MIC Hackathon: MCP Server for Theory–Experiment Settings

Vani Nigam*† and Achuth Chandrashekar‡

†*Department of Materials Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA*

‡*Department of Mechanical Engineering, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA*

*E-mail: vnigam@andrew.cmu.edu

Abstract

We present a modular framework powered by large language models (LLMs) that automates and streamlines key tasks in experimental and theoretical workflows using FerroSim and AFM data. The framework integrates digital twin microscopy with computational ferroelectric simulations to enable theory and experiment matching.

Introduction

Self Driving Laboratories(SDL) are being used for accelerating materials discovery and enhanced experimental settings. SDLs are being powered by simulations and Large Language Models(LLMs) for working on the logics of the AI driven experiments.[1] The AI Agents have been developed to develop technology using human in the loop experiments. By lever- aging human expertise through a reward-driven machine learning framework, we automate the optimization process, ensuring consistent, high-quality imaging across diverse samples

and conditions while reducing dependency on operator expertise and minimizing the risk of errors.[2] The simulation is implemented using an object-oriented approach, centering on a lattice-based model that calculates the spatio-temporal evolution of polarization. The computational workflow is divided into three primary phases: initialization, numerical integration, and post-processing.

Numerical Simulation Framework

Task Description

Simulation Initialization. The simulation environment is established through primary setup routines that define physical constraints and spatial geometry.

- setup field(): Initializes temporal parameters and the external driving force. It generates the time vector and a corresponding time-varying sine wave field, returning the global field distribution as a 2D NumPy array for spatially dependent calculations.
- setup lattice(): Instantiates the simulation grid by generating a collection of Lattice objects, each encapsulating local physical properties and state variables of discrete points in the material.
- makeCircle() (Static): Generates a binary mask as a uint8 2D array to define specific domains, facilitating the initialization of heterogeneous polarization states.

Governing Equations and Integration. The core of the simulation relies on solving the Landau–Khalatnikov (L–K) equation to model polarization switching dynamics.

- calcODE(): The primary solver engine that iterates through temporal and spatial dimensions to compute the derivative of polarization (dp/dt) and update polarization states (p_{new}).

- calDeriv(): Implements the mathematical derivative based on the L–K energy land- scape, taking current polarization, neighbor interactions, and external fields as inputs.
- getPNeighbors(): Calculates the sum of polarization values of nearest neighbors at time *t* to provide the necessary coupling term for the derivative calculation.
- knn search() (Static): Utilizes a k-nearest neighbors utility to identify spatial neigh- bors within a radius *R* for efficient spatial lookups.

Execution and Visualization. The runSim() method orchestrates the execution loop and aggregates data into a structured results dictionary. This includes global metrics (total polarization) and local responses (3D AFM tip–sample response). Post-simulation analysis is handled via:

- plot quiver(): Visualizes the orientation and magnitude of polarization across the lattice.
- plot summary(): Displays relationships between applied fields, total polarization, and simulated AFM responses.
- make interactive plot(): Enables dynamic exploration of the polarization switching process.

Table 1: Structure of the results dictionary output from runSim()

Data Key	**Physical Interpretation**
Polarization	1D array of total polarization over time dPolarization
	1D array of the derivative of total polarization
measuredResponse	3D array (*x, y, t*) of simulated AFM response line
polarization values	2D reshaped array for line-scan analysis

Automated Theory–Experiment Matching Loop

To bridge the gap between numerical simulations and experimental observations, the frame- work utilizes an automated loop managed by an LLM agent. By exposing simulation func- tions as tools through the Model Context Protocol (MCP), the system autonomously iterates through data acquisition, simulation, and comparative analysis.

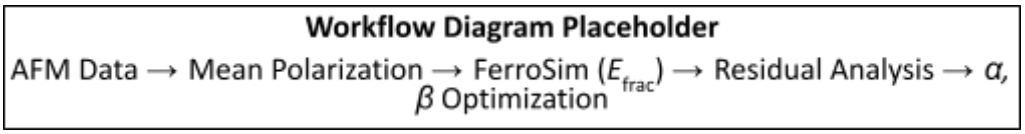

Figure 1: Workflow of the theory–experiment matching loop.

Numerical Simulation Framework

The agent executes the following steps to minimize the error between the simulated response and the physical sample:

1. **Data Extraction:** Retrieve AFM polarization data from an HDF5 file.
2. **Parameter Mapping:** Map the mean polarization to the effective field fraction E_{frac}.
3. **Dynamic Simulation:** Run the Landau–Khalatnikov simulation.
4. **Comparative Analysis:** Compute residuals between simulation and experiment.
5. **Optimization:** Refine free-energy coefficients α and β.

Technology Disclosure

Some parts of the paper were summarized by the Gemini AI assistant to ensure accuracy with the workflow. Some of the server codes for the Ferrosim and AFM were referenced from the github pages https://github.com/pycroscopy/DTMicroscope/tree/main, https:// github.com/ramav87/FerroSim and gemini coding assistant.

Acknowledgments

The authors thank the MIC Hackathon organizers for providing the platform to develop this MCP server integration.

References

Liu, Yu, Roger Proksch, Jason Bemis, et al. “Machine Learning-Based Reward-Driven Tuning of Scanning Probe Microscopy: Toward Fully Automated Microscopy.” ACS Nano 19, no. 21 (2025): 19659–69. https://doi.org/10.1021/acsnano.4c18760.

Mandal, Indrajeet, Jitendra Soni, Mohd Zaki, et al. “Autonomous Microscopy Experiments through Large Language Model Agents.” arXiv:2501.10385. Preprint, arXiv, July 7, 2025. https://doi.org/10.48550/arXiv.2501.10385.