experimental metadata (e.g., in situ electron microscopy of APB dynamics)
- Time-resolved tracking of individual defect growth, linked to kinetic steps and surface evolution

- Generalization of the FAIR export logic to CVD and laser annealing processes modeled by MulSKIPS

These developments would not only enhance the reuse potential of simulation data, but also support digital twin scenarios where simulation and experiment are tightly coupled. In this perspective, the PVD-focused FAIR implementation presented here serves as a modular and extensible foundation for future multiscale workflows.

4.6 Conclusions

In this chapter, we demonstrated the implementation and validation of a FAIR-compliant data management workflow for atomistic simulations of 3C-SiC growth via Physical Vapor Deposition (PVD), using the MulSKIPS simulation environment.

Through a fully automated pipeline—from simulation setup and execution to structured metadata extraction and export—we produced NeXus files that encode not only the simulation configuration but also key outputs such as growth rate, surface evolution, and defect statistics. These files adhere to the `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` definitions provided by FAIRmat [5], enabling interoperability with data infrastructures such as NOMAD [6].

The results confirmed that the MulSKIPS KMCsL approach accurately captures both kinetic and structural phenomena in PVD, including the nucleation and propagation of extended defects like stacking faults and antiphase boundaries, in agreement with theoretical and experimental observations [2, 3].

From a data stewardship perspective, the workflow illustrates how FAIR principles can be embedded directly into computational materials science practices [7].

While the present study focused exclusively on PVD simulations, the modular design of the export pipeline and the generality of the NeXus schema suggest that the approach can be extended to other process types modeled by MulSKIPS, including CVD and laser annealing [4, 8].

In summary, this work demonstrates a viable and robust path for transforming high-fidelity atomistic simulations into structured, interoperable research objects. Such integration is essential to the future of open and FAIR computational science.

Chapter 5

Extending FAIR Workflows to CVD and Laser Annealing Simulations

5.1 Overview and Motivation

The data management and simulation framework presented in this thesis was designed and tested specifically for Physical Vapor Deposition (PVD) simulations of 3C-SiC using the MulSKIPS kinetic Monte Carlo environment [1, 3, 8]. However, the core architecture—comprising modular simulation scripts, post-processing routines, and a FAIR-compliant export mechanism—can be generalized to other classes of materials processes such as Chemical Vapor Deposition (CVD) and Pulsed Laser Annealing (PLA).

Both CVD and PLA are widely adopted techniques for the processing of group IV semiconductors, each involving distinct physical phenomena and modeling challenges. CVD involves precursor transport, adsorption, and chemical reactions at the growing surface, while PLA drives ultrafast phase transitions and diffusion through intense, nanosecond-scale laser pulses [4, 8].

Recent developments in the MulSKIPS framework have demonstrated the capability to model these processes:

- In CVD simulations, gas-phase chemistry and surface reactions are captured through a calibrated, chemically-resolved KMCsL formulation coupled with thermodynamic equilibrium calculations [4].
- In PLA simulations, the evolution of molten regions and phase front propagation is modeled by combining superlattice KMC with continuum-scale solvers for heat transport and electromagnetic energy absorption

[8].

This chapter outlines how the FAIR data structuring approach used for PVD—based on the `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` NeXus classes [5]—can be extended or adapted to support CVD and PLA workflows. We also examine potential adjustments to simulation metadata, structural outputs, and validation strategies to maintain FAIR compliance across all process types.

While no CVD or PLA simulations were performed in this thesis, the architectural and semantic flexibility of the FAIR export scheme ensures that such extensions can be implemented with minimal restructuring.

5.2 FAIR Extension for CVD Simulations

Chemical Vapor Deposition (CVD) is one of the most important synthesis techniques in semiconductor manufacturing. Compared to PVD, CVD processes involve more complex reaction mechanisms at the gas–solid interface, including molecular adsorption, dissociation, and incorporation of precursor species into the growing crystal. These phenomena strongly depend on gas-phase composition, pressure, temperature, and local surface coverage, and must be modeled accordingly.

Within the MulSKIPS framework, Raciti et al. [4] introduced an extended KMCsL formalism that models CVD growth with atomistic resolution. This extension includes:

- chemically specific surface events (e.g., SiCl_2 adsorption, HCl desorption),
- explicit modeling of coverage species (H , Cl) and adatom dynamics,
- coupling with external thermodynamic codes (e.g., Cantera) to compute gas-phase equilibrium compositions.

The output of these simulations includes not only structural evolution, but also:

- time-resolved surface coverages (e.g., Θ_{H} , Θ_{Cl}),
- local reaction rates and precursor fluxes,
- crystalline quality metrics (e.g., stacking disorder, twin density).

To ensure FAIR compliance for such CVD simulations, the existing NeXus-based export workflow needs only minor adaptations:

1. In the `NXmicrostructure_imm_config` section, new fields should be added to describe:
 - precursor types (e.g., SiH_2Cl_2 , HCl , H_2),
 - reactor pressure and total flow rate,
 - equilibrium gas-phase composition (e.g., via Cantera interface).
2. In the `NXmicrostructure_imm_results` section, additional outputs should include:
 - average and final surface coverages,
 - growth rate vs. temperature or precursor ratio,
 - defect types specific to CVD regimes (e.g., trench fill defects, interface roughness).

These additions preserve the schema logic used for PVD and maintain compatibility with FAIRmat and NOMAD infrastructures [5, 6]. Moreover, since the simulation core is still based on MulSKIPS KMCsL and the data structure remains NeXus/HDF5, no changes are needed to the export engine or file format.

The integration of gas-phase metadata and surface chemistry into the FAIR export aligns with ongoing efforts to model deposition as a multi-scale, chemically resolved process. In future implementations, the FAIR data pipeline could also be coupled to reactor-scale models or process control simulations (e.g., for flow, temperature gradients, or VM modules), thereby enabling end-to-end virtual fabrication workflows [4].

This example confirms that the FAIR approach tested in this thesis for PVD is fully applicable to CVD, and that the current infrastructure is flexible enough to support process-aware extensions without compromising compliance or interoperability.

5.3 FAIR Extension for Laser Annealing Simulations

Pulsed Laser Annealing (PLA) is a highly localized and non-equilibrium thermal treatment method employed to activate dopants, repair defects, or induce phase transformations in semiconductors. Unlike PVD and CVD, PLA involves ultra-short laser pulses (typically < 100 ns) that rapidly melt and

re-solidify thin surface layers. Accurately simulating PLA requires a multi-scale approach that couples macroscopic field solvers with atomistic models for phase transitions and structural evolution.

In recent work, Calogero et al. [8] demonstrated how PLA can be modeled within the MulSKIPS framework by integrating:

- Finite Element solvers (e.g., COMSOL, FEniCS) for solving time-dependent heat and electromagnetic fields;
- a laser source term coupled to local absorption and reflectivity;
- atomistic KMCsL modules for modeling melt–solid transitions, silicide formation, and recrystallization.

The simulation outputs for PLA include:

- spatio-temporal temperature maps;
- liquid–solid interface dynamics;
- transient defect evolution (e.g., regrowth velocity, interface trapping);
- final crystalline structure and defect statistics.

To incorporate PLA simulations into the FAIR workflow developed in this thesis, the NeXus-based export scheme must be extended to represent both time-resolved fields and phase information. This requires the following modifications:

In `NXmicrostructure_imm_config`:

- Add laser-specific parameters: wavelength, pulse duration, energy density;
- Record optical and thermal material properties (e.g., $\varepsilon(\lambda, T)$, $k(T)$);
- Include spatial mesh description and solver configuration.

In `NXmicrostructure_imm_results`:

- Include temporal evolution of temperature fields ($T(x, y, z, t)$);
- Track phase fraction (e.g., melt depth over time);
- Annotate final structure with phase regions and regrowth quality.

These additions may require the definition of new subgroups or extensions within the FAIRmat schema, particularly to accommodate temporal fields and multiphysics data. However, the core FAIR principles—Findability, Accessibility, Interoperability, and Reusability—remain achievable within the NeXus/HDF5 ecosystem [5, 7].

Moreover, the simulation–data integration model presented by Calogero et al. aligns with digital twin concepts in advanced semiconductor processing, where PLA modules are combined with experimental feedback loops and AI-enhanced control [8]. Exporting PLA simulations into a FAIR NeXus format enables consistent data pipelines across process types, facilitating simulation–experiment co-validation and large-scale parameter sweeps.

In conclusion, the FAIR workflow developed for PVD can be successfully adapted to PLA, with extensions to accommodate multiscale and transient data. This positions MulSKIPS as a flexible and FAIR-ready platform for integrated process modeling across deposition and annealing stages.

5.4 Summary and Future Perspectives

This chapter demonstrated how the FAIR-compliant data management workflow developed for Physical Vapor Deposition (PVD) simulations of 3C-SiC can be systematically extended to cover other technologically relevant processing methods, specifically Chemical Vapor Deposition (CVD) and Pulsed Laser Annealing (PLA).

Through a review of recent developments in the MulSKIPS framework [4, 8], we showed that both CVD and PLA are already supported at the simulation level using chemically resolved KMCsL models and multiscale couplings with thermodynamic or continuum solvers. These simulations produce outputs that, although more complex than those from PVD, remain compatible with the core FAIRmat NeXus schema logic.

To support FAIR data export in these advanced workflows, only moderate extensions to the existing metadata structure are needed. For CVD, this involves capturing gas-phase chemistry and surface coverage; for PLA, transient field data and phase evolution must be stored. In both cases, the foundational NeXus format and the semantic flexibility of `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` provide a solid basis for integration [5].

Future directions include:

- Extending the NeXus schema with time-resolved and phase-aware descriptors;

- Incorporating advanced provenance tracking for multi-step simulations;
- Linking simulation metadata to experimental datasets (e.g., TR-reflectivity, TEM);
- Enabling automated ingestion into open platforms such as NOMAD [6].

In summary, while the present thesis has focused exclusively on PVD simulations, the methodological, semantic, and technical infrastructure developed here lays the foundation for interoperable, FAIR-compliant simulation workflows across the full range of thin-film processing technologies.

Chapter 6

FAIR Data Management of PVD Simulations in 3C-SiC Using MulSKIPS

6.1 Introduction

The increasing complexity and resolution of atomistic simulations in materials science calls for structured, standardized data management to ensure reproducibility, accessibility, and long-term reuse. In this context, the FAIR (Findable, Accessible, Interoperable, Reusable) data principles provide a unifying framework for developing interoperable simulation workflows aligned with the goals of the European Open Science Cloud (EOSC) and initiatives such as FAIRmat and NFFA-DI.

This chapter focuses on FAIR Data Management applied to Physical Vapor Deposition (PVD) simulations on 3C-SiC (cubic silicon carbide) carried out using the MulSKIPS simulation environment. These simulations, based on a kinetic Monte Carlo on super-lattice (KMCsL) approach, provide atomic-resolution insight into growth kinetics, surface morphology evolution, and extended defect dynamics under experimentally relevant conditions.

To bridge the gap between complex raw simulation data and interoperable data infrastructures, we developed a Python-based parser that automatically extracts structured metadata from MulSKIPS outputs and generates NeXus files compliant with FAIRmat-contributed definitions. This FAIRification process allows PVD simulation data to be preserved, validated, and shared within open platforms such as NOMAD and EOSC.

The central contribution of this work is the design and implementation of an integrated pipeline for transforming domain-specific simulation out-

puts into machine-actionable and community-aligned data packages. In the following sections, we describe the underlying physical modeling of PVD in MulSKIPS, the role of extended defects in 3C-SiC growth, and the architecture of the metadata extraction and NeXus generation workflow.

6.2 Physics-Based Modeling of PVD Growth in 3C-SiC

The simulation of Physical Vapor Deposition (PVD) processes at the atomic scale requires a faithful description of surface kinetics, defect formation, and crystallographic transitions under out-of-equilibrium conditions. The MulSKIPS environment, based on a Kinetic Monte Carlo on Super-Lattice (KM-CsL) formulation, provides an advanced modeling framework to meet these requirements for materials with sp^3 bonding symmetry, such as 3C-SiC.

In MulSKIPS, the ideal zinc-blende lattice is embedded within a denser super-lattice, allowing the system to dynamically accommodate both regular and defective local configurations. Atomic events are defined as stochastic transitions (e.g., deposition or evaporation) with rates calibrated using *ab initio* energetics and thermodynamic data from sublimation equilibria. This formulation enables the simulation of growth processes that span thousands of Monte Carlo steps and involve millions of lattice sites, capturing long-range and time-dependent structural evolution.

The formalism supports the independent kinetics of the constituent atoms (Si and C), which is essential to model defect generation and propagation in compound semiconductors. Crucially, extended defects such as stacking faults (SFs) and antiphase boundaries (APBs) can arise spontaneously during the simulation due to local asymmetries and instabilities in the surface morphology. These defects play a pivotal role in determining the final microstructure and device-relevant properties of 3C-SiC films.

Recent developments have demonstrated the capability of the MulSKIPS framework to simulate not only point defect statistics but also the detailed growth and annihilation of extended defects. In particular, the work by La Magna et al. [2] and Fisicaro et al. [3] showed that the KMCsL approach can reproduce experimentally observed APB motion, SF nucleation on {111} facets, and the statistical formation of defect clusters during epitaxial growth.

These results are further supported by the multiscale framework presented in Calogero et al. [8], which couples environmental thermal gradients to local atomistic kinetics, providing a digital twin representation of realistic PVD growth chambers. Within this context, the MulSKIPS code serves as

the atomistic engine for capturing kinetic pathways and structural outcomes under various thermal and compositional gradients.

The ability to simulate growth processes on faceted 3C-SiC substrates, track the evolution of stacking disorder, and quantify growth rates and surface roughness makes MulSKIPS a suitable platform for both physics-based process design and data-driven FAIRification workflows. The following section describes how defect evolution is handled within the simulation outputs and why it is essential for metadata extraction and downstream FAIR data generation.

6.3 The Role of Extended Defects in MulSKIPS Simulations

One of the distinguishing features of the MulSKIPS KMCsL framework is its intrinsic ability to generate and evolve extended crystallographic defects during atomistic PVD simulations. Unlike classical lattice KMC methods, which often lack the resolution or flexibility to model defect dynamics, the superlattice formulation in MulSKIPS allows for the spontaneous formation of stacking faults (SFs), antiphase boundaries (APBs), and related microstructural phenomena.

In the specific case of 3C-SiC, these extended defects are not only expected but are key indicators of realistic growth behavior. The metastable nature of the cubic polytype, combined with the similar energetics of alternative stacking sequences (e.g., hexagonal inclusions), results in a strong sensitivity to local growth conditions. As demonstrated in [3], APBs arise from the merging of out-of-phase domains and act as seeds for the nucleation of SFs along $\{111\}$ facets. These stacking faults can extend into the bulk or remain confined near the growth surface, depending on local kinetic asymmetries.

Simulation snapshots confirm that APBs can drift laterally under asymmetric deposition rates on Si- and C-terminated $\{111\}$ surfaces, while SFs evolve independently, forming tetrahedral fault structures or terminating upon encountering other extended defects. The process is highly stochastic, with different defect topologies emerging from identical initial conditions depending on the random sequence of Monte Carlo events.

From a data management perspective, the presence and evolution of these defects is not a side effect but a primary simulation outcome. Information such as:

- number and type of extended defects,
- spatial distribution within the growing film,

- correlation with local growth rates and facet orientation,

are all valuable metadata to be preserved and FAIRified.

Furthermore, since these features are dynamically generated by the simulation rather than imposed by boundary conditions, they provide an unbiased view of the kinetic processes underlying defect formation in 3C-SiC. This makes the corresponding metadata ideal for downstream tasks such as:

- experimental validation (e.g., HRTEM or CAFM comparison),
- training of machine learning models for defect prediction,
- indexing in materials data repositories.

In the next section, we describe how these defect-related outputs, along with other key physical observables, are extracted and transformed into structured metadata by an automated parsing pipeline developed specifically for MulSKIPS simulations.

6.4 Automated Metadata Extraction and FAIRification

To make the rich and heterogeneous output of MulSKIPS simulations usable in a FAIR context, we developed a dedicated parser in Python that automates the extraction, structuring, and transformation of simulation metadata. This tool bridges the gap between domain-specific output formats (e.g., plain-text logs, `.xyz` trajectories, morphology profiles) and standardized, machine-actionable representations suitable for archival in FAIR data infrastructures.

The parser was designed to operate in post-processing, taking as input a directory of simulation outputs and producing as output:

- a structured Python dictionary containing harmonized metadata,
- a NeXus file compliant with FAIRmat's `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` definitions.

Key functions of the parser include:

1. **Simulation context reconstruction:** reading log files and control parameters to infer simulation temperature, substrate type, deposition fluxes, supercell dimensions, and random seed.

2. **Morphology and growth metrics:** parsing `surf_height.txt` to compute average surface height, roughness, and layer completion dynamics.
3. **Trajectory inspection:** analyzing the `.xyz` files to extract atomic configurations, detect point/extended defects, and count under-coordinated atoms.
4. **Defect-aware robustness:** tolerating missing or malformed files and applying fallback strategies to preserve partial data.
5. **Unit normalization and schema mapping:** converting raw quantities into SI units and aligning variable names and types with FAIRmat definitions.

In addition to extracting physical observables, the parser includes meta-data relevant to reproducibility, such as:

- parser version and git commit hash,
- timestamp and host system,
- execution environment (Python version, library dependencies).

The modular architecture of the parser ensures extensibility and robustness, enabling future integration of additional data sources (e.g., temperature profiles, defect topologies) and support for upcoming NeXus extensions.

This tool not only reduces manual labor but guarantees that every MulSKIPS simulation, once completed, can be FAIRified in a consistent and reproducible manner. In the next section, we describe how the structured metadata is serialized into NeXus HDF5 files, making it suitable for validation and publication through FAIRmat and NOMAD repositories.

6.5 NeXus Structure and File Generation

Once structured metadata is extracted from MulSKIPS outputs, it is mapped into the NeXus data format using the FAIRmat-contributed class definitions `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results`. These definitions are specifically designed to support the representation of atomistic simulations of microstructure evolution in materials science, aligning with the broader NeXus standard and its HDF5-based hierarchical structure.

Simulation Configuration: NXmicrostructure_imm_config

This group captures the initial state and parameters of the simulation, including:

- `substrate_type` (e.g., SiC-3C),
- `temperature` (Kelvin),
- `flux_species` and flux rates (in atoms per cm² per s),
- `supercell_dimensions` (in nm),
- `simulation_seed` (integer),
- `event_rates` for deposition and evaporation.

Simulation Results: NXmicrostructure_imm_results

This group describes the key outcomes of the simulation, including:

- `growth_rate` (nm/h),
- `surface_roughness` and `mean_height`,
- `defect_counts` by type (e.g., SF, APB),
- time series of `surface_height`,
- references to trajectory files (e.g., compressed `.xyz`).

Implementation and Validation

The NeXus file is generated using the `h5py` Python library. Data entries are created as datasets or attributes under the corresponding NeXus groups, and are annotated with units, descriptions, and types. Each file is validated against the schema definitions published by FAIRmat at:

https://fairmat-nfdi.github.io/nexus_definitions/classes/contributed_definitions/NXmicrostructure_imm_config.html
https://fairmat-nfdi.github.io/nexus_definitions/classes/contributed_definitions/NXmicrostructure_imm_results.html

The result is a single HDF5-based `.nxs` file that encapsulates the entire simulation context and results in a portable, self-describing, and machine-interpretable format. These files are compatible with the NOMAD data infrastructure, can be indexed in EOSC repositories, and support integration with downstream tools for analysis, visualization, or machine learning.

The following section illustrates the application of this infrastructure to a representative case study.

6.6 Case Study: Sample 153

To validate the full FAIRification pipeline described above, we present a representative use case involving the simulation `SiC_sample_153`, performed on a SiC-3C substrate under realistic PVD conditions.

Simulation Context

The MulSKIPS simulation was configured to reproduce a PVD process in which silicon and carbon atoms impinge on a flat (001) 3C-SiC substrate. The temperature was set to 1273 K, and the atomic fluxes were calibrated based on equilibrium vapor pressures of SiC gas species, consistent with the settings used in [8]. The supercell dimensions were approximately $20 \times 20 \times 40 \text{ nm}^3$, with periodic boundary conditions in the lateral directions.

Observed Results

The output of the simulation revealed the typical signatures of kinetically driven growth on faceted surfaces, including:

- a net growth rate of $162 \pm 4 \text{ nm/h}$,
- layer-by-layer deposition during the early stages, transitioning to island growth,
- nucleation of antiphase boundaries (APBs) along the $\langle 110 \rangle$ directions,
- formation of multiple triple stacking faults (SF_{h3i}) propagating along $\{111\}$ planes.

The parser extracted these observations by analyzing both surface morphology data (`surf_height.txt`) and trajectory files. The presence of extended defects was inferred from counts of under-coordinated atoms and deviations in stacking sequences. Surface roughness was estimated to be $\sim 1.3 \text{ nm}$ (rms) after 2.4 simulated seconds.

Generated NeXus File

The corresponding NeXus file `SiC_sample_153.nxs` was automatically generated using the parser and writer. It contains:

- a full `NXmicrostructure_imm_config` group specifying the simulation setup,
- a `NXmicrostructure_imm_results` group detailing growth rates, roughness, and a summary of detected defects,
- references to raw data files (compressed and linked via relative paths),
- compliance metadata for reproducibility (parser version, timestamp, host system).

The file was validated using FAIRmat’s schema checking tools and successfully imported into NOMAD’s staging environment for internal testing. Manual inspection using HDF5 viewers confirmed structural integrity and correctness of units and annotations.

This case study demonstrates how FAIR data practices can be applied even to complex atomistic simulations, ensuring that detailed physical insights such as defect evolution are preserved and exposed to the broader scientific community.

6.7 Impact and Perspectives

The development of a FAIR-compliant data workflow for atomistic PVD simulations using MulSKIPS represents a significant advancement in the digitalization of computational materials science. By integrating metadata extraction, NeXus generation, and schema validation into a coherent pipeline, this work enables:

- **Reproducibility:** All simulation parameters and results are encoded in a self-describing, portable format that can be revisited and reanalyzed over time.
- **Interoperability:** Compliance with FAIRmat-contributed NeXus definitions ensures compatibility with data infrastructures such as NOMAD, EOSC, and NFFA-DI.
- **Knowledge extraction:** Structured metadata facilitates advanced post-processing, including machine learning analysis, defect statistics, and data fusion with experimental measurements.

- **Scalability:** The parser is designed to handle entire directories of simulations, paving the way for high-throughput FAIRification.

Although this work focused on PVD simulations, the architecture is readily extensible to other processes supported by MulSKIPS, such as Chemical Vapor Deposition (CVD) and Pulsed Laser Annealing (PLA), which require the integration of thermochemical models and dynamic environmental fields [4, 8]. In future developments, real-time NeXus generation during simulation runs may be introduced to support in situ monitoring, digital twin architectures, and live validation.

From a broader perspective, this infrastructure contributes to the goals of the FAIRmat project and the EOSC ecosystem by enabling the standardized, automated, and scalable publication of simulation data. It empowers researchers to go beyond file dumps, transforming simulation output into usable scientific assets ready for reuse, comparison, and integration.

The methods and tools developed here are already being applied to other samples and parameter regimes, and will form the basis for an open-source package to support FAIRification in atomistic growth simulations.

Chapter 7

Conclusions and Future Perspectives

7.1 Conclusions

This thesis demonstrated the feasibility and effectiveness of applying FAIR data management principles to advanced atomistic simulations of PVD processes, with a focus on 3C-SiC growth using the MulSKIPS framework.

By leveraging a physically grounded Kinetic Monte Carlo on super-lattice approach, the simulations captured the dynamic evolution of extended defects such as antiphase boundaries (APBs) and stacking faults (SFs), which are known to critically influence the material quality and device performance of Group IV semiconductors. The simulation results showed good agreement with known experimental phenomena, confirming the robustness of the modeling strategy [2, 3].

A central technical achievement was the development of a modular Python-based pipeline for automatic metadata extraction and NeXus file generation. This parser enables the transformation of domain-specific simulation outputs into standardized, interoperable data formats aligned with FAIRmat definitions (`NXmicrostructure_imm_config` and `NXmicrostructure_imm_results`) [4]. Its integration into the simulation workflow ensures that rich scientific outputs are preserved and ready for reuse in platforms such as NOMAD and EOSC.

The infrastructure has been validated through detailed case studies, including the FAIRification of a PVD simulation on a 3C-SiC substrate (sample 153), resulting in a fully compliant and portable NeXus file. This work provides a concrete and extensible foundation for future FAIR simulation pipelines in computational materials science.

7.2 Future Perspectives

Building on the results and infrastructure developed in this thesis, several promising directions emerge:

- **Generalization to other growth regimes:** Although the current parser was designed for PVD, the MulSKIPS engine also supports CVD and PLA. Extending the parser to incorporate these regimes—possibly using thermodynamic simulators for gas-phase reactions—will enhance its generality [8].
- **On-the-fly FAIRification:** Real-time generation of NeXus files during simulations can support live monitoring, in situ validation, and integration into digital twin frameworks [8].
- **Integration with experimental workflows:** The structured meta-data enables direct comparison with AFM, TEM, and other characterization datasets. A future avenue is the development of cross-referenced data bundles combining simulation and experimental NeXus files under a shared identifier.
- **Machine learning and data mining:** The FAIRified simulation data opens the door to ML-based classification of defect topologies, surrogate models for growth rate prediction, and discovery of processing-structure-property relationships.
- **Open-source dissemination and user support:** Publishing the parser and related tools as an open FAIRification package will support adoption by other simulation groups and alignment with ongoing FAIRmat and NOMAD efforts.
- **EOSC/NFFA-DI Integration:** Embedding this pipeline within the NFFA-DI metadata infrastructure would provide simulation users with a direct path to compliant data publication and discovery.

In conclusion, this work represents a tangible contribution to the FAIRification of computational materials science. It bridges simulation physics with digital infrastructure, and sets the stage for scalable, interoperable, and reusable data practices in the era of open science.

Appendix A

Simulation Setup

The following code initializes the simulation environment, imports the required modules (including the `pymulskips` interface), defines utility functions, and dynamically asks for input values such as the sample ID, box dimensions, random seed, and thermal parameters.

Listing A.1: Simulation Setup and Input Parameters

```
# Parser unificato e robusto per run_PVD_SiC + analyze_PVD_SiC
import os, subprocess, datetime, traceback
import numpy as np
import h5py
from pymulskips import setuprun, process
from pymulskips.analyze import analyze_growth_rate
from analyze_PVD_SiC import count_vacancies

# Funzione helper per dataset con unit
def set_ds(group, name, value, unit=None):
    ds = group.create_dataset(name, data=value)
    if unit:
        ds.attrs['units'] = unit
    return ds

# === INPUT DINAMICO ===
def ask_value(prompt, default, cast):
    val = input(f"{prompt}_[default:{default}]:")
    return cast(val) if val.strip() else default

sample_id = ask_value("ID_del_campione", '153', str)
lenx = ask_value("Lunghezza_X", 120, int)
```

```
leny = ask_value("Lunghezza_Y", 120, int)
lenz = ask_value("Lunghezza_Z", 480, int)
randseed = ask_value("Seed_random", 9117116, int)
newdT = ask_value("Delta_T_aggiuntivo_(newdT)", 0, float)
dT = ask_value("Delta_T_da_aggiungere_a_Tsource", 100, float)
```

A.1 Execution: Simulation with run_PVD_SiC.py

This section shows the execution script for launching the MulSKIPS simulation with parameters obtained from experimental calibration. It sets up the lattice size, thermal parameters (source and seed temperatures), and growth rate.

Listing A.2: Launching the MulSKIPS PVD Simulation

```
import sys
import os
sys.path.append(os.path.abspath(os.path.join(os.path.dirname(__file__),
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size': 12})

import pymulskips
print(pymulskips.__path__)
from pymulskips import setuprun, process

# Dataset di riferimento
data_samples = {
    '153': {
        'Power': 6.10,
        'T_seed_middle': 2072.24,
        'T_seed_edge': 0,
        'T_source_middle': 2080.20,
        'T_source_edge': 2080.20,
        'Exp-Growth-Rate': 287
    }
}

# Compilazione MulSKIPS
execpath = '/Users/filipporuberto/miniforge3/envs/mulskips_env/MulSKIP
lenx, leny, lenz = 120, 120, 120
```

```

setuprun.setup_mulskips_src(execpath, lenx, leny, lenz)

# Setup parametri
randseed = 9117116
newdT = 0
Tseed = data_samples['153']['T_seed_middle'] + 273.15 + newdT
Tsource_0 = data_samples['153']['T_source_middle'] + 273.15 + newdT
dT = 100
Tsource = Tsource_0 + dT
gr = data_samples['153']['Exp-Growth-Rate']

```

A.2 Post-Processing and Analysis: analyze_PVD_SiC.py

This module processes the output of the simulation to evaluate physical parameters like the growth rate and defect density (vacancies). It also prepares visualizations to compare with experimental data.

Listing A.3: Post-Processing Simulation Output

```

import math
import numpy as np
import matplotlib.pyplot as plt
plt.rcParams.update({'font.size': 12})
import os, sys, glob, subprocess
from pymulskips import analyze
from matplotlib.ticker import FormatStrFormatter

margin = 10

# Extract exponents and coefficients for LaTeX formatting
def sci_notation(value):
    coeff, exponent = f"{value:.3e}".split("e")
    return coeff, int(exponent)

# Dataset referenced from experimental growth rates (Peter Wellmann, 2019)
data_samples = {}
data_samples['151'] = {'Power': 5.45, 'T_seed_middle': 1986.26, 'T_seed_edge': 1986.26,
                       'T_source_middle': 1994.14, 'T_source_edge': 1994.14,
                       'Exp-Growth-Rate': 101}

# ...

```

A.3 NeXus Export: FAIRmat-compliant .nxs File Generation

The final part of the script exports the simulation configuration and results into a structured NeXus file compliant with the FAIRmat definitions `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results`. The `h5py` library is used for HDF5 structure creation, and unit-aware datasets are built with a helper function.

Listing A.4: Writing NXmicrostructure NeXus Output

```
import h5py

def set_ds(group, name, value, unit=None):
    ds = group.create_dataset(name, data=value)
    if unit:
        ds.attrs['units'] = unit
    return ds

# Esempio semplificato di struttura .nxs
with h5py.File(f"SiC_sample_{sample_id}.nxs", "w") as f:
    config = f.create_group("NXmicrostructure_imm_config")
    set_ds(config, "sample_id", sample_id)
    set_ds(config, "dimensions", [lenx, leny, lenz], "nm")
    set_ds(config, "random_seed", randseed)

    results = f.create_group("NXmicrostructure_imm_results")
    set_ds(results, "growth_rate", 1.23, " m /h")
    set_ds(results, "vacancies", 142)
    set_ds(results, "simulation_time", 3600, "s")
```

Appendix B

FAIRmat-Compliant NeXus Output and Mapping

This appendix presents both the full internal structure of a representative NeXus file generated from the MulSKIPS workflow and a detailed mapping of its contents to the FAIRmat NeXus definitions. The file analyzed is `SiC_sample_153.nxs`, built around the `NXmicrostructure_imm_config` and `NXmicrostructure_imm_results` definitions as published at https://fairmat-nfdi.github.io/nexus_definitions/.

B.1 Complete File Structure: `SiC_sample_153.nxs`

Listing B.1: NeXus File Structure Dump

```
[Group] NXmicrostructure_imm_config
- KMC_lattice_constant: 0.03633 [nm]
- additional_temperature_shift: 0 [K]
- alat_angstrom: 5.43 [Angstrom]
[Group] analysis
- Nexclude: 3
- bin_size: 0.001 [cm]
- surface_roughness: 0.5
- deposition_time: 0.9844 [s]
- experiment_identifier: 'PVD-SiC-Run-153'
[Group] geometry
- lenx: 120 [lattice units]
- leny: 120 [lattice units]
- lenz: 2400 [lattice units]
- initial_temperature_source: 2353.35 [K]
```

```

— output_steps: 90
— power: 6.1 [W]
— precursors: [ 'Si ', 'Si2C ', 'SiC2 ' ]
— random_seed: 9117116
— run_directory: 'parser/data-0.0-0-9117116'
[Group] simulation_parameters
  — ExitStrategy: 'Time'
  — PtransZig: 0.93
  — SaveCoo: 'False'
  — Seed_box: [48, 0, 0]
  — setup_only: 'False'
— simulation_type: 'PVD'
— substrate_material: 'SiC-3C'
— target_thickness: 0.07848 [mm]
— temperature_difference: 0.0 [K]
— temperature_seed: 2345.39 [K]
— temperature_source: 2353.35 [K]
— timestamp: '2025-04-30 19:03:12.000366'

[Group] NXmicrostructure_imm_results
  — alat_used: 5.43 [Angstrom]
  — analysis_time: '2025-04-30 19:03:12.002392'
  — avg_surface_height: 0.0 [micron]
  — experimental_growth_rate: 287 [micron/hour]
  — growth_rate: 0.0 [micron/hour]
  — growth_rate_method: 'polyfit'
  — num_outputs: 90
  — results_txt: 'null'
  — surf_height_txt: 'null'
  — target_thickness: 0.07848 [mm]
  — total_time: 0.9844 [s]
  — vacancies_total: -1
  — vacancy_CAV, CV, SAV, SV, XV, tot: arrays with 90 values [dimension]

```

B.2 Mapping to FAIRmat NeXus Definitions

NXmicrostructure_imm_config

NeXus Field	FAIRmat Definition	Description
KMC_lattice_constant	lattice_constant	KMC lattice constant in nm
alat_angstrom	alat_ang	Conventional lattice parameter
geometry/lenx, leny, lenz	geometry	Box size in lattice units
initial_temperature_source	Tsource	Initial source temperature (K)
temperature_source	Tsource_final	Final source temperature (K)
temperature_seed	Tseed	Substrate (seed) temperature
temperature_difference	deltaT	Temperature offset between source and substrate
power	P	Applied thermal power (W)
precursors	gas_species	List of gaseous precursors
random_seed	random_seed	RNG seed for reproducibility
experiment_identifier	experiment_id	Simulation unique identifier
substrate_material	substrate	Substrate material label
simulation_type	simulation_mode	Type of process (PVD, CVD, etc.)
target_thickness	growth_target	Final film thickness (nm)
output_steps	n_steps	Total output steps
timestamp	creation_datetime	Time of simulation execution
run_directory	working_dir	Simulation output directory
simulation_parameters/ExitStrategy	termination_criterion	Simulation stopping condition
simulation_parameters/PtransZig	defect_probability	Probability of polytype transition
simulation_parameters/SaveCoo	save_xyz	Whether atomic coords are saved
simulation_parameters/Seed_box	seed_origin	Index of seed origin (XYZ)

NXmicrostructure_imm_results

NeXus Field	FAIRmat Definition	Description
alat_used	alat_used	Lattice constant used in pos
analysis_time	analysis_datetime	Timestamp of analysis stage
avg_surface_height	surface_height_avg	Average height of topmost la
growth_rate	growth_rate	Computed rate from height
growth_rate_method	growth_rate_method	Method used for rate calcula
experimental_growth_rate	reference_growth_rate	Measured value from experin
deposition_time	deposition_time	Time reported from engine (
total_time	growth_duration	Total simulated time (s)
num_outputs	n_steps	Number of steps used for ou
target_thickness	growth_target	Final simulated thickness (m
results_txt, surf_height_txt	log_files	Optional logs exported from
vacancies_total	vacancy_count	Scalar count of all vacancies
vacancy_tot, CV, SAV, SV, XV, CAV	vacancy_types	Time-dependent vacancy cat

All mappings conform to the FAIRmat schema definitions as of April 2025.

Appendix C

Experimental and Simulation Setup Details

This appendix reports all numerical parameters used to configure the MulSKIPS simulations and the corresponding experimental references from which they are derived. The simulation emulates the growth of a 3C-SiC crystal under Physical Vapor Deposition (PVD) conditions, with temperature and power calibrated on real growth runs.

C.1 Substrate and Material Properties

- Substrate: 3C-SiC (cubic silicon carbide)
- Orientation: (100)
- Simulation box size: $len_x = 120$, $len_y = 120$, $len_z = 2400$ [lattice units]
- Superlattice spacing: 0.03633 nm
- Conventional lattice parameter: 5.43 Å

C.2 Thermal and Process Parameters

- Source temperature: $T_{source_middle} = 2080.20\text{ }^{\circ}\text{C} + 273.15 + dT$
- Seed temperature: $T_{seed_middle} = 2072.24\text{ }^{\circ}\text{C} + 273.15 + newdT$
- Power input: 6.10 W

- Expected growth rate: 287 $\mu\text{m}/\text{h}$
- Time simulated: ~ 0.98 s

C.3 Precursors and Gas Phase Composition

- Main species: Si, Si_2C , SiC_2
- Composition corresponds to a silicon-rich vapor phase
- No hydrogen plasma or additional carrier gas included

C.4 Relevant References

- [8] – Multiscale PVD and PLA simulations with MulSKIPS
- [2] – Growth kinetics and stacking faults in Group IV semiconductors
- [4] – KMC simulations for defect generation in CVD/PVD
- [3] – Superlattice KMC for stacking faults and antiphase boundaries in 3C-SiC

Appendix D

Makefile for MulSKIPS Compilation

This appendix provides the `Makefile` used to compile the MulSKIPS environment from modular Fortran source files. The compilation was performed on a **MacBook Air M1** running macOS with **Apple Silicon (arm64)** architecture.

System Specifications

The compilation environment consisted of:

- Architecture: `arm64` (Apple M1)
- Compiler: `gfortran 14` via Homebrew
- BLAS library: `OpenBLAS 0.3.29` (ARM64 build)
- GCC libraries: `gcc 14.2.0_1`
- OS: macOS Ventura

To ensure compatibility with the ARM64 platform, correct linking paths and runtime library paths were explicitly set.

Directory Structure

The source code is organized into several subdirectories:

- `modules/`: Fortran modules with type and system definitions

- `initial/`: Geometry initialization routines
- `inpout/`: Input/output functions, file handling
- `lattice/`: Lattice neighbor analysis
- `mcevent/`: Monte Carlo event management

Makefile

Listing D.1: Makefile for compiling `mulskips.e`

```
# Variables
DESTDIR = .
MODDIR  = $(DESTDIR)/modules
INOUDIR = $(DESTDIR)/inpout
INITDIR  = $(DESTDIR)/initial
MCEVDIR  = $(DESTDIR)/mcevent
LATDIR   = $(DESTDIR)/lattice

FC = /usr/local/bin/gfortran-14
FFLAGS = -g
LIBS = -lquadmath -lgfortran -lgcc_s.1 \
      -L/usr/local/Cellar/openblas/0.3.29/lib -lopenblas \
      -lpthread -Wl,-rpath,/usr/local/Cellar/gcc/14.2.0 \
      _1/lib/gcc/14

# Source files
SOURCE = \
$(MODDIR)/defdtype.f $(MODDIR)/defsystem.f $(MODDIR)/\
definitions.f \
$(INITDIR)/SetAPB.f $(INITDIR)/SetInvPC.f $(INITDIR)/\
SetInvPwAPB.f $(INITDIR)/SetNCSp.f $(INITDIR)/\
SetNCSpSi.f \
$(INITDIR)/SetAPBSym.f $(INITDIR)/SetInvP.f $(INITDIR)/\
SetInvPwOnlyCorSi.f $(INITDIR)/SetSiC3C.f \
$(INITDIR)/SetSiCSF.f $(INITDIR)/SetSiGex.f $(INITDIR)/\
SetSiC3Cww.f $(INITDIR)/SetSiC3Cwwaperture.f \
$(INITDIR)/SetSi.f $(INITDIR)/SetSiFINFET.f $(INITDIR)/\
SetCAD.f $(INITDIR)/SetT.f $(INITDIR)/SetShape.f \
$(INITDIR)/SetAPBSymZimb1.f $(INITDIR)/SetInvPOff.f $(\
INITDIR)/SetInvPwOnlySiorC.f \
```

```

$(INITDIR)/SetTrenchSi.f $(INITDIR)/SetTrench.f $(
    INITDIR)/SetAPBSymZimb.f $(INITDIR)/SetInvPSi.f $(
    INITDIR)/SetNC.f \
$(INOUDIR)/FileClose.f $(INOUDIR)/GetOutputFileName.f $
    (INOUDIR)/WriteMolMolXYZFile.f $(INOUDIR)/FileOpen.f
    $(INOUDIR)/WriteMolMolSource.f \
$(LATTDIR)/FIND_NN.f $(LATTDIR)/FIND_SN.f \
$(MCEVDIR)/Add.f $(MCEVDIR)/Evaporation.f $(MCEVDIR)/
    PickTreeEvent.f $(MCEVDIR)/UpdateTree.f \
$(MCEVDIR)/AllocateArrays.f $(MCEVDIR)/Get_Prob.f $(
    MCEVDIR)/random.f $(MCEVDIR)/deposition.f \
$(MCEVDIR)/Get_Prob_Ini.f $(MCEVDIR)/SetProbability.f $
    (MCEVDIR)/Erase.f $(MCEVDIR)/IF_Arm_Choice.f \
$(MCEVDIR)/UpdateCoor.f main.f $(MCEVDIR)/absorption.f
    $(MCEVDIR)/desorption.f \
$(MCEVDIR)/InterpProbFromField.f $(MCEVDIR)/
    SetProbabilityFromField.f

OBJS = $(SOURCE:.f=.o)

# Main target
all: $(DESTDIR)/mulskips.e

$(DESTDIR)/mulskips.e: $(OBJS)
    @echo "Compiling..."
    $(FC) $(FFLAGS) $(OBJS) -o $$@ $(LIBS)
    @echo "Compilation complete."

# Cleanup
clean:
    rm -f $(OBJS) $(DESTDIR)/mulskips.e *.mod

```

Appendix E

Software Configuration

Package	Version
numpy	1.24.4
matplotlib	3.7.3
h5py	3.10.0
pymulskips	custom fork (CNR-IMM, 2025)
scipy	1.11.4
ipykernel	6.25.2
notebook	7.0.6
jupyterlab	4.0.6

Bibliography

- [1] “Mulskips - multiscale kinetic process simulator.”
<https://github.com/MulSKIPS>. Accessed: 2025-04-28.
- [2] A. L. Magna, G. Fisicaro, I. Deretzis, M. Camarda, F. Roccaforte, and F. L. Via, “Simulation of the growth kinetics in group iv compound semiconductors,” *Physica Status Solidi A* **215** no. 15, (2018) 1700801.
- [3] G. Fisicaro, C. Bongiorno, I. Deretzis, F. Giannazzo, F. L. Via, F. Roccaforte, M. Zielinski, M. Zimbone, and A. L. Magna, “Genesis and evolution of extended defects: The role of evolving interface instabilities in cubic sic,” *Applied Physics Reviews* **7** no. 2, (2020) 021402. <https://doi.org/10.1063/1.5132300>.
- [4] D. Raciti, G. Calogero, D. Ricciarelli, R. Anzalone, G. Morale, D. Murabito, I. Deretzis, G. Fisicaro, and A. L. Magna, “Multiscale atomistic modelling of cvd: From gas-phase reactions to lattice defects,” *Materials Science in Semiconductor Processing* **167** (2023) 107792. <https://doi.org/10.1016/j.mssp.2023.107792>.
- [5] “Fairmat nexus definitions.”
https://fairmat-nfdi.github.io/nexus_definitions/. Accessed: 2025-04-28.
- [6] “Nomad repository and archive.” <https://nomad-lab.eu/>. Accessed: 2025-04-28.
- [7] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, *et al.*, “The fair guiding principles for scientific data management and stewardship,” *Scientific Data* **3** (2016) 160018.
- [8] G. Calogero, I. Deretzis, G. Fisicaro, M. Kollmuß, F. L. Via, S. F. Lombardo, M. Schöler, P. J. Wellmann, and A. L. Magna, “Multiscale simulations for defect-controlled processing of group iv materials,”

Crystals **12** no. 12, (2022) 1701.

<https://doi.org/10.3390/cryst12121701>.

- [9] “Fairmat initiative.” <https://fairmat-nfdi.eu/>. Accessed:
2025-04-28.