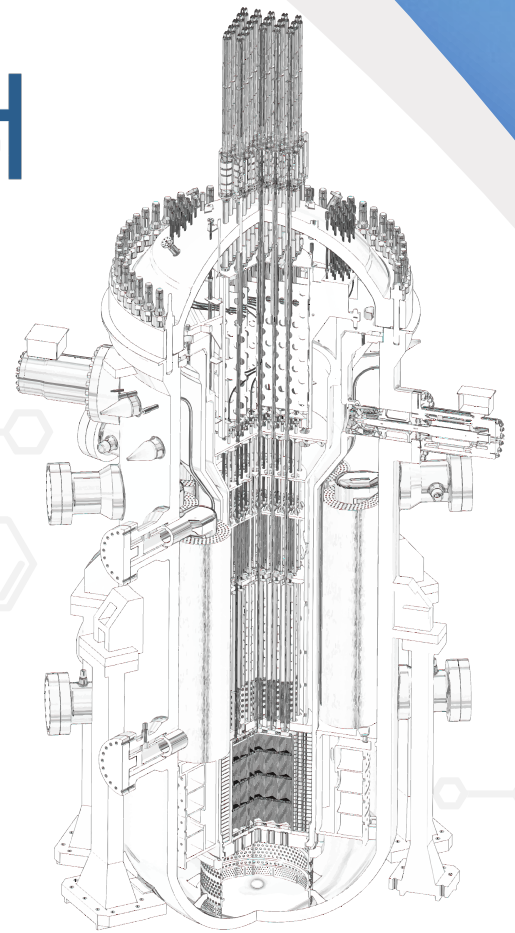
[B] 인공지능이 여는 원자로 해석의 새로운 패러다임

더 나은 세상을 위한 원자력기술,
국민과 세계가 지지하는 **한국원자력연구원**

MSR 3D 다물리 실시간 시뮬레이터 바이브 코딩 구현 사례

한국원자력연구원
장 성 동

2026.10.28.(수)



KAERI 한국원자력연구원 Korea Atomic Energy Research Institute



더 나은 세상을 위한 원자력기술
국민과 세계가 지지하는 한국원자력연구원

KAERI 한국원자력연구원 Korea Atomic Energy Research Institute

CONTENTS

01 개요

02 MSR 3D 다물리 실시간 시뮬레이터

03 에이전틱 엔지니어링(종합 오케스트레이션)

04 요약 및 결언

05 질의응답



[B] 인공지능이 여는 원자로 해석의 새로운 패러다임

01 개요

한국원자력연구원
Korea Atomic Energy Research Institute

01 개요

1. Andrej Karpathy: From Vibe Coding to Agentic Engineering w/ Stephanie Zhan, 26.4.30. (https://www.youtube.com/watch?v=96jN2OCOfLs&t=1160s)
2. 꼭 알아야할 안드레 카파시 30분 인터뷰 완전정리 – AI시대의 필수 인사이트!, 26.5.5. (https://www.youtube.com/watch?v=UNnsS03whx0&t=135s)
3. "AI, 인간 계층 4단계로 나눠", ... 오픈AI 창업자가 쏘아올린 논쟁, 중앙일보, 26.10.13. (https://www.koreadaily.com/article/20261013130045236)

인공지능의 3번째 흐름의 시작

안드레 카파시 인터뷰 내용 中



안드레 카파시(사진)이 지난 11일 X에 올린 AI 4단계 계층론이 AI격차 논쟁을 촉발시켰다. 사진은 지난 5월 앤트로픽에 합류했던 카파시의 모습.

OpenAI TESLA ANTHROP\C

"작년(2025년)에 ChatGPT 정도 수준에서 AI를 경험했다면,
12월(2025년)부터는 다시 봐야 한다."

→ 에이전트 성격의 일관성 있는 워크플로우 작동 시작

바이브 코딩
(Vibe Coding)

- 자연어 프롬프트와 감각적인 지시만으로 LLM에게 코드 작성을 맡긴 후 결과물만 수용해 소프트웨어를 빌드하는 패러다임
- 바닥을 높이는 것
  : 소프트웨어 생산 역량의 **'하한선'**을 대폭 끌어 올림

에이전틱 엔지니어링
(Agentic Engineering)

- AI 에이전트들을 조율하여 엔지니어링 수준을 유지하면서, 복잡한 대규모 소프트웨어를 설계 구축하는 전문 기술 방법론
- 천장을 높이는 것
  : 소프트웨어가 도달할 수 있는 **'상한선'**을 폭발적으로 확장함

프론티어 모델 기반 에이전틱 AI를 직접 활용해보는 경험 자산이 중요!

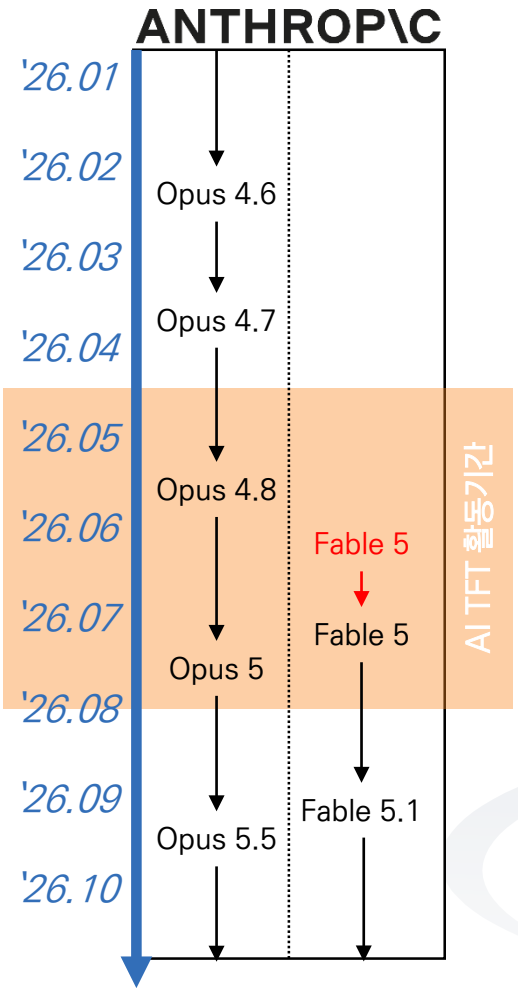
01 개요

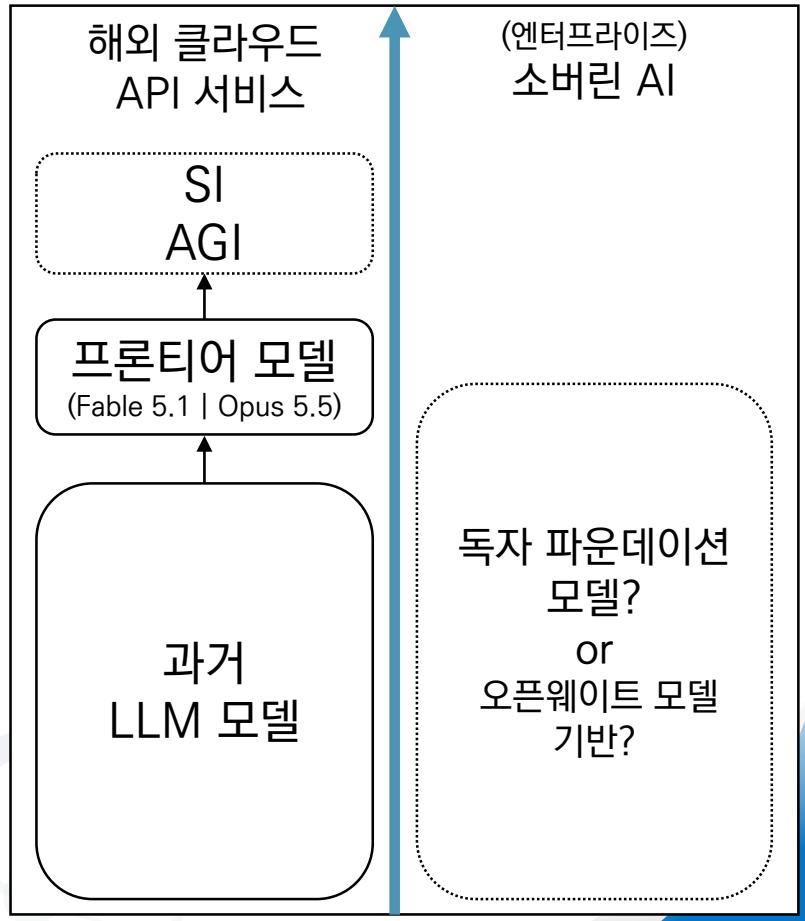
에이전틱 AI 경험의 필요성

프론티어 모델 기반 에이전틱 AI 활용의 제약사항

- 제조 및 첨단 기술(원자력·방산 분야 등)의 **보안 문제**
  - 핵심 설계 도면, 시뮬레이션 코드, 특허 기술 등 유출의 우려
- 프론티어 모델 **수출 통제**의 가능성
  - (예시) FABLE 5 모델
  : (26.06.09) 앤트로픽 MYTHOS급 프론티어 모델 출시
  : (26.06.12) 미국 정부의 FABLE 5 수출 통제(안전장치 우회 가능성)
  : (26.07.01) 안전장치 개선 후 FABLE 5 서비스 재개
- 연구 **분야별 통제** 가능성
  - (예시) 초기 FABLE 5 생물학/화학 키워드의 과도한 차단
  : 현재 정상적인 생화학 연구 관련 질문은 원활하게 처리
  : 화학 · 생물 · 방사능 · 핵 무기 제조, 독성 물질 합성 공정,
  병원체 개조 등 바이오안보 위험이 있는 위험 물질/악용 가능 정보 엄격 통제

AI 주권 확보가 최우선
제한된 환경 내 경험 자산 활용







[B] 인공지능이 여는 원자로 해석의 새로운 패러다임

02

MSR 3D 다물리 실시간 시뮬레이터

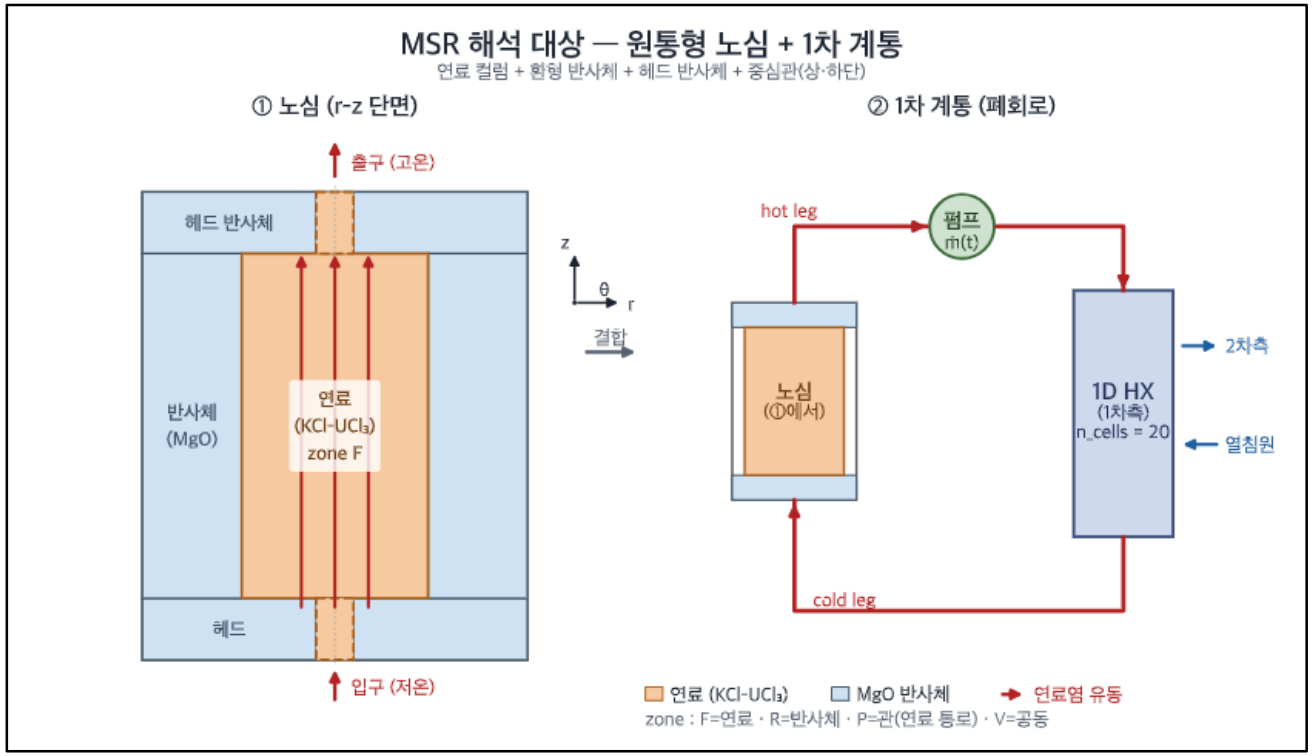
한국원자력연구원
Korea Atomic Energy Research Institute

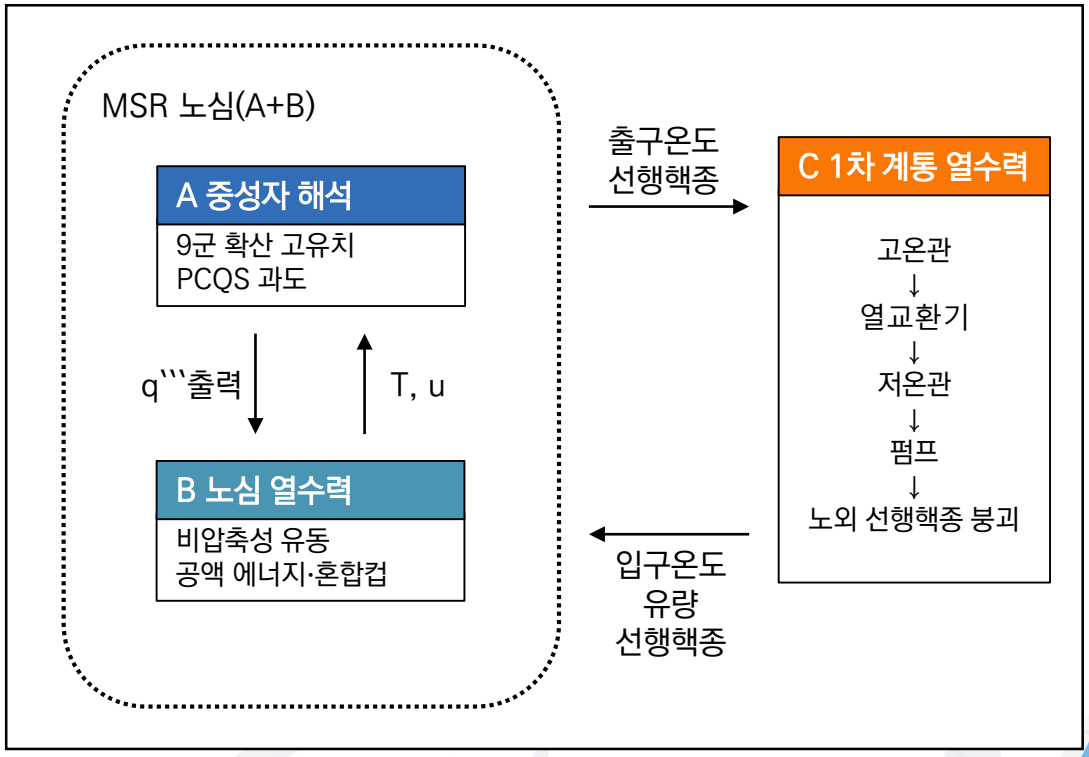
02 MSR 3D 다물리 실시간 시뮬레이터

MSR 문제 정의

시뮬레이터 해석 목표

- **펌프 유량 변동** 및 **외부 반응도 삽입**에 따른 용융염원자로 다물리 거동 해석을 위한 시뮬레이터





02 MSR 3D 다물리 실시간 시뮬레이터

MSR 예제 및 적용 방법론 요약

예제(100MWth CASE A – 노내·외 체류시간(~1초 | ~60초))

정격열출력	유량	노심 입출구 온도
100 MWth	606 kg/s	~ 623 ℃ | ~ 723 ℃

격자수 (rxθxz)	에너지군 | 지발군	노내·외 체류시간
8x4x14 (448)	9군 | 6군	~ 1 초 | ~ 60 초

형상 및 경계조건

- 형상
  - 연료 R = 31.4 cm
  - 연료 H = 58.0 cm
  - 반사체 20 cm 두께
- 경계조건
  - 중성자 (A)
    : 진공 경계조건
    : 선행핵종 재유입
  - 열유체 (B)
    : 하단 균일속도 유입
    : 입구온도=HX 출구온도
  - 1차 계통 (C)
    : 2차측 등온 열침원

A-중성자 동특성

- 9군 확산방정식: 원통좌표계
- 군별 Gauss-Seidel, 군 내 matrix-free BiCGstab(Jacobi 전처리)
- 순환연료 일반화 고유치: 지발원을 이류-붕괴 정상해로 치환해 적용
- 준정적 형상-진폭 분리: 매 시간단계 정·수반 고유치 재해석으로 형상 갱신, 수반 가중
- 온도 반응도: Cell별 반응단면적 선형 보정 후 섭동

C-1차 계통

- 집중정수·1D 폐루프: 노심→hot leg → 펌프 → HX cold leg, 전구간 균일 질량유량
- 펌프: 1차 관성지연(5s), implicit Euler
- HX: 1차측 20 셀 풍상 FV, implicit Euler 전진 sweep, 2차측 등온 열침원
- 배관: plug-flow 운반 지연 ring buffer
- 노외 지발중성자 선행핵종 해석적 붕괴 적용

B-노심 열유체

- 비압축 점성 Navier-Stokes: 원통좌표계
- 영방정식 난류 모형
- 3차원 에너지방정식: FV, 후방 Euler, SOR 반복
- 출구 상태는 에너지보존 mixing-cup 적분으로 1차 계통에 전달

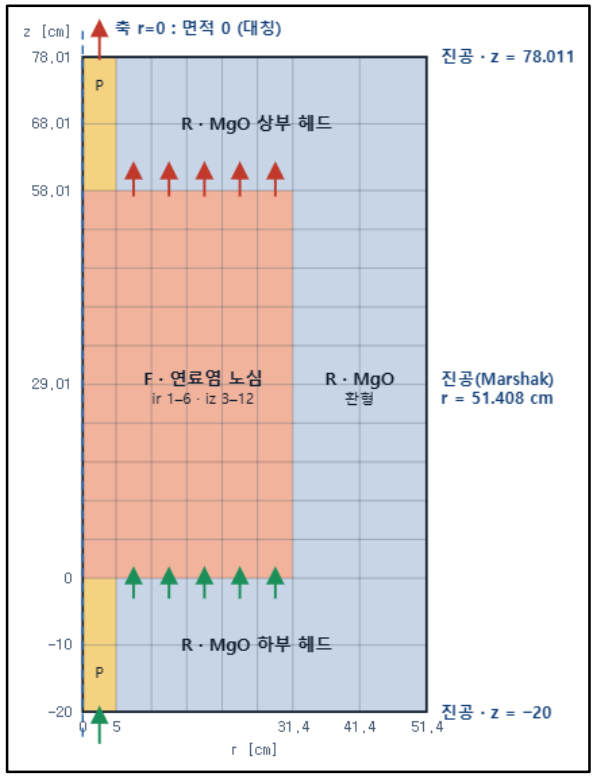
D-실시간 시뮬레이터

- Fortran 해석 내용을 C ABI로 노출, Python이 ctypes 인-프로세스 호출
- 세션형 시간전진: 정상 초기조건 확립 후 macro-step 단위 외부 구동
- 역방향 제어채널: 펌프 유량·외부 반응도를 운전원 입력으로 시나리오에 우선 적용
- PySide6 + pyqtgraph 계통도·상태·반응도 분해 패널

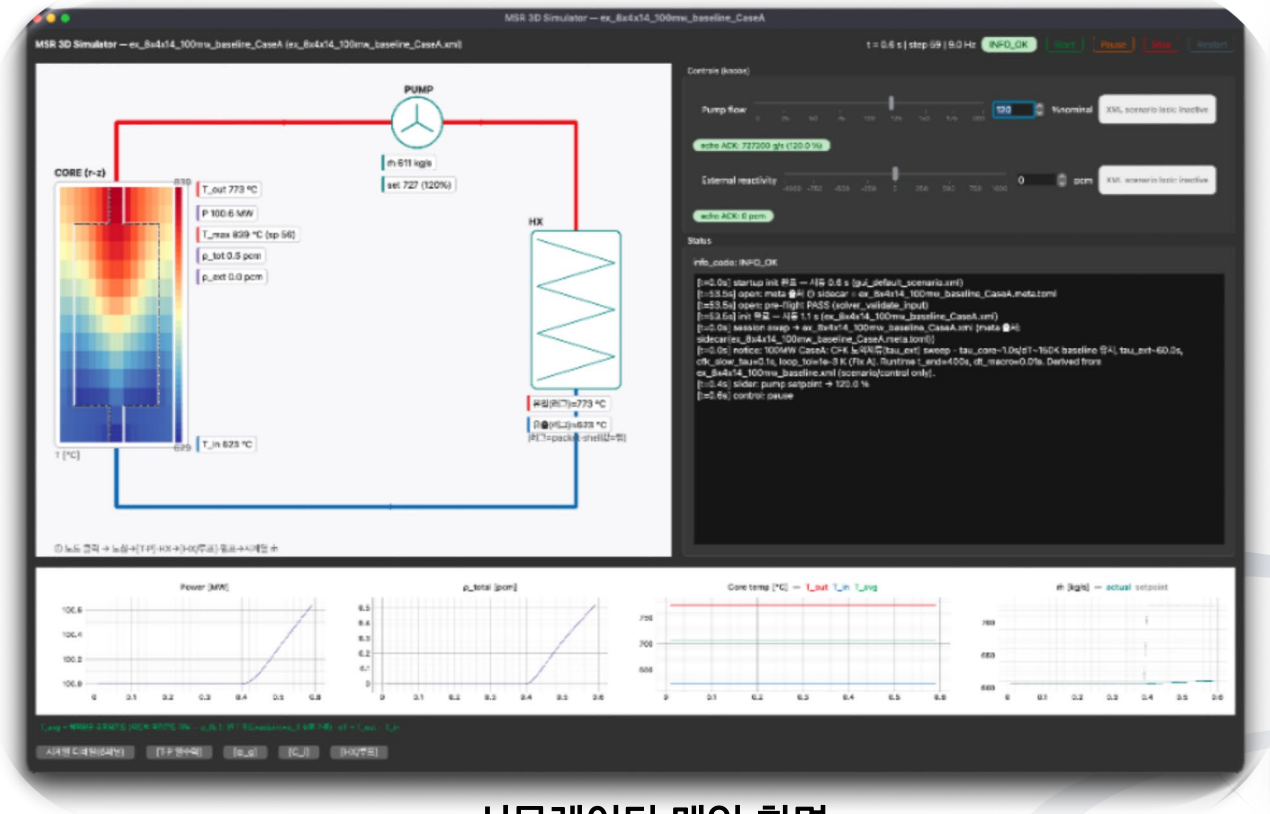
02 MSR 3D 다물리 실시간 시뮬레이터

MSR 시뮬레이터

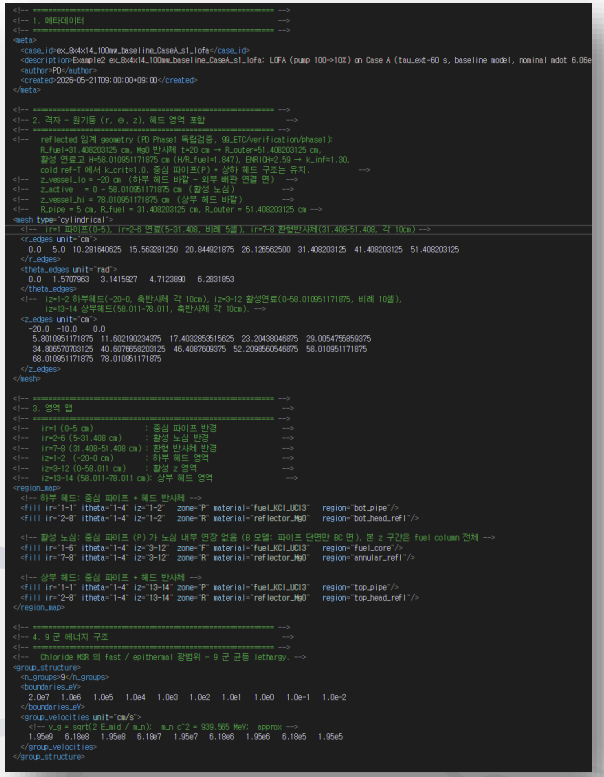
메인 화면 및 입력 파일 예시



원기둥형 단순 MSR 형상



시뮬레이터 메인 화면



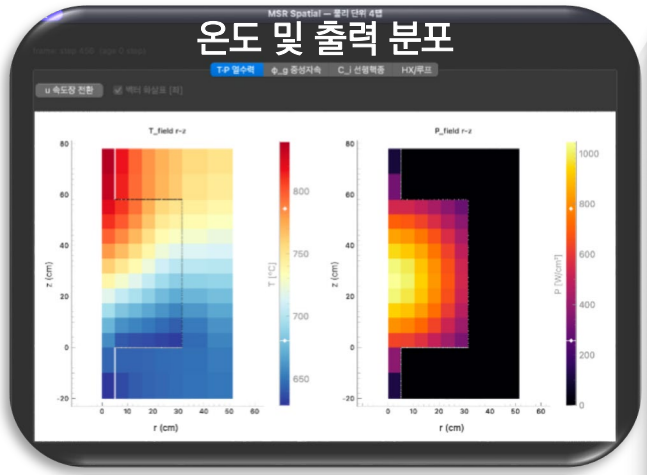
입력 파일(xml) 예시

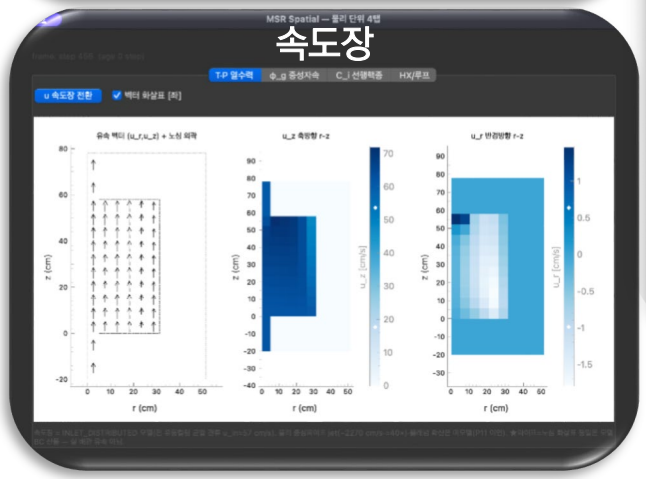
[B] 인공지능이 여는 원자로 해석의 새로운 패러다임

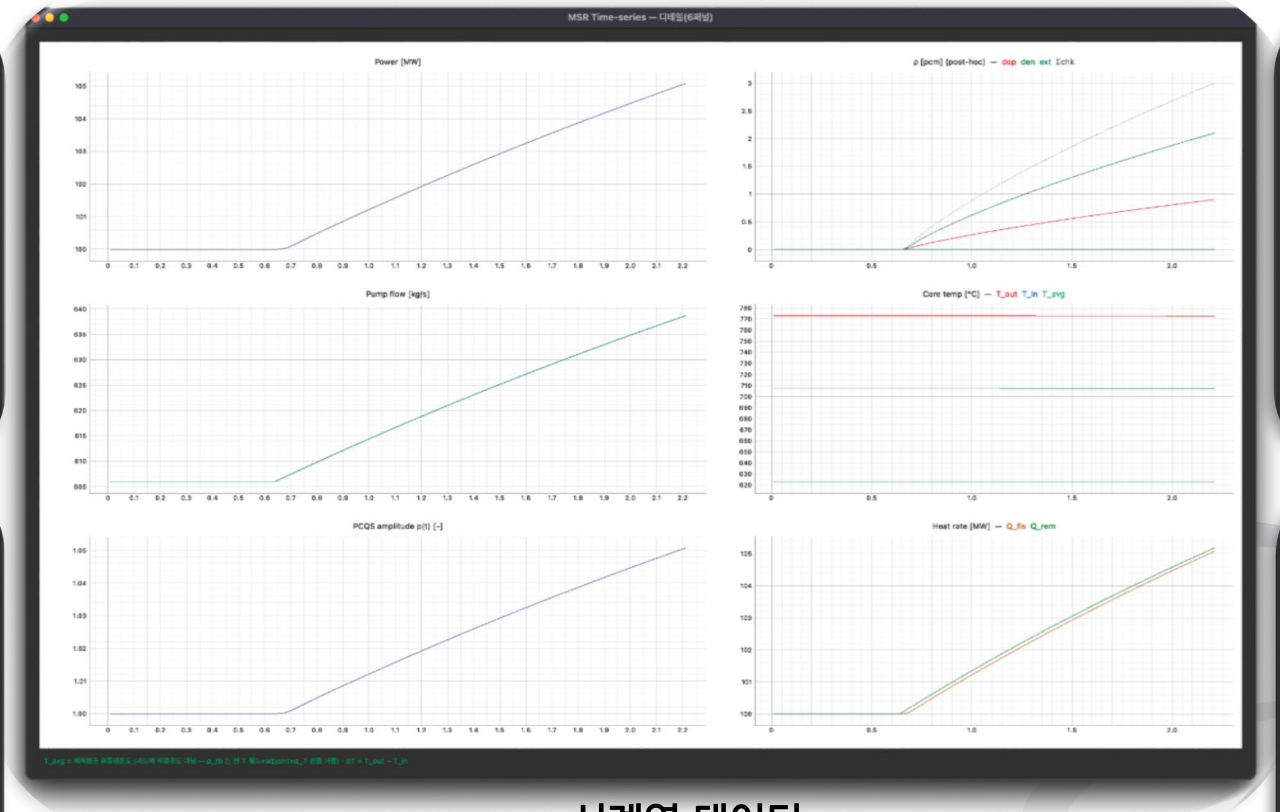
02 MSR 3D 다물리 실시간 시뮬레이터

MSR 시뮬레이터

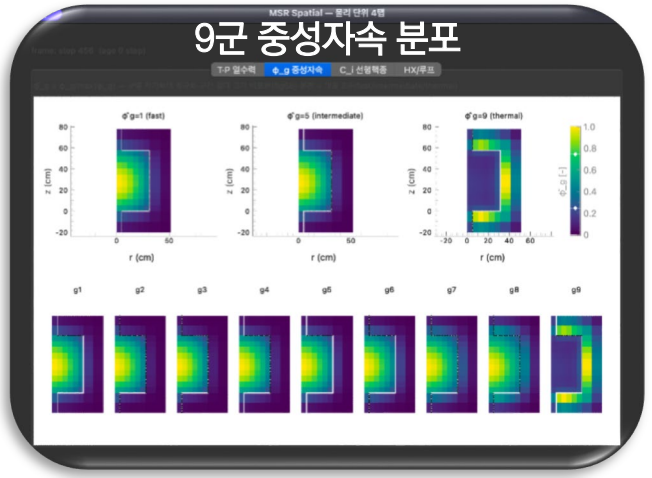
해석 결과(시계열 및 공간 데이터) 패널

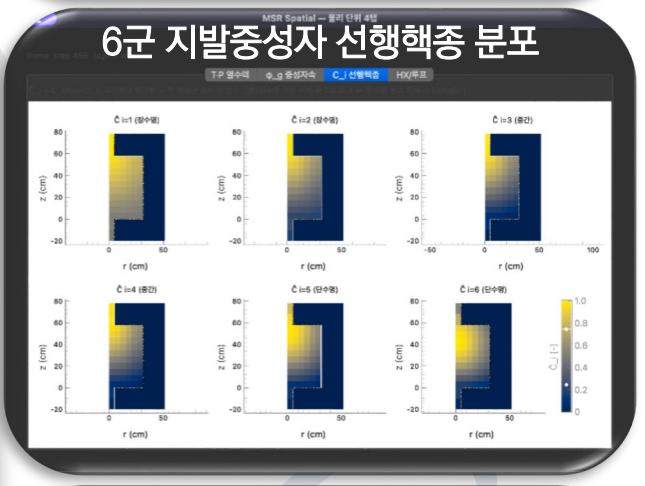


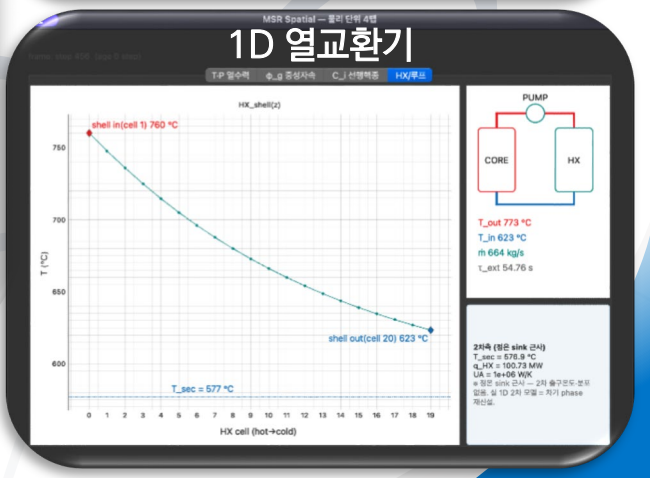




시계열 데이터
(출력/유량/반응도/노심 내 온도 정보)







동영상자료

참고용
(1kW)

동영상자료

참고용
(1kW
x8 격자 수)

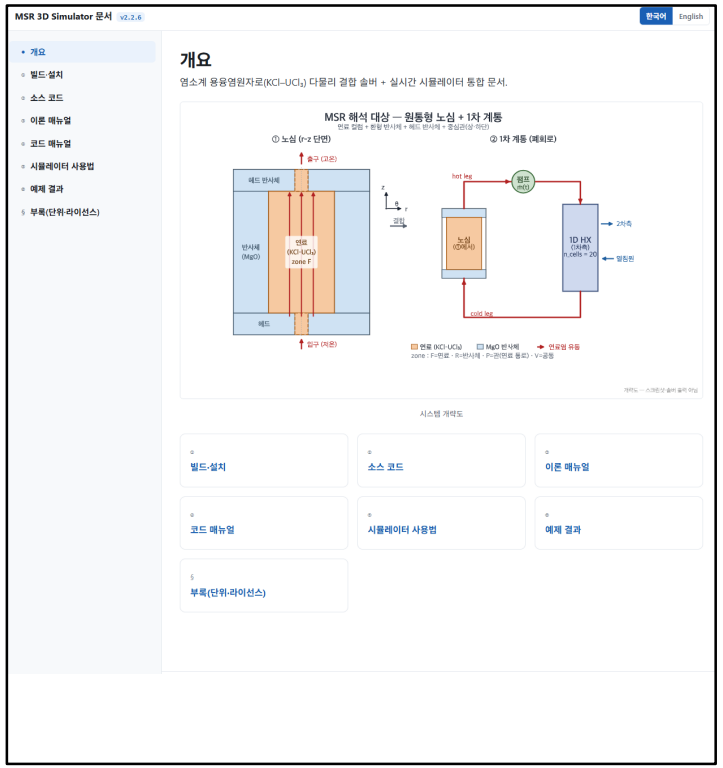
동영상자료

02 MSR 3D 다물리 실시간 시뮬레이터

MSR 시뮬레이터 매뉴얼

HTML 문서 작성(사용자/이론 매뉴얼-국문)

- 개요 | 빌드·설치 | 소스 코드 | 이론 매뉴얼 | 시뮬레이터 사용법 | 예제 결과 | 부록







02 MSR 3D 다물리 실시간 시뮬레이터

MSR 시뮬레이터 매뉴얼

HTML 문서 작성(사용자/이론 매뉴얼-국문)

- 개요 | 빌드·설치 | 소스 코드 | 이론 매뉴얼 | 시뮬레이터 사용법 | 예제 결과 | 부록





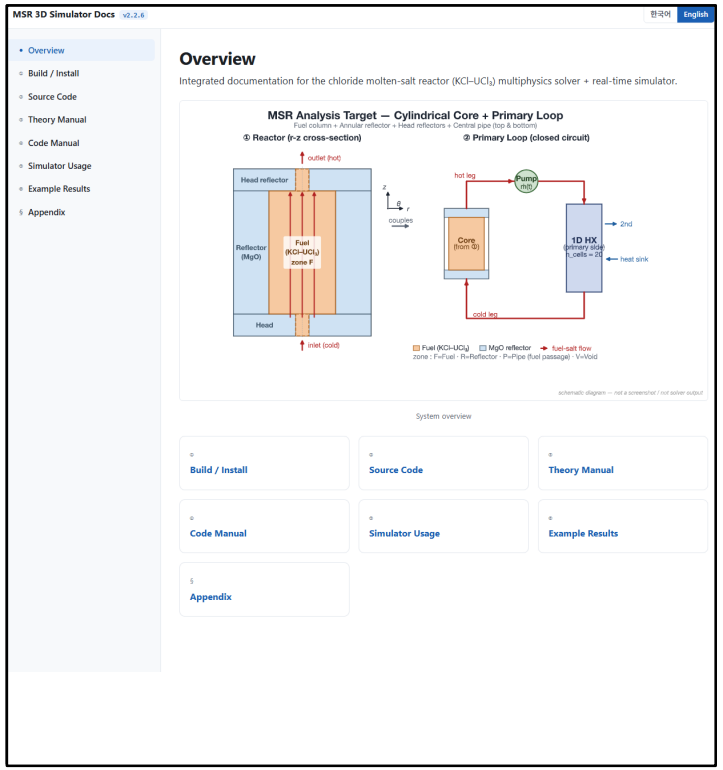


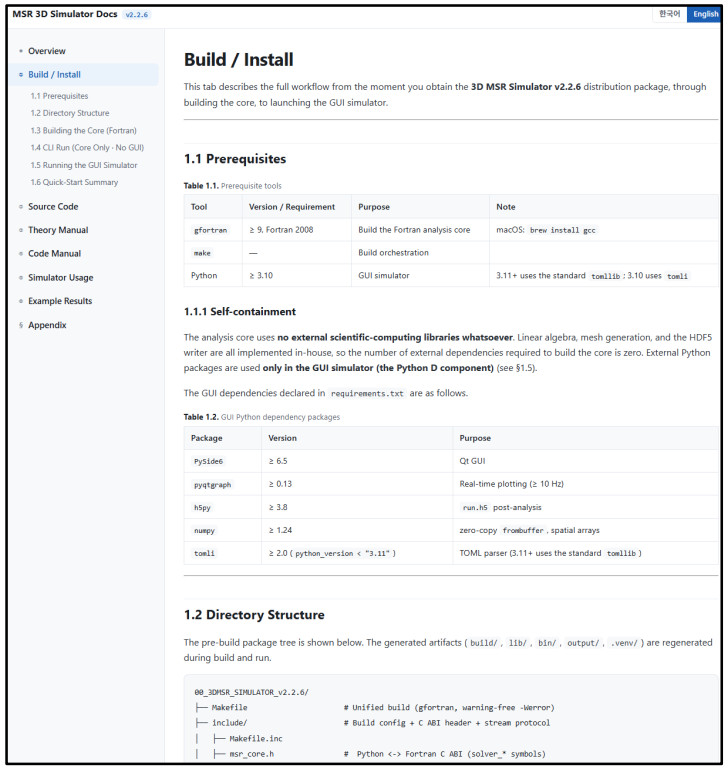
02 MSR 3D 다물리 실시간 시뮬레이터

MSR 시뮬레이터 매뉴얼

HTML 문서 작성(사용자/이론 매뉴얼-영문)

- Overview | Build/Install | Source Code | Theory Manual | Simulator Usage | Example Results | Appendix





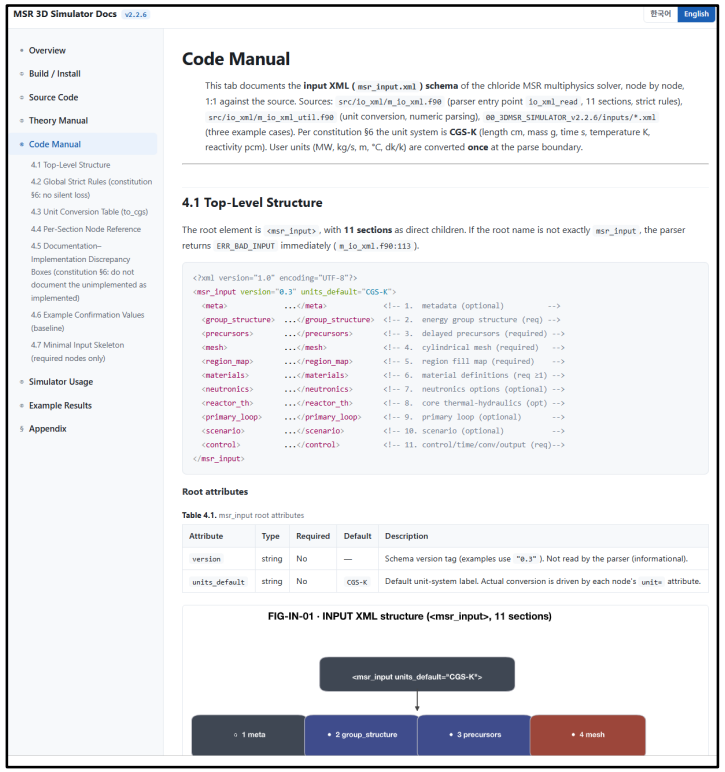


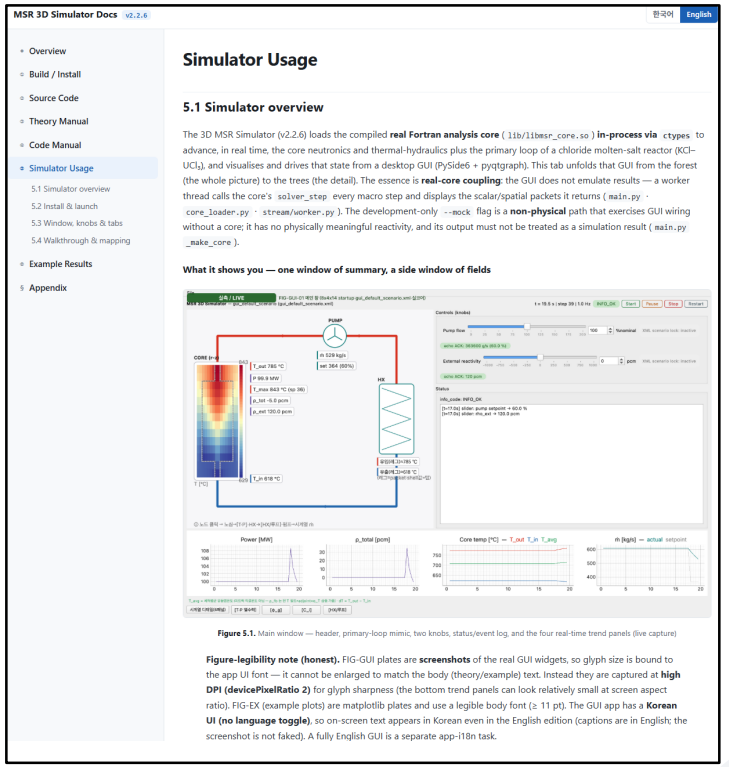
02 MSR 3D 다물리 실시간 시뮬레이터

MSR 시뮬레이터 매뉴얼

HTML 문서 작성(사용자/이론 매뉴얼-영문)

- Overview | Build/Install | Source Code | Theory Manual | Simulator Usage | Example Results | Appendix







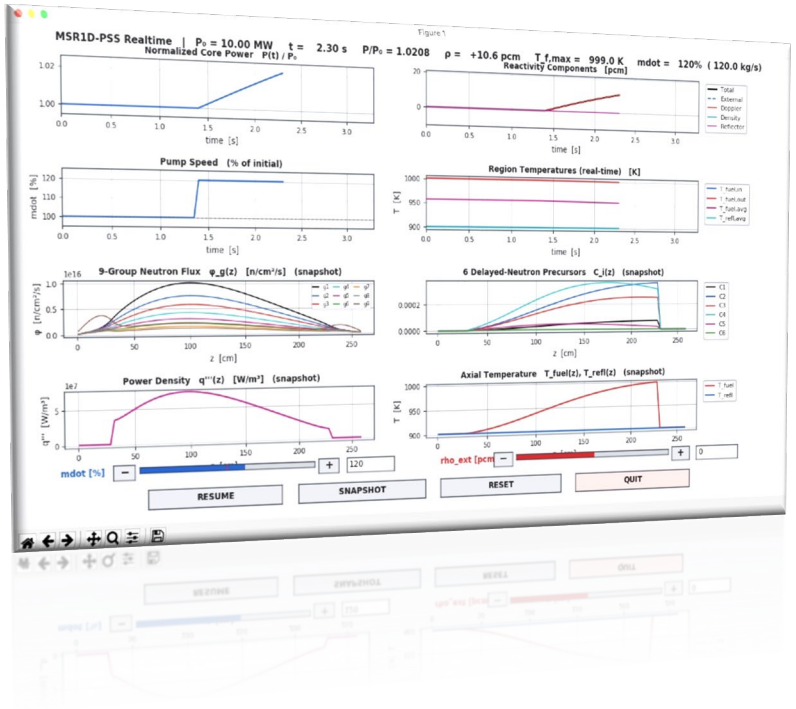
02 MSR 3D 다물리 실시간 시뮬레이터

MSR 시뮬레이터 구현 경험

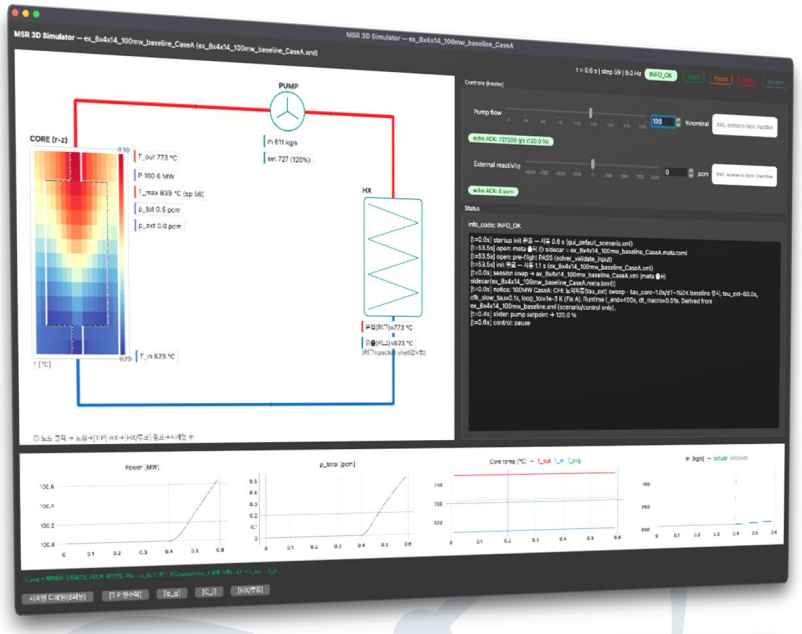
예제 풀이를 통한 차원 확장



0-D 모델



1-D 모델



3-D 모델



[B] 인공지능이 여는 원자로 해석의 새로운 패러다임

에이전틱 엔지니어링
(종합 오케스트레이션)

03

한국원자력연구원
Korea Atomic Energy Research Institute

03 에이전틱 엔지니어링(종합 오케스트레이션)

에이전틱 AI 활용법

배경지식

프롬프트 엔지니어링 (Prompt Engineering)

■ 어떻게 말할까?

- LLM이 최적의 답변을 출력하도록 입력 텍스트(질문, 지시사항, 예시)를 구조화하고 최적화하는 기법

단계적 사고 유도(CoT)
명확한 역할·목표·제약

컨텍스트 엔지니어링 (Context Engineering)

■ 무엇을 보여줄까?

- 모델이 추론 시점에 마주하는 전체 정보 환경(시스템 프롬프트, 대화 이력, RAG 검색 문서, 도구 정의)을 동적으로 설계하고 관리하는 기법

RAG 컨텍스트 압축(Compaction)
컨텍스트 윈도우 관리 인수인계서

하네스 엔지니어링 (Harness Engineering)

■ 어떻게 통제할까?

- LLM 주변에 안전장치, 도구(Tools), 실행 환경, 메모리 구조 등 외부 인프라를 구축하여 신뢰성 있는 실행을 보조하는 시스템 설계 기법

Agent = Model + Harness
권한·안전장치

루프 엔지니어링 (Loop Engineering)

■ 어떻게 반복·개선할까?

- [관찰 → 계획 → 행동 → 검증]의 제어 루프를 돌며 스스로 피드백을 받고 목표를 달성할 때까지 반복 실행하게 만드는 기법

랄프 엔지니어링 종료 조건 설정
자가 수정 설정

그래프 엔지니어링 (Graph Engineering)

■ 어떤 흐름·관계로 엮을까?

- 복잡한 업무를 수행하는 여러 에이전트들과 노드, 간선, 상태 기반의 협업 네트워크로 확장하는 기법

순환 그래프 멀티에이전트
품질관리 오케스트레이션

에이전틱 엔지니어링 (Agentic Engineering)

■ 어떻게 자율적으로 일하게 할까?

- 스스로 목표를 설정하고, 도구를 선택하며, 환경과 상호작용하며 자율적으로 문제를 해결하는 시스템을 설계하는 기법

자율적 루프 설정 종합 오케스트레이션
목표 지향성

03 에이전틱 엔지니어링(종합 오케스트레이션)

에이전틱 AI 활용법

스킬 적용 (3)

Karpathy-guidelines

Karpathy Guidelines

Behavioral guidelines to reduce common LLM coding mistakes, derived from Andrej Karpathy's observations on LLM coding pitfalls.

Tradeoff: These guidelines bias toward caution over speed. For trivial tasks, use judgment.

1. Think Before Coding

Don't assume. Don't hide confusion. Surface tradeoffs.

Before implementing:

- State your assumptions explicitly. If uncertain, ask.
- If multiple interpretations exist, present them - don't pick silently.
- If a simpler approach exists, say so. Push back when warranted.
- If something is unclear, stop. Name what's confusing. Ask.

2. Simplicity First

Minimum code that solves the problem. Nothing speculative.

- No features beyond what was asked.
- No abstractions for single-use code.
- No "flexibility" or "configurability" that wasn't requested.
- No error handling for impossible scenarios.
- If you write 200 lines and it could be 50, rewrite it.

Ask yourself: "Would a senior engineer say this is overcomplicated?" If yes, simplify.

3. Surgical Changes

Touch only what you must. Clean up only your own mess.

When editing existing code:

- Don't "improve" adjacent code, comments, or formatting.
- Don't refactor things that aren't broken.
- Match existing style, even if you'd do it differently.
- If you notice unrelated dead code, mention it - don't delete it.

When your changes create orphans:

- Remove imports/variables/functions that YOUR changes made unused.
- Don't remove pre-existing dead code unless asked.

The test: Every changed line should trace directly to the user's request.

4. Goal-Driven Execution

Define success criteria. Loop until verified.

Transform tasks into verifiable goals:

- "Add validation" → "Write tests for invalid inputs, then make them pass"
- "Fix the bug" → "Write a test that reproduces it, then make it pass"
- "Refactor X" → "Ensure tests pass before and after"

For multi-step tasks, state a brief plan:

```
1. [Step] → verify: [check]
```

find-skills

Find Skills

This skill helps you discover and install skills from the open agent skills ecosystem.

When to Use This Skill

Use this skill when the user:

- Asks "how do I do X" where X might be a common task with an existing skill
- Says "find a skill for X" or "is there a skill for X"
- Asks "can you do X" where X is a specialized capability
- Expresses interest in extending agent capabilities
- Wants to search for tools, templates, or workflows
- Mentions they wish they had help with a specific domain (design, testing, deployment, etc.)

What is the Skills CLI?

The Skills CLI (`npx skills`) is the package manager for the open agent skills ecosystem. Skills are modular packages that extend agent capabilities with specialized knowledge, workflows, and tools.

Key commands:

- `npx skills find [query]` - Search for skills interactively or by keyword
- `npx skills add <package>` - Install a skill from GitHub or other sources
- `npx skills check` - Check for skill updates
- `npx skills update` - Update all installed skills

Browse skills at: https://skills.sh/ v

How to Help Users Find Skills

Step 1: Understand What They Need

When a user asks for help with something, identify:

1. The domain (e.g., React, testing, design, deployment)
2. The specific task (e.g., writing tests, creating animations, reviewing PRs)
3. Whether this is a common enough task that a skill likely exists

Step 2: Check the Leaderboard First

Before running a CLI search, check the skills.sh leaderboard to see if a well-known skill already exists for the domain. The leaderboard ranks skills by total installs, surfacing the most popular and battle-tested options.

For example, top skills for web development include:

- `vercel-labs/agent-skills` — React, Next.js, web design (100K+ installs each)
- `anthropics/skills` — Frontend design, document processing (100K+ installs)

Step 3: Search for Skills

If the leaderboard doesn't cover the user's need, run the find command:

```
npx skills find [query]
```

For example:

- User asks "how do I make my React app faster?" → `npx skills find react performance`

frontend-design

This skill guides creation of distinctive, production-grade frontend interfaces that avoid generic "AI slop" aesthetics. Implement real working code with exceptional attention to aesthetic details and creative choices.

The user provides frontend requirements: a component, page, application, or interface to build. They may include context about the purpose, audience, or technical constraints.

Design Thinking

Before coding, understand the context and commit to a BOLD aesthetic direction:

- **Purpose**: What problem does this interface solve? Who uses it?
- **Tone**: Pick an extreme: brutally minimal, maximalist chaos, retro-futuristic, organic/natural, luxury/refined, playful/toy-like, editorial/magazine, brutalist/raw, art deco/geometric, soft/pastel, industrial/utilitarian, etc. There are so many flavors to choose from. Use these for inspiration but design one that is true to the aesthetic direction.
- **Constraints**: Technical requirements (framework, performance, accessibility).
- **Differentiation**: What makes this UNFORGETTABLE? What's the one thing someone will remember?

CRITICAL: Choose a clear conceptual direction and execute it with precision. Bold maximalism and refined minimalism both work - the key is intentionality, not intensity.

Then implement working code (HTML/CSS/JS, React, Vue, etc.) that is:

- Production-grade and functional
- Visually striking and memorable
- Cohesive with a clear aesthetic point-of-view
- Meticulously refined in every detail

Frontend Aesthetics Guidelines

Focus on:

- **Typography**: Choose fonts that are beautiful, unique, and interesting. Avoid generic fonts like Arial and Inter; opt instead for distinctive choices that elevate the frontend's aesthetics; unexpected, characterful font choices. Pair a distinctive display font with a refined body font.
- **Color & Theme**: Commit to a cohesive aesthetic. Use CSS variables for consistency. Dominant colors with sharp accents outperform timid, evenly-distributed palettes.
- **Motion**: Use animations for effects and micro-interactions. Prioritize CSS-only solutions for HTML. Use Motion library for React when available. Focus on high-impact moments: one well-orchestrated page load with staggered reveals (animation-delay) creates more delight than scattered micro-interactions. Use scroll-triggering and hover states that surprise.
- **Spatial Composition**: Unexpected layouts. Asymmetry. Overlap. Diagonal flow. Grid-breaking elements. Generous negative space OR controlled density.
- **Backgrounds & Visual Details**: Create atmosphere and depth rather than defaulting to solid colors. Add contextual effects and textures that match the overall aesthetic. Apply creative forms like gradient meshes, noise textures, geometric patterns, layered transparencies, dramatic shadows, decorative borders, custom cursors, and grain overlays.

NEVER use generic AI-generated aesthetics like overused font families (Inter, Roboto, Arial, system fonts), cliched color schemes (particularly purple gradients on white backgrounds), predictable layouts and component patterns, and cookie-cutter design that lacks context-specific character.

Interpret creatively and make unexpected choices that feel genuinely designed for the context. No design should be the same. Vary between light and dark themes, different fonts, different aesthetics. NEVER converge on common choices (Space Grotesk, for example) across generations.

IMPORTANT: Match implementation complexity to the aesthetic vision. Maximalist designs need elaborate code with extensive animations and effects. Minimalist or refined designs need restraint, precision, and careful attention to spacing, typography, and subtle details. Elegance comes from executing the vision well.

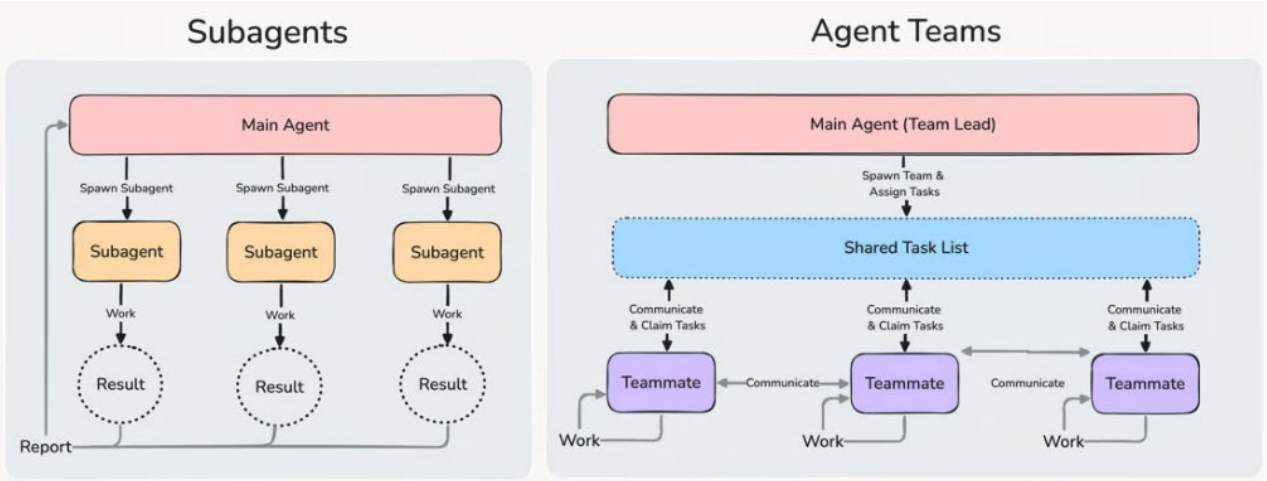
Remember: Claude is capable of extraordinary creative work. Don't hold back, show what can truly be created when thinking outside the box and committing fully to a distinctive vision.

03 에이전틱 엔지니어링

에이전틱 AI 활용법

종합 오케스트레이션

- (배경지식) 에이전트를 병렬로 실행하기



서브에이전트	• 한 세션 내에서 자신의 컨텍스트에서 부작업을 수행하고 요약을 반환하는 위임된 작업자 → 부작업이 검색 결과, 로그 또는 다시 참조하지 않을 파일 내용으로 주 대화를 넘칠 때
에이전트 뷰	• claude agents 로 열 수 있는 백그라운드 상태에서 실행 중인 세션을 디스패치하고 모니터링하는 하나의 화면, 연구 미리보기 → 여러 개의 독립적인 작업이 있고 작업을 넘기고, 상태를 한눈에 확인하고, 필요할 때만 개입하고 싶을 때
에이전트 팀	• 공유 작업 목록과 에이전트 간 메시징을 통해 조율되는 여러 세션으로, 리더가 관리합니다. 실험적이며 기본적으로 비활성됨 → Claude가 프로젝트를 여러 부분으로 나누고, 할당하고, 작업자들을 동기화 상태로 유지하기를 원할 때
프로젝트	• Claude.ai/code 또는 데스크톱 앱에서의 지속적인 대화. Claude는 threads라고 불리는 병렬 세션을 클라우드에서 시작하거나, 사용자가 요청할 때 Remote Control을 통해 사용자의 컴퓨터에서 시작하고, 각 세션에 프로젝트의 지침을 제공하며, 어느 것이 사용자를 필요로 하는지 표시함 → 작업이 여러 날 또는 주에 걸쳐 많은 작업을 포함하고, 사용자의 머신이 꺼져 있을 때도 계속 실행되어야 하며, 각 세션을 디스패치하고 추적하기보다는 한 번에 설명하고 싶을 때
동적 워크플로우	• 많은 서브에이전트를 실행하고 그 결과를 교차 검증하는 스크립트로, 한 번에 조율 하기에는 너무 큰 작업용 → 작업이 소수의 서브에이전트로는 조율할 수 없을 때, 또는 결과를 서로 검증하고 싶을 때

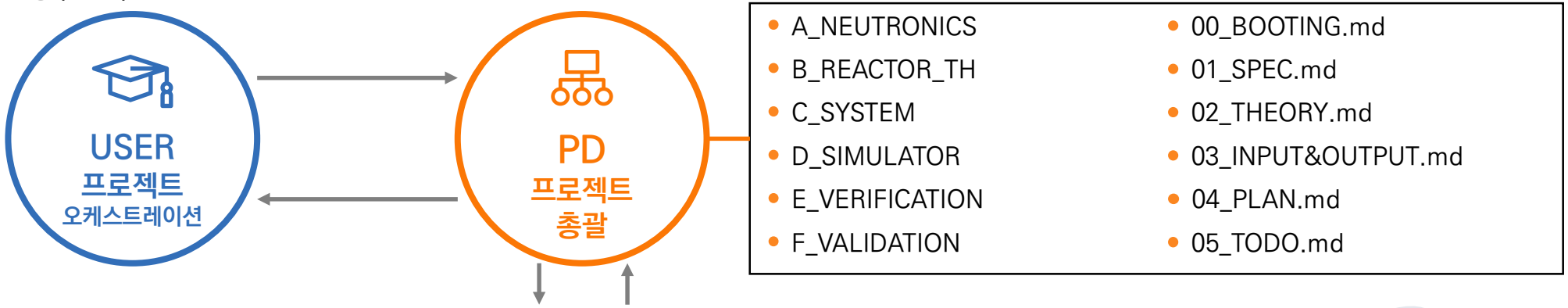
Code.claude.com/docs/ko/agents

03 에이전틱 엔지니어링(종합 오케스트레이션)

에이전틱 AI 활용법

종합 오케스트레이션

- 에이전트 팀 구성(예시)



책임자 A	책임자 B	책임자 C	책임자 D	책임자 E	책임자 F
디렉터리 A	디렉터리 B	디렉터리 C	디렉터리 D	디렉터리 E	디렉터리 F
A_NEUTRONICS	B_REACTOR_TH	C_SYSTEM	D_SIMULATOR	E_VERIFICATION	F_VALIDATION
원자로물리 • 중성자 확산 • 시간종속 (PCQS) • 지발중성자 선행핵종 이류	**노심 내 열수력** • 3D 용융염 유동·온도장	**1차 계통 열수력** • 펌프 모델 • 1D HX • 폐루프 압력 · 유량	**실시간 시뮬레이터** • PySide6+pyqtgraph • 펌프/반응도 조작 패널	**소스코드 확인** • 단위테스트 • 수렴성	**소스코드 검증** • 타당성 검증 • MSRE 벤치마크 등
▪ A_01_SPEC.md ▪ A_02_THEORY.md ▪ A_03_INPUT&OUTPUT.md ▪ A_04_PLAN.md ▪ A_05_TODO.md	▪ B_01_SPEC.md ▪ B_02_THEORY.md ▪ B_03_INPUT&OUTPUT.md ▪ B_04_PLAN.md ▪ B_05_TODO.md	▪ C_01_SPEC.md ▪ C_02_THEORY.md ▪ C_03_INPUT&OUTPUT.md ▪ C_04_PLAN.md ▪ C_05_TODO.md	▪ D_01_SPEC.md ▪ D_02_THEORY.md ▪ D_03_INPUT&OUTPUT.md ▪ D_04_PLAN.md ▪ D_05_TODO.md	▪ E_01_SPEC.md ▪ E_02_THEORY.md ▪ E_03_INPUT&OUTPUT.md ▪ E_04_PLAN.md ▪ E_05_TODO.md	▪ F_01_SPEC.md ▪ F_02_THEORY.md ▪ F_03_INPUT&OUTPUT.md ▪ F_04_PLAN.md ▪ F_05_TODO.md

동영상자료

동영상자료

동영상자료



[B] 인공지능이 여는 원자로 해석의 새로운 패러다임

요약 및 결언

04

한국원자력연구원
Korea Atomic Energy Research Institute

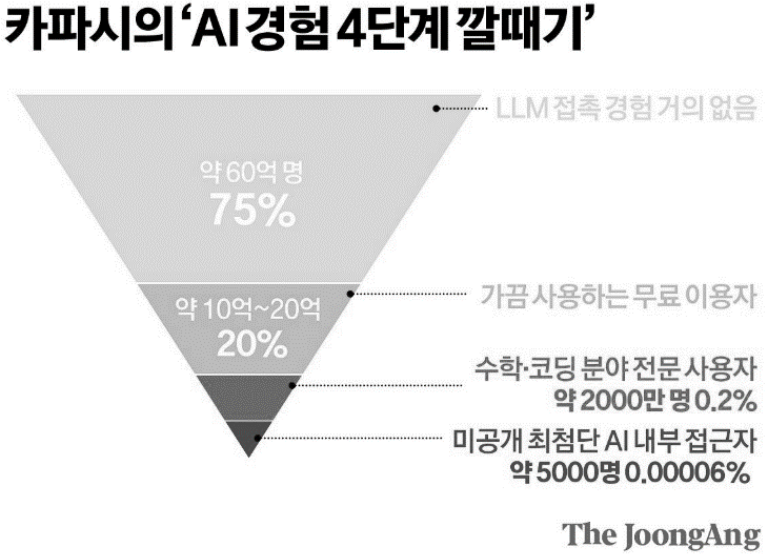
04 요약 및 결언

요약

인공지능이 여는 원자로 해석의 새로운 패러다임

- MSR 3D 다물리 실시간 시뮬레이터 바이브 코딩 구현 완료
- 에이전틱 AI 활용 시뮬레이터 매뉴얼(국문/영문) 작성 완료

1. "AI, 인간 계층 4단계로 나눠", … 오픈마 창업자가 쏘아올린 논쟁, 중앙일보, 26.10.13. (https://www.koreadaily.com/article/20261013130045236)



바이브 코딩
(Vibe Coding)

- 바닥을 높이는 것

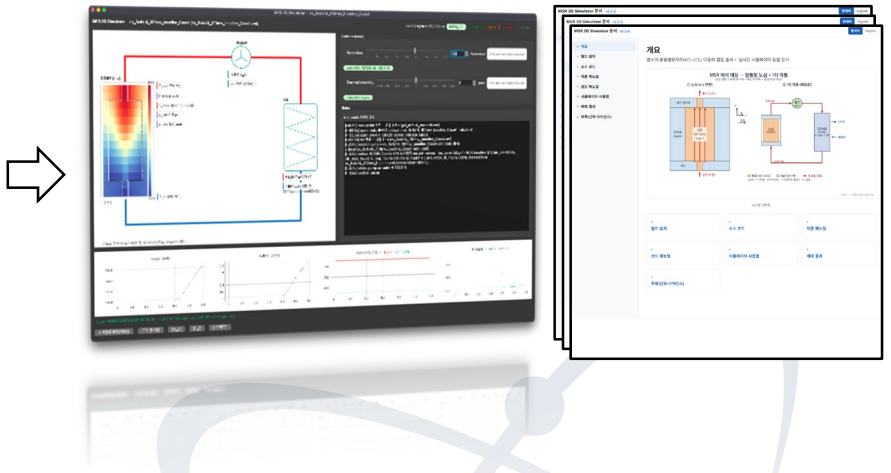
: 소프트웨어 생산 역량의 **'하한선'**을 대폭 끌어 올림

+

에이전틱 엔지니어링
(Agentic Engineering)

- 천장을 높이는 것

: 소프트웨어가 도달할 수 있는 **'상한선'**을 폭발적으로 확장함



에이전틱 AI 활용 시 연구 개발 가속화 가능성 확인

04 요약 및 결언

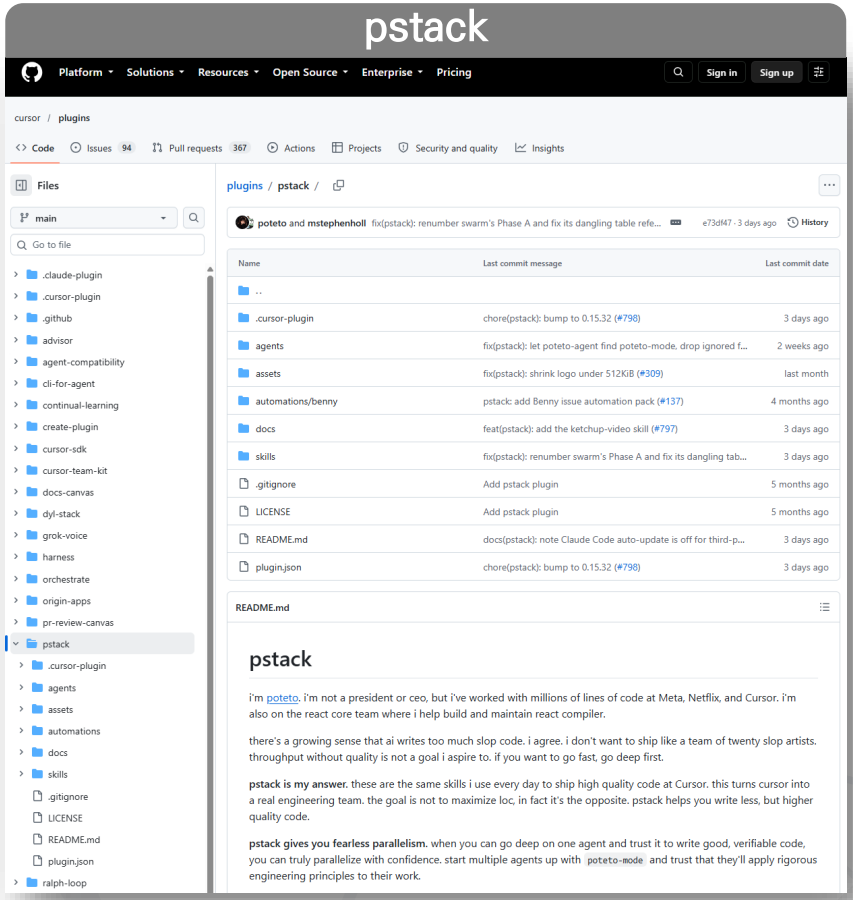
향후계획

① MSR 3D 다물리 실시간 시뮬레이터 검증 및 수정·보완

소스코드 진단



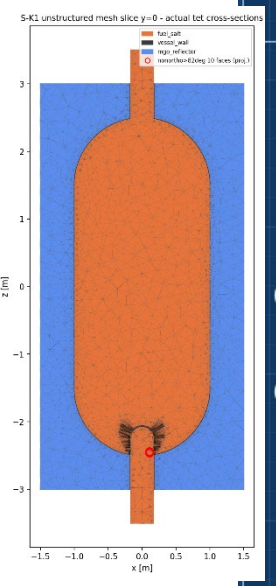
pstack

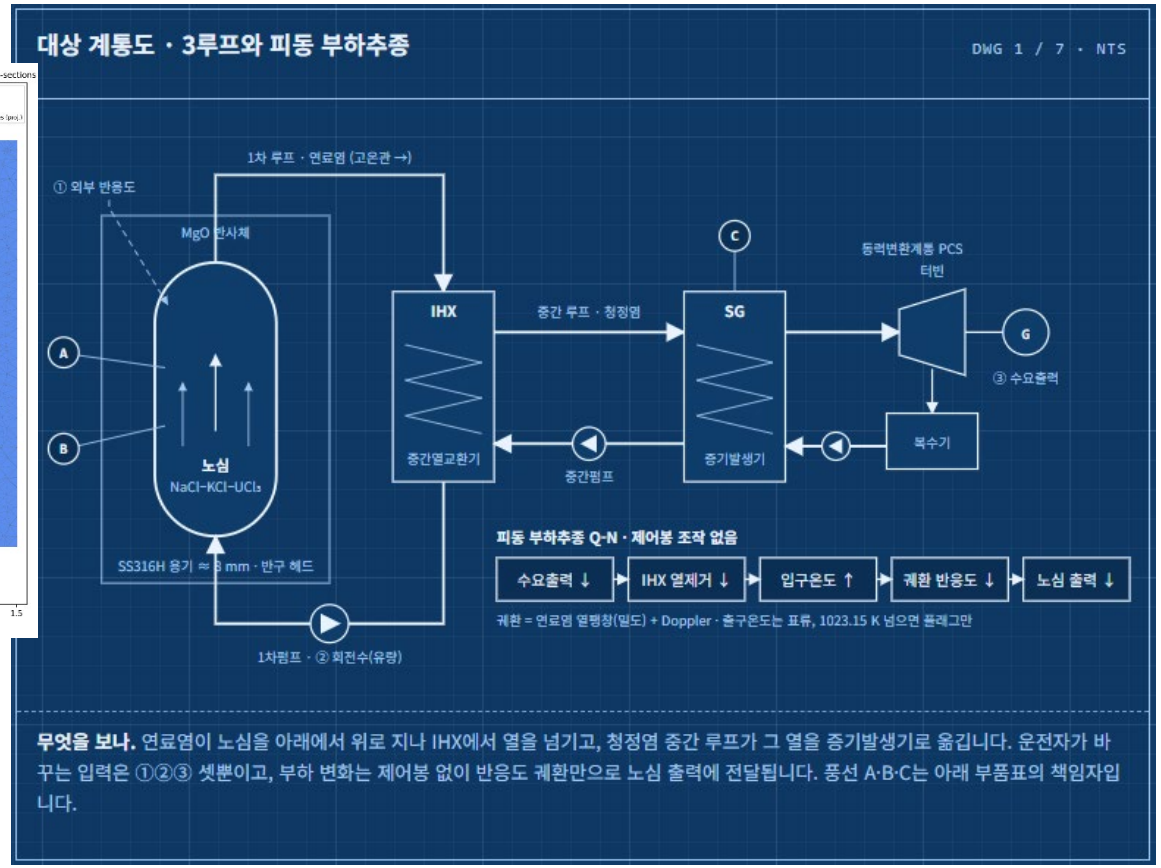


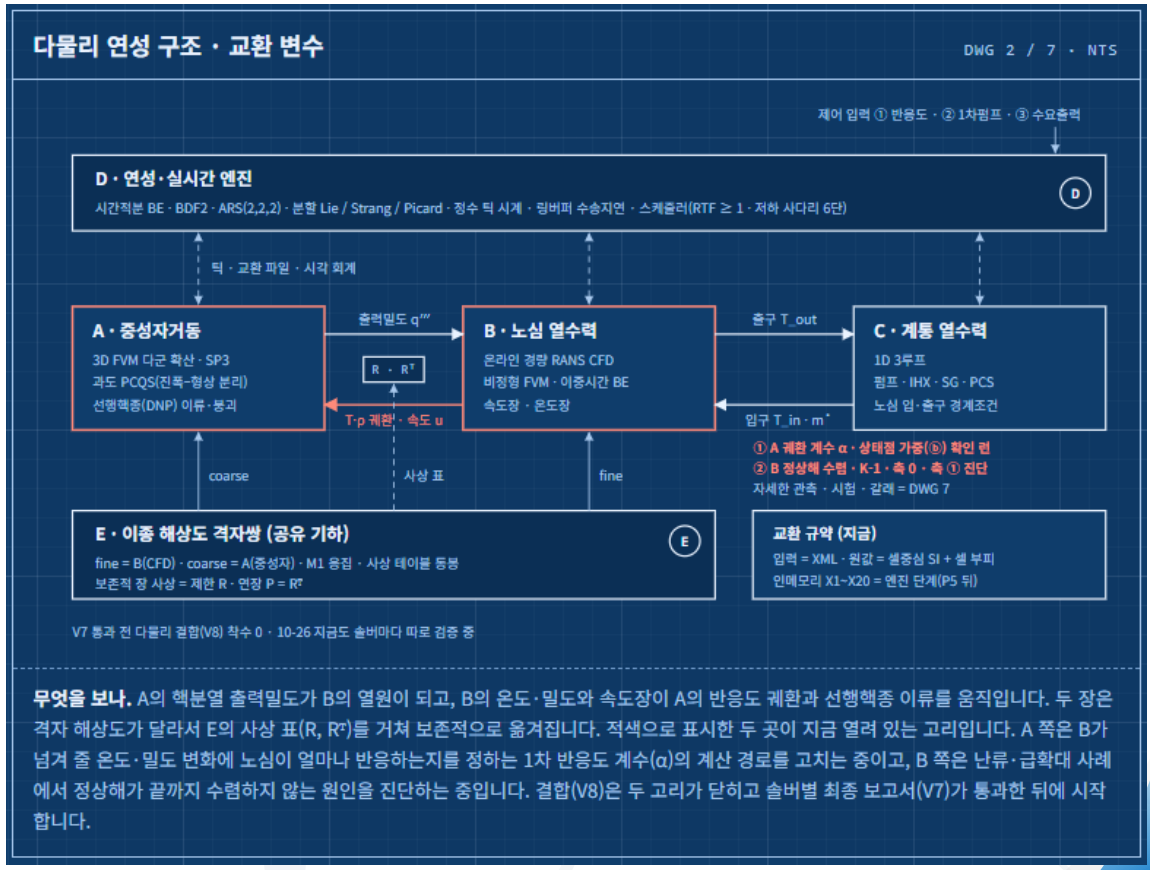
04 요약 및 결언

향후계획

② MSR 피동 부하추종 운전 해석을 위한 “바이브 코딩 + 에이전틱 엔지니어링” 경험 축적 계획







04 요약 및 결언

향후계획

③ On-premise 에이전틱 AI 활용

① 적합한 LLM 모델 선정 (uvx whichllm@latest)

```
Hardware Info
GPU 0: Apple M5 Max — 128.0 GB shared (budget 126.0 GB) — BW: 614 GB/s
CPU: Apple M5 Max — 18 cores
RAM: 128.0 GB
Disk free: 6903.9 GB
OS: darwin
VRAM headroom: 2.0 GB reserved per GPU
```

Recommended Models

#	Model	Quant	Fit / VRAM	Speed	Published	Score
1	deepseek-ai/DeepSeek-V4-Flash 284.0B (13.0Ba)	Q3_K_M	Full GPU 118.4 GB	16.9 tok/s ?	2026-04-22	95.3
2	Qwen/Qwen3.6-27B 27.8B	Q6_K	Full GPU 24.4 GB	11.2 tok/s ~	2026-04-21	89.0
3	Qwen/Qwen3-235B-A22B 235.1B (22.0Ba)	Q3_K_M	Full GPU 99.6 GB	20.4 tok/s ?	2025-04-27	88.0
4	Qwen/Qwen3.8-27B 27.8B	Q6_K	Full GPU 24.4 GB	11.2 tok/s ~	2026-08-05	88.0
5	google/gemma-4-31B-it 31.3B	Q6_K	Full GPU 27.4 GB	9.9 tok/s ~	2026-03-11	87.3
6	openai/gpt-oss-120b 116.8B (5.1Ba)	Q6_K	Full GPU 89.9 GB	22.1 tok/s ?	2025-08-04	86.6
7	MiniMaxAI/MiniMax-M2.5 228.7B (10.0Ba)	Q4_K_M	Full GPU 122.1 GB	17.9 tok/s ?	2026-02-12	86.4
8	google/gemma-4-26B-A4B-it 25.8B (3.8Ba)	Q8_0	Full GPU 27.0 GB	56.1 tok/s ?	2026-03-11	84.7
9	Qwen/Qwen3-30B-A3B 30.5B (3.0Ba)	Q8_0	Full GPU 31.6 GB	58.2 tok/s ?	2025-04-27	83.4
10	Qwen/Qwen3-Next-80B-A3B-Instruct 81.3B (3.0Ba)	Q6_K	Full GPU 62.9 GB	31.8 tok/s ?	2025-09-09	81.5

```
Speed:  ~ = estimated tok/s range,  ? = low-confidence/backend-sensitive tok/s
Top pick confidence: Medium (direct benchmark, gap +6.3, low-confidence speed)
Benchmark reference: 2026-05 curated snapshot; live AA / LiveBench / Aider merged when reachable.
Speed caution: Low-confidence speed estimates in top ranks: #1, #3
```

② Codex 설치하기 (curl –fsSL https::://chatgpt.com/codex/install.sh | sh)

③ Ollama 설치하기 (curl –fsSL https::://ollama.com/install.sh | sh)

④ On-premise 에이전틱 AI 구동하기 (ollama launch codex)

```
seongdong — ollama launch codex — ollama — ollama launch codex — 244×69

Last login: Wed Oct  7 09:59:23 on ttys017
[ SeongdongJANG😎  ~  ollama launch codex
Select model for Codex CLI: Type to filter...

  Recommended
    glm-5.3-flash:cloud
      Fast reasoning for coding and agentic workloads with 1M context and native image understanding
    glm-5.3:cloud
      Long-horizon coding and agentic engineering with deep reasoning and a 1M context
    deepseek-v4.1-flash:cloud
      Fast coding and agentic tool use with 1M context and native image understanding
    gemma4:31b-cloud
      Agentic workflows and multimodal reasoning
    gemma4:26b
      Agentic workflows and multimodal reasoning, ~19GB, (not downloaded)

  More
    gpt-oss
  ▸ gpt-oss:120b

↑/↓ navigate • enter select • ← back
```

동영상자료



[B] 인공지능이 여는 원자로 해석의 새로운 패러다임

질의응답

05

한국원자력연구원
Korea Atomic Energy Research Institute

KAERI 한국원자력연구원
Korea Atomic Energy Research Institute

감사합니다.

