

2020 한국전산유체공학회 통합학술대회

2020. 10. 20, 제주 매종글래드 호텔

오픈 소스 라이브러리를 활용한 CFD Workflow 자동화 기술 연구

Research on the Automation Technology for CFD Workflow Using Open Source Libraries

(주)넥스트폼

Convergence Technology Team

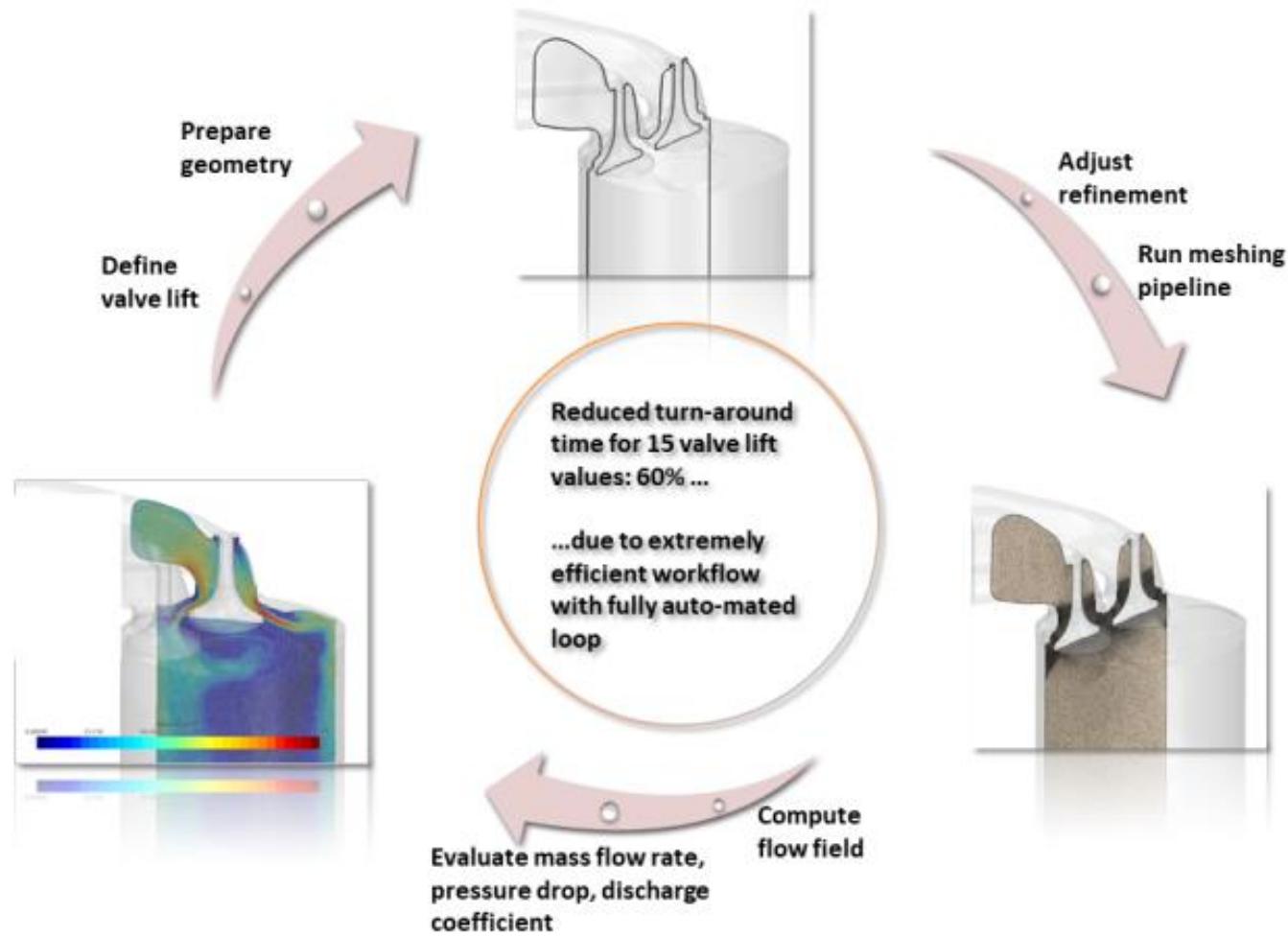
이 상 철



Contents

- 연구 배경
- 연구 개요
- 핵심 요소기술 소개
- 향후 추가 계획

연구 필요성



반복적, 시간 소모적 업무



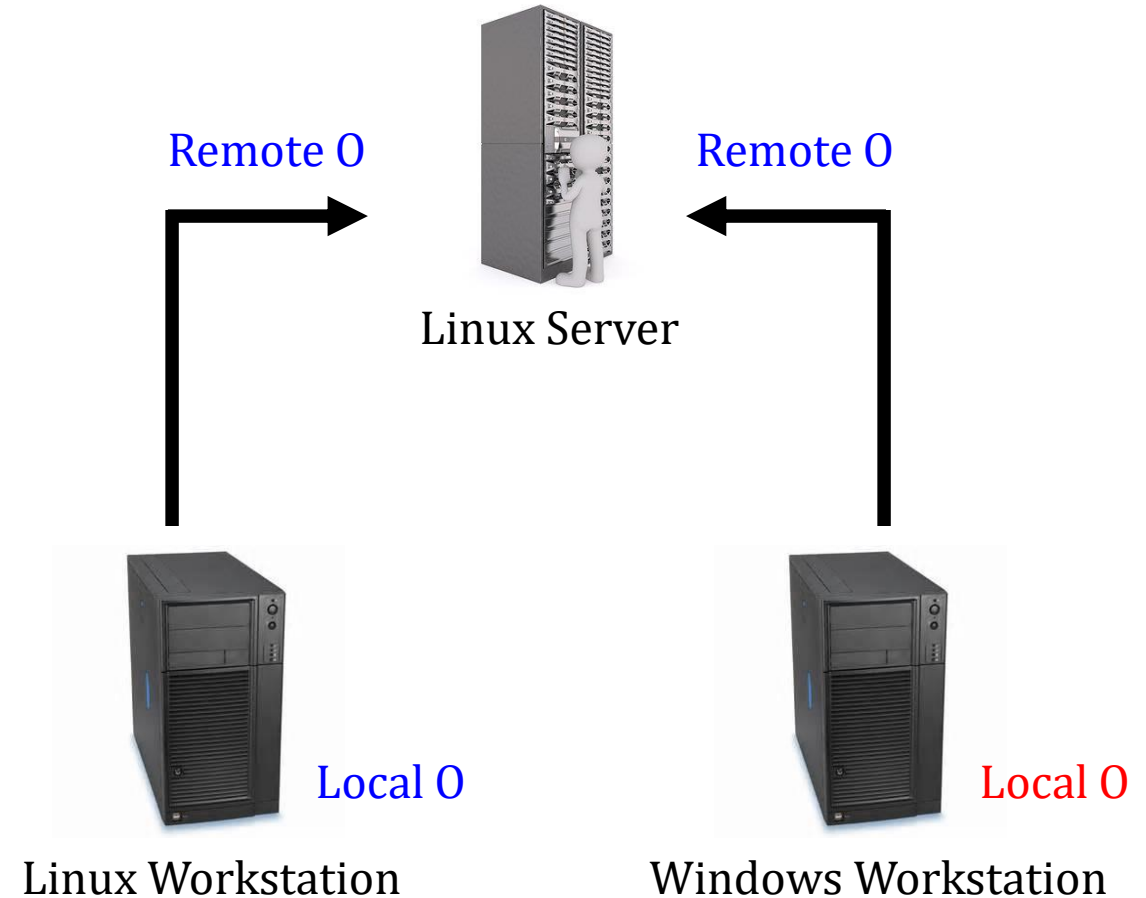
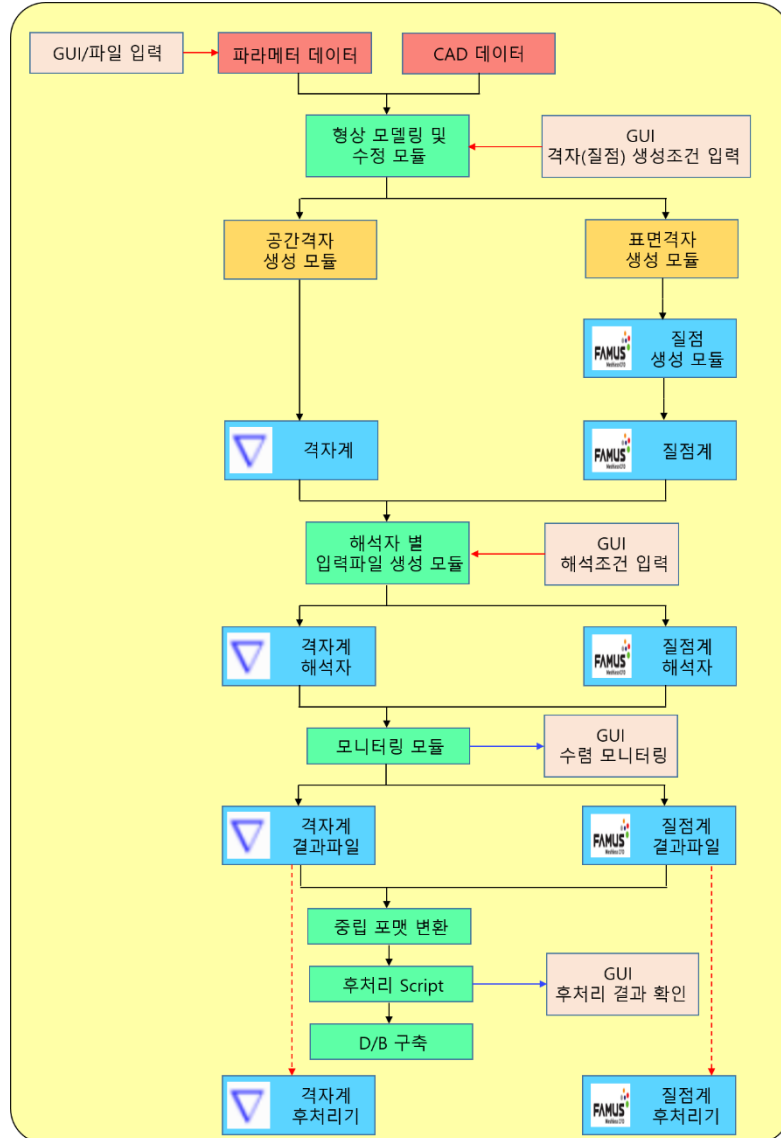
업무 자동화



효율적, 생산적 업무

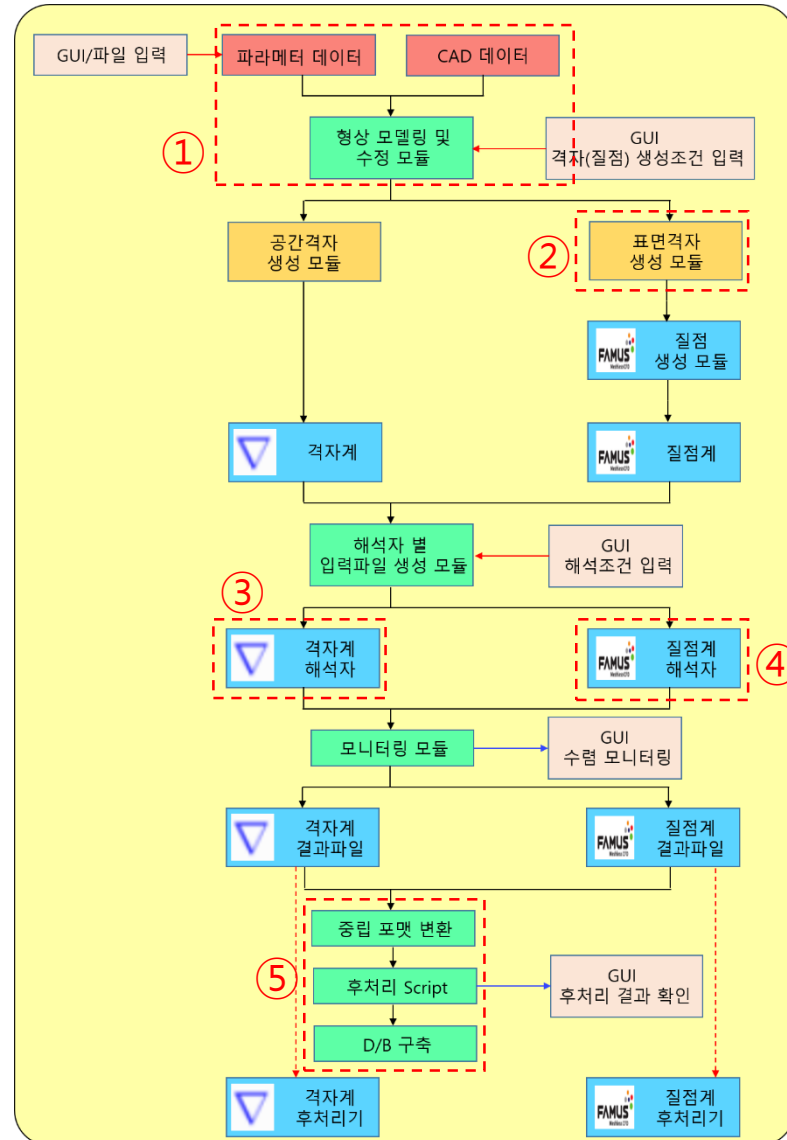
Routine CFD Workflow

연구 목표



→ 주 작업흐름 → GUI 입력 → GUI 출력

기 진행 개발기술



- ① 파라미터 기반 형상 모델링(OpenCascade)
- ② 표면격자 생성 모듈(Netgen)
- ③ 격자계 해석자(TSLAeroFoam)
- ④ 질점계 해석자(**FAMUS**)
- ⑤ 후처리 자동화 및 D/B 구축(VTK & MongoDB)

핵심 요소기술 소개

파라미터 기반 형상 모델링(OpenCascade)

[Missile DATCOM의 형상 정의]

Centerbody – Axisymmetric, Elliptic

Nose – Cone, Ogive, Power, Haack, Karman

Afterbody – Cone, Ogive

Finset – Hex, NACA, Arc

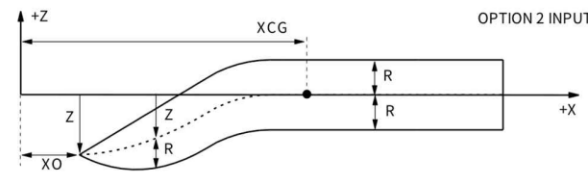
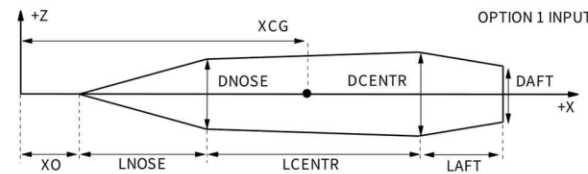
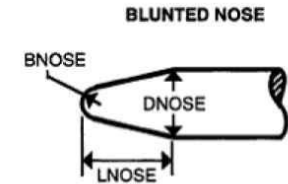
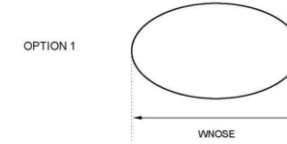
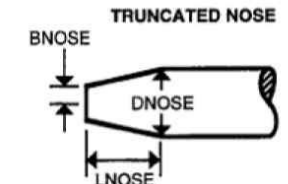
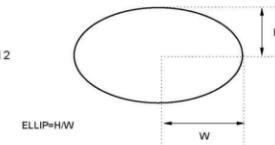


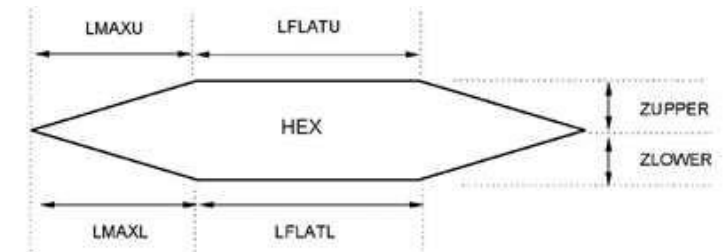
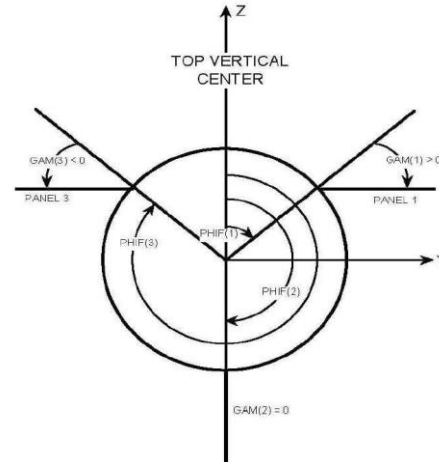
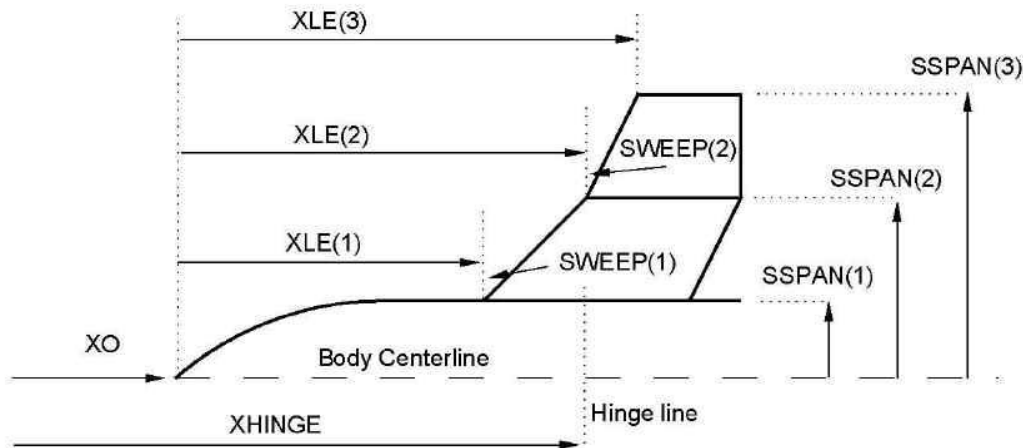
Figure 2. Axisymmetric Body Geometry Variables



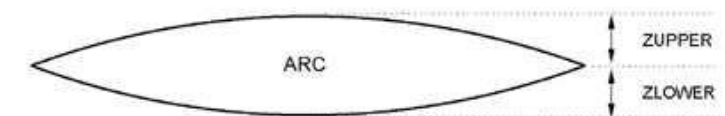
BNOSE = ###
TRUNC = .FALSE.



BNOSE = ###
TRUNC = .TRUE.



NOTE: All parameters must be input at each span station



NOTE: ARC section only allows ZUPPER and ZLOWER

파라미터 기반 형상 모델링(OpenCascade)(계속)

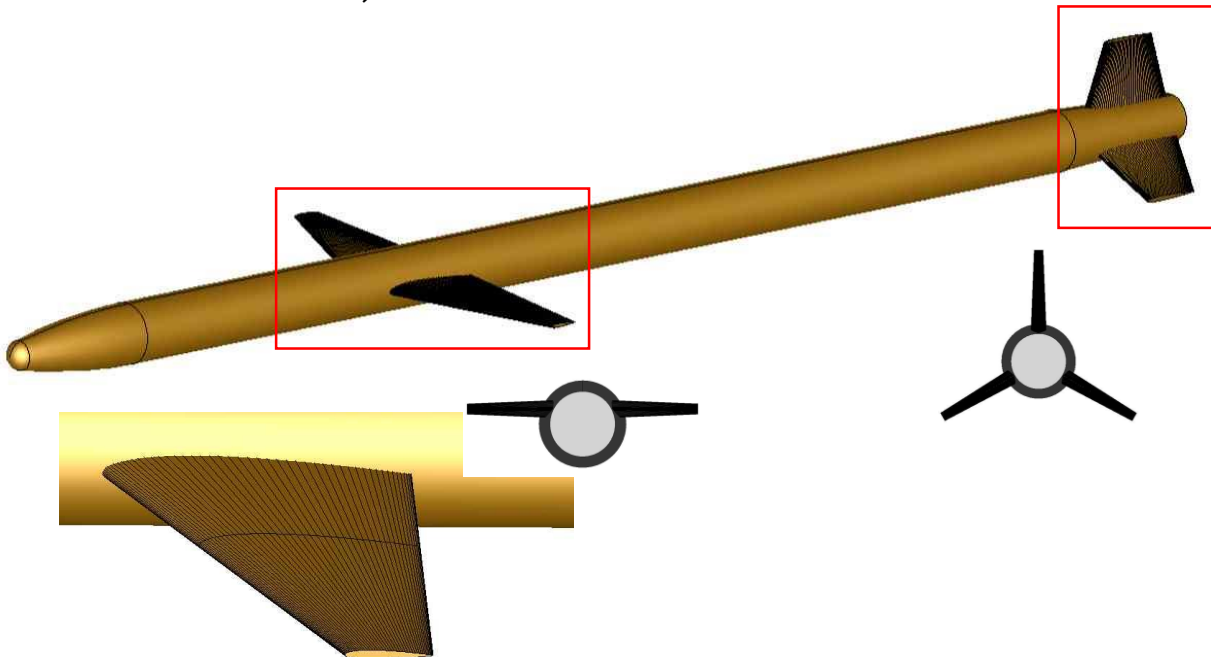
형상 #1

Centerbody – Axisymmetric (Ellipticity 1.0)

Nose – Ogive (blunted)

Afterbody – Cone

Finset – NACA, NACA



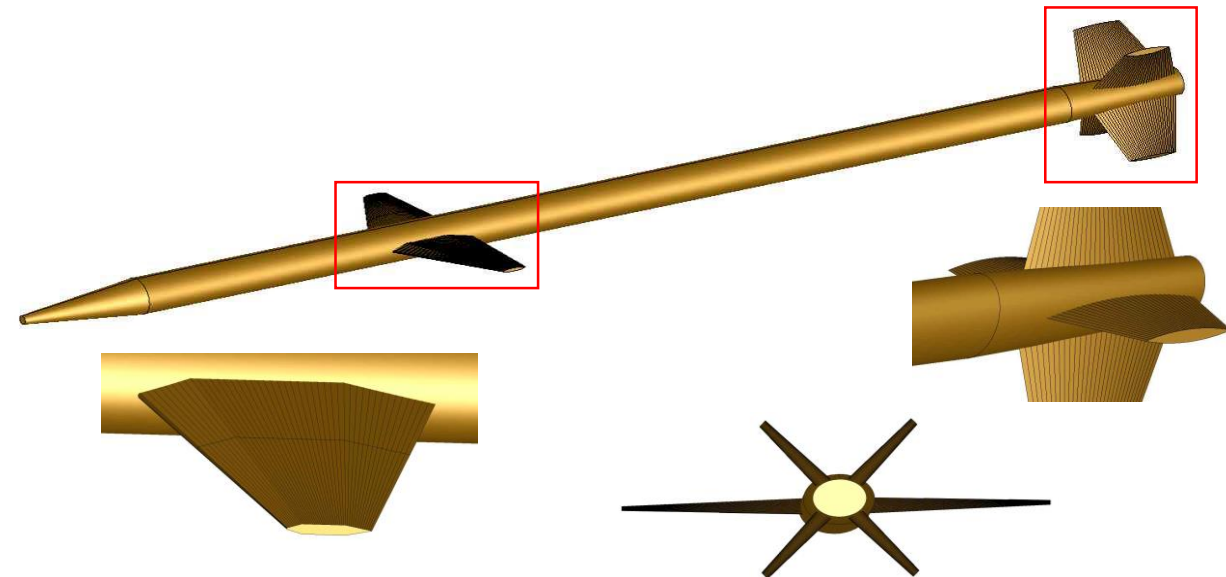
형상 #2

Centerbody – Elliptic (Ellipticity 0.5)

Nose – Cone (truncated)

Afterbody – Ogive

Finset – Hex, Arc



파라미터 기반 형상 모델링(OpenCascade)(계속)

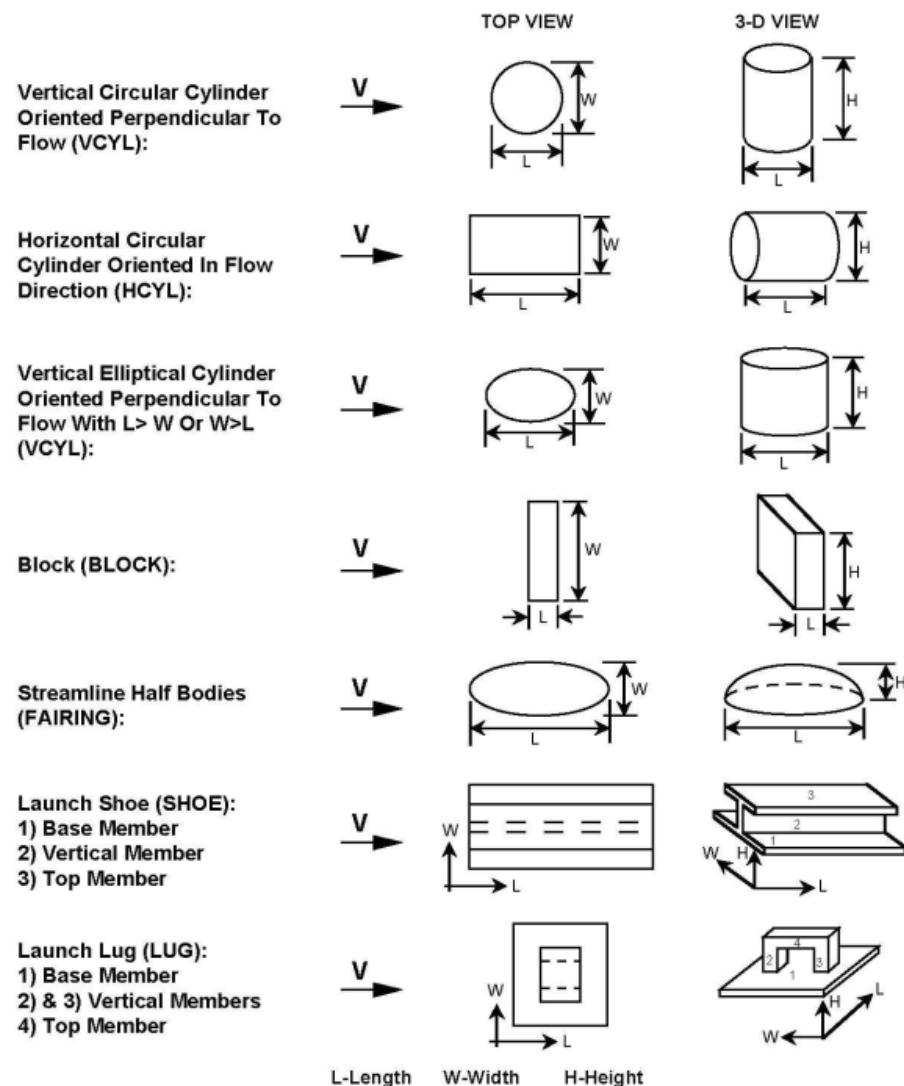
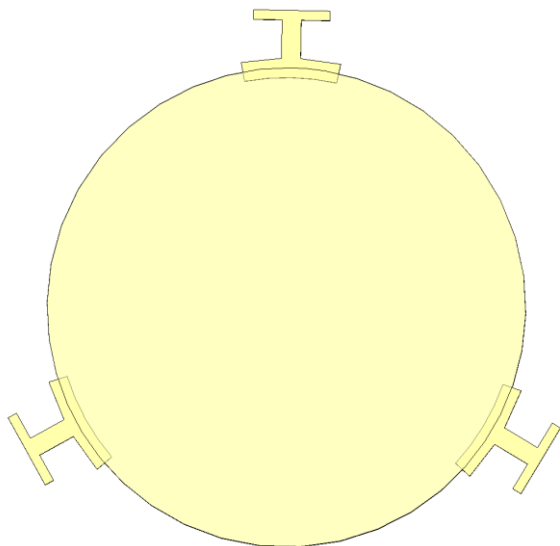
[Protuberance 형상 정의]

형상 - Length, Width, Height

부착 - 개수, 각도

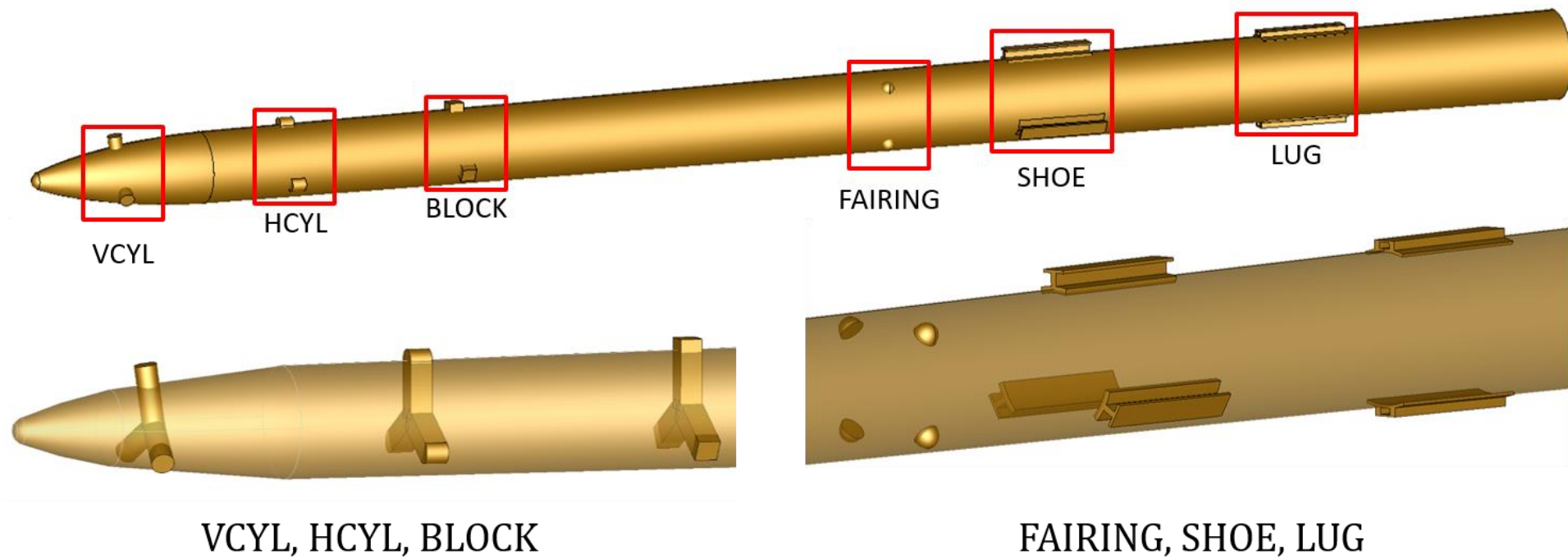
Offset - Missile 표면에서 이격

Missile Body 표면의 기울기를 고려하여 형상 생성

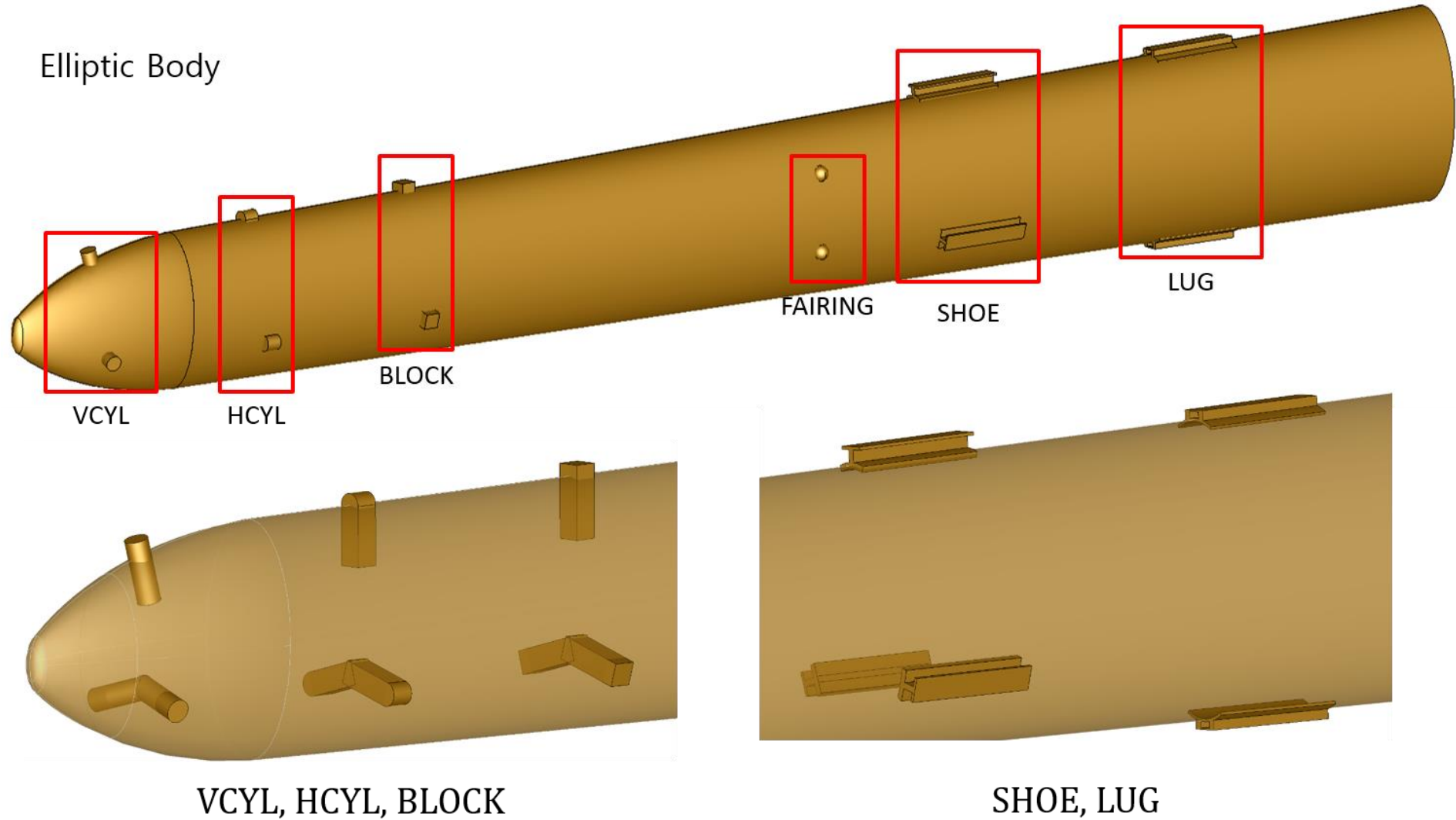


파라미터 기반 형상 모델링(OpenCascade)(계속)

Axisymmetric Body

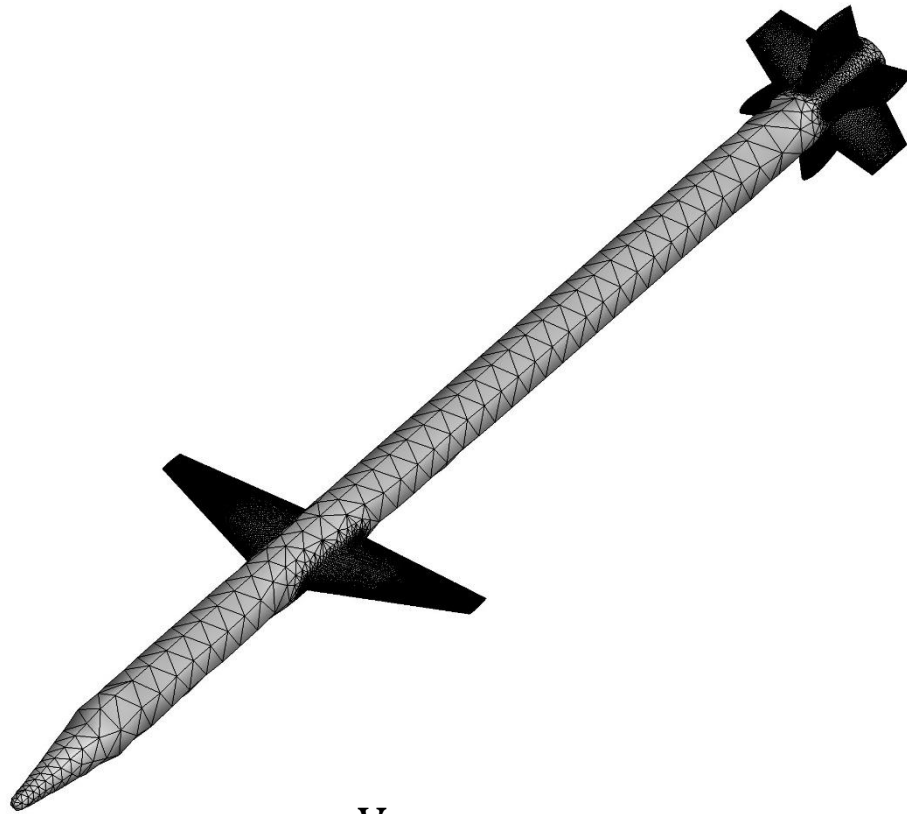


파라미터 기반 형상 모델링(OpenCascade)(계속)

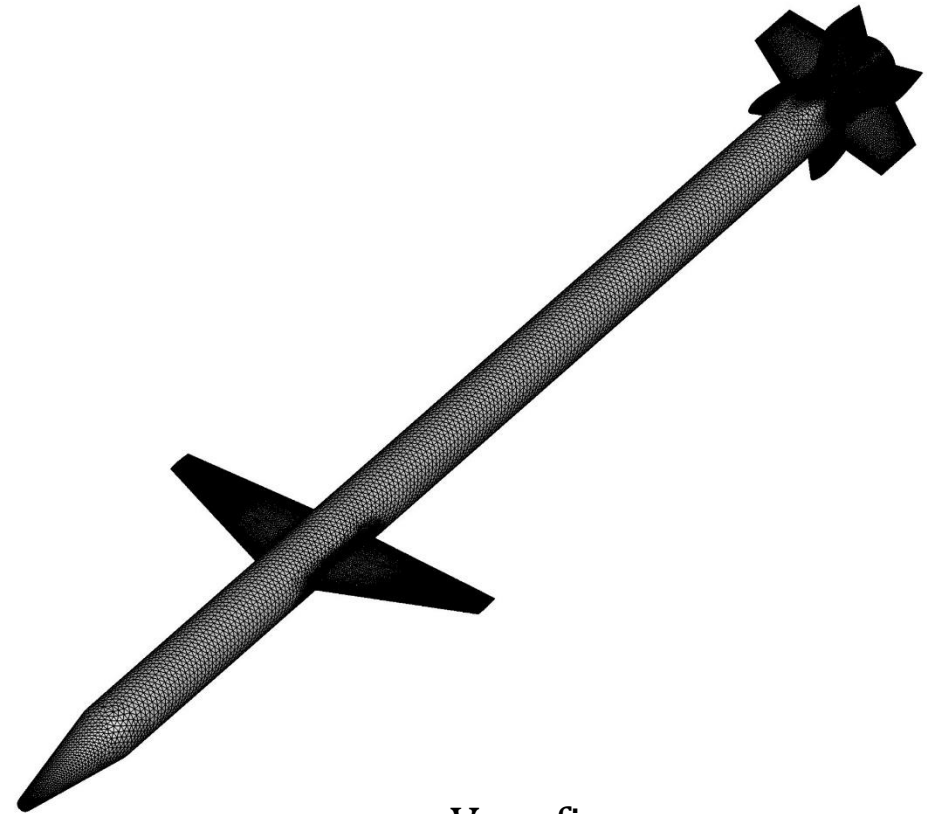


표면격자 생성 모듈(Netgen)

Fineness – Very coarse ~ Very fine (5 levels), User defined(본 옵션을 사용하면 Mesh size min/max값 이용)



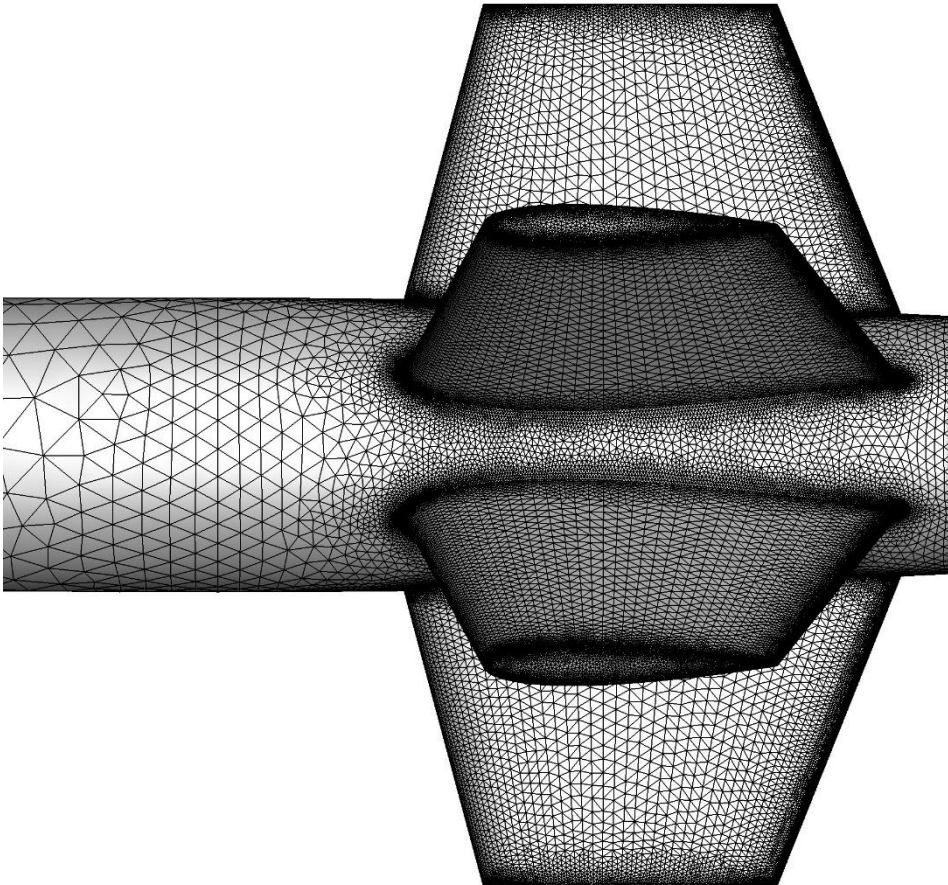
Very coarse



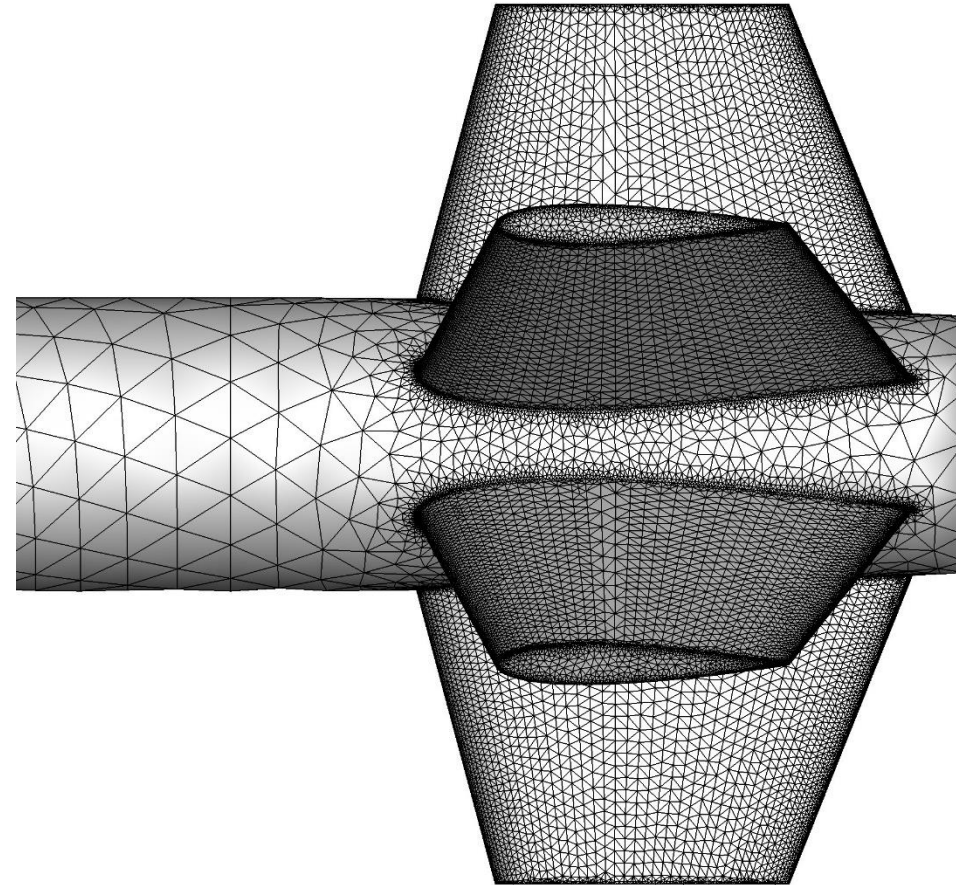
Very fine

표면격자 생성 모듈(Netgen)(계속)

Grading - 0 ~ 1 값, 인접한 두 격자의 크기 변화 설정



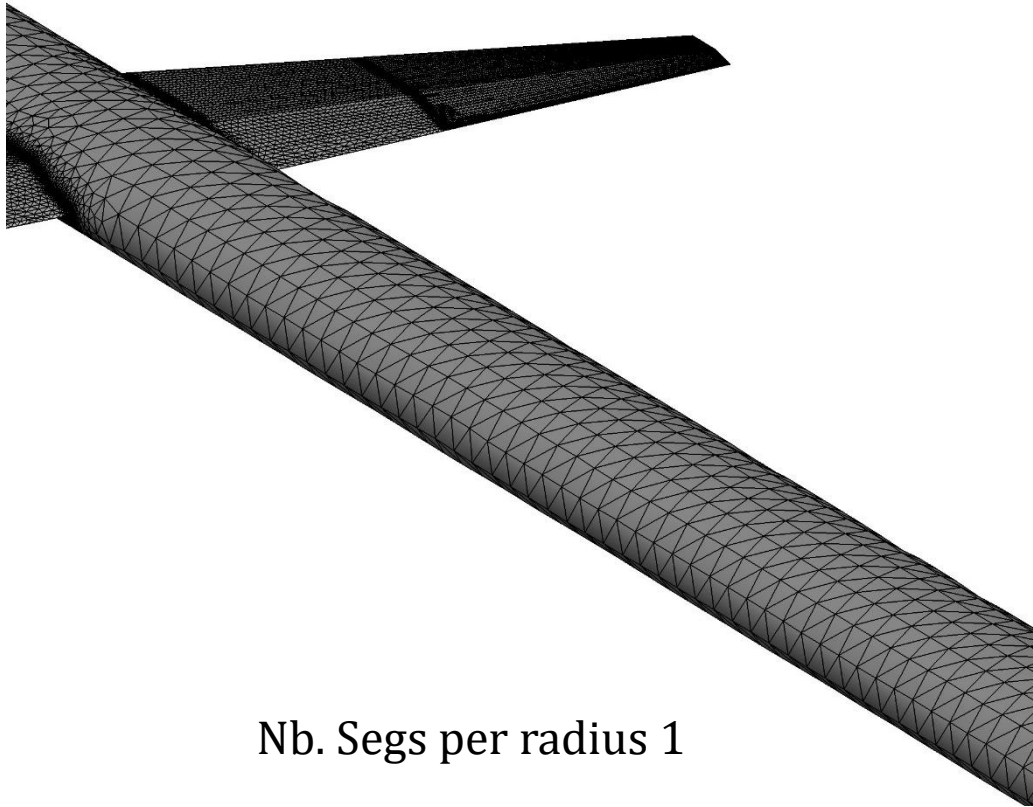
Grading 0.1



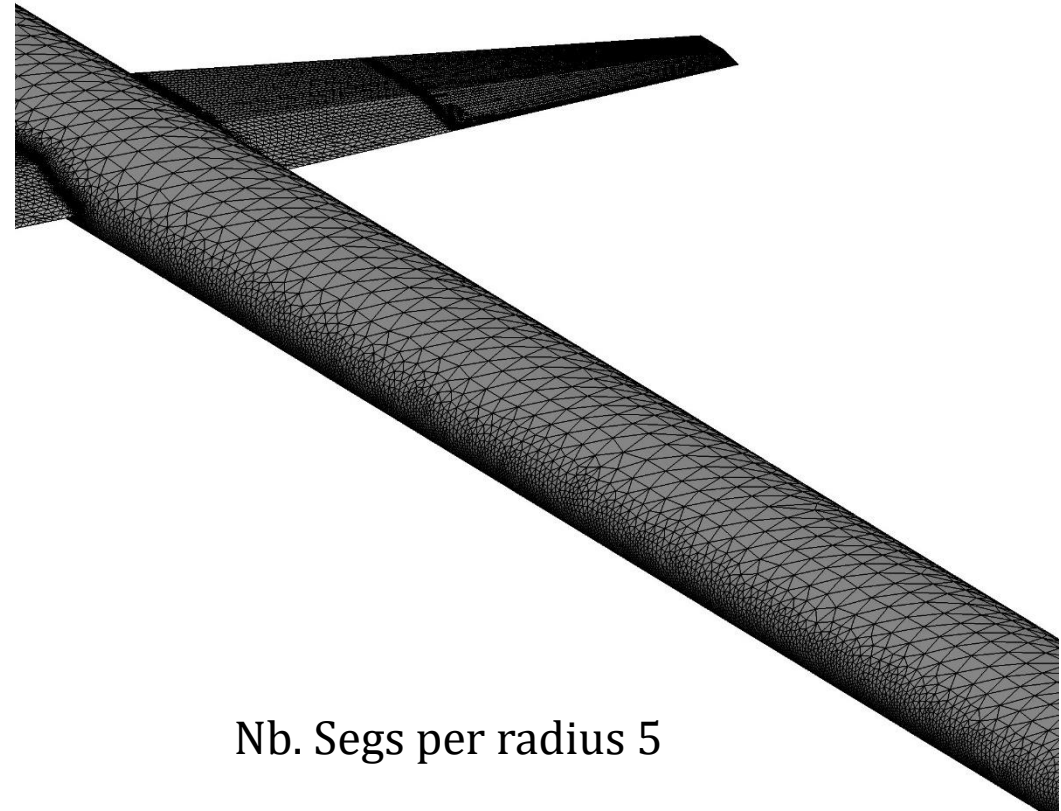
Grading 0.9

표면격자 생성 모듈(Netgen)(계속)

Nb. Segs per radius - 곡률에 따른 격자 밀집도 설정



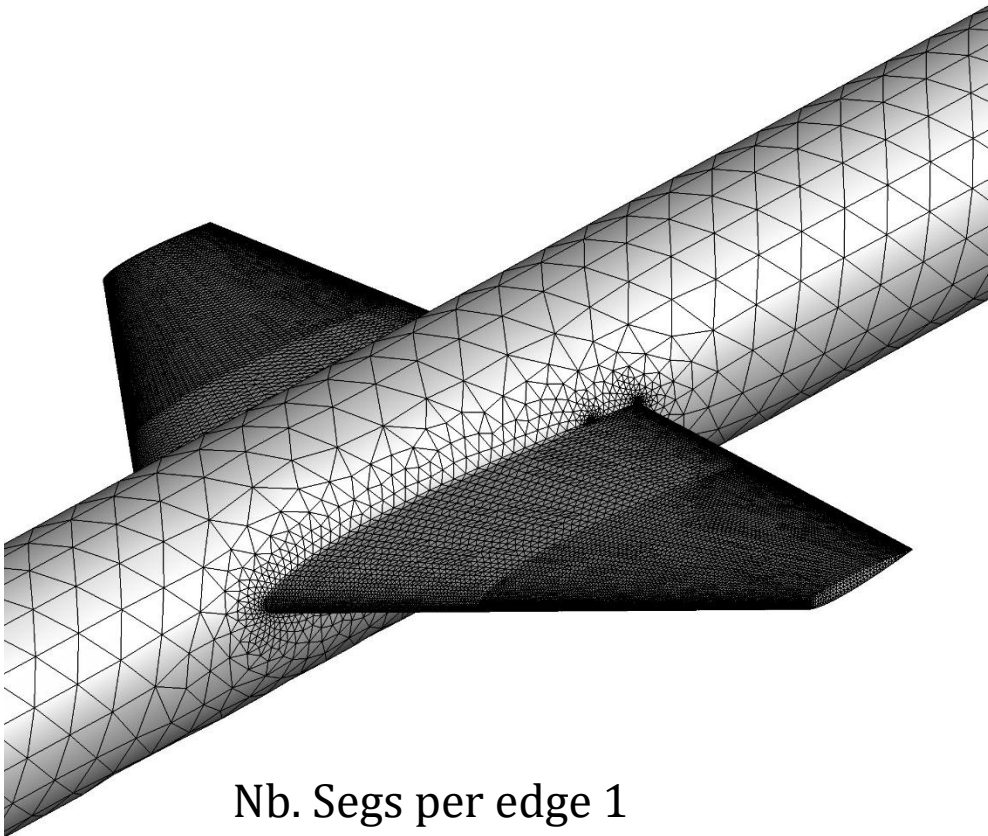
Nb. Segs per radius 1



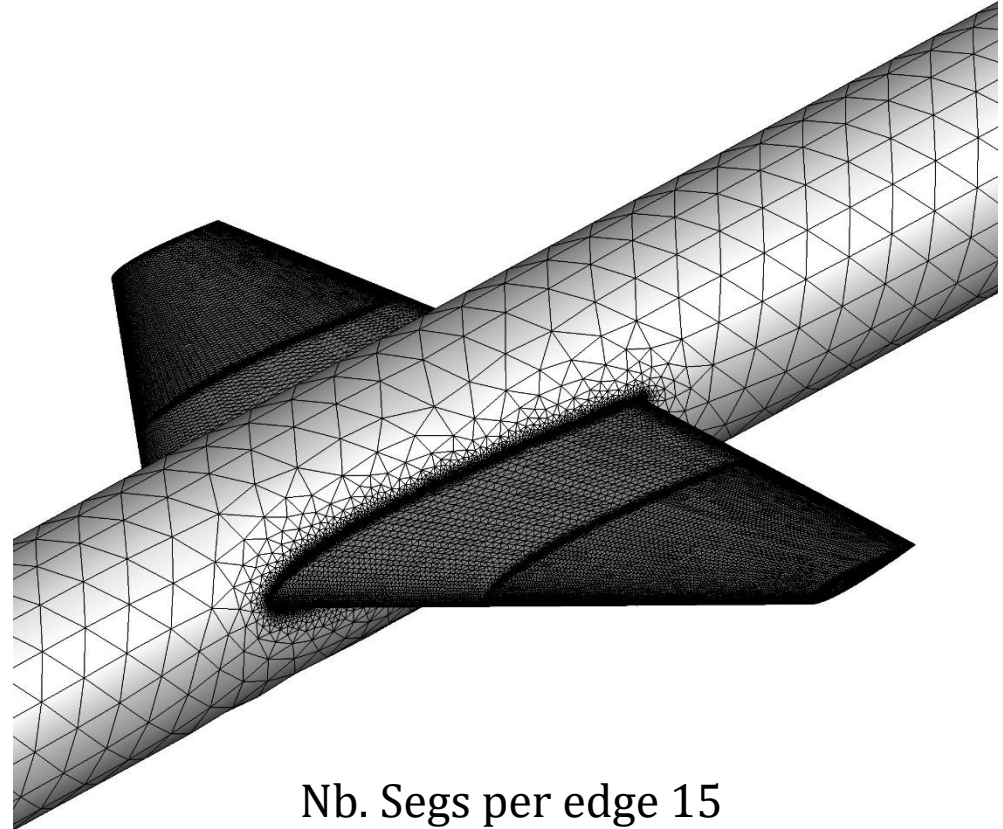
Nb. Segs per radius 5

표면격자 생성 모듈(Netgen)(계속)

Nb. Segs per edge – face edge의 격자 밀집도 설정

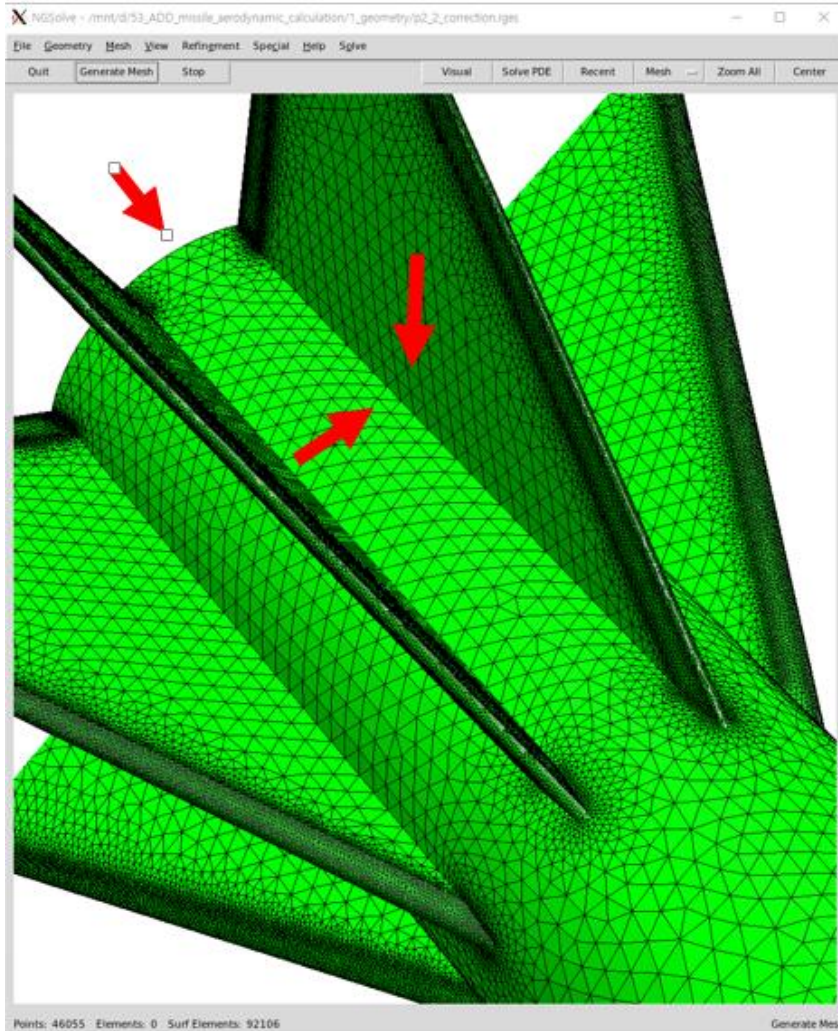


Nb. Segs per edge 1



Nb. Segs per edge 15

표면격자 생성 모듈(Netgen)(계속)



Netgen 격자의 문제점

Fin, Protuberance, Body 형상 STEP파일 불러오기



필요한 Edge 또는 Face의 Local refinement를 위해 형상 Explode
(Solid > Face > Edge 순으로 Explode)



Fin, Protuberance, Body 형상 Fuse



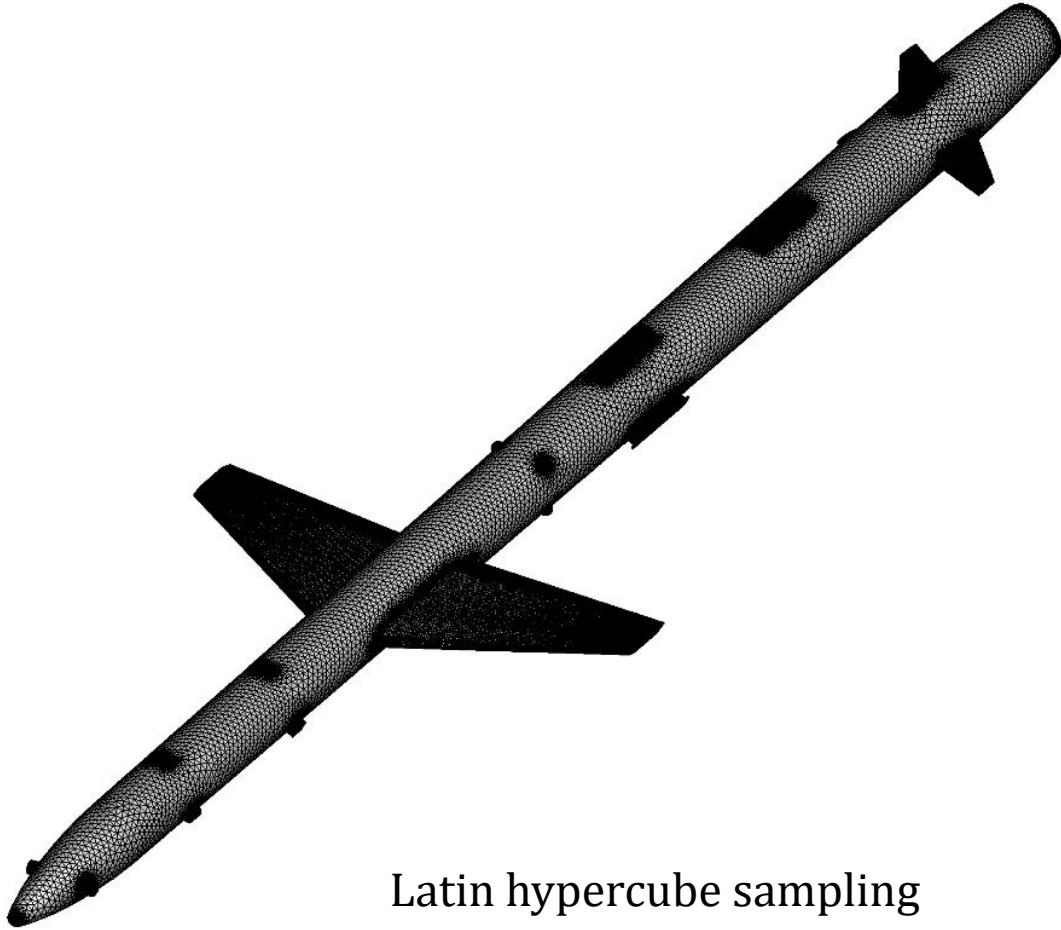
Explode된 Face나 Edge 중 Local refinement에 필요한
Face와 Edge만 따로 분류



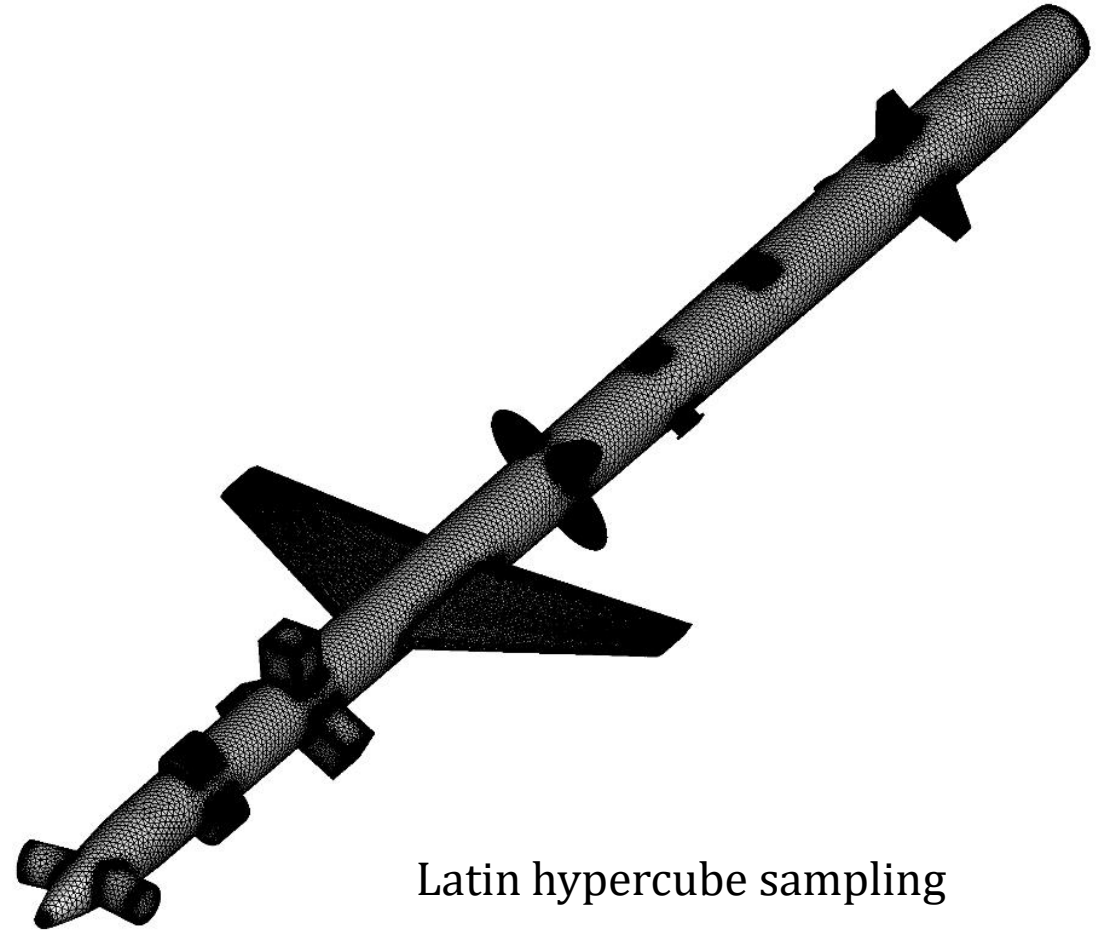
설정된 Mesh parameter와 Local refinement를 적용해서 격자 생성

격자 개선을 위한 절차

표면격자 생성 모듈(Netgen)(계속)

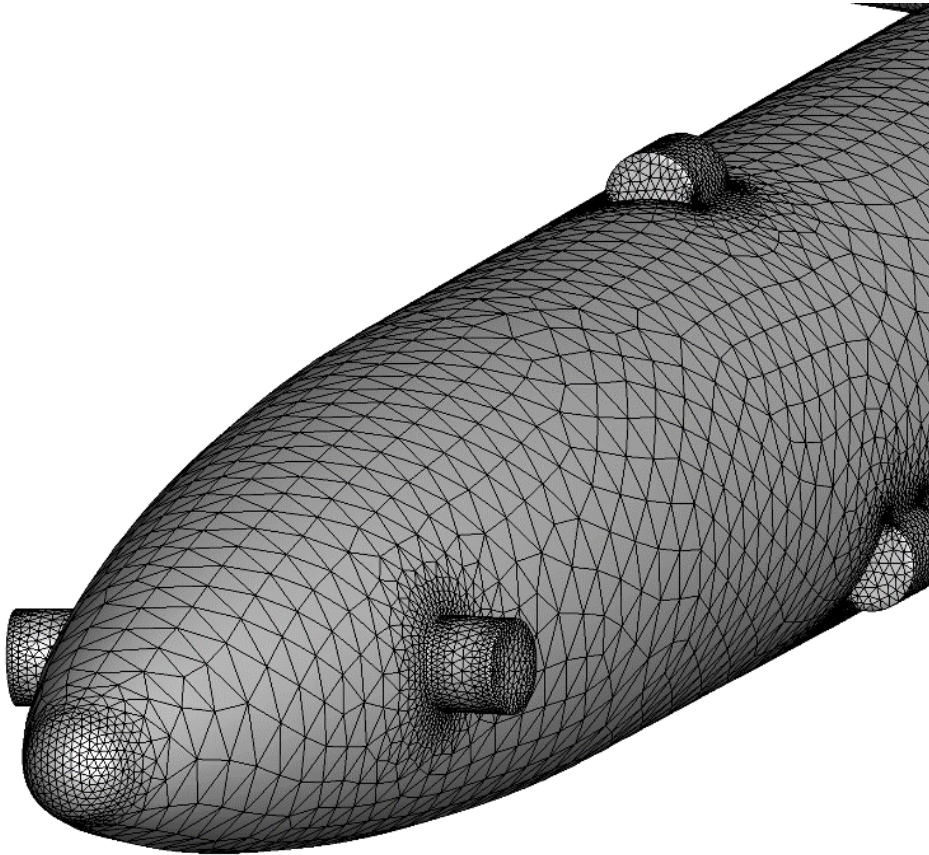


Latin hypercube sampling
표면격자 생성 #1

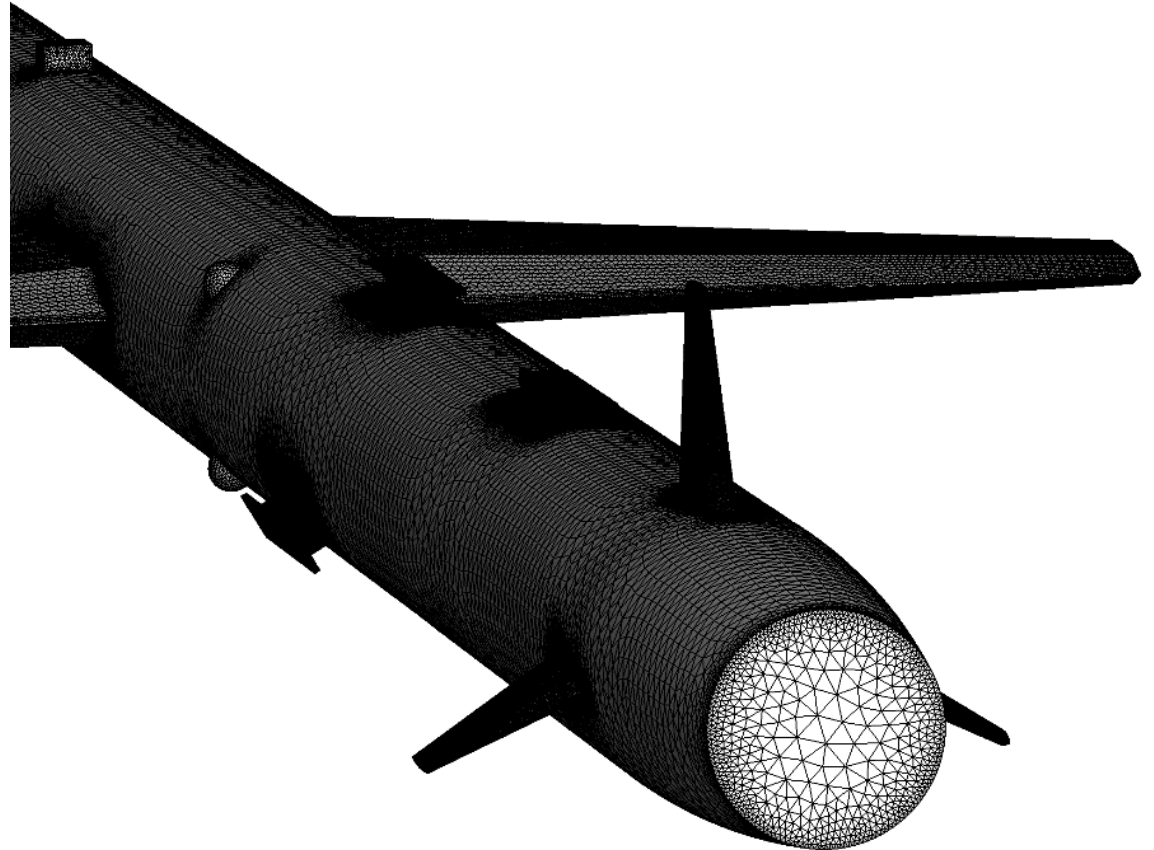


Latin hypercube sampling
표면격자 생성 #2

표면격자 생성 모듈(Netgen)(계속)

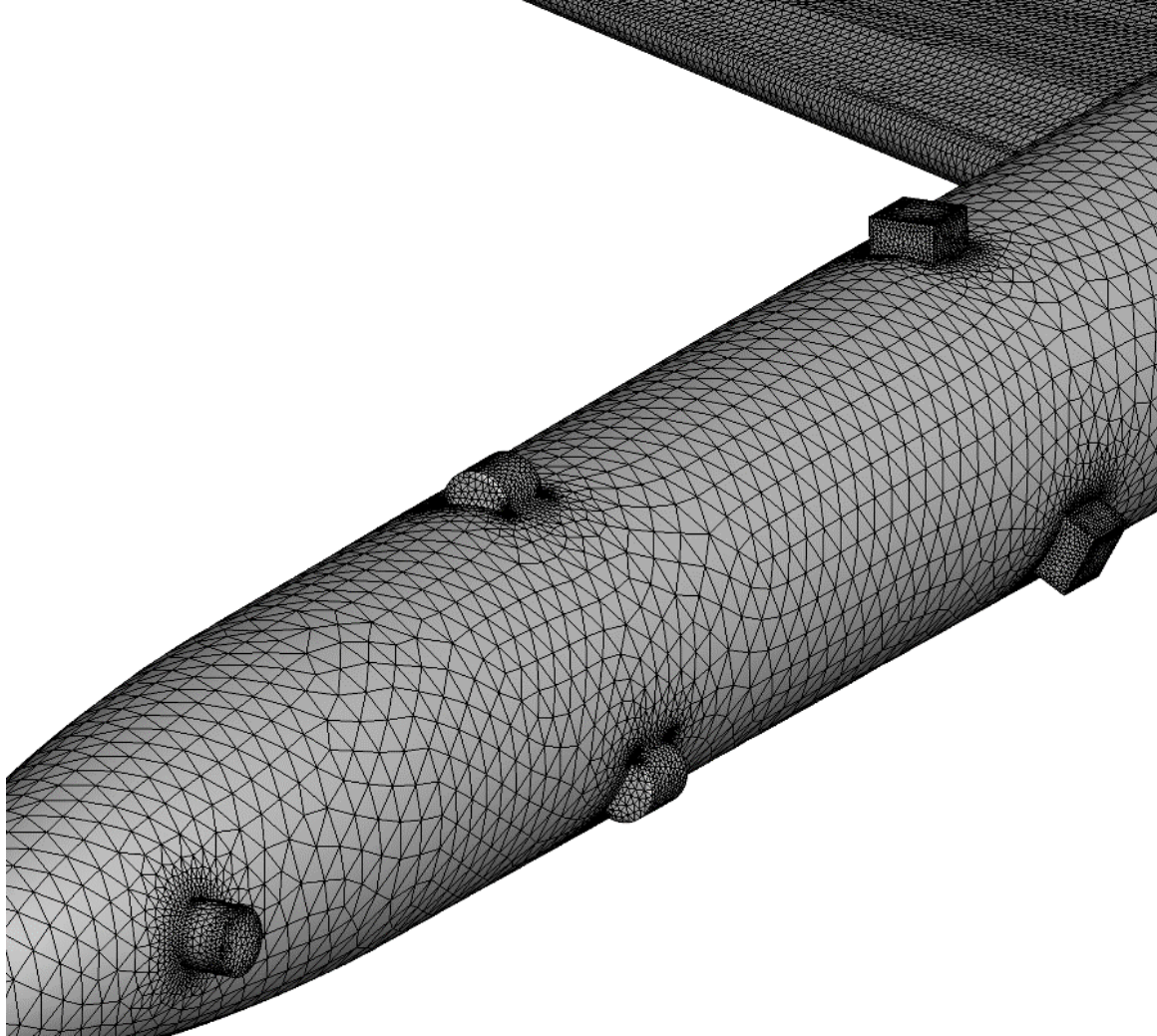


Nose Part

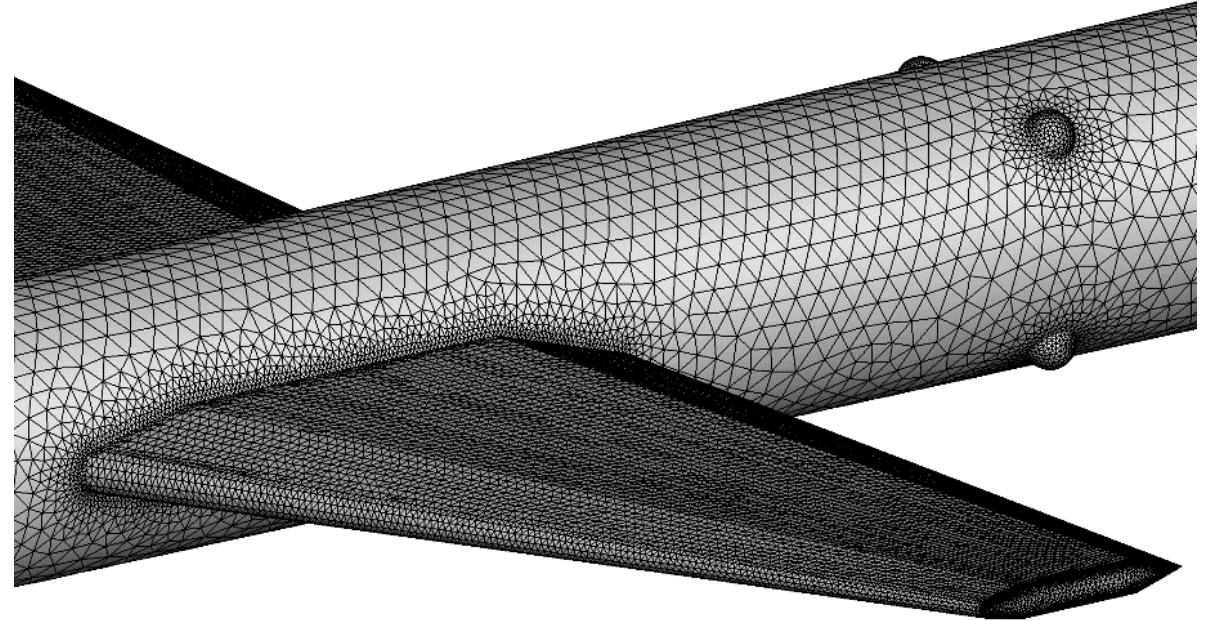


Afterbody Part

표면격자 생성 모듈(Netgen)(계속)

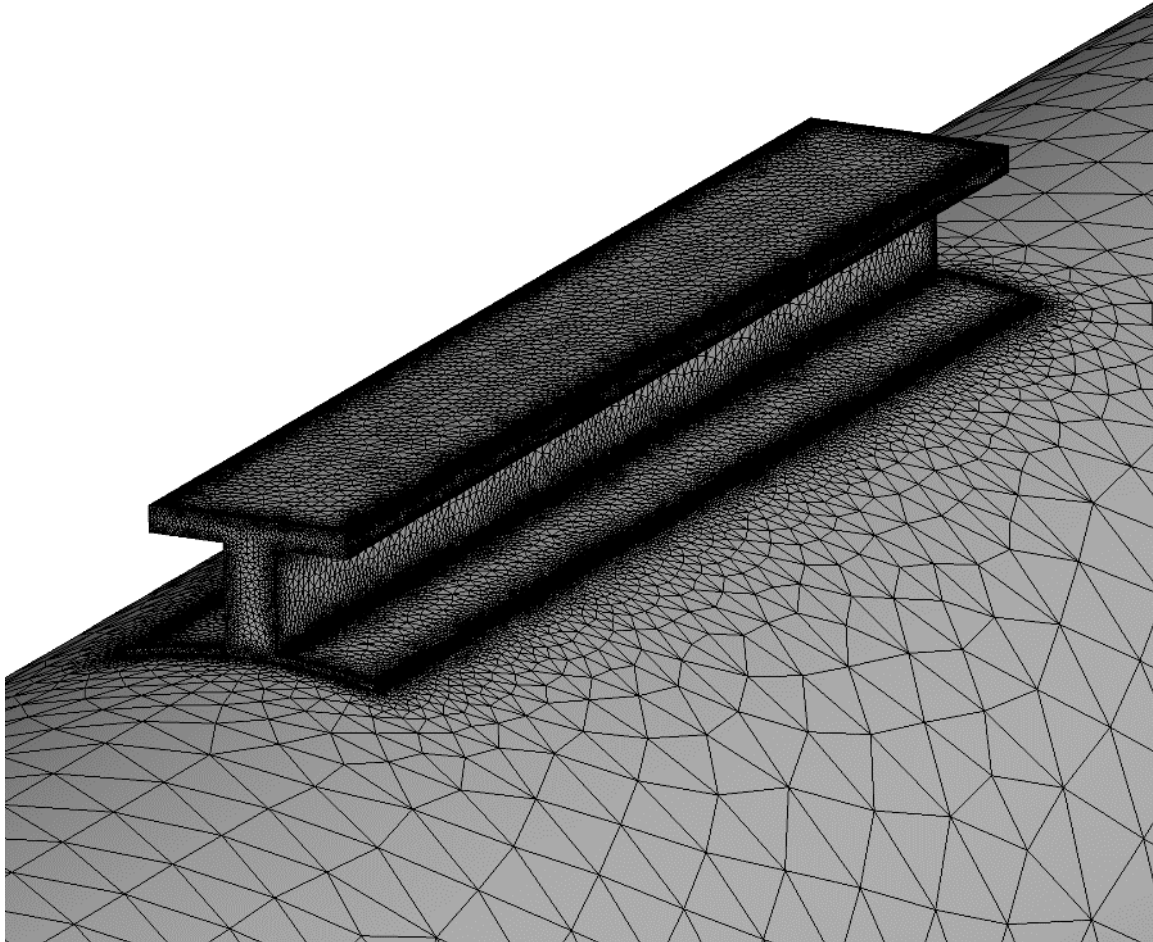


VCYL, HCYL and BLOCK

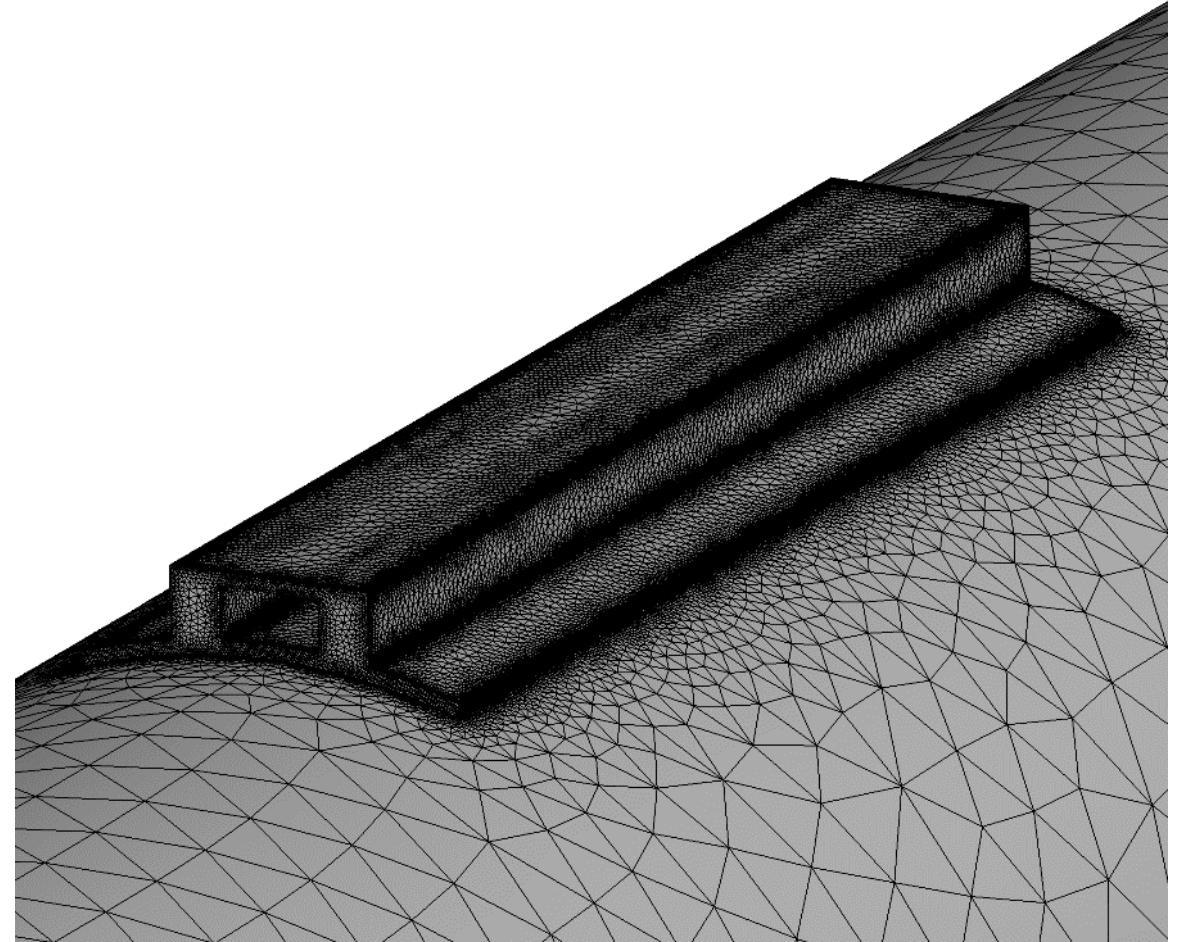


FIN and FAIRING

표면격자 생성 모듈(Netgen)(계속)



LAUNCH SHOE



LAUNCH LUG

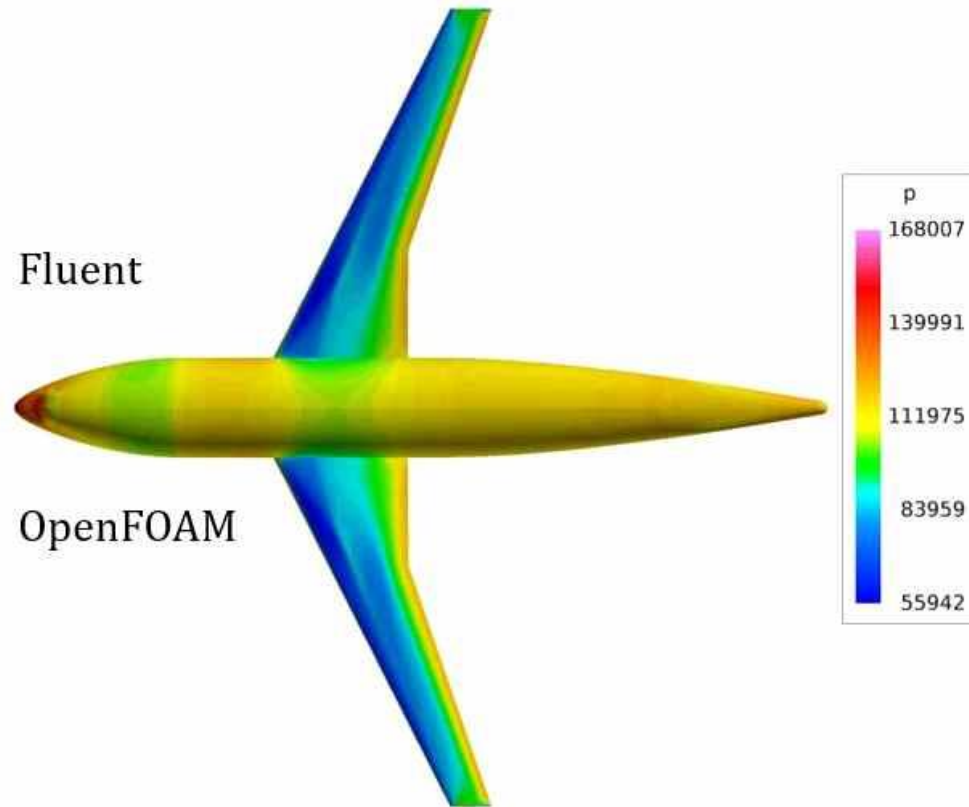
격자계 해석자(TSLAeroFoam)

- ☐ OpenFOAM을 기반으로 개발한 density based coupled 압축성 공력해석자
- ☐ 내재적 시간 적분을 위해 Implicit LU-SGS (Lower Upper Symmetric Gauss Seidel) 기법 적용
- ☐ 점성 플럭스 자코비안 처리를 위해 TSL (Thin Shear Layer) approximation 적용
- ☐ 공간 2차 정확도 확보 및 충격파 포착 능력 향상을 위해 Roe, AUSM+ 및 AUSM+Up 기법 적용
- ☐ Riemann invariant 경계조건 개발
- ☐ 오픈소스 난류 모델 라이브러리 활용으로 $k\omega$ -SST 및 SA 등 다양한 난류 모델 지원

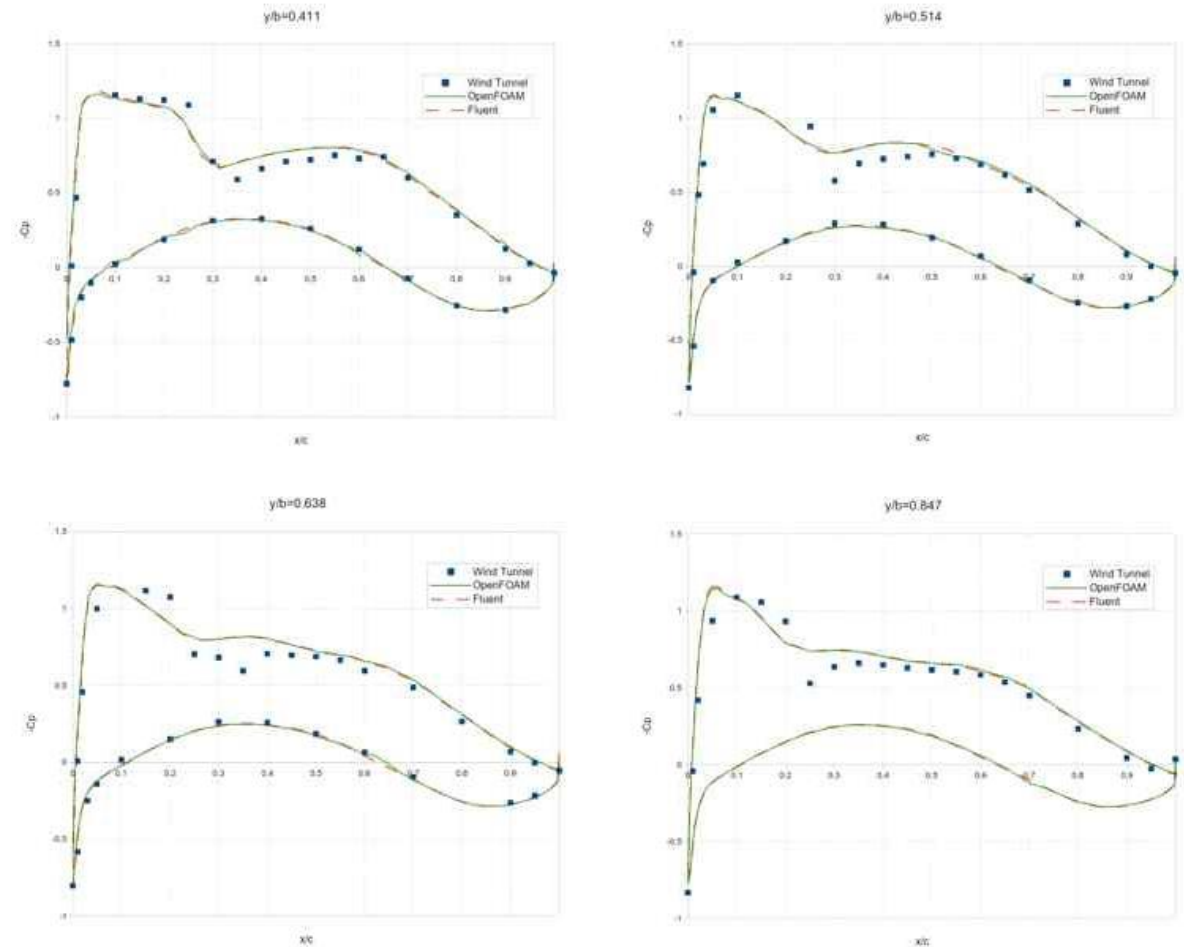
핵심 요소기술 소개

격자계 해석자(TSLAeroFoam)(계속)

천음속 3차원 항공기(Ma=0.75, AOA=0.49°)



압력 분포 비교



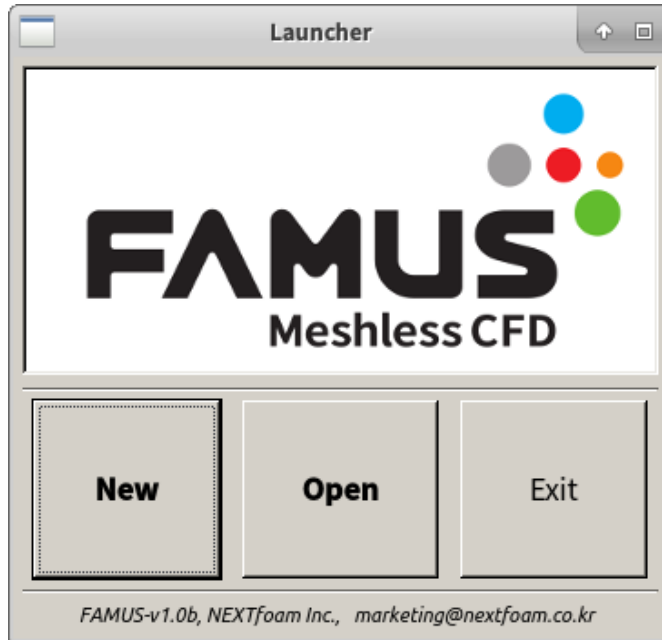
위치 별 압력 분포 비교

질점계 해석자(FAMUS)

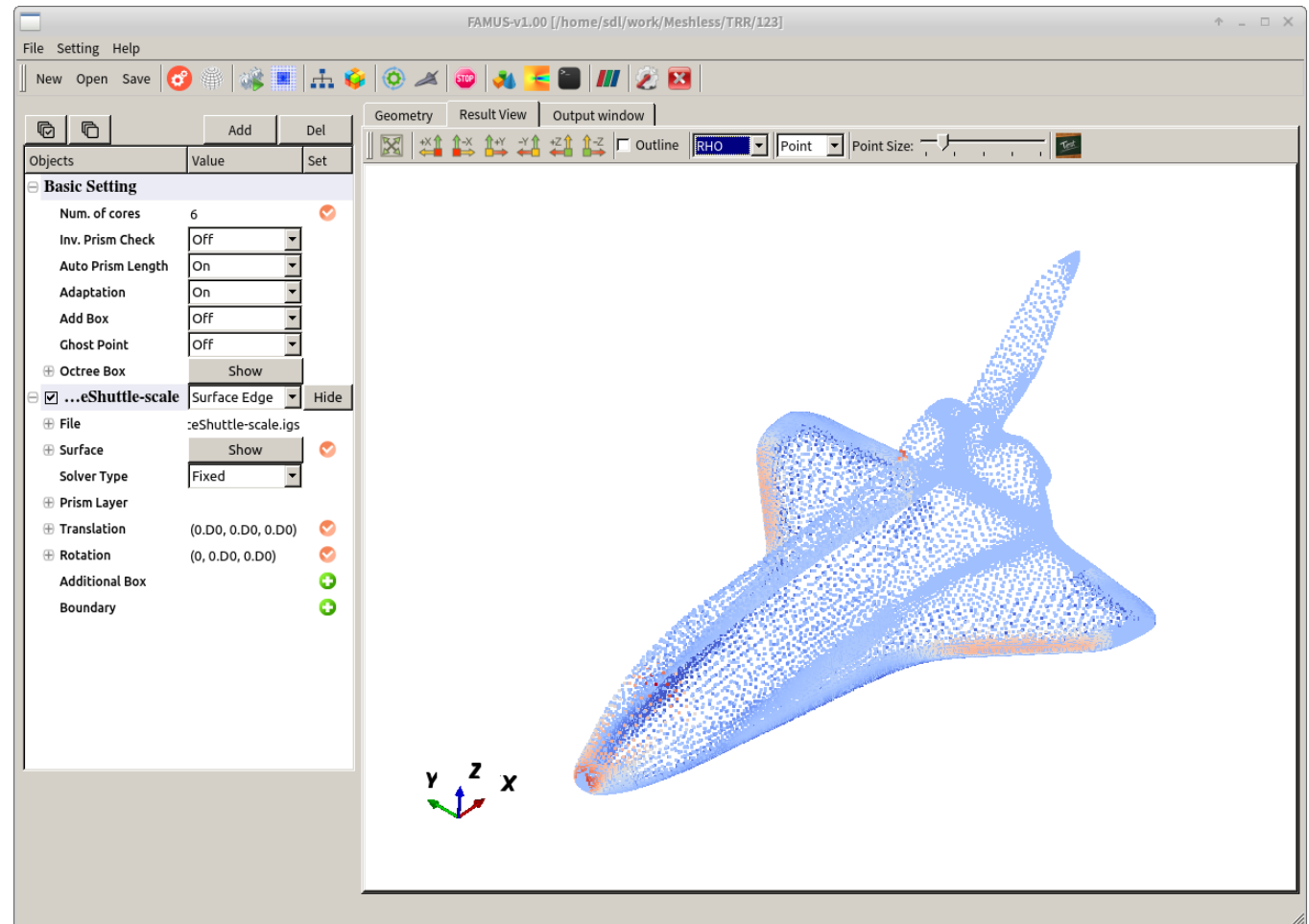
- ☐ 격자없이 질점계로 해석 영역을 생성하는 무격자 방식 적용
- ☐ 질점계 적용으로 인해 복잡한 형상 및 물체의 이동을 포함하는 해석영역 생성 용이
- ☐ 격자계 기반의 FVM과 동일한 수준의 보존성을 갖추기 위해 Geometric Conservation Law 충족
- ☐ 내재적 시간 적분을 위해 Implicit LU-SGS (Lower Upper Symmetric Gauss Seidel) 기법 적용
- ☐ 대류항 플럭스 계산기법 AUSMPW+를 무격자 기법에 적용
- ☐ 공간 재구성 기법을 위해 안정성이 높은 다차원 제한자(MLP) 사용

핵심 요소기술 소개

질점계 해석자(FAMUS)(계속)



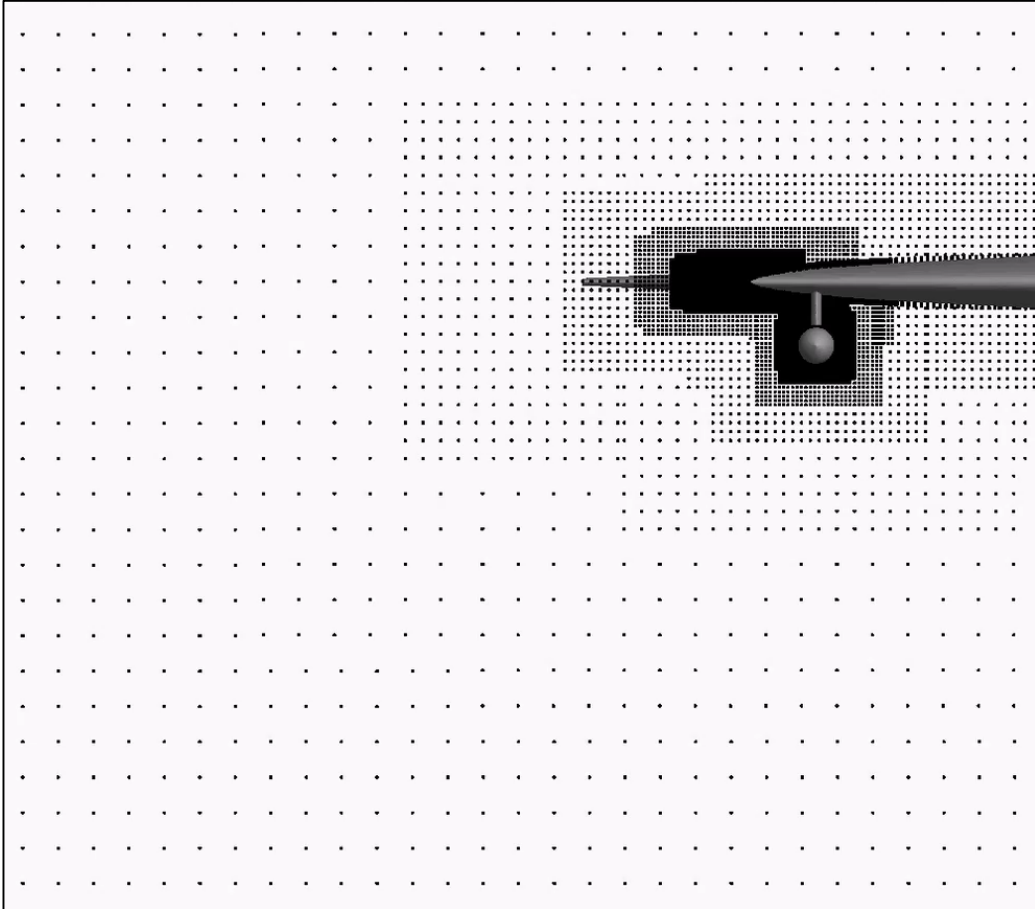
FAMUS launcher



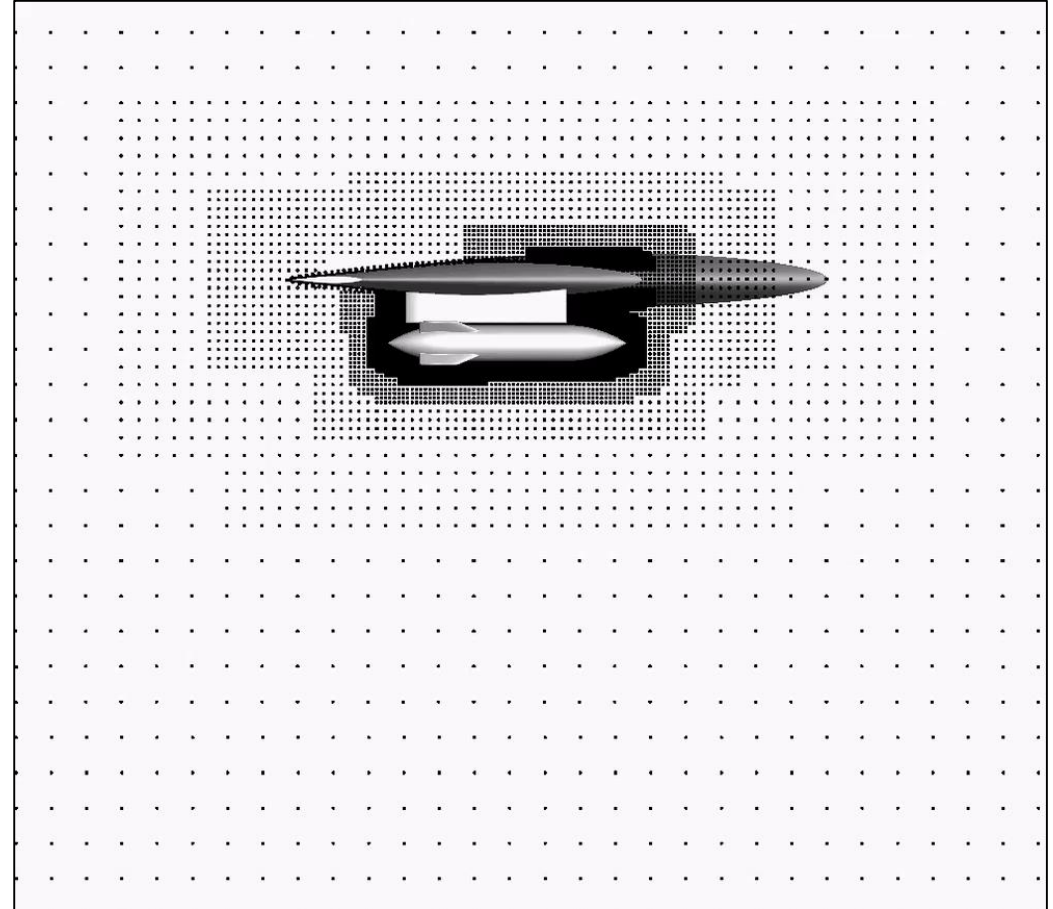
FAMUS GUI

질점계 해석자(FAMUS)(계속)

EGLIN Store Separation



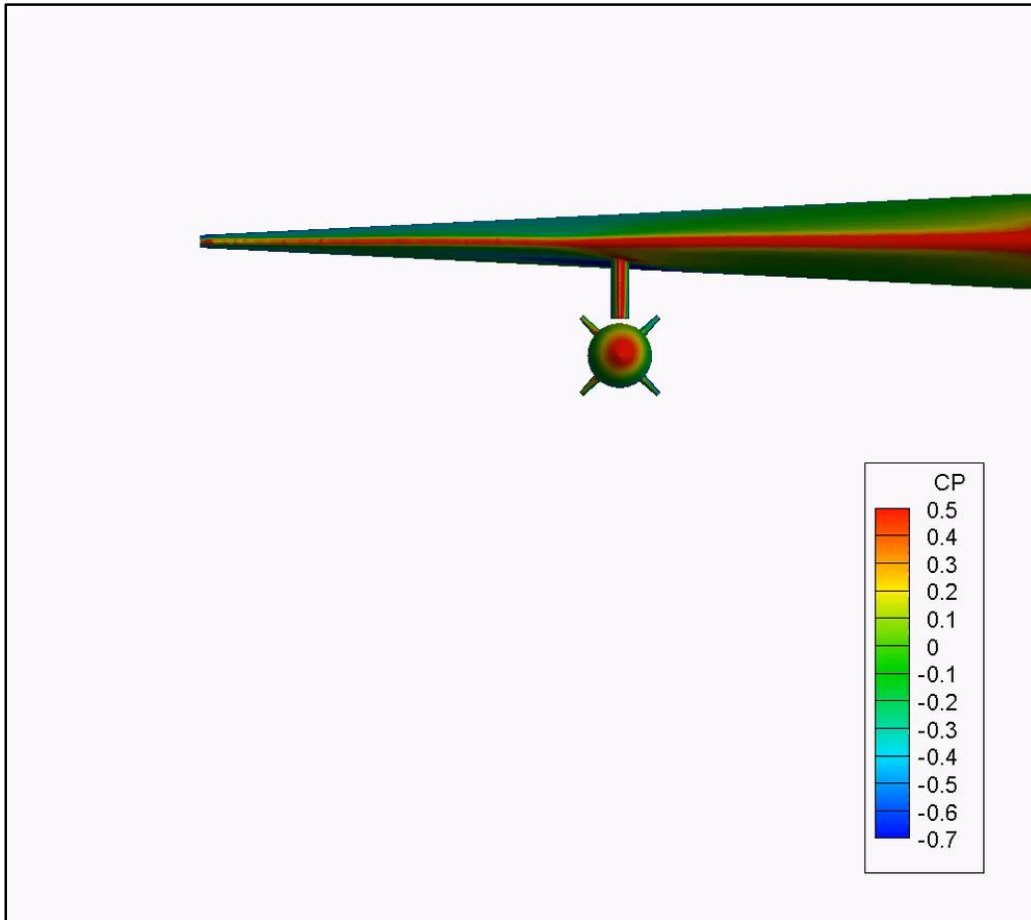
질점계 정면



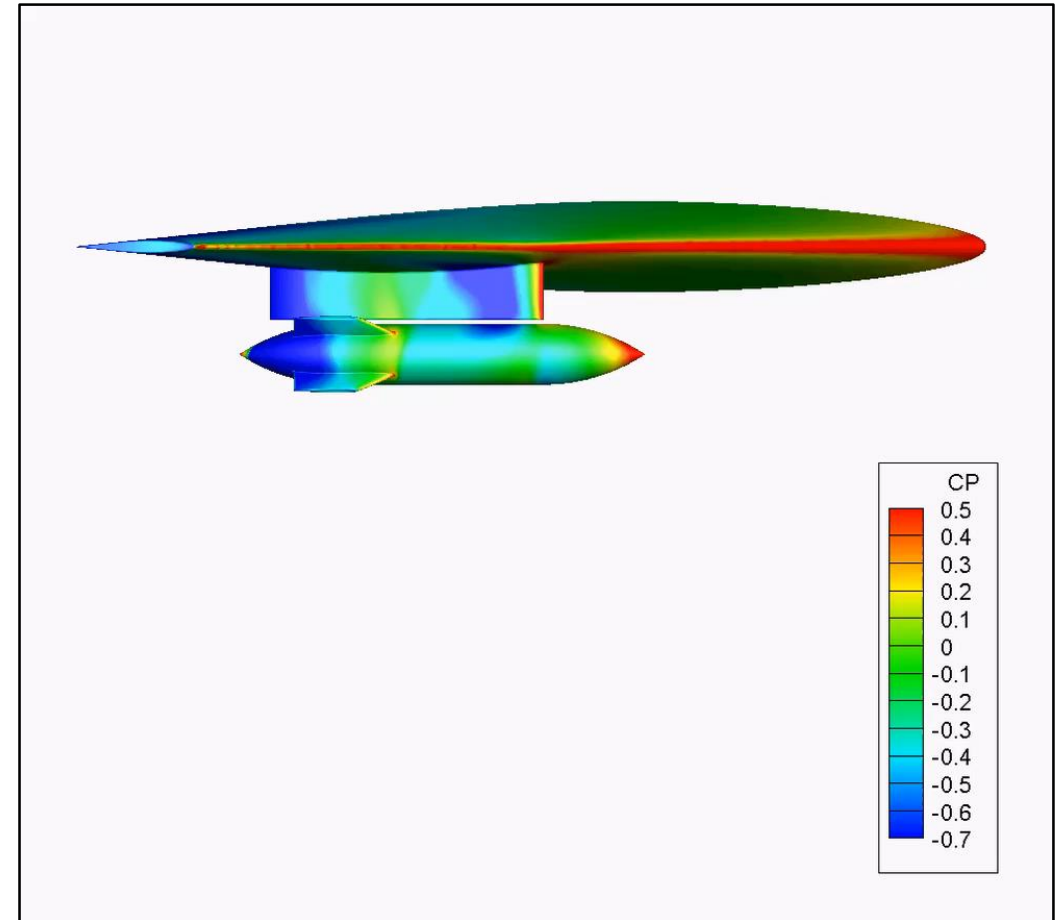
질점계 측면

질점계 해석자(FAMUS)(계속)

EGLIN Store Separation



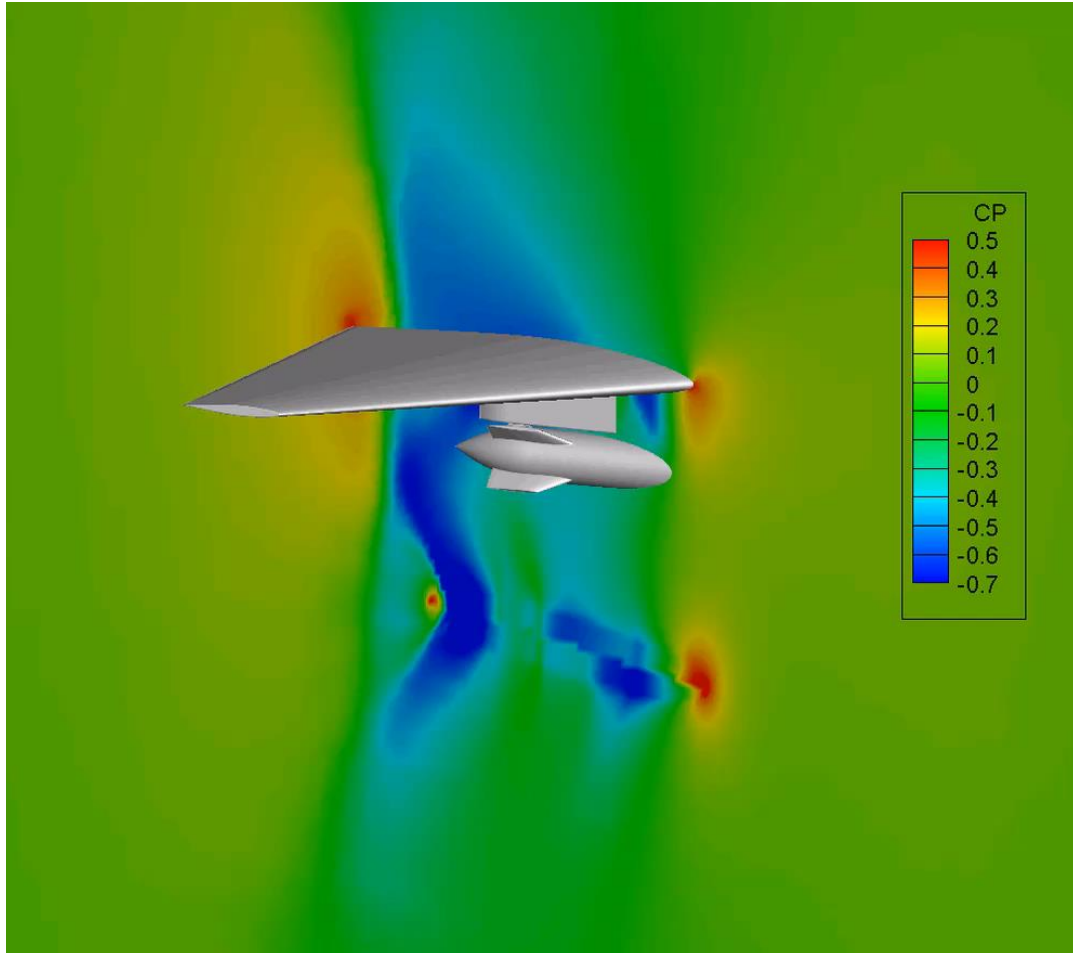
압력분포 정면



압력분포 측면

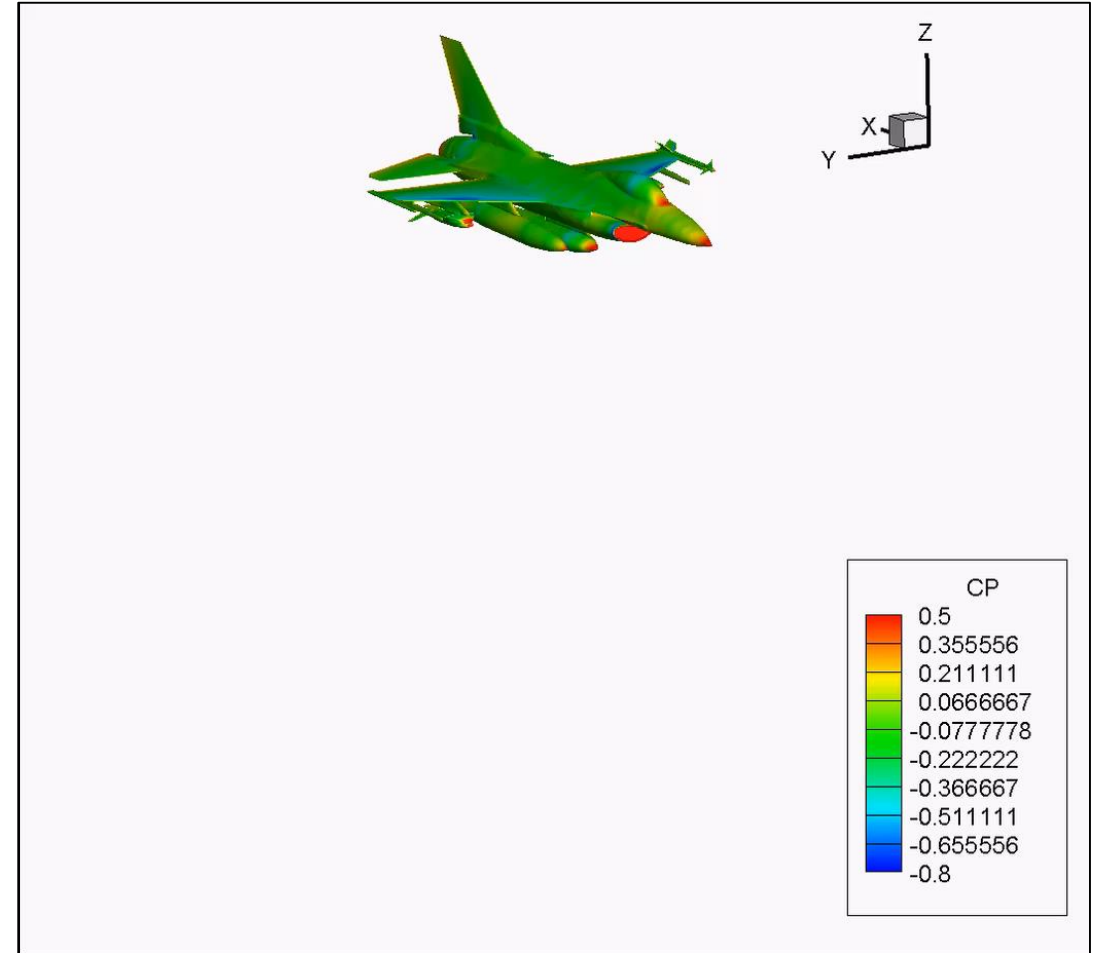
질점계 해석자(FAMUS)(계속)

EGLIN Store Separation



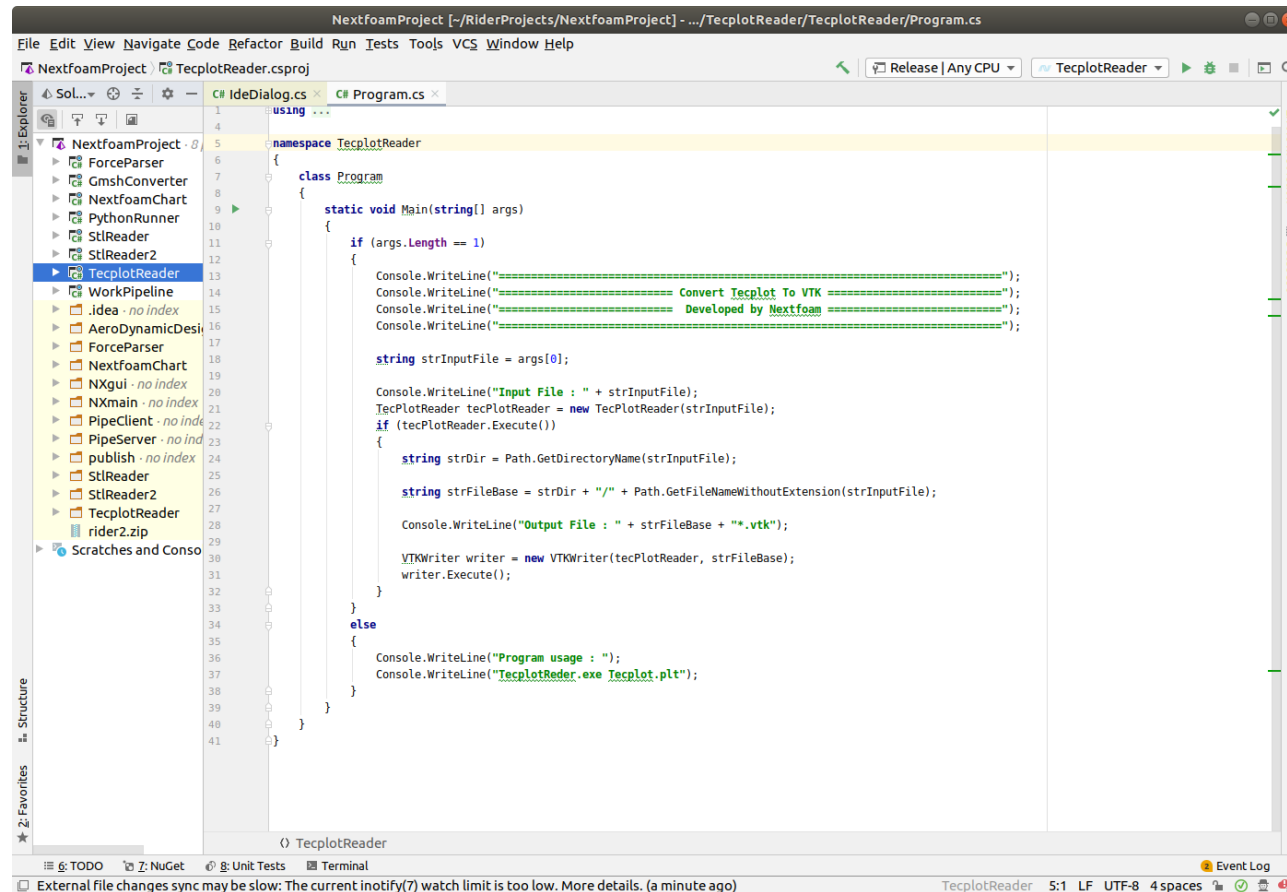
유동장 압력분포

F-16 Store Separation

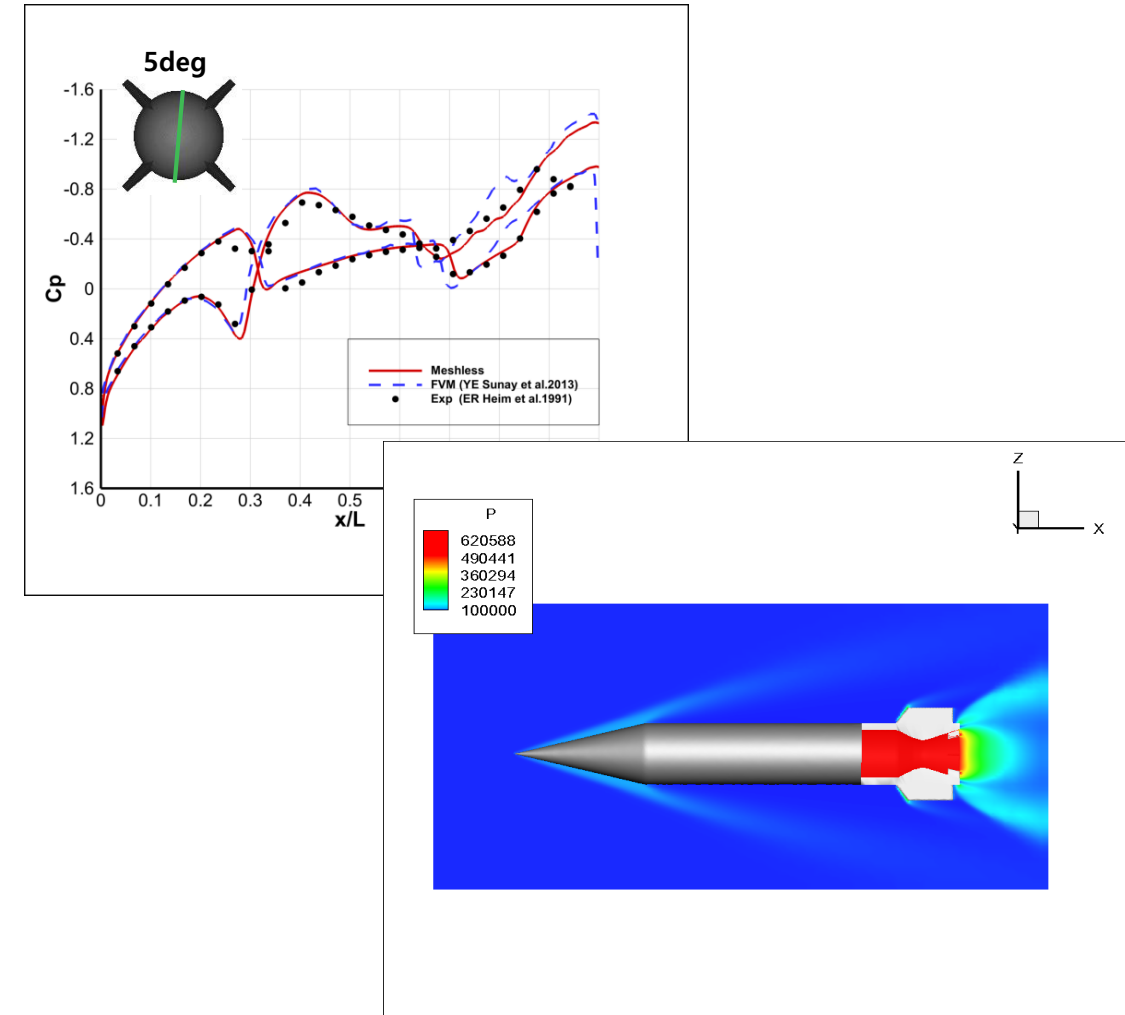


다수 무장 분리(압력분포)

후처리 자동화 및 D/B 구축(VTK & MongoDB)



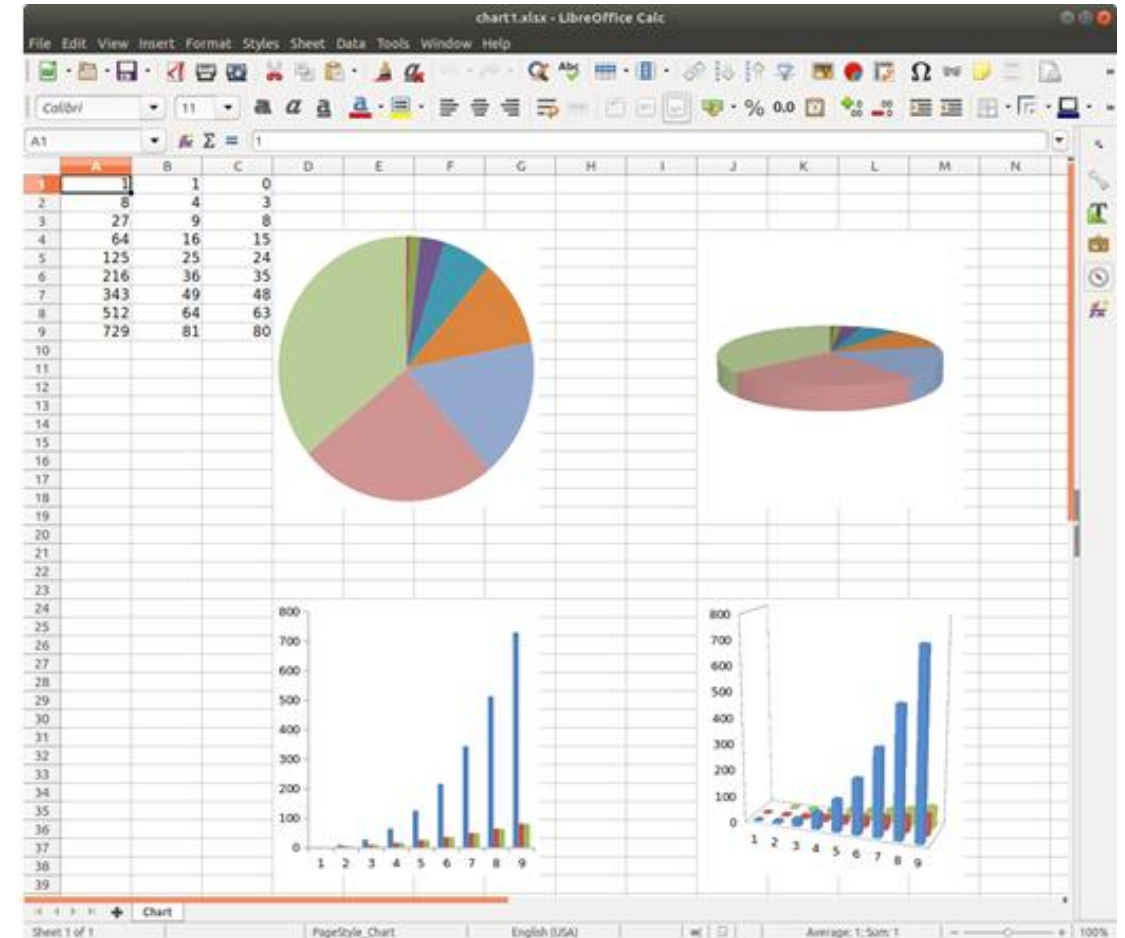
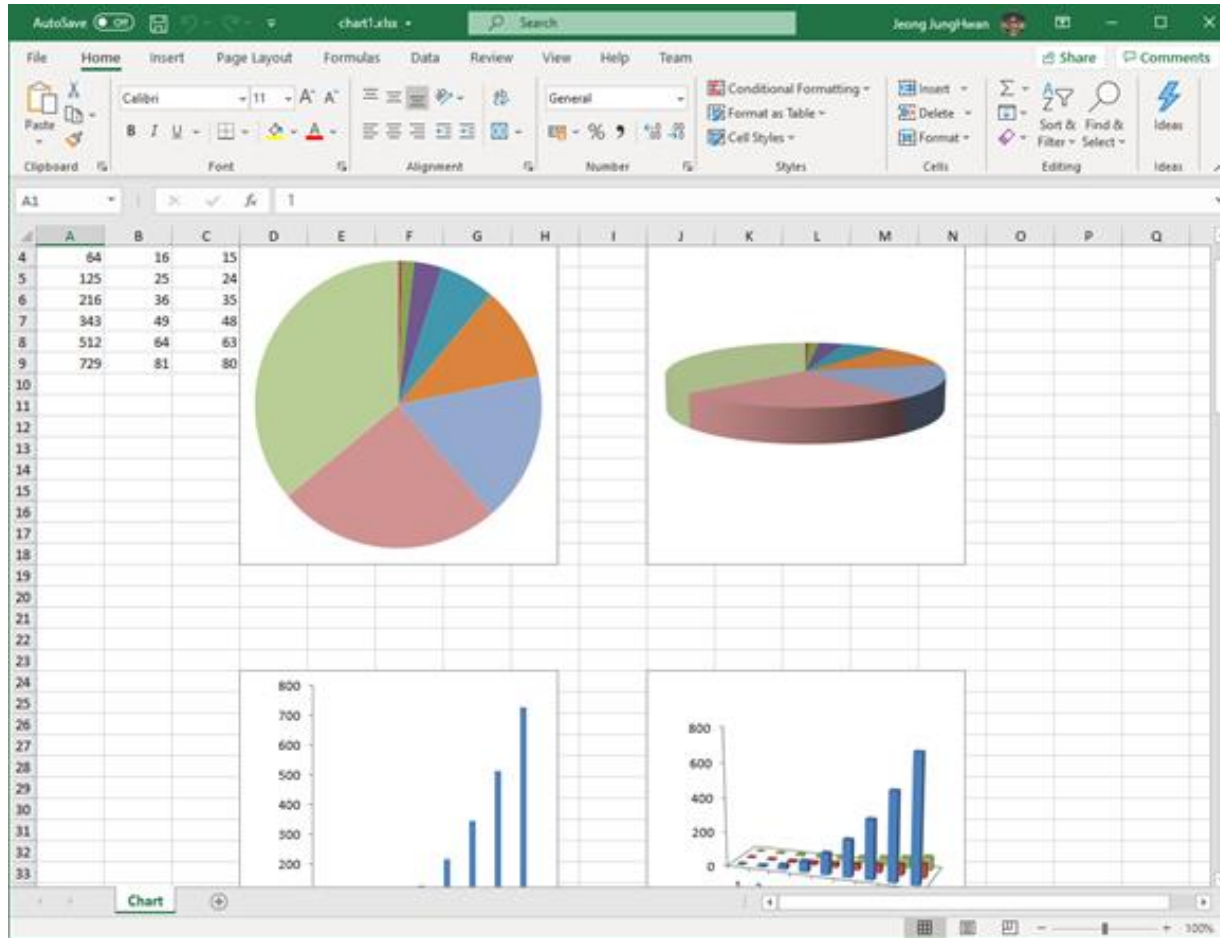
통합 포맷을 위해 개발된 VTK Converter



기 정의된 후처리 예시

핵심 요소기술 소개

후처리 자동화 및 D/B 구축(VTK & MongoDB)(계속)



객체 핸들링을 통한 보고서 자동 생성 예시

후처리 자동화 및 D/B 구축(VTK & MongoDB)(계속)

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:3000/public/adddata.html'. The page has a header with 'Upload Data' buttons. Below the header, there is a form with the following fields:

- Title:** A text input field containing 'Sample'.
- Contents:** A rich text editor with a toolbar and a text area containing 'This is for update.....'.
- Select upload directory:** A text input field with a 'Choose File' button and '106 files' displayed next to it.
- Writer:** A text input field containing 'jhjeong'.

At the bottom of the form are 'Add' and 'Cancel' buttons. A blue box with the text 'Javascript web uploader' is overlaid on the bottom right of the form.

The screenshot shows a 'File Upload' window with the following fields:

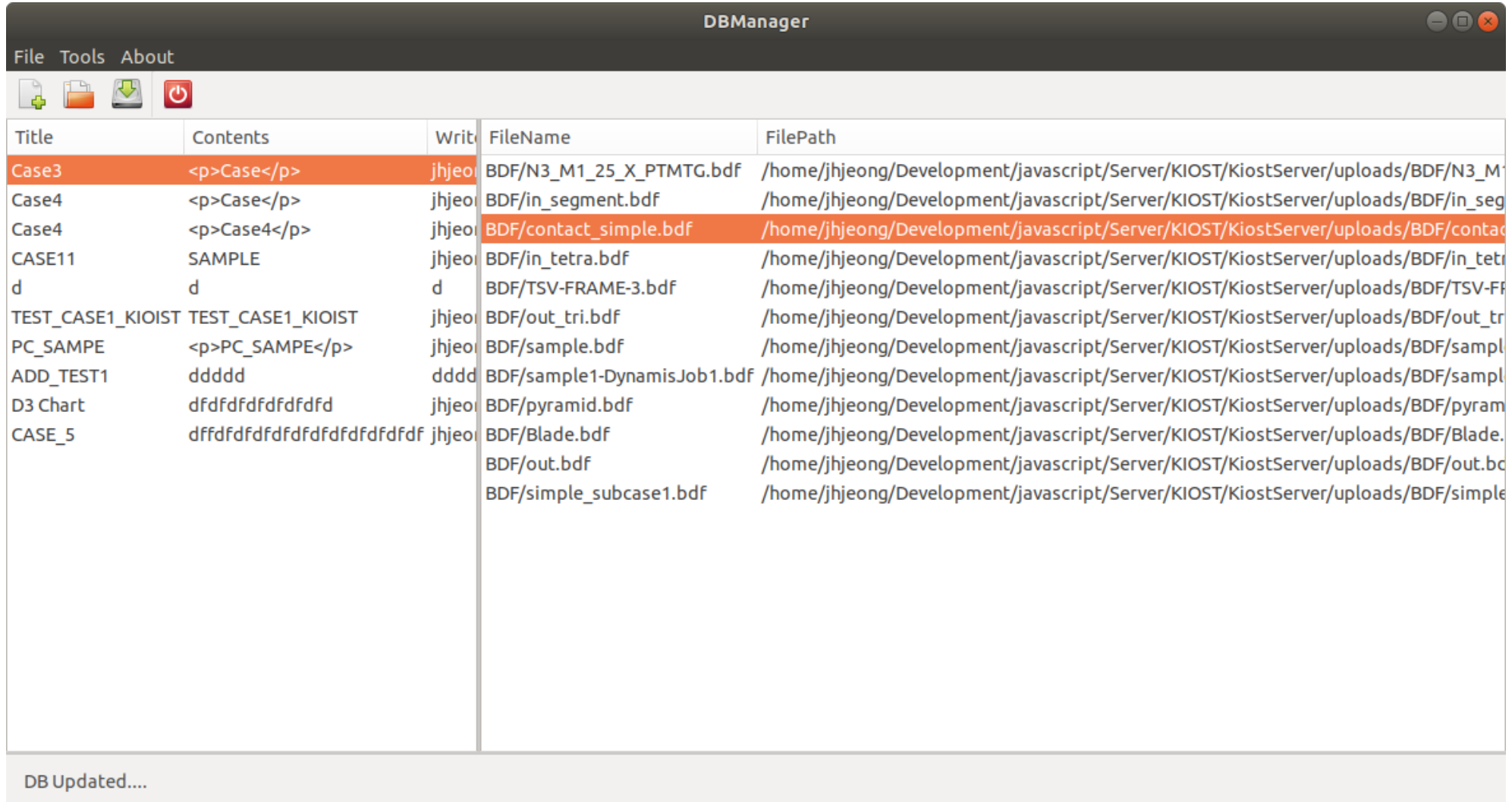
- Title:** A text input field containing 'Test Files'.
- Case:** A text input field containing 'Case 1'.
- Contents:** A text input field containing 'Test datas'.
- Writer:** A text input field containing 'jhjeong'.

Below the form is a 'Files' button. Underneath is a table listing uploaded files:

Title	Size	Path
Blade.bdf	56	/home/jhjeong/Data/BDF/Blade.b
Blade1.bdf	43	/home/jhjeong/Data/BDF/Blade1.
contact_simple.bdf	4998	/home/jhjeong/Data/BDF/contact
in_segment.bdf	2566801	/home/jhjeong/Data/BDF/in_segm
in_tetra.bdf	19482203	/home/jhjeong/Data/BDF/in_tetra
N3_M1_25_X_PTMTG.bdf	4173	/home/jhjeong/Data/BDF/N3_M1_
		/home/jhjeong/Data/BDF/out.bdf

At the bottom right is an 'Upload' button. A blue box with the text 'C# web uploader' is overlaid on the bottom left of the table.

후처리 자동화 및 D/B 구축(VTK & MongoDB)(계속)



The screenshot shows the DBManager application window. It has a menu bar with 'File', 'Tools', and 'About'. Below the menu bar is a toolbar with icons for adding, saving, and deleting files. The main area contains a table with the following columns: Title, Contents, Write, FileName, and FilePath. The table lists various database records, including Case3, Case4, CASE11, d, TEST_CASE1_KIOIST, PC_SAMPE, ADD_TEST1, D3 Chart, and CASE_5. The 'Case3' row is highlighted in orange. At the bottom of the window, there is a status bar that says 'DB Updated....'.

Title	Contents	Write	FileName	FilePath
Case3	<p>Case</p>	jhjeon	BDF/N3_M1_25_X_PTMTG.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/N3_M1_25_X_PTMTG.bdf
Case4	<p>Case</p>	jhjeon	BDF/in_segment.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/in_segment.bdf
Case4	<p>Case4</p>	jhjeon	BDF/contact_simple.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/contact_simple.bdf
CASE11	SAMPLE	jhjeon	BDF/in_tetra.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/in_tetra.bdf
d	d	d	BDF/TSV-FRAME-3.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/TSV-FRAME-3.bdf
TEST_CASE1_KIOIST	TEST_CASE1_KIOIST	jhjeon	BDF/out_tri.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/out_tri.bdf
PC_SAMPE	<p>PC_SAMPE</p>	jhjeon	BDF/sample.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/sample.bdf
ADD_TEST1	dddddd	dddddd	BDF/sample1-Dynamis.Job1.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/sample1-Dynamis.Job1.bdf
D3 Chart	dfdfdfdfdfdfdf	jhjeon	BDF/pyramid.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/pyramid.bdf
CASE_5	dfdfdfdfdfdfdfdfdfdf	jhjeon	BDF/Blade.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/Blade.bdf
			BDF/out.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/out.bdf
			BDF/simple_subcase1.bdf	/home/jhjeong/Development/javascript/Server/KIOST/KiostServer/uploads/BDF/simple_subcase1.bdf

DB Updated....

Database 조회 및 파일 다운로드를 위한 DB Manager


```

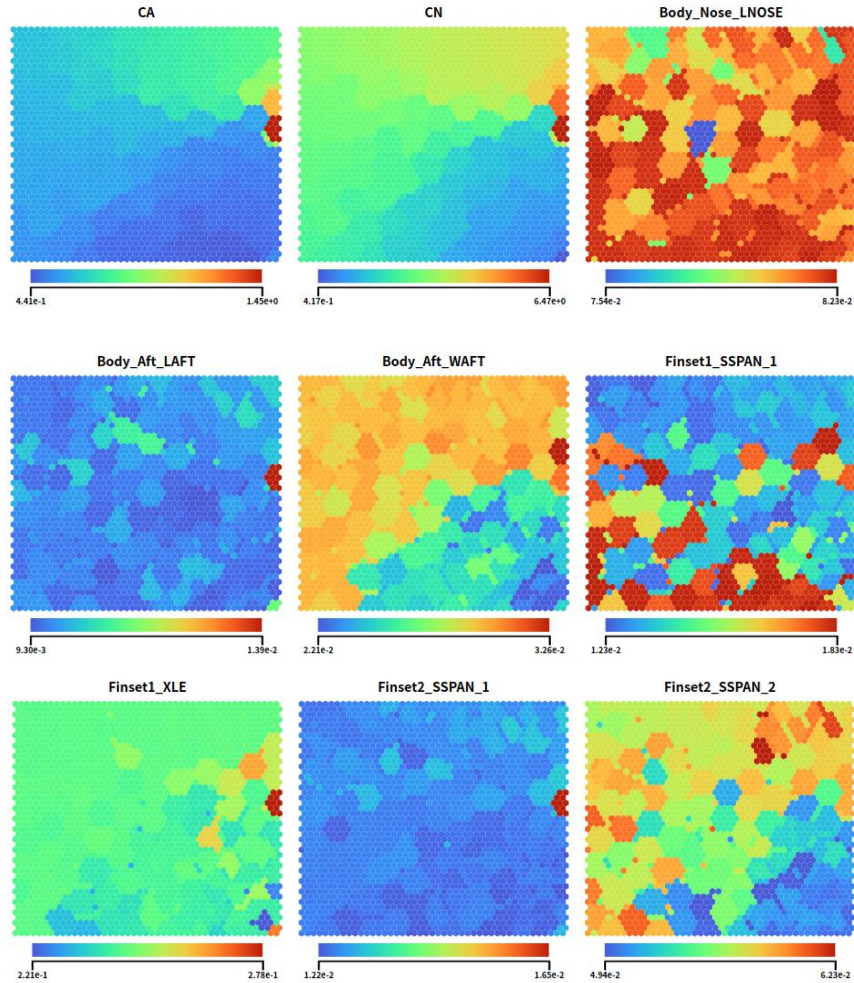
graph TD
    subgraph Inputs
        A[GUI/파일 입력] --> B[파라미터 데이터]
        C[CAD 데이터] --> D[형상 모델링 및 수정 모듈]
        E[GUI 격자(질점) 생성조건 입력] --> D
    end
    B --> D
    D --> F[공간격자 생성 모듈]
    D --> G[표면격자 생성 모듈]
    F --> H[격자계]
    G --> I[FAMUS 질점 생성 모듈]
    I --> J[FAMUS 질점계]
    H --> K[해석자 별 입력파일 생성 모듈]
    L[GUI 해석조건 입력] --> K
    K --> M[격자계 해석자]
    J --> N[FAMUS 질점계 해석자]
    M --> O[모니터링 모듈]
    N --> O
    O --> P[GUI 수렴 모니터링]
    O --> Q[격자계 결과파일]
    P --> R[질점계 결과파일]
    Q --> S[중립 포맷 변환]
    R --> S
    S --> T[후처리 Script]
    T --> U[GUI 후처리 결과 확인]
    T --> V[D/B 구축]
    V --> W[격자계 후처리기]
    V --> X[FAMUS 질점계 후처리기]
    W --> Y[최적화 모듈]
    X --> Y
    Y --> Z[POD 모듈]
    Z --> A
  
```

The flowchart illustrates the FAMUS system architecture, which is a process for generating and analyzing a computational mesh. The process begins with input from the GUI/file input, parameter data, and CAD data. These inputs feed into the shape modeling and modification module. This module then generates spatial and surface meshes. The spatial mesh is processed by the grid system, while the surface mesh is processed by the FAMUS point generation module. Both lead to the grid system and the FAMUS point system, respectively. These systems then feed into the analysis file generation module, which also receives input from the GUI analysis condition input. The analysis file generation module outputs to the grid system analyzer and the FAMUS point system analyzer. Both analyzers feed into the monitoring module, which also receives input from the GUI convergence monitoring. The monitoring module outputs to the grid system result file and the FAMUS point system result file. These result files then feed into the neutral format conversion module, which outputs to the post-processing script. The post-processing script feeds into the GUI post-processing result confirmation module and the D/B construction module. The D/B construction module outputs to the grid system post-processor and the FAMUS point system post-processor. Both post-processors feed into the optimization module, which then feeds into the POD module. The POD module feeds back into the GUI/file input, completing the loop.

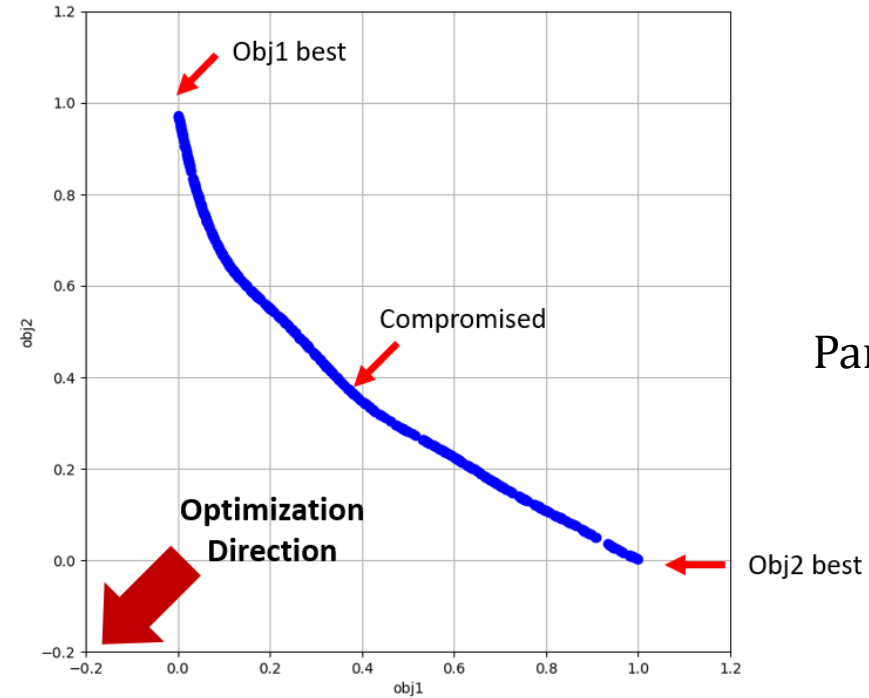
- 대체 모델(Surrogate model) : Kriging method
- 최적화 : 다목적 유전 알고리즘(MOGA)
- 데이터 마이닝 : SOM (Self Organizing Map) & ANOVA (Analysis of Variance)

- 유동장 재구성
- 최적화 및 역설계
- 데이터 합성

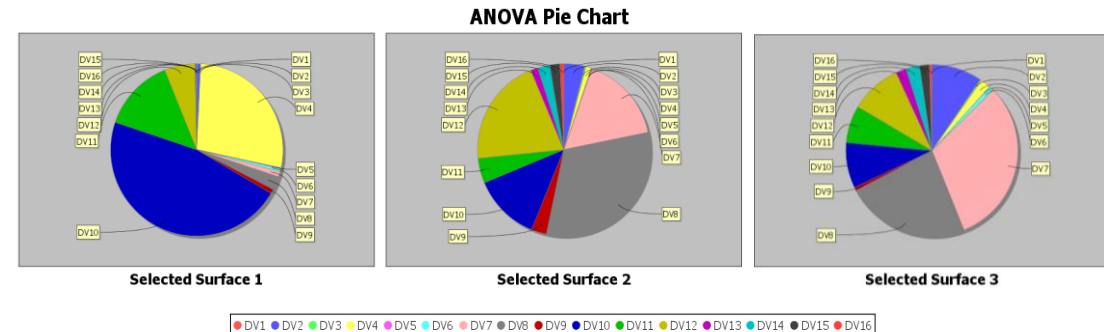
최적화 모듈



SOM 결과

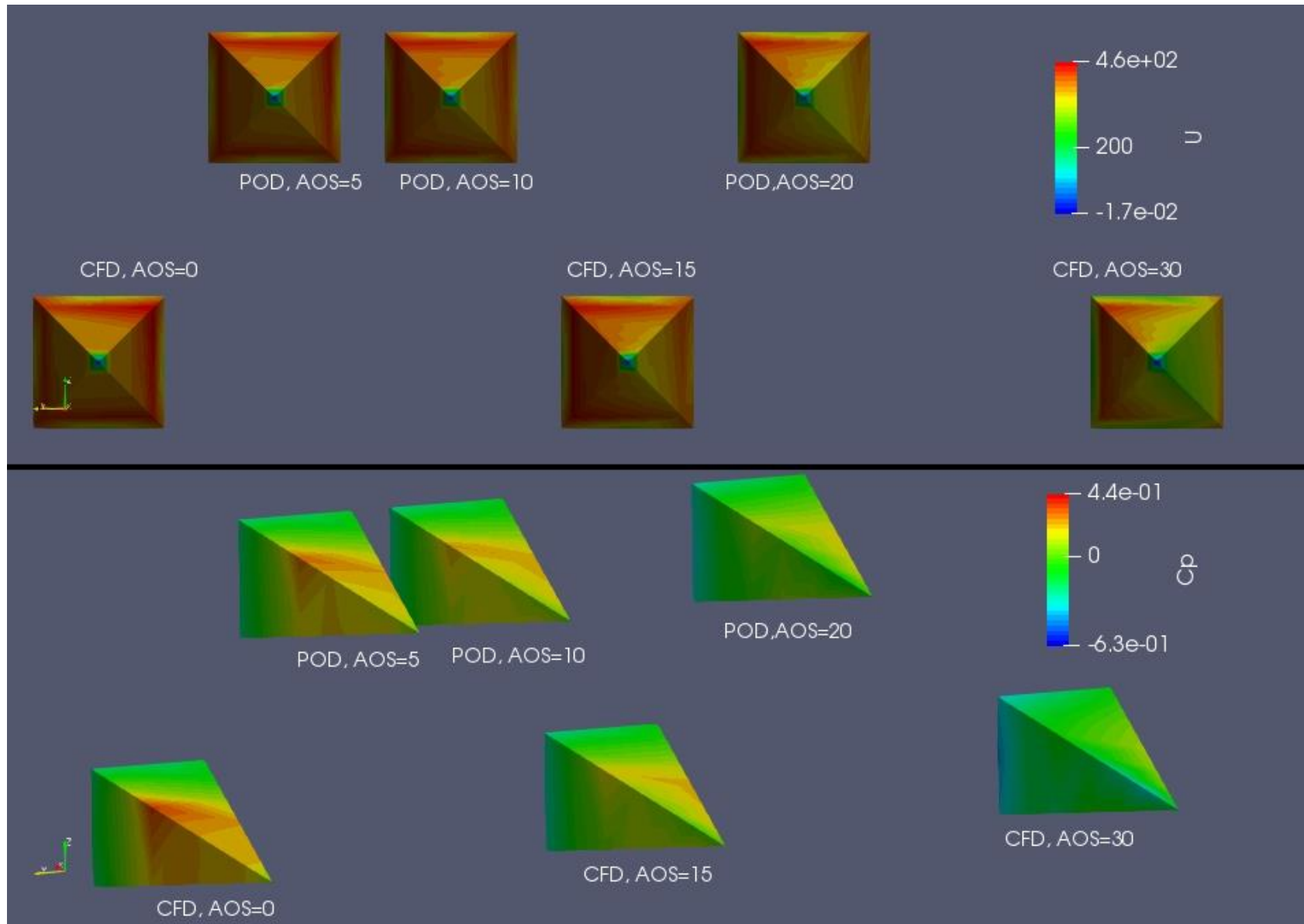


Pareto Solution 결과



ANOVA 결과

POD 모듈



Thanks for your attention.

NEXTfoam CO., LTD.

Convergence Technology Team

Sang Chul, Lee

E-mail: sclee@nextfoam.co.kr

Office: +82-70-8796-3016



www.nextfoam.co.kr