Evolutionary Coding Agents for Autonomous Optimization of Scientific Software and Metallurgical Design

Daniel Marchand[1], Tor S. Haugland[1]
[1]Organization, Address, City, Country (Arial 9)

Daniel.marchand@sintef.no

The rapid advancement of Large Language Models (LLMs) has transformed software development, offering capabilities ranging from code completion to automated refactoring. However, standard LLM usage often suffers from a lack of domain-specific rigor, particularly in high-precision scientific computing where functional correctness and performance are paramount. To address these limitations, recent breakthroughs in "Agentic" AI have coupled LLMs with Evolutionary Algorithms (EAs). In this framework, the LLM acts as a high-dimensional mutation operator—proposing intelligent variations of code—while a rigorous evaluation step filters out incorrect solutions. This methodology, exemplified by the AlphaEvolve agent [1], has recently demonstrated state-of-the-art capabilities in mathematical discovery and algorithmic optimization by treating software improvement as an evolutionary process.

In this work, we leverage the **OpenEvolve** platform [2] to adapt these evolutionary coding agents to the specific demands of materials science and atomic-scale simulations. Rather than developing a new evolutionary engine, we utilize OpenEvolve's distributed controller loop where a "Prompt Sampler" generates candidate modifications based on code history, an ensemble of LLMs synthesizes the functional code, and a domain-specific "Evaluator" rigorously tests performance against scientific benchmarks. This implementation moves beyond simple snippet generation to a closed-loop system capable of iterative refinement and self-correction.

We demonstrate the configuration of this framework for two distinct, high-performance applications. The first targets the optimization of **eT** [3], a state-of-the-art electronic structure program used for coupled cluster simulations. These simulations are computationally intensive, often requiring significant runtime resources for molecular property calculations. The agent architecture is designed to explore optimizations in tensor contraction operations, memory management, and symmetry handling—areas where manual tuning is often intractable. The second application utilizes **Kawin** [4], an open-source Kampmann-Wagner Numerical (KWN) model for phase precipitation. Here, the evolutionary agent is deployed to optimize material design parameters rather than source code structure. By evolving the input parameters—specifically chemical composition and artificial aging temperature cycles—the system seeks to maximize the yield strength of Aluminum-Magnesium-Silicon alloys (AA6xxx).

The implementation establishes the critical integration between the open-source evolutionary core and the complex build systems of scientific simulation tools. By formalizing the evaluation pipelines and fitness functions for these disparate physics domains, we demonstrate a generalized methodology for autonomous scientific software improvement. This framework provides a robust foundation for identifying non-intuitive optimizations in legacy scientific codebases, paving the way for self-optimizing simulations that adapt to new hardware and physical constraints.

References

[1] A. Novikov et al., *AlphaEvolve: A coding agent for scientific and algorithmic discovery*, arXiv preprint arXiv:2506.13131 (2025).

[2] A. Sharma, *OpenEvolve: Open-source implementation of AlphaEvolve*, GitHub Repository (2025). https://github.com/codelion/openevolve.

[3] S. D. Folkestad et al., *eT 1.0: An open source electronic structure program with emphasis on coupled cluster and multilevel methods*, J. Chem. Phys. 152, 184103 (2020).

[4] N. Ury et al., *Kawin: An open source Kampmann–Wagner Numerical (KWN) phase precipitation and coarsening model*, Acta Materialia 255, 118988 (2023).