자연어 기반 CFD 국부 격자 refinement 에이전트

정승한*† · 최준석* · 양선웅*
*한양대학교 ERICA 기계공학과

A Natural Language-Guided Agent for Local Mesh Refinement in CFD

Seunghan Jung*†, Junseok Choi*, Sunwoong Yang*
*Dept. of Mechanical Eng., Hanyang Univ.

Key Words: 인공지능 에이전트(Artificial Intelligent agent), 전산유체역학(Computational Fluid Dynamics)

Abstract

This study presents a quality-driven remeshing agent linking natural-language requests to SALOME/SMESH. A local Qwen LLM is confined to two decisions — interpreting user intent and selecting a remeshing strategy from a measurement-grounded decision table — while deterministic Python code assembles, executes, and assesses the mesh. Poor-quality elements are grouped into spatial clusters with measured element size, enabling targeted refinement that leaves sound regions untouched. Deterministic guards (element budget, single-family changes, physical validity) and mandatory user approval bound every modification. For a wing geometry, the agent produced a 1.4×10^6-element hybrid prism–tetrahedral mesh in 2 min 18 s and, within at most three autonomous cycles, reduced the defective-element rate by 43–55% while improving face non-orthogonality from 37.0–39.8° to 27.2–27.7°. All runs ran on a single 8 GB GPU. Plans, quality reports, MED files, and execution traces are preserved, making the process transparent, reproducible, and auditable.

1. 서 론

전산유체역학(computational fluid dynamics, CFD) 해석 공정에서 격자 생성은 해의 정확도와 수렴성을 좌우하는 핵심 단계이지만, 동시에 가장 많은 시간과 인적 자원이 소모되는 병목 구간이다. 특히 외부 공력 형상과 같이 곡률 변화가 크고 trailing edge처럼 국부적으로 얇은 영역을 포함하는 기하에서는 경계층 해상을 위한 viscous layer 삽입, 목표 y^+에 부합하는 첫 셀 높이 결정, 성장률 및 최대·최소 셀 크기의 조합 등 다수의 파라미터를 동시에 조율해야 한다. 이러한 판단은 대체로 해석자의 경험적 직관에 의존하며, 파라미터 조합이 부적절할 경우 층 간 self-intersection이나 국부적 셀 왜곡으로 인해 격자 생성 자체가 실패한다. 그 결과 실무에서는 격자 생성–확인–수정의 시행착오가 반복되고, 형상이 변경될 때마다 동일한 과정을 처음부터 되풀이해야 한다.

최근 대규모 언어 모델(large language model, LLM)을 CFD 워크플로에 접목하려는 연구가 활발히 진행되고 있다. MetaOpenFOAM[1]과 ChatCFD[2]는 자연어 지시로부터 해석 조건과 솔버 설정 파일을 생성할 수 있음을 보였고, CFDAgent[3]는 전처리 단계까지 포함하는 다중 에이전트 구성을 제시하였다. 그러나 이들 연구에서 격자는 정해진 절차로 한 번 생성되는 대상이며, 생성된 격자의 품질을 정량 지표로 진단하고 그 측정값에 근거해 재격자화 전략을 선택하고 실행하는 폐루프는 다루어지지 않았다. 또한 대부분 대형 LLM 모델을 전제하여, 형상 데이터의 외부 반출이 제한되는 환경에서 소형 로컬 모델만으로 동작 가능한지는 검증되지 않았다.

본 연구는 오픈소스 전처리 플랫폼인 SALOME의 헤드리스 파이썬 실행 환경과 로컬 구동 LLM(Ollama 기반 Qwen)을 결합한 자동 격자 생성 에이전트를 제안한다. 제안 시스템은 Fig. 1과 같이 (i) 자연어 요청을 해석하여 초기 하이브리드 격자를 생성하는 단계, (ii) 셀·면 단위 품질 지표를 산출하고 불량 요소를 공간 군집으로 특정하는 진단 단계, (iii) 진단 결과에 근거해 국부 재격자화 전략을 선택·실행하는 단계로 구성되며, 종료 조건을 만족할 때까지 (ii)–(iii)를 반복한다. 이때 LLM의 역할은 자연어 해석과 결정표 기반 전략 선택의 두 지점으로 국한되고, 출력은 JSON 스키마와 선택형 답안으로 제약되며, 수치 코드 조립·규칙 준수·격자 품질 계산은 결정적 코드가 담당한다.

본 논문의 기여는 다음과 같다. 첫째, 격자 품질 실측에 근거해 국부 재격자화 전략을 선택하는 폐루프 에이전트 구조를 제시하였다. 둘째, 네 가지 국부 세분화 기법의 동일 조건 비교를 통해 전략 선택 문제를 "불량 군집의 덮임 범위 결정"으로 정식화 하였다. 셋째, 소형 로컬 LLM의 출력을 선택형으로 재설계함으로써 8 GB급 GPU(NVIDIA GeForce RTX 5050)만으로 신뢰성 있는 판단이 가능함을 보였다.

2. 이론 및 실험

2.1 에이전트 구성과 설계 원칙

제안 시스템은 Fig.1과 같이 (i) 자연어 해석 및 초기 격자 생성, (ii) 품질 진단, (iii) 전략 선택 및

† Presenting Author, jshan0407@hanyang.ac.kr

재격자화의 세 부분으로 구성되며, 진단 결과가 종료 조건을 만족할 때까지 (ii)-(iii)를 반복하는 피드백 루프를 형성한다. 설계의 핵심 원칙은 LLM과 결정적(deterministic) 코드의 역할 분담이다. 사전 실험에서 소형 로컬 LLM(Qwen3.5:9B)은 보기가 주어진 선택 문제에서는 높은 정확도를 보였으나, 수치 계산과 다수 제약의 동시 준수, 그리고 긴 구조화 출력의 완성에서 반복적으로 실패하였다. 이에 따라 본 시스템은 LLM의 개입을 (a) 자연어 요청의 해석과 (b) 재격자화 전략의 선택이라는 두 판단 지점으로 국한하고, 스크립트 생성 · 수치 조립 · 규칙 준수 · 품질 계산은 전부 코드가 수행하도록 하였다.

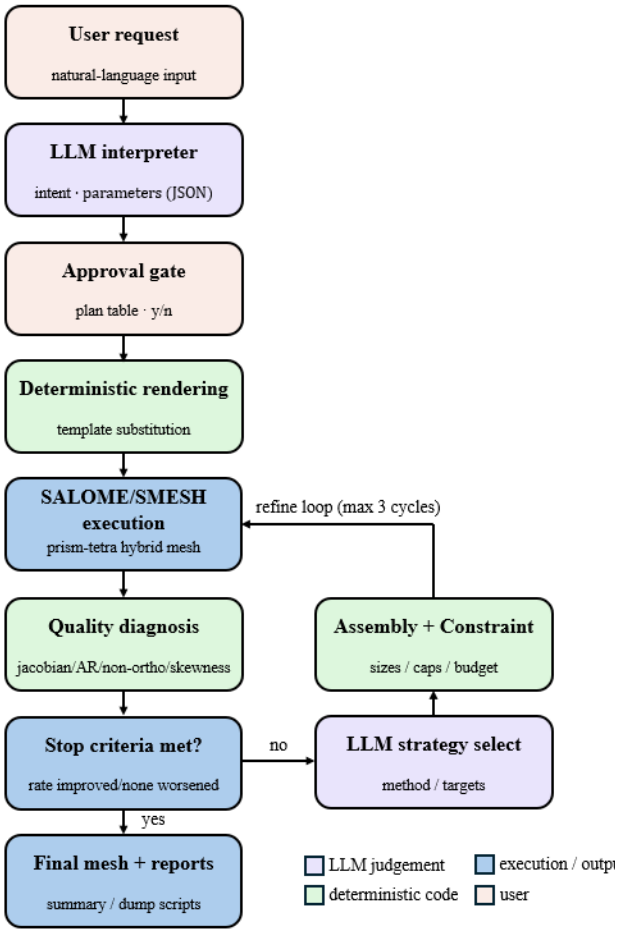

Fig. 1 Overall architecture of remeshing agent

2.2 자연어 해석과 초기 격자 생성

사용자의 한국어 요청은 Local LLM에 의해 의도(생성/개선/확인/질문), 파라미터 값, 국부 세분화 요청을 담은 JSON으로 변환되며, 허용 키 필터와 범위 검증을 거쳐 승인 표로 제시된다. 사용자가 승인하면 검증된 템플릿에 값을 치환하는 방식으로 SALOME 파이썬 스크립트가 코드로 생성된다. LLM이 스크립트를 직접 작성하지 않으므로 경계 조건 그룹 누락이나 존재하지 않는 API 호출과 같은 생성 오류가 구조적으로 차단된다. 격자는 벽면 viscous layers와 tetra 배경의 하이브리드로 생성되며, 1D–2D–3D 알고리즘을 분리 적용하여 leading edge와 trailing edge의 1D node 밀집을 독립 제어한다.

2.3 격자 품질 진단

격자 완성 직후 두 계층의 품질 지표가 자동 산출된다. 셀 단위로는 jacobian 값과 aspect ratio를 SMESH (SALOME의 mesh 모듈) 필터로 선별하되, 낮은 jacobian이 불가피한 벽면 프리즘은 제외하여 실제 개선 대상인 사면체만 추린다. 면 단위로는 유한체적 해석의 수치 정확도와 직결되는 non-orthogonality와 skewness를 coupling으로 직접 계산한다. 이웃 cell 중심 P, N의 연결 벡터 **d**, 공유면의 unit normal vector $\hat{\mathbf{n}}$, 면 중심 f_c, 연결선과 면의 교점 f_i에 대해

$$\theta_f = cos^{-1}(|d \cdot \hat{n}|/|d|) \quad (1)$$

$$s_f = |f_c - f_i|/|d| \quad (2)$$

로 정의되며 내부 면 전체(기본 격자 기준 약 2.9×10^6개)에 대해 수 초 내에 산출된다. Skewness는 **d**가 면과 평행에 가까울 때 식 (2)가 발산하는 정의상 특성이 있어 평균·최대 대신 상위 5% 값과 기준 초과 개수로 보고한다.

선별된 불량 요소는 위치 기반으로 군집화 되며, 각 군집에는 경계 상자, 요소 수, 그리고 구성 요소 부피로부터 역산한 등가 변 길이의 중앙값(실측 크기)이 부여된다. 격자 크기 지정이 상한(upper bound)으로 동작하는 SMESH의 특성상, 실측 크기를 초과하는 처방은 무효가 되므로 이 실측 값이 모든 처방의 기준이 된다.

2.4 Remeshing 전략과 결정표 기반 선택

국부 세분화 기법으로 box, shell, point, sphere의 네 가지 크기 제어와 경계층 파라미터 조정(layers)을 지원한다. 동일 조건 비교 실험(3.1절) 결과, 품질 개선 량은 기법의 형상이 아니라 "remeshing 시 설정한 형상의 부피"에 지배되며, 급격한 크기 전이는 오히려 불량을 양산함을 확인하였다. 이에 따라 표면 밀집 군집의 기본 처방인 shell은 군집 상자를 안쪽부터 30%씩 커지는 중첩되는 상자들로 감싸 격자 크기 변화를 완만하게 강제한다.

Remeshing 전략은 에이전트가 진단 결과로 부터 결정한다. 그 내부에서 LLM은 결정표에 기반한 선택만을 담당하며, 출력은 method, targets 두 가지의 선택형 탑안으로 제한된다. 상황 변수 (표면/배경, 형상 비, 고립 여부)와 판정 단서(직전 기

법, 불량률 추이, 정체 여부)는 코드가 계산하여 우선순위형 결정표와 함께 제시하고, 좌표·크기·전개·개수 상한 등 값의 조립은 코드가 담당한다. 조립 단계에는 군집 수 상한, 사이클당 단일 파라미터 계열, 격자 수 예산(사이클당 2×10^6개), 물리 경고 검사의 결정적 가드가 내장되어, LLM 출력의 변동성이 실행 안전성에 영향을 주지 않는다.

2.5 Remeshing 사이클 종료 판정과 재현성

사이클 종료는 불량률 개선과 aspect ratio, 심각 불량(Jacobian < 0.2) 비악화의 동시 충족으로 판정하며, 최대 3사이클과 정체 감지의 안전장치를 둔다. 심각 불량(Jacobian < 0.2)의 절대 개수는 prism–tetra 접합부의 기하학적 형태가 만드는 구조적 양으로서 어떤 처방으로도 감소하지 않음을 반복 실측으로 확인하였다. 따라서 이 지표는 감소를 요구하는 대상에서는 제외하고, 악화 여부만 확인하는 사이클 정지 조건으로 사용하였다. 모든 사이클의 계획 JSON, 품질 보고, 실행 스크립트는 실행 폴더에 보존되어 전 과정이 재현 가능하다.

2.6 초기 격자 생성 결과

Wing 형상(chord 0.12 m, half span 0.3 m)에 대해 자연어 요청부터 품질 진단까지 사용자 입력 2회(요청, 승인)로 2분 18초에 완료되었으며, 프리즘 311,610개와 사면체 1,089,204개의 하이브리드 격자(총 1.40×10^6 요소, 음수·영 부피 요소 0개)가 얻어졌다. 동일 요청의 반복 실행에서 요소 수가 완전히 일치하여 결정적 생성의 재현성을 확인하였다.

2.7 자동 개선 루프

Table 1은 초기 파라미터를 달리한 두 실행의 사이클별 품질 이력이다. 실행 A에서는 1사이클만에 종료 조건이 충족되어 최종적으로 불량률이 0.617%에서 0.354%로, face non-orthogonality θ_{95}가 37.0°에서 27.7°, skewness s_{95}가 1.11에서 0.35로 줄어드는 개선을 얻었으며, 총 소요는 8분 1초, 사용자 개입은 승인 2회였다. 실행 B에서는 사이클마다 서로 다른 지표(aspect ratio, Jacobian < 0.2 개수)가 종료를 연기하여 3사이클의 점진 심화가 수행되었고, 불량률이 0.563%에서 0.253%로 55% 감소하였다. 동일 시스템이 초기 조건에 따라 상이한 사이클 수로 반응한 것은, 루프가 고정 절차의 재생이 아니라 실측 판정에 의해 구동됨을 보여준다. 두 실행의 전 사이클(총 4회)에서 LLM의 전략 제안은 모두 첫 시도에 유효한 답안으로 수렴하였고, 매 사이클 LLM이 지목한 대상은 코드 생성 단계의 제약에 따라 밀집도 상위 2곳으로 자동 제한되었다. 제약은 군집 수 상한, 사이클 당 하나의 파라미터만 수정, 추가 격자 수 2×10^6개 이하, 물리적으로 불가능한 형상은 제외와 같은 내용으로 적용된다. 아래의 Fig. 2는 실행의 개선 전후 불량 요소 분포를 나타낸다.

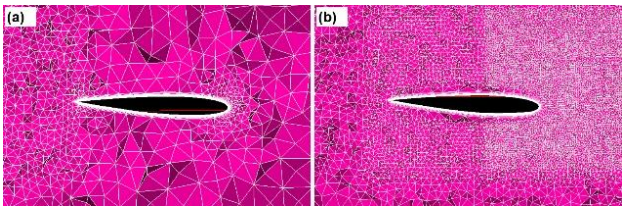

Fig. 2 Meshing results: before(left) and after(right)

Table1에서 N은 전체 격자 요소 수, rate는 벽면 prism을 제외한 tetra 중 jacobian이 0.4보다 작은 요소의 비율 (불량률), θ_{95}, s_{95}는 각각 식 (1), (2)로 정의된 면 non-orthogonality와 skewness의 상위 5% 값을 나타내며, 사이클 0은 개선 전 초기 격자를 의미한다.

Table 1 Quality history of two autonomous runs

Run	Cycle	N(10^6)	Failure rate(%)	s_{95}	θ_{95}(°)
A	0	1.40	0.617	1.111	37.0
	1	2.97	0.354	0.354	27.7
B	0	1.44	0.563	1.762	39.8
	1	2.87	0.335	0.377	28.3
	2	3.65	0.271	0.348	27.2
	3	4.11	0.253	0.343	27.2

3. 결 론

3.1 세분화 전략의 동일 조건 비교

동일 초기 격자와 동일 불량 군집에 대해 네 가지 크기 제어 기법을 비교한 결과(Table 2), 품질 개선은 기법의 형상이 아니라 적용한 형상이 품질이 낮은 격자를 얼마나 많이 덮었는지에 비례하였다. Point 형상 remeshing은 비용이 거의 없으나(+0.2%) 군집을 덮지 못해 불량률이 정체되었고(0.617에서 0.607%), 군집을 여유 있게 덮은 sphere 형상은 최고 품질(0.379%)을 최대 비용(+112.6%)으로 얻었다. 군집 형상과 일치하는 shell(겹겹이 쌓여 점점 커지는 rectangular box형상)이 비용 대비 효율에서 가장 우수하였으며(+48.1%에 0.457%), 반경을 군집 크기로 줄인 sphere는 shell과 불량률 감소가 유사한 지점으로 수렴하였다. 이 결과는 전략 선택 문제를 "덮임 범위의 결정"으로 정식화할 수 있음을 시사하며, 2.4절 결정표의 실측 근거가 된다.

Table 2 Refinement strategies compared on the identical initial mesh (ΔN: element increase)

Strategy	ΔN(%)	Rate(%)	s_{95}	θ_{95}(°)
Initial mesh	-	0.617	1.111	37.0
Point	+0.2	0.607	1.106	36.9
Sphere(tight)	+43.1	0.483	0.446	30.7
Shell	+48.1	0.457	0.426	30.2
Sphere(Large)	+112.6	0.379	0.356	28.2

3.2 LLM 판단의 신뢰성과 로컬 구동

개발 과정에서 축적된 전략 제안 31회의 tracking에서 LLM 선택의 오류는 0회였던 반면, local LLM의 서술형 JSON 출력의 완성 실패가 병목이었으며 skill이 늘수록 Qwen의 remeshing 성공률이 88%에서 46%로 오히려 하락하였다. 이는 skill 추가가 생성해야 할 구조화 출력의 길이와 프롬프트 내 스키마 분량을 함께 증가시키기 때문으로, 실제 실패는 값의 오류가 아니라 괄호 불일치와 같은 구조적 미완성이 대부분이었다. 즉 성공률을 선택 정확도와 생성 완성도의 곱으로 볼 때, skill 증가는 전자를 유지한 채 후자만을 저하시킨다. 출력을 2.4절의 선택형 답안으로 재설계한 후에는 5종 전 문항이 첫 시도에 수렴하였고(문항당 1~4분), 이는 2.7절 실전 루프의 전 사이클에서 재현되었다. 이러한 결과는 소형 로컬 LLM의 한계를 출력 인터페이스의 재설계로 보완할 수 있음을 보여준다. 전 과정은 8 GB VRAM의 GPU (NVIDIA GeForce RTX 5050) 한 장에서 수행되어(생성 속도 18.5 token/s) 형상 데이터의 외부 유출 없이 동작한다.

3.3 연구 결과와 향후 방향성

본 연구는 로컬 LLM 에이전트와 결정적 실행 계층의 결합을 통해 자연어 기반 자동 재격자화라는 새로운 접근을 제시하였다. Wing 외부 유동 격자에 대한 적용에서 제안 시스템은 사용자 승인 2회만으로 약 8분 내에 생성-진단-재격자화의 전 주기를 완주하였으며, 서로 다른 초기 조건의 두 실행에서 불량률을 약 43~55% (0.617 → 0.354%, 0.563 →0.253%)로 감소시키고 face non-orthogonality를 37.0 - 39.8°에서 27.2 - 27.7°로 개선하였다. 이는 판단을 LLM에 국한하고 수치 조립 · 검증 · 실행을 코드에 위임하는 역할 분담 구조가 8 GB급 GPU(NVIDIA GeForce RTX 5050)에서 구동되는 소형 로컬 모델만으로도 신뢰성 있는 품질 개선 루프를 이끌 수 있음을 실증한 것이다. 자연어 인터페이스는 격자 생성에 요구되던 전문 지식의 장벽을 낮추어 공학적 의도와 계산 실행 사이의 간극을 해소하며, 전 과정의 로컬 수행은 형상 데이터의 보안 요구를 함께 충족한다. 향후에는 축적된 격자 생성 · 개선 실측 사례를 활용한 미세조정(fine-tuning)과 skill 체계의 고도화를 통해 로컬 LLM의 판단 능력 자체를 향상시켜, 실무에서 실제 사용 가능한 재격자화 에이전트로 발전시키고자 한다.

후 기

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. RS-2024-00355857).

참고문헌

(1) Chen, Y., Zhu, X., Zhou, H. and Ren, Z., 2024, “MetaOpenFOAM: An LLM-Based Multi-Agent Framework for CFD,” arXiv Preprint, arXiv:2407.21320.

(2) Fan, E., Hu, K., Wu, Z., Ge, J., Miao, J., Zhang, Y., Sun, H., Wang, W. and Zhang, T., 2026, “ChatCFD: A Large Language Model-Driven Agent for End-to-End Computational Fluid Dynamics Automation with Structured Knowledge and Reasoning,” *Adv. Intell. Discov.*, Vol. 2, No. 3, e202500174.

(3) Xu, Z., Wang, L., Wang, C., Chen, Y., Luo, Q., Yao, H.-D., Wang, S. and He, G., 2025, “CFDAgent: A Language-Guided, Zero-Shot Multi-Agent System for Complex Flow Simulation,” *Phys. Fluids*, Vol. 37, No. 11, 117124.