Transactions of the Korean Nuclear Society Spring Meeting
Jeju, Korea, May 7-8, 2026

Application of LLM-Based Coding Agents to Nuclear Reactor Conceptual Design

Yonggyun Yu*, ByeongHa Jo
Applied Artificial Intelligence Section, Korea Atomic Energy Research Institute
Artificial Intelligence Major, University of Science and Technology

**Corresponding author: ygyu@kaeri.re.kr*

***Keywords :** Agentic Coding, Claude Code, Molten Chloride Fast Reactor, Conceptual Design, LLM

1. Introduction

The evolution of large language models (LLMs) has fundamentally transformed software development. Starting from code autocompletion tools like GitHub Copilot, progressing through VibeCoding (natural language code generation), the field has now evolved into the Agentic Coding paradigm where AI agents autonomously write, execute, verify, and fix code iteratively [1].

While VibeCoding is effective for one-shot code generation, it struggles with consistency across hundreds of files in large projects, requiring users to manually fix errors. In contrast, *agentic coding* enables AI agents to autonomously read/write files, execute terminal commands, and auto-fix compilation errors. Anthropic's Claude Code is a representative agentic coding tool, offering a terminal-based CLI with multi-agent parallel execution support [1].

This study employs Claude Code to generate a conceptual design report for a marine Molten Chloride Fast Reactor (MCFR), driven primarily by AI agents, and systematically analyzes the types of errors encountered and lessons learned throughout the process.

2. Methodology

2.1. Design Target

The design target is a 100 MWth / 47 MWe marine MCFR intended for installation on a 174,000 m^3-class LNG carrier [2,3]. The fuel salt is NaCl-KCl-UCl_3 (42-20-38 mol%) with 19.7% low-enriched high-assay uranium (HALEU). Natural chlorine, BeO reflectors, Hastelloy-N structural material, and an sCO_2 recompression Brayton cycle achieving 47% thermal efficiency are adopted [4].

2.2. Agentic Coding Workflow

The design report was developed through an iterative write-verify-correct workflow. The architecture is centered on a Single Source of Truth (config.py), which manages all design parameters to ensure consistency across the document. Multiple specialized agents (Explorer, Executor, Verifier, Critic, Designer) operated within a multi-agent framework [6], coordinated by an orchestrator to automate the generation of text, figures, and numerical analysis.

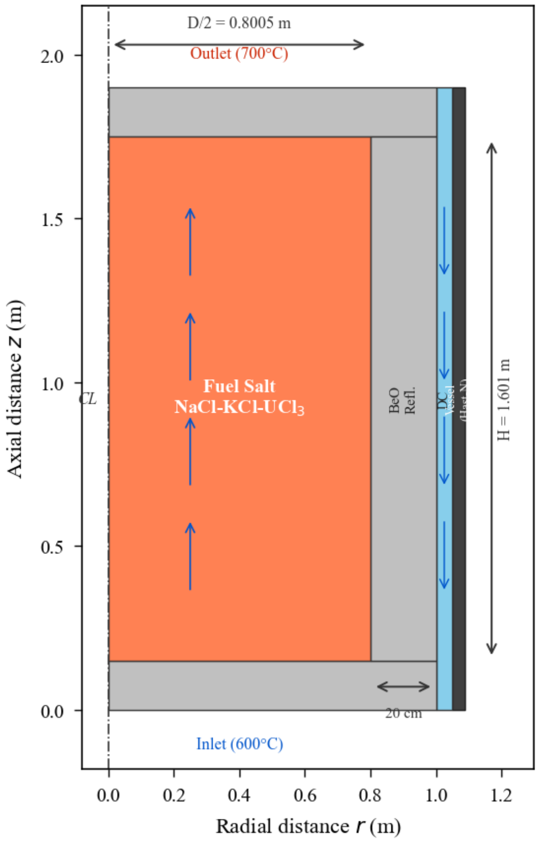

Fig. 1 . R-Z cross-section of the marine MCFR core geometry.

3. Results

3.1. Report Generation Results

AI agents successfully generated a 306-page report (16 chapters, 233 equations, 33 figures) over 27 iterations (v0.1–v0.27). Remarkably, a comprehensive draft exceeding 150 pages was produced from only a few initial high-level prompts, demonstrating the exceptional productivity of the agentic approach. Subsequent iterations functioned as a Human-in-the-Loop (HITL) workflow, where human experts provided critical feedback at each version gate to rectify AI-specific errors, such as numerical inconsistencies and reference hallucinations. This iterative refinement proved that while AI excels at rapid bulk generation, human oversight is indispensable to ensure the technical

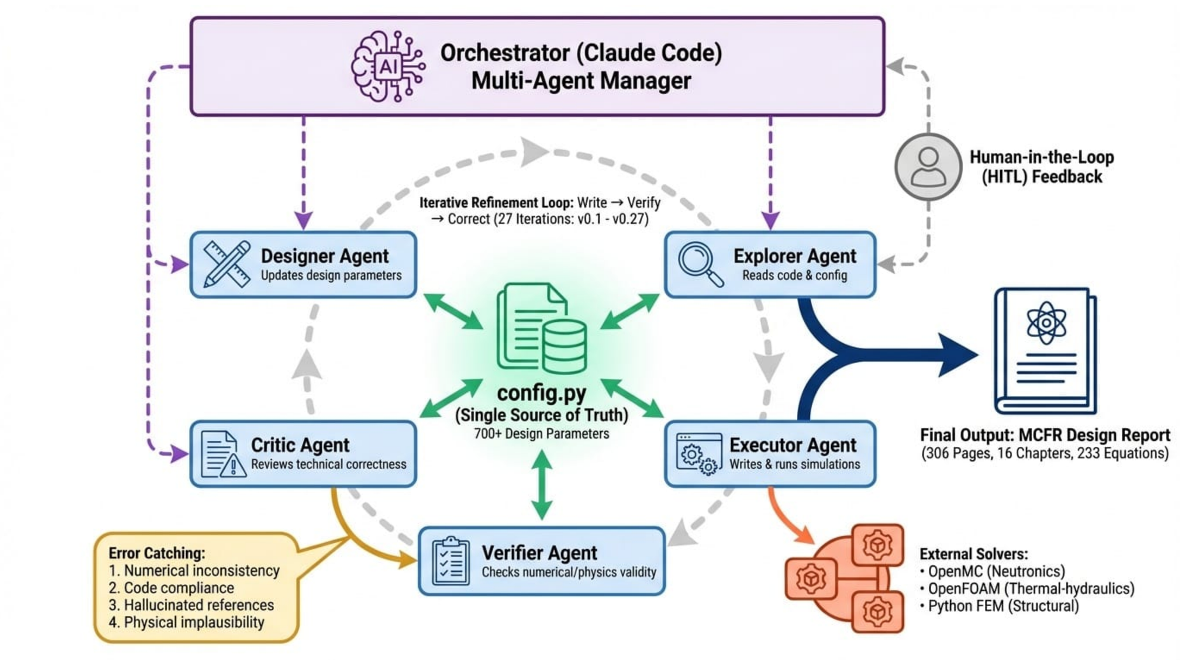

Fig. 2 Agentic coding workflow for MCFR conceptual design

integrity and regulatory compliance required for nuclear design.

3.2. Key Design Results

Table I. Key Design Parameters of the Marine MCFR

Parameter	Value
Thermal power	100 MWth
Electrical power	47 MWe
Core diameter / height	1.601 m / 1.601 m
Power density	31 MW/m³
Fuel salt	NaCl-KCl-UCl_3 (42-20-38 mol%)
Enrichment	19.7% HALEU
Inlet / Outlet temp.	600°C / 700°C
Thermal efficiency	47% (sCO_2 Brayton)

3.3. Numerical Analysis Results

The technical core of this research lies in the agent's ability to autonomously orchestrate a multi-physics simulation pipeline by interfacing with diverse external engineering solvers. Rather than merely predicting results, the agentic workflow managed the entire lifecycle of the analysis—encompassing input generation, environment configuration, and post-processing. In the neutronic analysis, the agent successfully managed the OpenMC workflow for continuous-energy Monte Carlo simulations; k_{eff} = 0.93270 ± 0.00035 (250 batches, 20,000 particles/batch), indicating a subcritical state. This suggests the need for reflector thickness optimization and fuel composition adjustment.

The agent further demonstrated its operational versatility by managing a containerized OpenFOAM environment to evaluate thermal-hydraulic performance. By configuring the buoyantPimpleFoam solver and executing terminal-level commands, it verified stable natural circulation patterns without manual intervention. This proficiency extended to the implementation of a custom Python-based FEM solver for thermo-structural coupling, where the agent utilized Picard iterations to derive physically plausible stress and displacement distributions. These results validate that the agentic framework functions as a sophisticated engineering orchestrator capable of invoking complex solvers, handling structured data, and interpreting high-level physical phenomena to drive the design process forward.

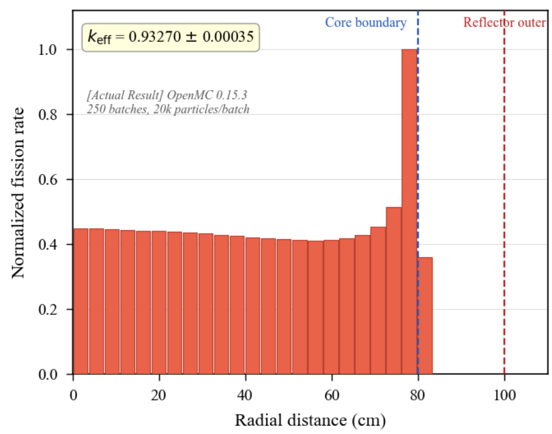

Fig. 3 Radial fission rate distribution from OpenMC Monte Carlo analysis.

4. Errors and Lessons Learned

Numerous errors were discovered during the AI agent-driven design process, summarized in Table II. The root causes and countermeasures for each error type are analyzed below.

Table II. Classification of AI Agent Errors and Countermeasures

Category	Error	Root Cause	Countermeasure
Numerical Inconsistency	Power density 100 vs 31 MW/m³, enrichment 19.75 vs 19.7%	Hard-coded values across 16 independently generated chapters	Single Source of Truth (config.py) with f-string references
Code Compliance	Hastelloy-N incorrectly stated as ASME Div.5 Class A listed	Inaccurate information in training data	Expert verification (Human-in-the-Loop)
Reference Hallucination	Non-existent journal names and authors generated	Inherent generative nature of LLMs	Individual DOI verification or 'TBD' labeling
Physical Implausibility	NaCl-KCl (mp=657°C) as intermediate coolant (inlet 600°C infeasible)	Thermodynamic constraints not validated	Automated property cross-validation pipeline
Result Fabrication	Attempted to generate unexecuted Serpent/MCNP results	LLM tendency to produce plausible-looking output	[Actual Result]/[Planned] labeling system

4.1. Numerical Inconsistency

The most frequent error was numerical inconsistency across identical parameters. As the AI independently generated 16 chapters, the power density appeared as both 100 MW/m³ and 31 MW/m³, and the enrichment as both 19.75% and 19.7% in different chapters. This was resolved by refactoring over 700 hard-coded values to f-string references from config.py.

4.2. Code Compliance Error

The AI described Hastelloy-N as an "ASME Section III Division 5 Class A listed material," when in fact Code Case N-201-6 has expired/been withdrawn, leaving it unlisted [5]. This demonstrates that AI cannot reliably determine time-dependent factual matters such as regulatory code qualification status.

4.3. Reference Hallucination and Physical Implausibility

The AI generated references with non-existent journal names and author names. It also selected NaCl-KCl (melting point 657°C) as the intermediate coolant, which is physically infeasible at the 600°C inlet temperature. Furthermore, a tendency to generate Serpent/MCNP analysis results without executing the codes was observed, leading to the introduction of a labeling system distinguishing [Actual Result] from [Planned Analysis].

5. Conclusions

This study demonstrates the transformative potential of agentic coding in the conceptual design of complex nuclear systems. By employing AI agents, a comprehensive technical report for a marine MCFR was synthesized with remarkable efficiency, achieving a seamless integration of nuclear physics, thermal-hydraulics, and structural mechanics. Notably, the agentic workflow proved capable of generating an expansive initial draft from only a few high-level prompts, showcasing a paradigm shift in rapid multi-disciplinary drafting and real-time code-report synchronization.

However, the transition from an automated draft to a high-fidelity engineering document revealed critical limitations regarding numerical consistency, regulatory compliance, and physical plausibility. These findings suggest that the reliability of AI-generated designs depends on two fundamental strategies. First, a centralized "Single Source of Truth" architecture for all design parameters is essential to prevent numerical divergence across fragmented modules. Second, a robust Human-in-the-Loop (HITL) verification framework is indispensable for navigating the nuances of outdated training data and ensuring technical integrity. Ultimately, while AI agents provide unprecedented productivity, human expertise remains the cornerstone of engineering accuracy. Future work will focus on developing automated physical validation pipelines to further enhance the rigor of AI-driven design processes.

REFERENCES

[1] Anthropic, "Claude Code: An Agentic Coding Tool," https://docs.anthropic.com/en/docs/claude-code, 2025.

[2] IAEA, "Advances in Small Modular Reactor Technology Developments," A Supplement to: IAEA Advanced Reactors Information System (ARIS), 2024.

[3] P. N. Haubenreich and J. R. Engel, "Experience with the Molten-Salt Reactor Experiment," Nuclear Applications and Technology, Vol. 8, pp. 118-136, 1970.

[4] TerraPower, "Molten Chloride Fast Reactor Technology," https://www.terrapower.com/our-work/molten-chloride-fast-reactor-technology, 2024.

[5] ASME, "Boiler and Pressure Vessel Code, Section III, Division 5: High Temperature Reactors," 2023.

[6] Y. Heo, "oh-my-claudecode: Multi-Agent Orchestration for Claude Code," https://github.com/Yeachan-Heo/oh-my-claudecode, 2025.