

연차보고서	<input checked="" type="checkbox"/> 일반형		기술료	
	<input type="checkbox"/> 통합형(총괄)	<input type="checkbox"/> 통합형(세부)		정수[] 비정수[]
중앙행정기관명	산업통상자원부	사업명	세부사업명	수소충전 인프라 안전관리 핵심기술 개발
전문기관명	한국에너지기술평가원		내역사업명	에너지기술개발사업
연구개발과제명(과제번호)	수소충전소 화재폭발시 피해저감을 위한 방호벽 설계기술 및 안전기준 개발 (No. 20215810100020, RS-2021-KP002433)			

2024년 연차보고서 부록 :

수소증기운 폭발사고 해석코드(radXiFoam) 사용자 매뉴얼 v2.0

2024. 11. 13

강형석, 최근상

한국원자력연구원

목 차

1. 서 론	(3)
1-1. 개정 이력	(3)
1-2. OpenFOAM 소개 및 설치	(3)
1-3. radXiFoam v1.0 개요 및 설치	(3)
2. radXiFoam 해석용 격자모델 생성방법	(4)
2-1. SRI Test 4-02 실험장치 격자모델	(4)
2-2. blockMesh 프로그램 사용법	(6)
3. SRI Test 4-02 해석용 입력자료	(10)
3-1. 물성치(constant 디렉토리) 입력	(10)
3-2. 초기 및 경계조건(0 디렉토리) 입력	(20)
3-3. 계산제어(system 디렉토리) 입력	(31)
4. radXiFoam 실행방법	(42)
5. 해석결과 분석방법	(47)
6. 참고문헌	(57)
7. 부록	(58)
7-1. radXiFoam 계산 순서도	(58)
7-2. alphaBetaCalc.py	(59)
7-3. overPressurePlot.py	(60)
7-4. radXiFoam v2.0 설치방법	(61)
7-5. HSL 수소충전소 모형 실험장치 CAD 방식 격자모델링 기법	(62)
7-6. OF-Interface 프로그램 사용방법	(83)

1. 서론

1-1. 개정 이력

본 사용자 매뉴얼은 Version 2.0으로 Version 1.0[1]에 아래의 세 가지를 부록으로 추가 하였다.

- radXiFoam v2.0 설치 방법.
- 영국 Health & Safety Laboratory(HSL)이 수행한 수소충전소 모의 장치(Mock-Up Facility)에서의 수소 증기운 폭발실험[2]을 검증해석 시 필요한 격자모델[3]을 SALOME[4] 소프트웨어로 CAD 방식으로 생성하는 기법.
- 영국 HSL의 수소충전소 모의장치에서의 수소 증기운 폭발실험을 검증해석 시 필요한 입력 파일들을 OF-Interface 프로그램을 사용하여 자동으로 생성하는 기법.

1-2. OpenFOAM 소개 및 설치

본 사용자 매뉴얼은 OpenFOAM-v2112[5] 기반으로 개발한 수소증기운 폭발사고 해석용 솔버(Solver)인 radXiFoam v1.0[1]의 사용법을 기술하였다. OpenFOAM은 “Open-source Field Operation And Manipulation” 의 약자이며, C++ 프로그래밍 언어로 구현된 Computational Fluid Dynamics(CFD) 소프트웨어 패키지이다. radXiFoam 설치하기 위해서는 Linux Operating System(OS)에 OpenFOAM-v2112를 먼저 설치한 후 사용자 환경설정을 구성해야 한다. 본 매뉴얼에서 언급한 리눅스 OS는 Ubuntu 20.04 LTS[6] 이며, OpenFOAM-v2112를 설치하는 과정은 아래와 같다.

- OpenFOAM-v2112.tgz와 ThirdParty-v2112를 아래 위치에서 다운로드한다.
 - <https://www.openfoam.com/download/release-history>
- OpenFOAM-v2112의 상세 설치과정과 방법은 아래 위치의 Build.me을 참고한다.
 - <https://develop.openfoam.com/Development/openfoam/-/blob/master/doc>
- OpenFOAM-v2112를 구동하기 위해서는 터미널 창에서 아래의 명령어를 실행해야 한다.
 - `source /오픈폼 설치 디렉토리/OpenFOAM-v2112/etc/bashrc`
 - 위 명령어는 /홈디렉토리/etc/.bashrc 파일에 alias 이름을 간단하게 만들어서 사용하는 것이 편리하다.

1-3. radXiFoam v1.0 개요 및 설치

radXiFoam v1.0의 소스와 예제는 github에 업로드 및 공개되어 있다. 다운로드 및 설치하는 과정은 아래와 같다.

- radXiFoam 소스파일(SRC.zip)을 아래 위치에서 다운로드한다.
 - <https://github.com/ksm0226/radXiFoam/SRC.zip>
- SRC.zip의 압축을 풀고, 해당 디렉토리의 터미널 창에서 그림 1과 같이 컴파일한다.
 - OpenFOAM-v2112 구동용 환경설정 명령어(of2112) 실행
 - Allwclean.sh 실행
 - Allwmake.sh 실행

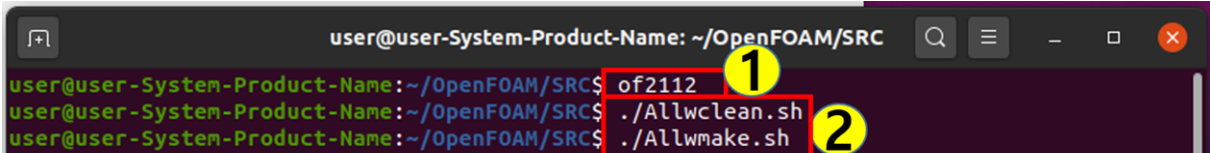


그림 1. radXiFoam 컴파일을 위한 명령어 입력

- 만약, 컴파일 단계에서 permission 관련 에러가 발생하면 아래와 같이 해결한다.
 - `chmod +x *.sh`
- 컴파일 성공여부 확인 방법
 - 컴파일이 완료되면 그림 2와 같이 radXiFoam 실행파일의 위치가 나타난다.

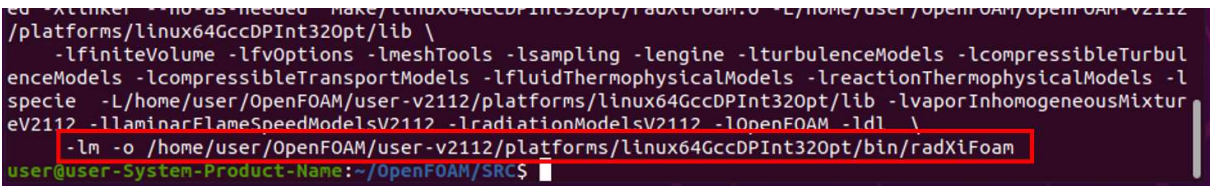
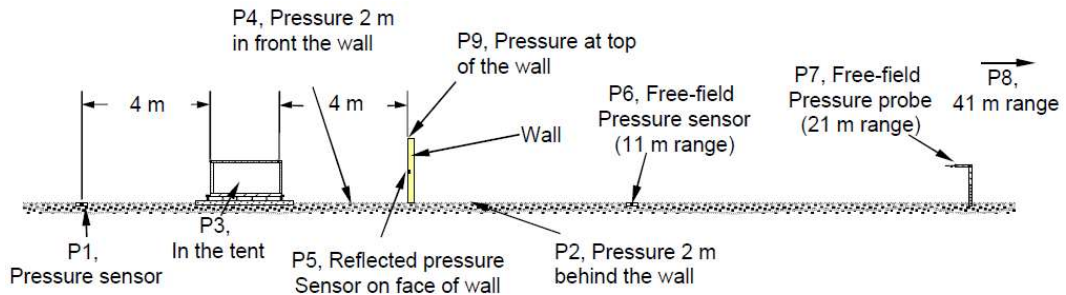


그림 2. radXiFoam 컴파일 완료 메시지 화면

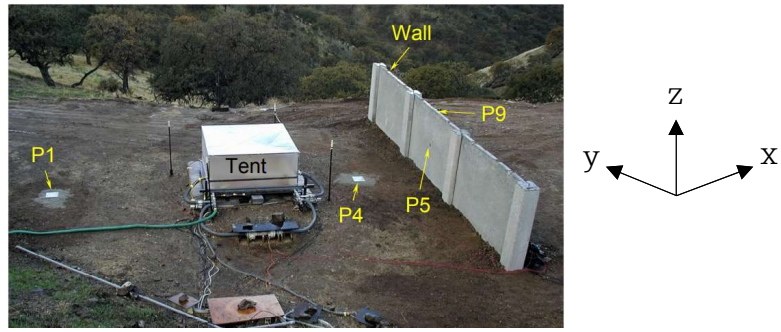
2. radXiFoam 해석용 격자모델 생성방법

2-1. SRI Test 4-02 실험장치 모의 격자모델

radXiFoam 코드를 실행시키기 위해서는 다차원 격자모델이 필요하다. 본 매뉴얼에서는 미국 SRI(Stanford Research Institute)에서 수행한 수소증기운 폭발실험 4-02[7]을 해석하기 위해서 사용되었던 격자모델[8,9]을 이용하여 설명하고자 한다. SRI Test 4-02 실험장치의 전체적인 개략도는 그림 3에 나타나 있고, 아를 모의하여 생성된 격자모델과 적용된 경계조건은 그림 4와 같다. 격자모델은 수소-공기 혼합기체가 위치한 Tent 바닥면의 중앙에 있는 전기스파크 지점을 기준으로 1/2 대칭조건이 적용되어 생성되었다.



(a) 실험장치 개략도



(b) 방호벽 모양 및 위치

그림 3. SRI 실험장치 개략도 및 방호벽 모양 [7]

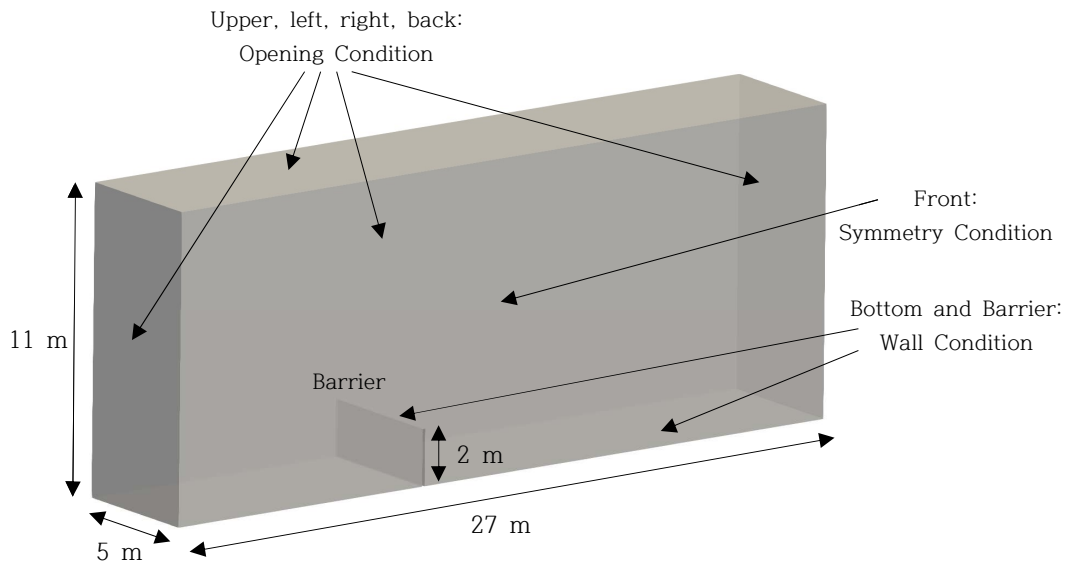


그림 4. SRI Test 4-02 해석용 격자모델 [8,9]

2-2. blockMesh 프로그램 사용법

본 절에서는 OpenFOAM-v2112에 내장된 격자생성 프로그램인 blockMesh[5]의 사용법을 SRI 4-02 실험장치 모의용 격자모델을 예제로 사용하여 설명하고자 한다. blockMesh 프로그램을 제어하는 입력파일은 “case\system\blockMeshDict” 이며, 여기서 case는 해석을 수행하기 위하여 새로 생성한 디렉토리의 이름이다.

2-2-1. 해석영역 지정용 블록좌표 생성방법

해석대상 실험장치의 영역을 격자모델로 만들기 위해서 방호벽(Wall)과 Tent 공간의 위치를 고려해서 그림 5와 같이 7개의 직육면체 블록(block)으로 먼저 나누었다. 그다음, 7개 블록의 꼭지점에 해당하는 지점(point)의 좌표를 그림 10과 같이 정의한 후, blockMeshDict 파일에 표 1과 같이 입력하였다.

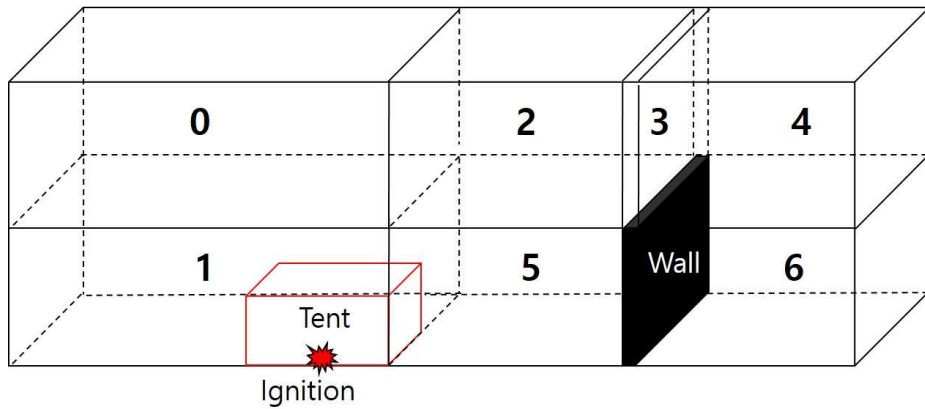


그림 5. 격자모델 생성을 위한 블록 구성도

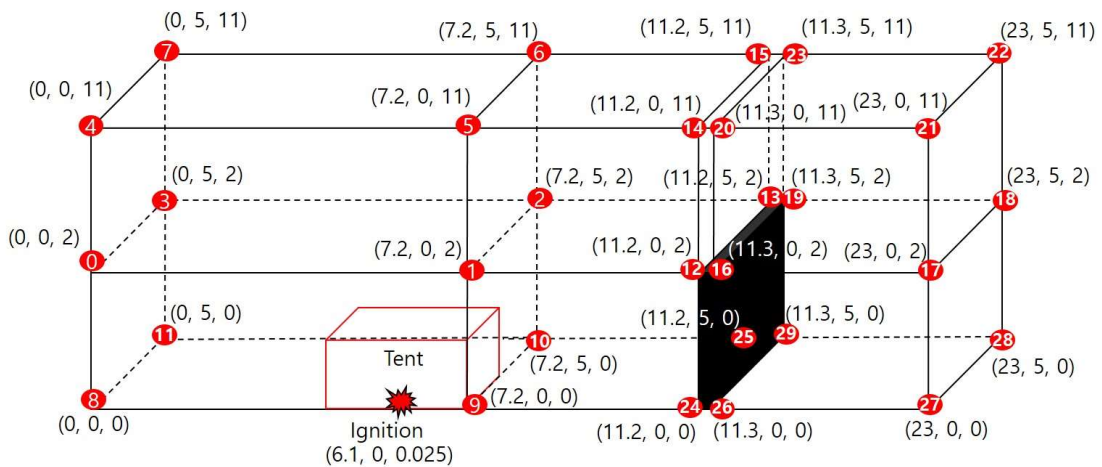


그림 6. 격자모델 생성용 블록 꼭지점 좌표 지정도

표 1. 격자모델 생성용 블록 꼭지점 좌표 입력

```
19 vertices
20 (
21 //Block 0, Point 0, 1, 2, 3, 4, 5, 6, 7
22 (0 0 2)
23 (7.2 0 2)
24 (7.2 5 2)
25 (0 5 2)
26 (0 0 11)
27 (7.2 0 11)
28 (7.2 5 11)
29 (0 5 11)
30
31 //Block 1 Point 8, 9, 10, 11
32 (0 0 0)
33 (7.2 0 0)
34 (7.2 5 0)
35 (0 5 0)
36
37 //Block 2, Point 12, 13, 14, 15
38 (11.2 0 2)
39 (11.2 5 2)
40 (11.2 0 11)
41 (11.2 5 11)
42
43 //Block 3
44
45 //Block 4 Point 16, 17, 18, 19, 20, 21, 22, 23
46 (11.3 0 2)
47 (23 0 2)
48 (23 5 2)
49 (11.3 5 2)
50 (11.3 0 11)
51 (23 0 11)
52 (23 5 11)
53 (11.3 5 11)
54
55 //Block 5 Point 24, 25
56 (11.2 0 0)
57 (11.2 5 0)
58
59 //Block 6 Point 26, 27, 28, 29
60 (11.3 0 0)
61 (23 0 0)
62 (23 5 0)
63 (11.3 5 0)
64 );
```

2-2-2. 블록 및 격자 셀 생성방법

blockMeshDict 파일에 입력된 총 30개의 좌표 중에서 8개씩 묶어서 직육면체 블록(hexahedral block)을 정의할 수 있다. 그 예는 표 2의 ①에 나타나 있다. 여기서 “hex (0 1 2 3 4 5 6 7)”의 숫자는 꼭지점 좌표계의 번호를 의미하는데, blockMeshDict 파일에 정의된 순서대로 “0”부터 번호가 차례대로 지정된다. 표 2에서 방호벽 내부의 영역은 고려하지 않았다. 블록의 x, y, z 방향으로 격자 셀(Cell) 개수를 표 2의 ②와 같이 지정할 수 있다. 표 2의 ③은 전체 격자 셀 사이의 크기 비율을 조정하는 인자이며, “1”은 각 축 방향으로 분포한 전체 셀 중에서 첫 번째 셀 길이와 마지막 셀 길이의 비율을 정의하는 인자이다. 즉 “1”은 셀 길이가 균일하고, “1”보다 큰 숫자나 작은 숫자는 셀 길이가 그 숫자의 비율만큼 증가하거나 감소한다. 이와 관련된 상세한 설명은 OpenFOAM-v2112의 사용자 매뉴얼[2]에 있다.

표 2. 블록 생성용 좌표번호, 셀 개수 및 셀 길이 비율 지정 입력

```

70 blocks
71 (
72   hex (0 1 2 3 4 5 6 7) (72 50 90) simpleGrading (1 1 1)
73   hex (8 9 10 11 0 1 2 3) (72 50 20) simpleGrading (1 1 1)
74   hex (1 12 13 2 5 14 15 6) (40 50 90) simpleGrading (1 1 1)
75   hex (12 16 19 13 14 20 23 15) (1 50 90) simpleGrading (1 1 1)
76   hex (16 17 18 19 20 21 22 23) (117 50 90) simpleGrading (1 1 1)
77   hex (9 24 25 10 1 12 13 2) (40 50 20) simpleGrading (1 1 1)
78   hex (26 27 28 29 16 17 18 19) (117 50 20) simpleGrading (1 1 1)
79
80
81);

```

2-2-3. 블록의 경계면 생성방법

직육면체 블록이 생성되면, 그 블록의 한 면(Face)이 격자모델의 외부 경계나, 내부의 벽면(Wall Surface)과 접촉하고 있는 경우에는, 표 3과 같이 경계면의 형식(Type)을 지정해주어야 한다. 그러기 위해서는 먼저 표 3의 ①과 같이 사용자가 경계면의 이름을 지정하고, ②와 같이 blockMesh 프로그램에서 지원하는 경계면 Type(wall, symmetry, patch 등)을 지정한 후, ③과 같이 그 경계면에 해당하는 4개 꼭지점의 좌표번호를 입력해야 한다. 꼭지점의 좌표번호는 표 1과 같이 입력된 순서대로의 번호이다. 더 상세한 설명은 OpenFOAM-v2112의 사용자 매뉴얼[2]에 있다.

표 3. 생성된 블록의 경계면 지정 입력

```

89 boundary
90 (
91     walls //barrier + ground
92     {
93         type wall;
94         faces
95         (
96             (8 9 10 11)
97             (9 24 25 10)
98             (24 12 13 25)
99             (12 16 19 13)
100            (16 19 29 26)
101            (26 27 28 29)
102        );
103    }
104
105    symmetry
106    {
107        type symmetryPlane;
108        faces
109        (
110            (0 1 4 5)
111            (8 9 1 0)
112            (1 12 14 5)
113            (9 24 12 1)
114            (12 16 20 14)
115            (16 17 21 20)
116            (26 27 17 16)
117
118        );
119    }
120 }
121
122 open
123 {
124     type patch;
125     faces
126     (
127         (0 3 7 4)
128         (8 11 3 0)
129         (4 5 6 7)
130         (5 14 15 6)
131         (14 20 23 15)
132         (20 21 22 23)
133         (21 22 18 17)
134         (17 18 28 27)
135         (3 2 7 6)
136         (11 10 2 3)
137         (6 15 13 2)
138         (13 19 23 15)
139         (19 18 22 23)
140         (10 25 13 2)
141         (29 28 18 19)
142     );
143 }
144 );

```

2-2-4. blockMesh 프로그램 실행방법

blockMesh 프로그램의 입력파일인 “blockMeshDict” 에 대한 작성을 완료한 후, 터미널 창에서 명령어 “blockMesh” 를 입력하여 실행하면, 그림 7과 같은 정보가 출력되며 격자모델이 생성된다. 그림 7에서 ①은 생성된 블록의 개수와 블록 내 격자의 최소/최대 크기, ②는 생성된 전체 격자모델의 기하학적 영역과 셀의 개수, ③은 격자모델 경계면의 이름이다.

```

Creating polyMesh from blockMesh
Creating patches
Creating cells
Creating points 1th scale (1 1 1)
Block 0 cell size :
i : 0.1 .. 0.1
j : 0.1 .. 0.1
k : 0.1 .. 0.1
Block 1 cell size :
i : 0.1 .. 0.1
j : 0.1 .. 0.1
k : 0.1 .. 0.1
Block 2 cell size :
i : 0.1 .. 0.1
j : 0.1 .. 0.1
k : 0.1 .. 0.1
Block 3 cell size :
i : 0.1 .. 0.1
j : 0.1 .. 0.1
k : 0.1 .. 0.1
Block 4 cell size :
i : 0.1 .. 0.1
j : 0.1 .. 0.1
k : 0.1 .. 0.1
Block 5 cell size :
i : 0.1 .. 0.1
j : 0.1 .. 0.1
k : 0.1 .. 0.1
Block 6 cell size :
i : 0.1 .. 0.1
j : 0.1 .. 0.1
k : 0.1 .. 0.1
Writing polyMesh with 0 cellZones
-----
Mesh Information
boundingBox: (0 0 0) (23 5 11)
nPoints: 1307691
nCells: 1264000
nFaces: 3835280
nInternalFaces: 3748720
Patches
patch 0 (start: 3748720 size: 13500) name: walls
patch 1 (start: 3762220 size: 25280) name: symmetry
patch 2 (start: 3787500 size: 47780) name: open
End

```

그림 7. 격자모델 생성과정에서 출력되는 정보

2-2-5. 격자모델 정보 확인방법

격자모델이 생성되면, 그 정보는 “case/constant/polyMesh” 디렉토리 안에 저장된다. 이 정보들은 터미널 창에서 “checkMesh” 명령어를 입력하면, 그림 8과 같이 생성된 ①격자모델을 구성하는 셀 종류와 개수, ②격자모델의 셀과 면 등의 topology 상태, ③격자모델 기하학적인 상태의 정보가 출력된다.

```
Create time
Create mesh for time = 0
Time = 0
Mesh stats
  points:      1307691
  faces:      3835280
  internal faces: 3748720
  cells:      1264000
  faces per cell: 6
  boundary patches: 3
  point zones: 0
  face zones: 0
  cell zones: 0
Overall number of cells of each type:
  hexahedra:  1264000
  prisms:     0
  wedges:     0
  pyramids:   0
  tet wedges: 0
  tetrahedra: 0
  polyhedra:  0
Checking topology...
  Boundary definition OK.
  Cell to face addressing OK.
  Point usage OK.
  Upper triangular ordering OK.
  Face vertices OK.
  Number of regions: 1 (OK).
Checking patch topology for multiply connected surfaces...
  Patch      Faces   Points  Surface topology
  walls      13500  13821  ok (non-closed singly connected)
  symmetry   25280  25641  ok (non-closed singly connected)
  open       47780  48191  ok (non-closed singly connected)
Checking faceZone topology for multiply connected surfaces...
  No faceZones found.
Checking basic cellZone addressing...
  No cellZones found.
Checking geometry...
  Overall domain bounding box (0 0 0) (23 5 11)
  Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
  Mesh has 3 solution (non-empty) directions (1 1 1)
  Boundary openness (-8.40688e-17 -5.3065e-16 -1.07589e-15) OK.
  Max cell openness = 3.46945e-16 OK.
  Max aspect ratio = 1 OK.
  Minimum face area = 0.01. Maximum face area = 0.01. Face area magnitudes OK.
  Min volume = 0.001. Max volume = 0.001. Total volume = 1264. Cell volumes OK.
  Mesh non-orthogonality Max: 0 average: 0
  Non-orthogonality check OK.
  Face pyramids OK.
  Max skewness = 3.5094e-13 OK.
  Coupled point location match (average 0) OK.
Mesh OK.
End
```

그림 8. 생성 후 저장된 격자모델 정보 출력화면

3. SRI Test 4-02 입력자료

radXiFoam 코드는 해석을 위해 입력해야 하는 변수들을 문자 편집기를 사용해서 작성해야 하며, 그 입력변수들의 특성에 따라서 총 3개의 디렉토리로 그룹화되어 있다. 추후에는 문자 편집기 대신 GUI(Graphical User Interface)를 활용하는 방식이 개발될 계획이다.

3-1. 물성치(constant 디렉토리) 입력

“constant” 디렉토리에는 radXiFoam 해석에 사용되는 입력인자 중에서 물성치(properties)와 격자모델 정보 등이 그림 9와 같이 모여있는 디렉토리이다.

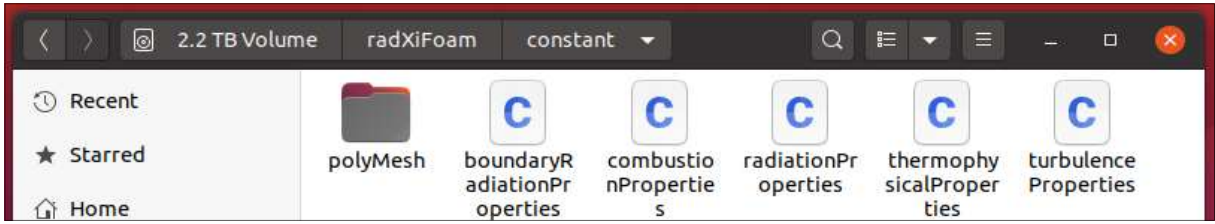


그림 9. 물성치(constant 디렉토리) 내용 화면

3-1-1. polyMesh

본 디렉토리는 “blockMesh” 명령어가 실행되어 생성된 격자모델의 정보가 저장되는 곳이다.

3-1-2. thermophysicalProperties

본 파일은 해석에 사용되는 유체의 열역학적 정보를 표 4와 같이 입력하는 파일이다. 그 파일 내의 세부 입력인자에 대한 설명은 다음과 같다.

표 4. thermophysicalProperties

```

1 /*----- C++ -----*/
2
3 //
4 // Field Operation and Manipulation
5 //
6 //
7 //
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "constant";
14     object       thermophysicalProperties;
15 }
16 // ***** //
17
18 thermoType 1
19 {
20     type        heheuPsiThermo;
21     mixture     vaporInhomogeneousMixture;
22     transport   sutherland;
23     thermo      janaf;
24     equationOfState perfectGas;
25     specie      specie;
26     energy      absoluteEnthalpy;
27 }
28
29 stoichiometricAirFuelMassRatio 2
30 stoichiometricAirFuelMassRatio [0 0 0 0 0 0] 7.9364;
31
32 fuel
33 { 3
34     specie
35     {
36         molWeight      2.01594;
37     }
38     thermodynamics
39     {
40         Tlow           200;
41         Thigh          3500;
42         Tcommon        1000;
43         highCpCoeffs   ( 3.3372792 -4.94024731e-05 4.99456778e-07 -1.79566394e-10 2.00255376e-14 -950.158922 -3.20502331 );
44         lowCpCoeffs    ( 2.34433112 0.00798052075 -1.9478151e-05 2.01572094e-08 -7.37611761e-12 -917.935173 0.683010238 );
45     }
46     transport
47     {
48         As             6.362e-07;
49         Ts             72;
50     }
51 }
52

```

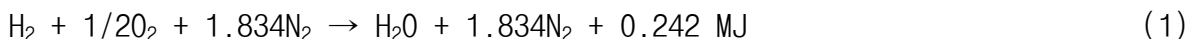
○ thermoType

본 입력인자는 radXiFoam 코드에서 열유체 해석에 사용되는 변수들에 적용될 모델을 지정하고 그룹화 해놓은 것이다. 그 세부내용은 아래와 같으며, 더 상세한 내용은 참고 문헌[2,3]에 기술되어 있다.

- type: 열전달 해석에 사용될 열역학 에너지 패키지(package)가 “heheuPsiThermo” 임을 정의한다. 이 패키지는 압축성유동에 적용되는 enthalpy가 사용된다.
- mixture: 수소증기운 폭발해석에 참여하는 혼합기체의 유형이 “vaporInhomogeneousMixture” 임을 정의하며, 수소-공기-수증기의 혼합물이 비균일하게 섞여있는 경우에 적용된다.
- transport: 유동해석에 사용될 molecular viscosity의 모델이 “sutherland” 임을 정의한다.
- thermo: 열전달 해석에 사용될 specific heat(Cp), enthalpy(h), entropy(s)의 모델이 “JANAF” 임을 정의한다. 해당하는 OpenFOAM-v2112의 소스[9]는 janafThermo.C, janafThermo.H, janafThermoI.C 이다.
- equationOfState: 열유체 해석에서 필요한 밀도를 구할 때 사용하는 상태방정식의 유형이 “perfectGas” 임을 정의하며, 이상기체법칙을 따름을 의미한다.
- specie: 증기운 폭발해석에 참여하는 혼합기체의 성분이 “화학 종(chemical specie)” 으로 정의되고 수송방정식을 풀어서 해석됨을 정의한다.
- energy: 에너지방정식에서 주요 변수로 “absoluteEnthalpy(h)” 가 사용됨을 정의한다.

○ stoichiometricAirFuelMassRatio

본 입력은 수소증기운 폭발해석에서 연소열 계산의 기준이 되는 이상적인 stoichiometric 조건에서 수소-산소 화학반응식(식 (1))을 기준으로 계산한 산소와 수소의 질량비율이다.



제목은 “stoichiometricAirFuelMassRatio” 이지만, 실제 입력에서는 Air 대신 산소(Oxygen)가 사용된 이유는 radXiFoam이 대기 중의 수증기를 모의하기 위해서 OpenFOAM-v2112의 “veryInhomogeneousMixture.C” 수정하는 과정에서 혼합기체 중의 산소의 질량분율을 식 (2)와 같이 정의했기 때문이다. 식 (2)의 “stoicRatio().value()” 가 입력으로 주어지는 “stoichiometricAirFuelMassRatio” 값을 사용한다. 즉, 공기 중의 질소는 따로 고려되고 있기 때문에 산소와 수소의 질량비율만 고려한 것이다. 식 (2)에서 “fu” 는 식 (3)과 같이 정의되며, 연소반응이 일어나지 않으면 “fu” 와 “ft” 는 동일한 값이 된다.

$$\text{scalar ox} = 1 - \text{ft} - \text{n2} - \text{wv} - (\text{ft}-\text{fu}) * \text{stoicRatio}().\text{value}() \quad (2)$$

$$\text{scalar fu} = \text{b} * \text{ft} + (1.0-\text{b}) * \text{fres}(\text{ft}, \text{stoicRatio}().\text{value}()) \quad (3)$$

○ fuel

수소증기운 폭발해석에서 연소반응의 연료에 해당하는 수소의 화학종 정보 및 열유동 해석에 사용되는 물성치 값을 정의한다.

- specie: 화학 종으로 취급되는 수소의 몰 분자량을 정의하며, 2.01594 g/mol 이다.

- thermodynamics: 열전달 해석에 사용되는 수소의 C_p , h , s 의 값을 계산하는 JANAF 다항식 모델의 계수를 지정한다. JANAF 모델이 적용될 수 있는 영역의 최고 온도(Thigh) 및 최소 온도(Tlow) 값을 지정하고, 그 값에 따른 다항식 계수(식 (4))에 적용될 수 있는 수치를 지정한다.

$$C_p = (((a[4]*T + a[3])*T a[2])*T + a[1])*T + a[0]) + \text{EquationOfState}::Cp(p, T) \quad (4)$$

- transport: 수소 molecular viscosity(μ)에 대한 sutherland 모델에 입력될 계수와 상수(식 (5))를 지정한다.

$$\mu = A_s \times \frac{\sqrt{T}}{(1 + \frac{T}{T_s})} \quad (5)$$

○ oxidant

수소증기운 폭발해석에서 연소반응의 산화제로 사용되는 산소의 화학종 정보 및 열유동 해석에 사용되는 물성치의 값을 정의한다. 그 형식은 fuel인 수소와 동일하며, 몰분자량과 계수 값들이 수소와 다르다.

○ burntProduct

수소증기운 폭발해석에서 연소반응의 생성물인 수증기(H₂O)의 화학종 정보 및 열유동 해석에 사용되는 물성치의 값을 정의한다. 그 형식은 fuel인 수소와 동일하며, 몰분자량과 계수 값들이 수소와는 다르다.

○ waterVapor

수소증기운 폭발해석에서 연소반응이 발생하기 전에 대기 중의 존재하는 수증기는 초기 조건으로 지정하며, 몰분자량과 물성치 등은 “burntProducts” 인 수증기와 동일하다.

3-1-3. combustionProperties

본 파일은 표 5와 같이 연소해석에 필요한 기본 모델인 층류화염속도, 점화, 연료공기 당량비 등에 필요한 데이터를 지정한다.

표 5. combustionProperties

```
Open 2.2 TB Volume ~/OpenFOAM/data/radXiFoam/constant Save
17
18 laminarFlameSpeedCorrelation powerLaw; 1
19
20 fuel Hydrogen; 2
21
22 equivalenceRatio equivalenceRatio [0 0 0 0 0 0] 1.1; 3
23
24
25 SuModel unstrained; //transport;
26 sigmaExt sigmaExt [0 0 -1 0 0 0] 100000; 4
27 XiModel transport; //algebraic;
28 XiCoef XiCoef [0 0 0 0 0 0] 0.62;
29 XiShapeCoef XiShapeCoef [0 0 0 0 0 0] 1;
30 uPrimeCoef uPrimeCoef [0 0 0 0 0 0] 1;
31
32
33 ignite yes;//no; 5
34 ignitionSites
35 (
36 {
37     location (6.1 0 0.025);
38     diameter 0.06;
39     start 0.0;
40     duration 0.002;
41     strength 50;
42 }
43 );
44
45 ignitionSphereFraction 1; 6
46 ignitionThickness ignitionThickness [0 1 0 0 0 0] 0.001;
47 ignitionCircleFraction 0.5;
48 ignitionKernelArea ignitionKernelArea [0 2 0 0 0 0] 0.001;
49
50
51 powerLawCoeffs 7
52 {
53     Hydrogen
54     {
55         w 2.1;
56         alpha 2.1;
57         beta -0.13;
58     }
59 }
60 }
```

- laminarFlameSpeedCorrelation
본 입력인자는 층류화염속도(S_u) 모델을 “powerLaw” 로 지정하는 것이며, powerLaw의 세부내용은 표 5의 (7)에 기술되어 있다.
- fuel
연소반응식의 연료가 “Hydrogen” 인 것을 지정한다.

○ equivalenceRatio

층류화염속도 상관식(식 (6))에 포함되어 있는 fuel-air equivalence ratio(ϕ) 값을 지정하며, 이상적인 화학반응식(식 (1))의 (수소/공기) 질량비율 대비 해석하고자 하는 혼합기체의 (수소/공기) 질량비율을 나타낸다. 이 값은 부록 1에 기술되어 있는 alphaBetaCalc.py 프로그램을 사용해서 구할 수 있다. 더 상세한 내용은 참고문헌 [10,11]에 기술되어 있다.

$$S_u = S_{u,o} \left(\frac{T_u^o}{T_o} \right)^\alpha \left(\frac{p}{p_o} \right)^\beta \quad (6)$$

○ SuModel

층류화염속도(S_u) 값을 구하기 위해서 사용될 모델을 지정한다. 총 3개의 모델인 “unstrained”, “equilibrium”, “transport” 중에서 한 가지를 선택할 수 있다.

- unstrained: 본 모델이 지정되면 “ $S_u = S_{u,o}$ ” 가 입력되며, $S_{u,o}$ 값은 표 5의 powerLawCoeffs/Hydrogen 아래의 W 값인 2.1 이다.
- equilibrium: 본 모델이 지정되면 “ $S_u = S_{inf}$ ” 가 입력되며, S_{inf} 값은 아래 식 (7)에 의해서 계산된다. Sigmas은 난류변형률이며 “bEqn.H” 루틴에서 계산되고, 사용자 입력으로 표 5와 같이 지정된다.

$$S_{inf} = S_{u,o} \times (1 - \text{Sigmas} / \text{SigmaExt}) \quad (7)$$

- transport: 본 모델이 지정되면 S_u 값을 계산하기 위한 수송방정식(transport equation)이 적용되며, 자세한 내용은 “bEqn.H” [4]에서 파악할 수 있다.

○ SuModel

층류화염속도(S_u) 값을 계산하는 모델이 “transport” 일 때, 그 수송방정식의 오른쪽에 있는 소스 항의 계산에 사용되는 변수이다. 본 변수의 도입목적은 난류 변형률이 특정 값 이상으로 증가하면 화염전파가 더 가속되지 않고 오히려 소멸되는 현상을 표현하기 위함이다. 자세한 내용은 “bEqn.H” [4]와 참고문헌[12]에 나타나 있다.

○ XiModel

“flame wrinkling(ξ)” 값을 계산하기 위해서 사용하는 모델을 지정한다. 총 3개의 계산모델인 “fixed”, “algebraic”, “transport” 중에서 한 가지를 선택할 수 있다. ξ 는 식 (8)과 같이 정의되며, 화염경계면 전파를 모의하는 수송방정식의 오른쪽에 있는 화염속도 소스 항에 위치한다. 자세한 내용은 “bEqn.H” [4]에서 파악할 수 있다.

$$\xi = \frac{S_t}{S_u} \quad (8)$$

- fixed: “fixed” 모델이 지정되면 ξ 값으로 사용자 입력에 의한 상수가 사용된다.
- algebraic: “algebraic” 모델이 지정되면 ξ 값을 구하기 위해서 식 (9)가 적용된다. 자세한 내용은 “bEqn.H” [4]와 참고문헌[13]에서 파악할 수 있다. 식(8)에서 사용되는 “XiCoef” 와 “XiShapeCoef” 는 OpenFOAM에서 추천하는 값인 “0.62” 와 “1” 이 사용되었다. “uPrimeCoef” 에 대한 기본 값은 “1” 이며, “uPrimeCoef” 값을 사용해서 Turbulent Velocity(u') 값이 식 (10)와 같이 계산된다.

$$\xi = \text{scalar}(1) + (\text{scalar}(1) + (2 * \text{XiShapeCoef}) * (\text{scalar}(0.5) - b)) * \text{XiCoef} * \sqrt{u / (S_u + S_{uMin})} * \text{Reta} \quad (9)$$

$$\text{volScalarField } u_p(uPrimeCoef * \sqrt{(2.0/3.0) * \text{turbulence} \rightarrow k()}) \quad (10)$$

- transport: “transport” 모델이 지정되면 ξ 값을 구하기 위해서 수송방정식이 적용되며, 자세한 내용은 “bEqn.H” [4]와 참고문헌[13]에서 파악할 수 있다.

○ ignite

점화모델(ignite) 사용 여부에 대해서 “yes” 또는 “no” 로 지정한다. “yes” 로 지정하면 “ignitionSites” 모델이 작동하면서 연소계산이 시작된다.

- ignitionSites: 연소현상이 시작되는 지점, 범위, 시작 시간, 지속 시간 및 크기에 대해서 지정한다. SRI Test 4-02에서는 전기스파크 점화장치가 사용되었고, 이를 모의하기 위해서 개발된 Spark Ignition Model[6,14]을 바탕으로 필요한 입력을 지정하였다. 자세한 것은 OpenFOAM-v2112 소스의 “ignite.H” [9]와 참고문헌[6,14]에 기술되어 있다.
- ignitionSphereFraction: 연소계산 영역이 3차원인 경우에 점화커널이 구형이라고 가정하는 비율 인자이다. 본 입력인자의 값이 1이면 전체가 구형이고, 1보다 작으면 일부분만 구형으로 가정된다. SRI Test 4-02에서는 사용된 Spark Ignition Model은 점화영역이 완전 구형으로 가정되었다[6,14].
- ignitionCircleFraction: 연소계산 영역이 2차원인 경우에 점화커널이 원형이라고 가정되는 비율 인자이다.
- ignitionKernalArea: 연소계산 영역이 1차원인 경우에 점화커널이 평면이라고 가정하고 본 입력인자가 사용된다.

○ powerLawCoeffs

“laminarFlameSpeedCorrelation” 로 “powerLaw” 가 지정되었을 때, 식 (5)에 적용되는 S_{uo} , α , β 값을 지정한다.

3-1-4. turbulenceProperties

본 파일은 표 6과 같이 난류유동 해석에 필요한 기본적인 입력을 지정한다.

표 6. turbulenceProperties

```
1 /*-----* C++ *-----*/
2 =====
3 \\ Field | OpenFOAM: The Open Source CFD Toolbox
4 \\ Operation | Version: v2112
5 \\ And | Web: www.OpenFOAM.com
6 \\ Manipulation |
7 /*-----*
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     location     "constant";
14     object       turbulenceProperties;
15 }
16 // *****
17
18 simulationType RAS; //LES; 1
19
20 RAS
21 {
22     RASModel     kOmegaSST; //kEpsilon; 2
23
24     turbulence   on;
25
26     printCoeffs  on;
27 }
28
29
30 LES 3
31 {
32 }
33
34 // *****
```

○ simulationType

본 입력인자는 난류유동 계산을 위해서 적용되는 모델의 유형을 지정한다. “RAS”는 “Reynolds Averaged Simulation” 이고 “LES”는 “Large Eddy Simulation”을 의미한다. 수소증기운 폭발해석은 효율적으로 계산하기 위해서 “RAS”를 사용할 것을 추천한다.

○ RAS

- RASModel: “Reynolds Averaged Simulation(RAS) turbulence modelling” 기반의 난류모델을 지정한다. 수소증기운 폭발해석은 KOmegaSST 모델[6,15]을 사용할 것을 추천한다.
- turbulence: 난류유동 계산에 대한 활성화 여부를 지정한다. “on”이 지정되면 활성화되고, “off”가 지정되면 비활성화된다.
- printCoeffs: 난류유동 계산 시 적용된 모델의 계수를 radXiFoam 실행화면에 출력할 지를 지정하는 입력이다. “on”이 지정되면 화면출력이 실행된다.

3-1-5. radiationProperties

본 파일은 표 7과 같이 난류유동 해석에 필요한 기본적인 입력을 지정한다.

표 7. radiationProperties

```
17
18 radiation on; ①
19
20 radiationModel P1; //none; ②
21
22 // Number of flow iterations per radiation iteration
23 solverFreq 10; ③
24
25 absorptionEmissionModel vaporGreyMeanAbsorptionEmission; //constantAbsorptionEmission ④
26
27 constantAbsorptionEmissionCoeffs ⑤
28 {
29     absorptivity absorptivity [0 -1 0 0 0 0] 0.01;
30     emissivity emissivity [0 -1 0 0 0 0] 0.01;
31     E E [1 -1 -3 0 0 0] 0;
32 }
33
34 vaporGreyMeanAbsorptionEmissionCoeffs ⑥
35 {
36     lookUpTableFileName none;
37
38     EhrrCoeff 0.2;
39
40     wv
41     {
42         Tcommon 200;
43         invTemp true;
44         Tlow 200;
45         Thigh 2500;
46
47         loTcoeffs
48         (
49             0
50             0
51             0
52             0
53             0
54             0
55         );
56         hiTcoeffs
57         (
58             -0.23093
59             -1.12390e3
60             9.4153e6
61             -2.99885e9
62             0.51382e12
63             -1.868e10
64         );
65     }
66 }
```

- radiation
복사열전달 계산에 대한 활성화 여부를 지정한다. “on” 이 지정되면 활성화되고, “off” 가 지정되면 비활성화된다.
- radiationModel
복사열전달 계산에 사용될 모델을 지정한다. 수소증기운 폭발해석에서는 “P1” 모델에 대해서만 검증해석을 수행하였고, 자세한 내용은 참고문헌[5,16]에 기술되어 있다.
- solverFreq
복사열전달 계산을 수행하는 빈도를 지정한다. 표 7과 같이 “10” 으로 지정하면, 유동장에 대한 반복계산(iteration)을 10회 수행할 때마다 복사열전달 계산을 1회 수행하게 된다. radXiFoam 코드는 수치해석 기법이 implicit 기반으로 개발되었기 때문에 수렴 조건을 만족시킬 때까지 유동장에 대한 반복계산을 계속 수행하게 된다.

○ absorptionEmissionModel

복사열전달 계산 시 필요한 열복사 흡수 및 방출계수에 대한 모델을 지정한다. 지정할 수 있는 모델은 “constantAbsorptionEmission” 또는 “vaporGreyMeanAbsorptionEmission” 이다. 각 모델에 대한 설명은 아래 “마”, “바” 에 기술되어 있고, 두 모델의 큰 차이는 계수의 온도 의존성이다. radXiFoam에 의한 수소증기운 폭발해석에서는 대기 중에 존재하는 수증기를 초기조건으로 모의한 후, 그 수증기에 의한 열복사 흡수 및 방출현상을 복사열전달 모델과 에너지방정식으로 모의하기 때문에 “vaporGreyMeanAbsorptionEmission” 모델을 사용할 것을 추천한다. 자세한 내용은 참고문헌[16]에 기술되어 있다.

○ constnatAbsorptionEmission

열복사 흡수, 방출 및 방출기여에 대한 계수를 지정한다.

○ vaporGreyAbsorptionEmission

열복사 흡수 및 방출계수가 온도에 따라서 변화하는 것을 모의하는 모델에 필요한 인자를 지정한다. 자세한 내용은 “vaporGreyMeanAbsorptionEmission.C” [4]와 참고문헌 [16]에서 파악할 수 있다.

- lookUpTableFileName: radXiFoam 코드에서는 외부 파일을 사용하지 않으므로 “none” 로 지정한다.
- EhrCoeff: 열복사 현상에 의한 열방출율(Q_r)과 연소(화학)반응에 인한 열방출율(Q_{chem})의 비율을 지정한다.
- wv: 수증기(Water Vapor)에 의한 열복사 흡수 및 계수 모델에 필요한 인자들을 지정한다. 자세한 내용은 참고문헌[16]에서 파악할 수 있다.

3-1-6. boundaryRadiationProperties

본 파일은 표 8과 같이 복사열전달 해석에 필요한 격자모델 경계면에서의 입력을 지정한다. 각 경계면의 이름과 열복사의 흡수 및 방출계수를 지정한다. “type” 에서 지정된 “lookup” 은 입력된 계수를 사용하겠다는 의미이다. 본 입력과 “0” 디렉토리의 G 파일의 입력이 서로 중복되면, G 파일의 입력이 우선적으로 복사열전달 계산에 적용된다.

표 8. boundaryRadiationProperties

```

1 /*----- C++ -----*/
2 |=====|
3 | \ \ \ \ | F i e l d | OpenFOAM: The Open Source CFD Toolbox
4 | \ \ \ \ | O p e r a t i o n | Version: v1906
5 | \ \ \ \ | A n d | Web: www.OpenFOAM.com
6 | \ \ \ \ | M a n i p u l a t i o n |
7 /*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format        ascii;
12     class         dictionary;
13     location      "constant";
14     object        boundaryRadiationProperties;
15 }
16 // *****
17
18 "tentWall" // ".*" //
19 {
20     type          lookup;
21     emissivity    0.074;
22     absorptivity  0.074;
23 }
24
25 "walls"
26 {
27     type          lookup;
28     emissivity    0.001;
29     absorptivity  0.010;
30 }
31
32 "symmetry"
33 {
34     type          lookup;
35     emissivity    0.001;
36     absorptivity  0.001;
37 }
38
39 "open"
40 {
41     type          lookup;
42     emissivity    0.001;
43     absorptivity  0.001;
44 }
45
46 // *****

```

3-2. 초기 및 경계조건(0 디렉토리) 입력

“0” 디렉토리에는 radXiFoam 해석에 필요한 15개 입력인자 또는 필드(field)의 초기상태(initial state)와 경계면(boundary surface)에서의 정보가 그림 10과 같이 모여 있는 디렉토리이다. 여기서, 필드는 격자모델 내 각 셀에서 radXiFoam에 의해 계산되는 주요 변수를 따로 모아놓은 그룹을 의미한다.

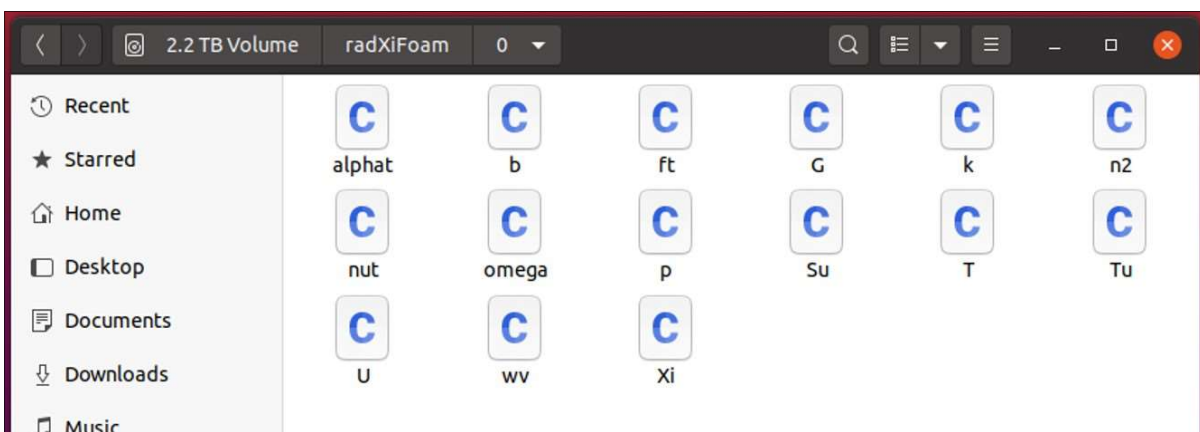


그림 10. 초기 및 경계조건(0 디렉토리) 내용 화면

3-2-2. T (Temperature)

본 입력인자는 표 10과 같이 격자모델 내 해석대상 열·유동장의 온도 필드에 대한 초기조건과 각 경계면에서의 온도 조건을 지정하며, 물리적 차원은 [K] 이다. 입력방식은 “U(Velocity)” 와 동일하다. 비균일한 온도분포 경우에는 “system/setFieldsDict” 파일을 사용해서 설정할 수 있다.

표 10. T(Temperature) 초기 및 경계조건

```
8 FoamFile
9 {
10   version      2.0;
11   format       ascii;
12   class        volScalarField;
13   location     "0";
14   object       T;
15 }
16 // *****
17
18 dimensions    [0 0 0 1 0 0 0];
19
20 internalField  uniform 293.15; // 20 degC
21
22 boundaryField
23 {
24   walls
25   {
26     type        fixedValue;
27     value       uniform 293.15;
28   }
29   symmetry
30   {
31     type        symmetryPlane;
32   }
33   open
34   {
35     type        uniformInletOutlet;
36     uniformInletValue  293.15;
37     value       uniform 293.15;
38   }
39 }
40 }
```

3-2-3. Tu (Temperature of Unburned Gas)

본 입력인자는 표 11과 같이 격자모델 내 해석대상 열·유동장에서 연소반응이 발생하지 않은 혼합기체의 온도 필드에 대한 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 [K] 이다. 입력방식은 “U(Velocity)” 와 동일하다.

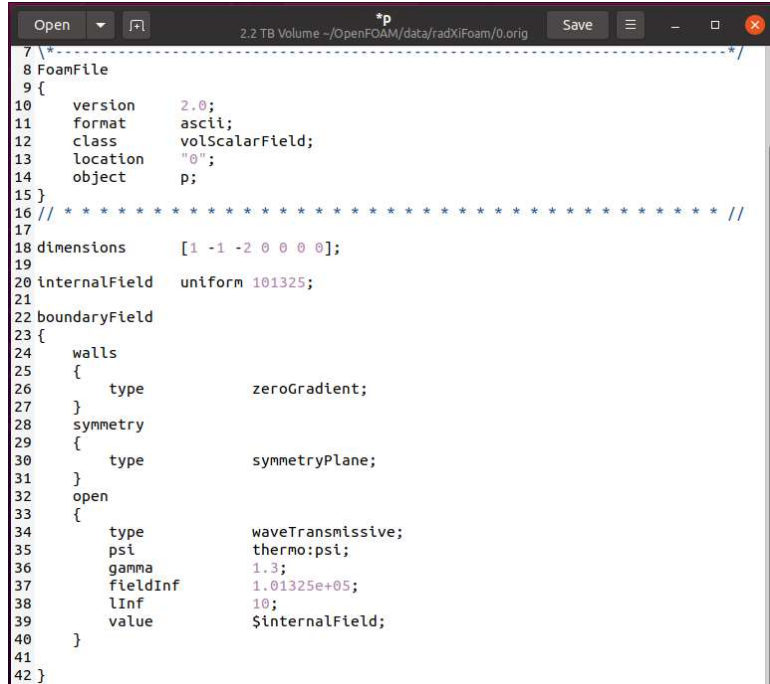
표 11. Tu (Temperature of Unburned Gas) 초기 및 경계조건

```
8 FoamFile
9 {
10   version      2.0;
11   format       ascii;
12   class        volScalarField;
13   location     "0";
14   object       Tu;
15 }
16 // *****
17
18 dimensions    [0 0 0 1 0 0 0];
19
20 internalField  uniform 293.15;
21
22 boundaryField
23 {
24   walls
25   {
26     type        fixedValue;
27     value       uniform 293.15;
28   }
29   symmetry
30   {
31     type        symmetryPlane;
32   }
33   open
34   {
35     type        uniformInletOutlet;
36     uniformInletValue  293.15;
37     value       uniform 293.15;
38   }
39 }
40 }
```

3-2-4. p (Pressure)

본 입력인자는 표 12와 같이 격자모델 내 해석대상 열·유동장에서 압력 분포에 대한 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 $[\text{kg}/\text{m} \cdot \text{s}^2]$ 이다. 입력 방식은 “가. 속도 필드”와 동일하다. 그림 20에서 경계면 “open”에서 value가 “internalField”가 인 것은 위 부분에서 지정된 “internalField uniform 101325” 값을 사용하겠다는 의미이다.

표 12. p (Pressure) 초기 및 경계조건



```
7 /*-----*/
8 FoamFile
9 {
10   version      2.0;
11   format       ascii;
12   class        volScalarField;
13   location     "0";
14   object       p;
15 }
16 // ***** //
17
18 dimenstons    [1 -1 -2 0 0 0];
19
20 internalField uniform 101325;
21
22 boundaryField
23 {
24   walls
25   {
26     type      zeroGradient;
27   }
28   symmetry
29   {
30     type      symmetryPlane;
31   }
32   open
33   {
34     type      waveTransmissive;
35     thermo:psi;
36     gamma     1.3;
37     fieldInf  1.01325e+05;
38     lInf      10;
39     value     $internalField;
40   }
41 }
42 }
```

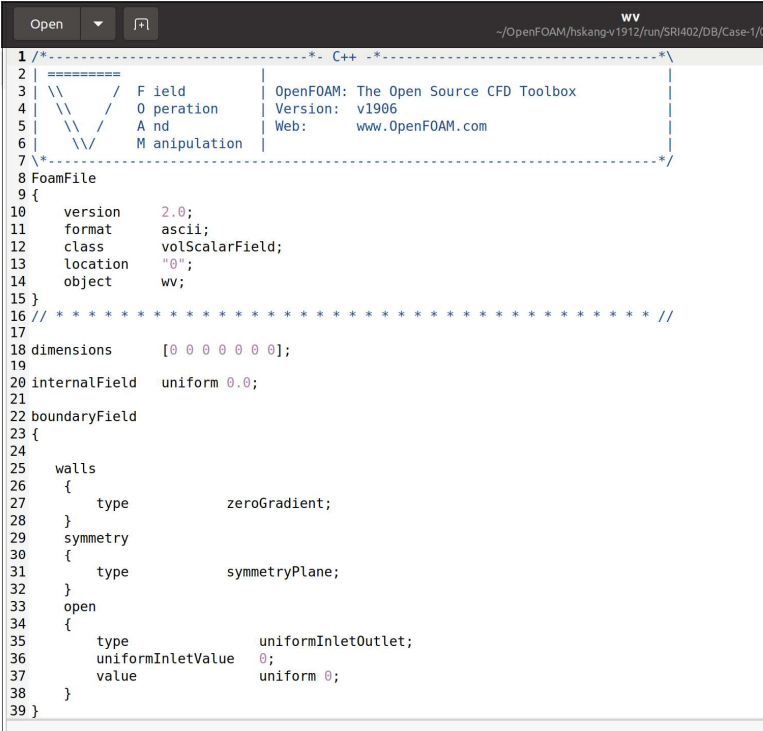
3-2-5. b (Regress Variable)

본 입력인자는 표 13과 같이 수소화염 경계면의 이송현상을 모의하는 수송방정식 (bEqn.H)의 주요 변수인 Regress Variable b에 대한 해석대상 열·유동장 내의 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 무차원이다. “b=1”은 화염이 도달하지 않아서 미연소 상태임을 의미한다. 입력방식은 “가. U(velocity)”와 동일하다.

3-2-7. wv (Mass Fraction of Water Vapor)

본 입력인자는 표 15와 같이 해석대상 열·유동장 내에서 대기 중에 존재하는 수증기의 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 질량 비율이기 때문에 무차원이다. 입력방식은 “U(Velocity)”와 동일하다. 특이사항은 동일한 수증기 양이라도 대기 중에서의 질량분율과 수소-공기 혼합기체가 존재하는 Tent 영역에서의 수증기 질량분율은 다르다. 따라서 이러한 비균일한 분포는 “system/setFieldsDict” 파일을 사용하여 지정할 수 있다. “wv” 값을 계산하는 방법은 부록 1의 “alphaBetaCalc.py”에 기술되어 있다. SRI Test 4-02에서는 실험조건에 수증기의 양이 명시되지 않아서 “0”으로 가정하였다.

표 15. wv (Mass Fraction of Water Vapor) 초기 및 경계조건



```
1 /*-----*-- C++ -*-----*\
2
3 | \ \ \ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox
4 | \ \ \ O | o p e r a t i o n | Version: v1906
5 | \ \ \ A | a n d | Web: www.OpenFOAM.com
6 | \ \ \ M | m a n i p u l a t i o n |
7 /*-----*--\
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        volScalarField;
13     location     "0";
14     object       wv;
15 }
16 // *****
17
18 dimensions      [0 0 0 0 0 0];
19
20 internalField   uniform 0.0;
21
22 boundaryField
23 {
24     walls
25     {
26         type      zeroGradient;
27     }
28     symmetry
29     {
30         type      symmetryPlane;
31     }
32     open
33     {
34         type      uniformInletOutlet;
35         uniformInletValue  0;
36         value            uniform 0;
37     }
38 }
39 }
```

3-2-8. n2 (Mass Fraction of Nitrogen)

본 입력인자는 표 16과 같이 해석대상 열·유동장 내에서 대기 중에 존재하는 질소의 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 질량 비율이기 때문에 무차원이다. 입력방식은 “가. 속도 필드”와 동일하다. 특이사항은 동일한 질소 양이라도 대기 중의 질량분율과 수소-공기 혼합기체가 존재하는 Tent 영역 내부의 질소 질량분율이 다르다. 따라서 이러한 영역에 따른 비균일한 분포는 “system/setFieldsDict” 파일을 사용하여 지정할 수 있다. “n2” 값을 계산하는 방법은 부록 1의 “alphaBetaCalc.py”에 기술되어 있다.

표 16. n2 (Mass Fraction of Nitrogen) 초기 및 경계조건

```

7 \*-----*
8 FoamFile
9 {
10   version      2.0;
11   format       ascii;
12   class        volScalarField;
13   location     "0";
14   object       n2;
15 }
16 // *****
17
18 dimensions    [0 0 0 0 0 0];
19
20 internalField  uniform 0.767;
21
22 boundaryField
23 {
24
25   walls
26   {
27     type        zeroGradient;
28   }
29   symmetry
30   {
31     type        symmetryPlane;
32   }
33   open
34   {
35     type        uniformInletOutlet;
36     uniformInletValue  0.767;
37     value         uniform 0.767;
38   }
39 }

```

3-2-9. Su (Laminar Flame Speed) 및 Xi (Flame Wrinkling)

본 입력인자는 표 17, 18과 수소화염 경계면의 이송현상을 모의하는 “Regress Variable b”의 수송방정식(bEqn.H)의 오른쪽에 있는 소스 항에 사용되는 S_u 와 ξ ($=S_t/S_u$)에 대한 해석대상 열·유동장 내에서 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 각각 [m/s], 무차원이다. 입력방식은 “가. 속도 필드”와 동일하다. S_u 는 수소농도에 따라서 그 값이 달라지며, 참고문헌[11]의 제시된 그래프를 활용하여 지정할 수 있다. SRI Test 4-02 입력에서 수소가 없는 대기 영역을 포함한 모든 영역을 나타내는 “Internal Field”에서의 S_u 값을 0.5 m/s로 지정한 이유는 실험에서 수소연소가 발생하면 Tent 영역 밖으로 연소되지 않은 수소-공기 혼합기체가 연소된 영역의 높은 압력에 의해서 밀려나오기 때문에 그 혼합기체의 연소반응 계산을 위해서 지정하였다. 수소-공기 혼합기체가 없는 영역에서는 S_u 값이 존재하더라도 수소-공기 연소반응이 일어나지 않기 때문에 입력된 값이 해석에 사용되지 않는다. ξ 의 수학적 의미는 난류화염속도와 층류화염속도의 비율이며, ξ 를 계산하는 방법은 제1절, 다, (6)에 기술되어 있다.

표 17. Su (Laminar Flame Speed) 초기 및 경계조건

```
Open  Save  2.2 TB Volume ~/OpenFOAM/data/radXiFoam/0.orig  Su
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volScalarField;
13  object       Su;
14 }
15 // ***** //
16
17 dimensions    [0 1 -1 0 0 0];
18
19 internalField uniform 0.5;
20
21 boundaryField
22 {
23     walls
24     {
25         type          zeroGradient;
26     }
27     symmetry
28     {
29         type          symmetryPlane;
30     }
31     open
32     {
33         type          uniformInletOutlet;
34         uniformInletValue 0.1;
35         value         uniform 0.1;
36     }
37 }
38
```

표 18. Xi (Flame Wrinkling) 초기 및 경계조건

```
Open  Save  2.2 TB Volume ~/OpenFOAM/data/radXiFoam/0.orig  *Xi
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volScalarField;
13  object       Xi;
14 }
15 // ***** //
16
17 dimensions    [0 0 0 0 0 0];
18
19 internalField uniform 1;
20
21 boundaryField
22 {
23     walls
24     {
25         type          zeroGradient;
26     }
27     symmetry
28     {
29         type          symmetryPlane;
30     }
31     open
32     {
33         type          uniformInletOutlet;
34         uniformInletValue 1;
35         value         uniform 1;
36     }
37 }
38
```

3-2-10. G (Incident Radiation Intensity)

본 입력인자는 표 20, 21과 같이 수소-공기 연소반응 발생 시 복사열전달 계산에 필요한 “Incident Radiation Intensity(G)” 의 해석대상 열·유동장 내에서 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 $[kg/s^3]$ 이다. 입력방식은 “U(Velocity)” 와 동일하다. type” 는 적용되는 경계조건에 대한 상세한 내용은 참고문헌[2]에 기술되어 있다.

표 19. G (Incident Radiation Intensity) 초기 및 경계조건

```

1 /*----- C++ -----*/
2
3 Field      | OpenFOAM: The Open Source CFD Toolbox
4 Operation  | Version: v1806
5 And       | Web: www.OpenFOAM.com
6 Manipulation
7
8 FoamFile
9
10 version    2.0;
11 format     ascii;
12 class      volScalarField;
13 object     G;
14
15 // *****
16
17 dimensions [1 0 -3 0 0 0 0];
18
19 internalField uniform 0;
20
21 boundaryField
22 {
23     walls
24     {
25         type      MarshakRadiation;
26                 T;
27         emissivity 0;
28     }
29     value      uniform 0;
30 }
31
32 {
33     symmetry
34     {
35         type      symmetryPlane;
36     }
37 }
38
39 {
40     type      uniformInletOutlet;
41     value      uniform 0;
42 }

```

3-2-11. k (Turbulence Kinetic Energy) 및 ω (Omega)

본 입력인자는 표 20,21과 같이 수소-공기 연소반응으로 인해서 발생하는 난류유동을 모의하기 위해서 도입된 “k- ω Shear Stress Transport(SST)” 모델의 주요 인자인 k, ω 의 해석대상 열·유동장 내에서 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 각각 $[m^2/s^2]$, $[1/s]$ 이다. 입력방식은 “U(Velocity)” 와 동일하다. 표 20, 21에서 입력인자들의 값이 각각 0.02 인데, 이것은 대기 중에 불고 있는 바람으로 인한 난류유동을 모사하기 위해서 가정하였다. Tent 내부와 Ignition Model 영역 내부의 k, ω 은 “system/setFieldsDict” 파일을 사용하여 따로 지정할 수 있다. “type” 에 적용되는 경계조건에 대한 상세한 내용은 참고문헌[2,3,18]에 기술되어 있다.

표 20. k (Turbulence Kinetic Energy) 초기 및 경계조건

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        volScalarField;
13     location     "0";
14     object       k;
15 }
16 // *****
17 //
18 dimensions      [0 2 -2 0 0 0];
19
20 internalField   uniform 1e-2;
21
22 boundaryField
23 {
24     walls
25     {
26         type      kqRWallFunction;
27         value     uniform 1e-2;
28     }
29     symmetry
30     {
31         type      symmetryPlane;
32     }
33     open
34     {
35         type      uniformInletOutlet;
36         uniformInletValue  1e-2;
37         value       uniform 1e-2;
38     }
39 }
40
41 }

```

표 21. ω (Omega) 초기 및 경계조건

```

8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        volScalarField;
13     object       omega;
14 }
15 // ***** //
16
17 dimensions      [0 0 -1 0 0 0];
18
19 internalField   uniform 1e-2;
20
21 boundaryField
22 {
23     walls
24     {
25         type      omegaWallFunction; //omegaWallFunction; //zeroGradient;
26         value     $internalField; ;
27     }
28     symmetry
29     {
30         type      symmetryPlane;
31     }
32     open
33     {
34         type      uniformInletOutlet;
35         uniformInletValue  1e-2; // 75;
36         value       uniform 1e-2; // 75;
37     }
38 }
39 }

```

3-2-12. nut (Turbulence Viscosity)

본 입력인자는 표 22와 같이 난류모델에 의해서 계산되는 난류 점성계수(ν_t)의 격자모델 내 해석대상 열·유동장에 대한 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 $[\text{kg}/\text{m} \cdot \text{s}]$ 이다. 입력방식은 “U(Velocity)” 와 동일하다.

표 22. nut(Turbulent Viscosity) 초기 및 경계조건

```
Open 2.2 TB Volume ~/OpenFOAM/data/radXFOAM/0.0rig Save
nut
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volScalarField;
13  location     "0";
14  object       nut;
15 }
16 // *****
17
18 dimensions    [0 2 -1 0 0 0 0];
19
20 internalField uniform 0;
21
22 boundaryField
23 {
24   walls
25   {
26     type      nutUSpaldingWallFunction; //nutUSpaldingWallFunction;
27     value     uniform 0;
28   }
29   symmetry
30   {
31     type      symmetryPlane;
32   }
33   open
34   {
35     type      calculated;
36     value     uniform 0;
37   }
38 }
39
40 }
```

3-2-13. alphasat (Turbulence Thermal Diffusivity)

본 입력인자는 표 23과 같이 격자모델 내 해석대상 유동장의 난류유동에 의한 열 확산(Turbulence Thermal Diffusivity)에 대한 초기조건과 각 경계면에서의 경계조건을 지정하며, 물리적 차원은 $[\text{kg}/\text{m} \cdot \text{s}]$ 이다. 입력방식은 “U(Velocity)” 와 동일하다.

표 23. alphasat(Turbulent Thermal Diffusivity) 초기 및 경계조건

```
Open 2.2 TB Volume ~/OpenFOAM/data/radXFOAM/0.0rig Save
alphat
1 /*----- C++ -----*/
2
3 Field      OpenFOAM: The Open Source CFD Toolbox
4 Operation  Version: v1806
5 And        Web:      www.OpenFOAM.com
6 Manipulation
7
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volScalarField;
13  location     "0";
14  object       alphat;
15 }
16 // *****
17
18 dimensions    [1 -1 -1 0 0 0 0];
19
20 internalField uniform 0;
21
22 boundaryField
23 {
24   walls
25   {
26     type      compressible::alphatWallFunction;
27     Prt       0.85;
28     value     uniform 0;
29   }
30   symmetry
31   {
32     type      symmetryPlane;
33   }
34   open
35   {
36     type      calculated;
37     value     uniform 0;
38   }
39 }
40 }
```

3-3. 계산제어(system 디렉토리) 입력

“system” 디렉토리에는 radXiFoam 해석을 위해서 필요한 설정 파일들과 진행 과정을 제어하기 위한 파일들이 그림 11과 같이 모여있는 디렉토리이다.



그림 11. 계산제어(System 디렉토리) 내용 화면

3-3-1. controlDict

본 파일의 내용(표 24, 25)은 천이과정으로 진행되는 radXiFoam 해석의 전반적인 진행 과정과 입력 및 출력 형태를 제어한다.

표 24. controlDict 입력

```
Open [icon] *controlDict
2.2 TB Volume ~/OpenFOAM/data/radXiFoam/system
17
18 application radXiFoam;
19
20 startFrom latestTime;
21
22 startTime 0;
23
24 stopAt endTime;
25
26 endTime 0.1;
27
28 deltaT 1e-5;
29
30 writeControl adjustableRunTime;
31
32 writeInterval 0.001;
33
34 purgeWrite 0;
35
36 writeFormat binary;
37
38 writePrecision 6;
39
40 writeCompression off;
41
42 timeFormat general;
43
44 timePrecision 6;
45
46 runtimeModifiable true;
47
48 adjustTimeStep yes;
49
50 maxCo 0.8;
51
52 maxDeltaT 0.0002;
53
```

- application
해석에 사용될 OpenFOAM solver의 이름을 지정한다.
- startFrom
천이과정 해석에서 기존의 결과를 초기조건으로 사용하여 재시작할 때의 명령어이다.

○ startTime

천이과정 해석에서 시작 시간을 지정할 때 사용되는 명령어이며, “0” 또는 임의 시간이 지정될 수 있다.

○ stopAt 및 endTime

천이과정으로 시작된 해석이 종료될 때의 시간으로 “endTime” 을 사용하라는 명령어이다. “endTime” 은 실제 종료 시간을 지정하는 명령어이다.

○ deltaT

천이과정 해석에서 계산시간 간격을 지정하는 명령어이며, 일반적으로 계산시간 간격이 연소유동장 해석결과에 상당한 영향을 미칠 수 있기 때문에 매우 주의해서 결정해야 한다. 본 radXiFoam 코드에서는 기본 값으로 “1e-5 s” 을 추천하고, 변동 가능한 계산시간 간격(adjustTimeStep)과 “maxCo” 값에 따라서 계산 중에 자동으로 계산시간 간격이 조정되는 방법을 추천한다. 자세한 것은 참고문헌[2,6]에 기술되어 있다.

○ writeControl 및 writeInterval

천이과정 해석에서 중간 결과를 저장하는 방법을 지정하는 명령어이며, “adjustableRunTime” 과 “0.001” 로 지정하면, 시작 시간을 기준으로 0.001 s 마다 중간결과를 저장하게 된다. 다른 방법으로는 iteration 수를 기준으로 하는 “timeStep” 방법도 있다[2].

○ purgeWrite 및 writeFormat

“purgeWrite” 는 천이과정 해석에서 중간결과를 저장할 때 저장공간을 관리하기 위해 사용하는 명령어이다. “0” 이면 모든 중간결과를 저장하고, “0” 이 아니면 가장 최근의 결과만 저장한 후, 이전의 모든 결과들은 삭제한다. “writeFormat” 은 저장되는 파일의 형식을 지정하는 명령어로 “binary” 는 이진법 형식의 파일로 저장하고, “ascii” 는 Text 파일 형식으로 저장한다.

○ writePrecision 및 writeCompression

“writePrecision” 는 저장되는 해석결과 파일 내 변수 숫자의 소수점이하의 자리 수를 지정하는 명령어이며, 파일형식이 “ascii” 일 때만 적용된다. “writeCompression” 은 저장파일의 압축 여부를 “on” 또는 “off” 로 지정하는 명령어이다.

○ timeFormat 및 timePrecision

“timeFormat” 는 저장되는 해석결과 파일 내의 시간 값의 형식을 지정하는 명령어이다. “general” 은 지수 형식, “fixed” 는 고정 소수점 형식으로 저장한다. “timePrecision” 는 시간 값의 숫자의 소수점이하의 자리 수를 지정하는 명령어이다.

○ `runTimeModifiable` 및 `adjustTimeStep`

“`runTimeModifiable`”은 해석이 진행되는 중간에 `controlDict` 파일의 수정가능 여부를 “`true`” 또는 “`false`”로 지정하는 명령어이다. “`adjustTimeStep`”은 Solver가 자동으로 계산시간 간격을 조정할 수 있도록 허용하는 명령어를 “`yes`” 또는 “`no`”로 지정한다. “`yes`”이면 Solver는 “`deltaT`”, “`maxCo`”, “`maxDeltaT`”의 값을 사용하여 계산시간 간격의 크기를 자동으로 조정하게 된다.

○ `maxCo` 및 `maxDeltaT`

“`maxCo`”는 Courant 수의 최대 값을 지정하며, “`maxDeltaT`”는 계산시간 간격(`deltaT`)의 최대 값을 지정한다. $Courant(=U \cdot \Delta t / \Delta x)$ 수는 일반적으로 “1” 이하로 지정한다. 그 의미는 해석유동장에서 유체의 속도를 고려하여 모든 셀에서 차분방정식이 계산될 수 있도록 계산시간 크기를 조정하는 것이다. 자세한 사항은 참고문헌[2,17]에 기술되어 있다.

○ `functions`

본 입력은 해석이 진행될 때 사용자에게 의해 추가되는 함수를 실행한다. 일반적으로 사용자가 해석대상 열·유동장 내 국부지점에서 특정 변수 값을 추적하고자 할 때 사용한다.

○ `sensorP`

본 입력세트는 계산된 압력(Pressure) 결과를 7개의 위치에서 0.0002 s 간격으로 출력하여 파일로 저장하는 명령어이다. “`sensorP`”의 이름을 가진 파일이 “`case/postProcess/` 디렉토리에 생성된다.

- `enabled`: 본 함수의 작동 여부에 대한 명령어를 “`true`” 또는 “`false`”로 지정한다.
- `probes`: 압력 데이터를 추출할 때 사용하는 방식을 지정하며, “`probes`”는 국부지점에서 데이터를 하나씩 추출할 때 사용하는 방법이다. 자세한 사항은 참고문헌[2]에 기술되어 있다.
- `writeControl` 및 `writeInterval`: 천이과정으로 진행되는 해석에서 압력 데이터의 결과를 읽고 출력파일에 기록할 때 시간 간격을 지정하는 명령어이다. “`adjustableRunTime`”과 “0.0002”는 Solver에 의해서 해석이 진행되는 시간에 따라서 데이터 출력이 조정되며, 그 간격이 0.0002 s 라는 의미이다.
- `probeLocations`: “`blockMeshDict`” 파일에 의해서 생성된 격자모델 내에서 압력을 추출할 국부지점을 지정하는 명령어이다.
- `fields`: 해석결과 중에서 추출할 데이터가 `p`(Pressure) 임을 지정한다.

표 25. controlDict - functions 입력 방법

```

54 functions
55 {
56     sensorP 1
57     {
58         enabled true;
59         type probes;
60         writeControl adjustableRunTime;
61         writeInterval 0.0002;
62         probeLocations 2 (
63             (1.01 0.00 0.01) // p1
64             (6.15 0.95 0.01) // p3
65             (9.2 0.00 0.01) // p4
66             (11.19 0.00 1) // p5
67             (11.20 0.00 2.01) // p9
68             (13.3 0.00 0.01) // p2
69             (17.1 0.00 0.01) // p6
70             (27.1 0.00 1.01) // p7
71         );
72         fields (
73             p 3
74         );
75     }
76     sensorT
77     {
78         enabled true;
79         type probes;
80         writeControl adjustableRunTime;
81         writeInterval 0.0002;
82         probeLocations (
83             (6.09 0.00 0.025) // Ign
84             (5.530 0.570 0.025) // Ion1
85             (5.460 0.640 0.025) // Ion2
86             (5.390 0.141 0.025) // Ion3
87             (6.100 0.000 1.015) // Ion4
88             (6.100 0.000 0.215) // Ion5
89             (6.100 0.000 0.430) // Ion6
90         );
91         fields (
92             T
93         );
94     }
95 }
96 }
97 }
98 }
99 }

```

3-3-2. fvSchemes

본 파일의 내용(표 26)은 물리적인 지배방정식(식 (11))을 수치계산을 위한 차분방정식으로 변환할 때 사용한 방법을 지정하는 것이며, 해석하고자 하는 물리적 현상에 맞춰서 주요 변수들의 차분방법을 지정한다. 자세한 사항은 참고문헌[2,3]에 기술되어 있다.

$$\frac{\partial(\rho U)}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \mu \nabla U = -\nabla p \quad (11)$$

- ddtSchemes

지배방정식(식 (11))의 맨 왼쪽의 시간 항의 차분 방법이다. OpenFOAM에서는 기본적으로 Euler 방법을 제공한다.

- gradSchemes

지배방정식(식 (11))의 Gradient 항 (∇p)의 차분 방법이다. OpenFOAM에서는 기본적으로 Gauss linear 방법을 제공한다.

- divSchemes

지배방정식(식 (11))의 Divergence 항 ($\nabla \cdot \phi U$)의 차분 방법이다. OpenFOAM에서는 기본적으로 제공되는 방법보다는 사용자가 직접 지정하는 것을 추천하고 있다.

- $\text{div}(\phi, U)$: " $\nabla \cdot UU$ "의 Divergence 항을 차분화 할 때 "Gauss linearUpwindV cellLimited leastSquares 1" 방법을 사용한다. ϕ 는 일반적으로 격자 셀 면에서 정의되는 속도벡터를 의미하며[3,9], $\text{div}(\phi, U)$ 는 운동량 방정식의 대류항을 나타낸다.
- $\text{div}(\phi, k)$: " $\nabla \cdot kU$ "의 Divergence 항을 차분화 할 때 "Gauss upwind" 방법을 사용한다. $\text{div}(\phi, k)$ 는 난류모델의 운동에너지 방정식의 대류항을 나타낸다.

표 26. fvSchemes 입력

```

Open [v] 2.2 TB Volume ~/OpenFOAM/data/radXiFoam/system Save [≡] - □ ×
17
18 ddtSchemes
19 {
20     default Euler;
21 }
22
23 gradSchemes
24 {
25     default Gauss linear;
26 }
27
28 divSchemes
29 {
30     default none;
31     div(phi,U) Gauss linearUpwindV cellLimited leastSquares 1;
32     div(phi,K) Gauss linearUpwind cellLimited leastSquares 1;
33     div(phi,d,p) Gauss upwind;
34     div(meshPhi,p) Gauss upwind;
35     div(phi,v,p) Gauss upwind;
36     div(phi,k) Gauss upwind;
37     div(phi,kU) Gauss upwind;
38     div(phi,epsilon) Gauss upwind;
39     div(phi,omega) Gauss linearUpwind cellLimited leastSquares 1;
40     div(phi,omegaU) Gauss upwind;
41     div(phi,R) Gauss upwind;
42     div(R) Gauss linear;
43     div(phiXi,Xi) Gauss upwind;
44     div(phiXi,Su) Gauss upwind;
45     div(phiSt,b) Gauss limitedLinear01 1;
46     div(phi,ft_wv_n2_b_ha_hau) Gauss multivariateSelection
47     {
48         n2 limitedLinear01 1;
49         wv limitedLinear01 1;
50         ft limitedLinear01 1;
51         b limitedLinear01 1;
52         ha limitedLinear 1;
53         hau limitedLinear 1;
54         ea limitedLinear 1;
55         eau limitedLinear 1;
56     };
57     div(U) Gauss linear;
58     div((Su*n)) Gauss linear;
59     div((U+((Su*Xi)*n))) Gauss linear;
60     div(((rho*nuEff)*dev2(T(grad(U)))))) Gauss linear;
61 }
62
63 laplacianSchemes
64 {
65     default Gauss linear limited corrected 0.5;
66 }
67
68 interpolationSchemes
69 {
70     default linear;
71 }
72
73 snGradSchemes
74 {
75     default limited corrected 0.5;
76 }
77 wallDist
78 {
79     method meshWave;
80 }

```

○ laplacianSchemes

지배방정식(식 (11))의 확산 항 중에서 “ $\nabla^2 U$ ” 차분 방법이다. OpenFOAM에서는 기본적으로 “Gauss linear limited corrected 0.5” 방법을 제공한다.

○ interpolationSchemes

차분방정식을 푸는 과정에서 해석대상 열.유동장 내 특정 지점에서 계산된 변수(예: 온도, 압력 등) 값들 이용하여 특정 지점의 사이의 값을 추정할 때 사용하는 보간방법이다. OpenFOAM에서는 기본적으로 “선형방법(linear)”을 제공한다.

○ snGradSchemes

차분방정식을 형성하는 과정에서 격자 셀 면에 수직인 성분이 필요한 경우가 있을 때 사용하는 차분방법이다. OpenFOAM에서는 기본적으로 “limited corrected 0.5” 방법을 제공한다.

○ wallDist

난류유동을 해석하기 위해서 RAS 계열의 $k-\omega$ SST 모델을 적용하면, 기본적으로 Wall Function을 사용하기 때문에 벽면으로부터 거리 정보가 필요하며, 계산하는 방법이 “meshWave” 이다. 자세한 사항은 참고문헌[2,3,18]에 기술되어 있다.

3-3-3. fvSolution

본 파일(표 27)은 차분방정식을 해석대상 격자모델에 적용하여 생성된 행렬식(Matrix)을 풀 때 필요한 Solver의 종류, 해를 구하는 과정을 가속시켜주는 예조건화(Preconditioner)와 Smoother, 수렴기준(Tolerance)을 지정한다. 본 매뉴얼에서는 SRI Test 4-02 해석을 위해서 사용되었던 Solver 등을 간략히 열거한다. 자세한 내용은 참고문헌[2,3]에 기술되어 있다.

○ Solvers

지배방정식에 따른 차분방정식들을 행렬식으로 만든 후, 해를 구하기 위해서 적용한 solver, smoother, preconditioner, tolerance, relTol를 기술하였다. 관련 이론과 특성은 참고문헌[2,3,19,20]에 상세히 기술되어 있다.

- PCG/PBiCGStab: 행렬식을 풀 때 적용하는 Solver 중 하나인 “PreConditioned (Bi)Conjugate Gradient Solver” 을 적용한다. 대칭행렬은 PCG, 비대칭행렬은 PBiCG를 주로 사용한다.
- PCG/PBiCGStab: 본 Solver는 DILU(Diagonal Incomplete LU decomposition), Gauss-Seidel 방법으로 행렬식을 푸는 것이다.
- 위에서 언급한 Solver 외에 다중격자기법의 Solver인 GAMG(Geometric agglomerated Algebraic Multi-Grid) solver 등의 장점과 특성은 참고문헌[3,19,20]에 기술되어 있다.

표 27. fvSolution 입력

```

17
18 solvers
19 {
20     p
21     {
22         solver          PCG;
23         preconditioner  DIC;
24         tolerance       1e-06;
25         relTol          0.1;
26     }
27
28     pFinal
29     {
30         $p;
31         tolerance       1e-06;
32         relTol          0;
33     }
34     Su
35     {
36         solver          smoothSolver;
37         smoother        GaussSeidel;
38         preconditioner  DIC;
39         tolerance       1e-06;
40         relTol          0.1;
41     }
42
43     SuFinal
44     {
45         $$Su;
46         tolerance       1e-06;
47         relTol          0;
48     }
49     rho
50     {
51         solver          smoothSolver;
52         preconditioner  DIC;
53         tolerance       1e-05;
54         relTol          0.1;
55     }
56 }
57
58     rhoFinal
59     {
60         $rho;
61         tolerance       1e-05;
62         relTol          0;
63     }

```

○ smoother

smoother는 격자모델의 특성으로 인해서 행렬식의 해를 구하는 과정이 느려지는 것을 방지하기 위해서 사용기법이다. 특히 난류유동장 해석에서 자주 사용된다. 자세한 것은 참고문헌[3,9,20]에 기술되어 있다.

- GaussSeidel: 본 기법은 Jacobi 방법에서 개선된 것이며, 행렬식의 대각행렬을 “0” 이 되지 않도록 만드는 방법을 사용한다. 자세한 사항은 참고문헌[19,20]에 기술되어 있다.
- 그 외 다른 기법으로 DIC(Diagonal Incomplete-Cholesky), DICGaussSeidel, DILU(Diagonal Incomplete-LU), DILUGaussSeidel 등이 있다.

○ preconditioner

smoother는 차분방정식의 행렬식을 풀기 전에 추가적인 조치를 취하여 해를 구하는 과정이 보다 쉽고 빨리 진행되도록 하는 방법이다. 자세한 사항은 참고문헌[19]에 기술되어 있다.

- DIC/DILU: Preconditioner 기법 중의 하나로 대칭행렬에는 DIC(Diagonal Incomplete-Cholesky), 비대칭행렬에는 DILU(Diagonal Incomplete-LU)를 적용한다.
 - FDIC: 대칭행렬에 적용되는 DIC 기법을 더 빨리 실행시키는 기법인 FDIC(Faster Diagonal Incomplete-Cholesky) 이다.
- tolerance
- OpenFOAM에서는 행렬식의 해를 구할 때 iterative method(반복적 기법)을 사용하기 때문에 어떤 변수의 해(예: p, rho 등)가 iteration이 진행해도 그 값이 고정되지 않고, 조금씩 변할 수 있다. 따라서 이러한 특성 때문에 iteration에 따른 수렴된 해의 기준(예: 소수점 이하 5자리 까지는 동일해야 함)을 제시하고, 그 기준이 만족되면 수렴된 해가 구해진 것으로 판단한다.
- relTol (Relative Tolerance)
- Relative Tolerance는 iteration이 진행함에 따라 어떤 변수의 해가 변하는 정도를 평가 할 때, 어떤 시점(예: 첫 번째 iteration 시 구해진 값)을 기준으로 평가하는 것이다. 즉, 두 번째 iteration으로 구해진 해를 첫 번째 iteration의 해로 나누는 것이다. 단, relTol 값이 “0” 으로 지정되면, 본 옵션을 사용하지 않음을 의미한다.
- *Final
- 본 옵션은 iteration의 마지막 단계에서 만족시켜야 할 기준을 제시한다.
- PIMPLE
- radXiFoam 코드 내의 운동량 방정식은 Navier-Stokes Equation이기 때문에 대류항이 비선형이고, 속도와 압력이 서로 연계되어 있다. 따라서 이런 특성을 가진 운동량 방정식을 풀기 위해서 먼저 운동량 값을 가정한 후 압력을 계산한 다음, 계산된 압력 값을 사용해서 최종적으로 운동량을 계산하는 방법인 PIMPLE(PISO-semi-implicit method for Pressure-Linked Equations) 알고리즘(그림 11)을 도입하였다. 이 PIMPLE 알고리즘을 실행하기 위해서 필요한 계수들을 fvSolution 내에서 표 28과 같이 지정해야 한다.
- nCorrectors: PIMPLE 및 PISO loop(그림 11)에서 corrector를 몇 번 수행할 지를 지정한다.
 - nNonOrthogonalCorrectors: 차분방정식의 Laplacian 항과 관련하여 유체의 유동방향과 격자 셀의 표면이 직각으로 만나지 않아서 보정 작업을 수행 시 필요한 계수를 지정한다.
 - momentumPredictor: 압력방정식(그림 11)을 풀기 전에 운동량을 예측하는 단계를 수행할 것인지에 대한 여부를 “yes” 또는 “no” 로 지정한다.
 - nOuterCorrectors: 천이과정 계산에서 특정한 time(t)에서 그 다음 단계의 time(t+ Δt)으로 진행하기 전에 PIMPLE loop를 반복하는 횟수를 지정한다.

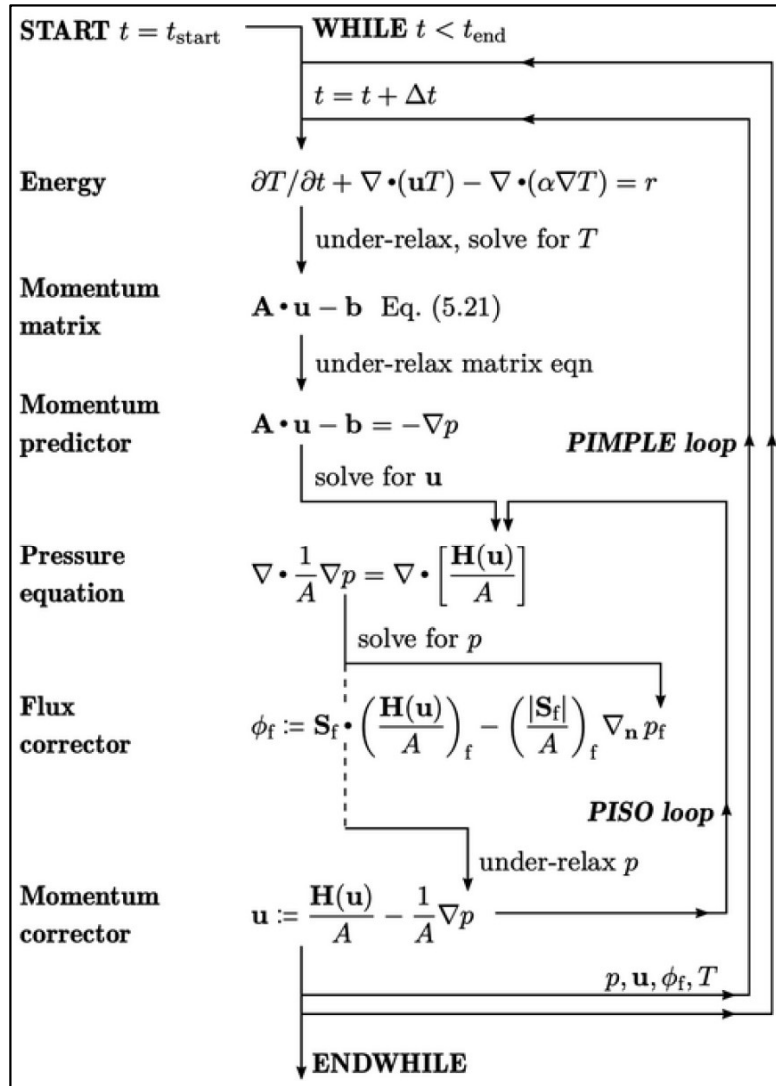


그림 11. PIMPLE Algorithm 계산 순서도

표 28. fvSolution - PIMPLE/relaxationFactors 입력

```

127 PIMPLE
128 {
129     nCorrectors 1;
130     nNonOrthogonalCorrectors 1;
131     momentumPredictor yes;
132     nOuterCorrectors 2;
133 }
134
135 relaxationFactors
136 {
137     fields
138     {
139         p 0.7;
140         pFinal 1;
141     }
142     equations
143     {
144         G 0.5;
145         U 0.8;
146         "(ft|wv|n2)" 0.8;
147         "(Su|Xi|b|ea|eau|ha|hau)" 0.8;
148         "(k|kU|omega|nuTilda|epsilon|omegaU)" 0.8;
149         UFinal 1;
150         "(ft|wv|n2)Final" 1;
151         "(Su|Xi|b|ea|eau|ha|hau)Final" 1;
152         "(k|kU|omega|nuTilda|epsilon|omegaU)Final" 1;
153     }
154 }
155 }

```

○ relaxationFactors

radXiFoam 코드는 수소증기운 폭발현상을 해석할 때 천이과정으로 해석해야 하므로 모든 차분방정식에 “시간 미분항(time derivative term)” 이 있다. OpenFOAM에서는 “시간 미분항” 을 이산화할 때 implicit 기법을 사용하기 때문에 시간간격 크기(time step size)를 explicit 기법보다 크게 할 수 있는 장점이 있지만, 새로운 시간(new time step)에서 구하는 변수는 이전 시간(previous time)에서 계산된 변수의 값을 부분적으로만 사용하기 때문에 수렴된 해를 구하는데 어려움이 있을 수 있다. 이와 같은 implicit 기법의 단점을 보완하기 위해서 “relaxation factor” 를 도입하여, 이전 시간에서의 계산된 변수 값에서 변화하는 정도를 그림 12와 같이 사용자가 0 - 1 사이에서 지정하도록 하였다. “relaxation factor” 를 주는 방법은 표 13처럼 필드 단위로 줄 수도 있고, 각 차분방정식 별로 줄 수도 있다.

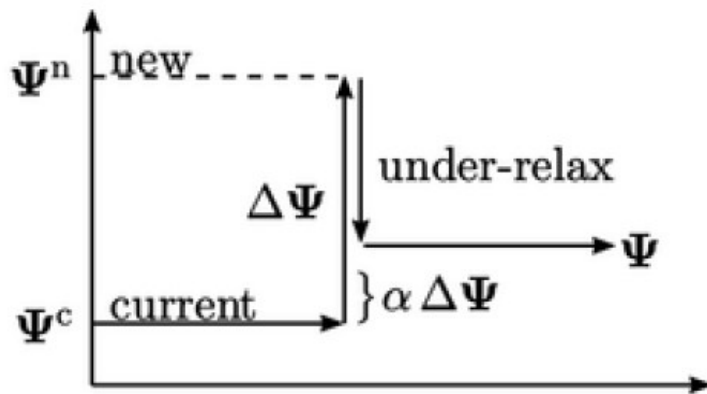


그림 12. Relaxation Factor 개념도

3-3-4. decomposeParDict

본 파일(표 29)은 병렬계산을 위해서 격자모델을 몇 개로 세분화 시킬 지에 대한 정보를 지정한다. 사용자는 계산용 하드웨어의 CPU 개수와 총 격자 셀 수를 참고하여, 약 10,000 cells/CPU 비율로 격자모델을 세분화하면 타당할 것으로 추정된다. “method scotch” 는 격자모델 세분화 시에 CPU 코어 사이의 경계의 수를 최소화 시킬 수 있는 방법으로 알려져 있다[2].

표 29. decomposeParDict 입력

```

1 /*-----* C++ -*-----*/
2
3 \ \ \ \ \ Field      | OpenFOAM: The Open Source CFD Toolbox
4 \ \ \ \ \ O peration | Version: v2112
5 \ \ \ \ \ A nd        | Web: www.OpenFOAM.com
6 \ \ \ \ \ M anipulation|
7 /*-----*
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        dictionary;
13  object       decomposeParDict;
14 }
15 // *****
16
17 numberOfSubdomains 8;
18
19 method       scotch;
20

```

3-3-5. setFieldsDict

본 파일(표 29)은 전체 해석대상 열·유동 영역에 대한 초기조건을 비균일하게 지정할 때 사용하는 입력이다. SRI Test 4-02에서는 수소-공기 연소반응이 Tent 내부에서만 발생하기 때문에 “setFieldsDict” 파일을 사용하여 화학종 조성관련 입력을 지정하였고, 실험장치 전기스파크에 의한 점화과정을 모의한 Spark Ignition Model을 위한 온도, 압력 및 난류에디 운동에너지(k)와 소멸율(s^{-1}) 등의 입력을 지정하였다. 본 입력인자 설정에 관한 자세한 내용은 참고문헌[6,14]에 기술되어 있다.

표 29. setFieldsDict 입력

```
8 FoamFile
9 {
10   version      2.0;
11   format       ascii;
12   class        dictionary;
13   location     "system";
14   object       setFieldsDict;
15 }
16 // ***** //
17 defaultFieldValues
18 (
19
20           volScalarFieldValue ft                0
21           volScalarFieldValue n2              0.767
22           volScalarFieldValue b                1
23 );
24 regions
25 (
26   boxToCell
27   {
28     box (5 0 0) (7.2 1.1 1);
29     fieldValues
30     (
31           volScalarFieldValue ft                0.029
32           volScalarFieldValue n2              0.745
33           volScalarFieldValue b                1
34           volScalarFieldValue k                150
35     );
36   }
37   sphereToCell
38   {
39     centre (6.1 0 0.025);
40     radius 0.06;
41
42     fieldValues
43     (
44           volScalarFieldValue p                105000
45           volScalarFieldValue T                1000
46           volScalarFieldValue omega           500
47           volScalarFieldValue k                337.5
48     );
49   }
50 );
```

○ defaultFieldValues

본 파일을 통해서 비균일한 분포를 주고자 하는 화학종 입력인 ft, n2, b에 대해서 먼저 기본 값을 지정한다. 본 파일에서 지정되지 않은 변수들의 기본 값은 “case/0” 디렉토리에서 설정한 값들이 사용된다.

○ regions

비균일한 영역의 이름과 범위를 먼저 지정한 다음, 변수들의 값을 지정한다.

- boxToCell: SRI Test 4-02 Tent 영역을 OpenFOAM에서 제공하는 “box” 옵션[2,3]을 사용하여 지정한 후, 화학종 조성관련 및 난류유동 초기조건을 지정하였다.
- sphereToCell: Spark Ignition Model이 차지하는 영역인 구(sphere)를 중심좌표와 반경으로 지정한 후, 열수력 및 난류유동 초기조건을 지정하였다.

4. radXiFoam 실행방법

본 장에서는 사용자가 radXiFoam 코드로 SRI Test 4-02 입력자료를 제2장, 제3장과 같이 작성한 후에 case 디렉토리에서 수소증기운 폭발해석을 실행하는 과정을 설명하고자 한다.

4-1. 실행 옵션

본 절에서는 완성된 SRI Test 4-02 입력자료를 사용하여 “1개 CPU” 를 사용하여 Serial Run 실행 또는 “다수의 CPUs” 를 사용하여 Parallel Run 실행을 하는 방법을 설명하고자 한다. 수소증기운 폭발해석은 폭발이 발생하는 영역의 격자 셀을 매우 작게 생성해야 하므로 전체 격자 수가 상당히 많아지게 된다. 따라서 병렬계산을 실행하는 것이 총 계산시간을 단축할 수 있다.

4-1-1. 환경설정 및 Serial Run 실행방법

○ OpenFOAM 환경설정 및 radXiFoam 입력생성

리눅스 환경에서 터미널을 오픈 한 후에 OpenFOAM-v2112 환경설정을 위한 명령어를 그림 13과 같이 “of2112” 입력하여 실행한 후, radXiFoam 해석 준비를 위해서 (1) blockMesh, (2) checkMesh, (3) setFields 명령어를 실행한다.

○ Serial Run(직렬계산) 실행

SRI Test 4-02 입력자료가 있는 case 디렉토리에서 그림 13과 같이 “radXiFoam” 명령어를 입력하면, “1 CPU” 를 사용하는 직렬계산(Serial Run)이 시작된다.

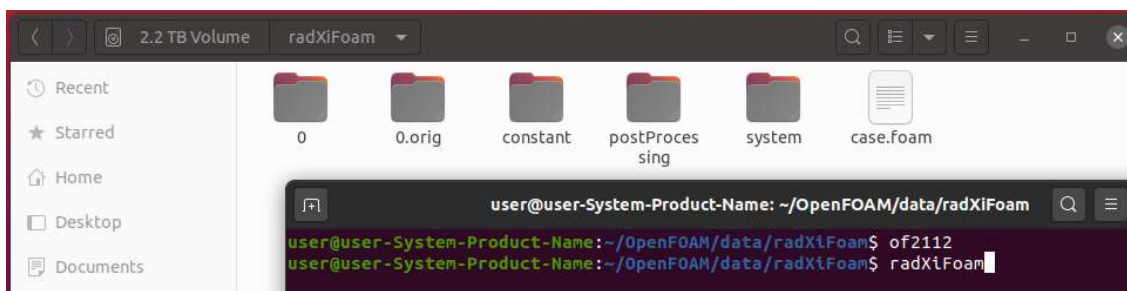


그림 13. radXiFoam 직렬계산 실행 화면

4-1-2. 병렬계산 실행방법

○ 격자모델 decompose 실행

병렬계산을 위해서는 먼저, OpenFOAM 환경설정 및 radXiFoam 입력생성을 수행한 다음, 생성된 격자모델을 병렬계산용으로 세분화 하기 위해서 decomposePar 명령어(그림 14)를 실행하면 “case\system\decomposePar” 파일에 따라서 격자모델이 분할되고 “case\WOW*” 디렉토리 내 초기조건이 입력된 후, 지정된 병렬코어 개수에 해당하는 디

렉토리가 생성된다. 이때 병렬계산에 사용되는 하드웨어의 CPU 코어 개수에 맞춰서 격자 모델을 분할해야 한다.

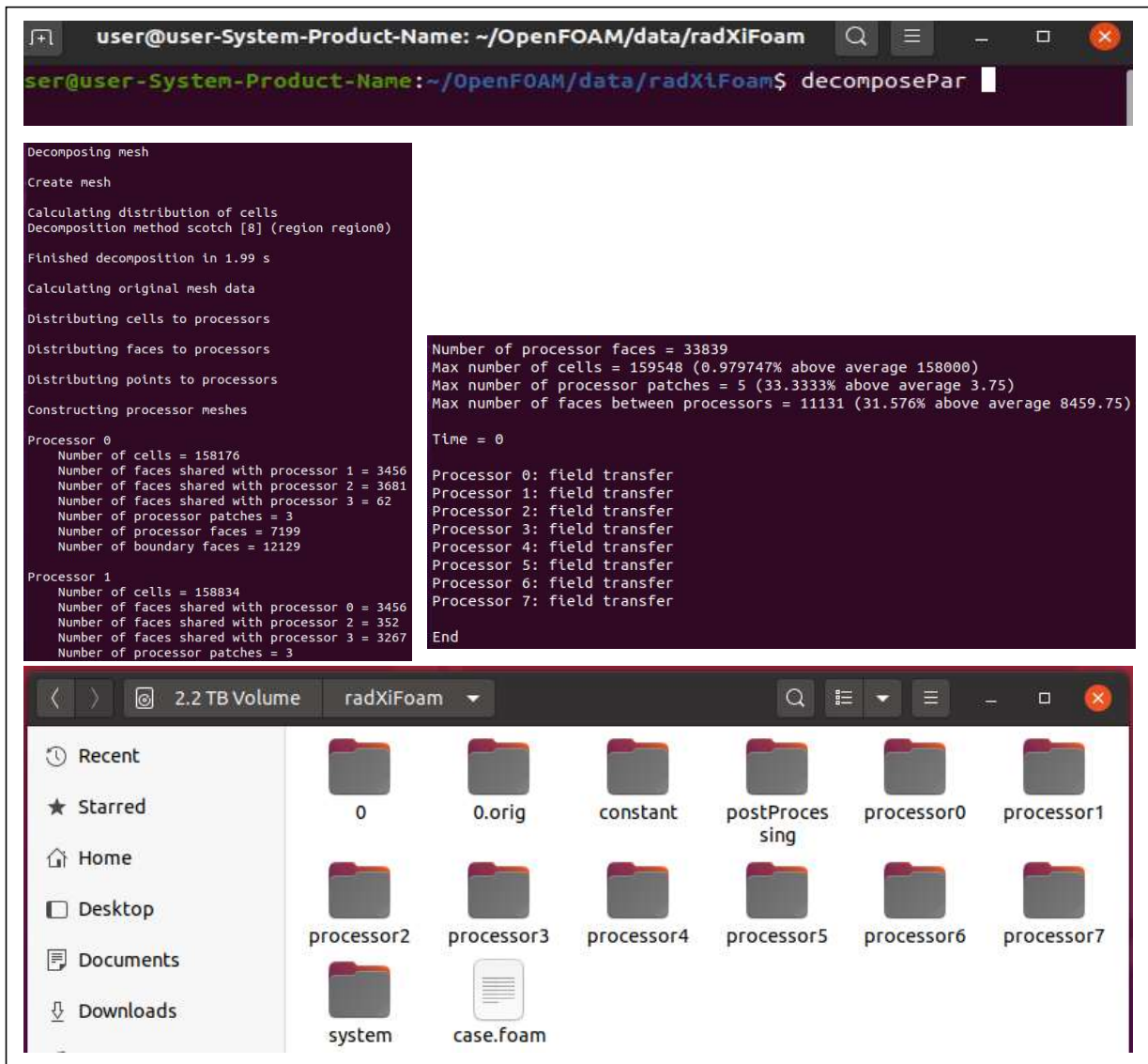


그림 14. 병렬계산용 격자모델 분할 명령어, 진행과정 및 결과 화면

○ 병렬계산 실행

decomposePar 명령어에 의해서 격자모델에 대한 분할작업이 완료되면, 그림 15과 같이 병렬계산 실행 명령어인 “mpirun -np 8 radXiFoam -parallel” 를 입력하면 계산이 시작된다. 만약, 화면에 출력되는 정보를 파일로 저장하고 싶을 때는 “mpirun -np 8 radXiFoam -parallel >> mylog” 명령어를 입력하면 된다. 여기서, mylog는 저장되는 파일의 이름이다. 병렬계산이 진행되면서 생성되는 중간결과 및 최종결과는 격자모델이 분할된 디렉토리에 그대로 저장된다. 따라서 이렇게 분할된 결과들을 한 개로 합쳐서 천이 과정 시간에 따라서 저장할 수 있는데, 그 명령어는 “reconstructPar” 이다. 그러나 본 매뉴얼에서는 이 방법보다는 해석결과 분석 소프트웨어인 “ParaView” 에서 병렬계산의 결과파일을 읽어들이는 옵션 “Decomposed Case” 을 선택하여 사용할 것을 추천한다.

```

user@user-System-Product-Name: ~/OpenFOAM/data/radXiFoam
user@user-System-Product-Name:~/OpenFOAM/data/radXiFoam$ mpirun -np 8 radXiFoam -parallel

```

그림 15. radXiFoam 병렬계산 실행 명령어

4-2. 실행 화면 및 출력정보 확인

4-2-1. 실행 명령어 입력 후 초기화 정보 출력

본 절에서는 radXiFoam 실행 명령어를 입력하면 화면에 출력되는 초기화 정보(그림 16)에 대해서 설명하고자 한다.

```

// ***** //
Create time

Create mesh for time = 0

PIMPLE: no residual control data found. Calculations will employ 2 corrector loops

Reading combustion properties

Found ignition cells:
1(324060)

Ignition on
Reading thermophysical properties

Selecting thermodynamics package
{
    type            heheuPsiThermo;
    mixture         vaporInhomogeneousMixture;
    transport       sutherland;
    thermo          janaf;
    equationOfState perfectGas;
    specie          specie;
    energy          absoluteEnthalpy;
}

min(b) = 1

Reading field U

Reading/calculating face flux field phi

Creating turbulence model

Selecting turbulence model type RAS
Selecting RAS turbulence model kOmegaSST
Selecting patchDistMethod meshWave
RAS
{
    RASModel        kOmegaSST;
    turbulence      on;
    printCoeffs    on;
    alphaK1         0.85;
    alphaK2         1;
    alphaOmega1     0.5;
    alphaOmega2     0.856;
    gamma1         0.555556;
    gamma2         0.44;
    beta1           0.075;
    beta2           0.0828;
    betaStar        0.09;
    a1              0.31;
    b1              1;
    c1              10;
    F3              false;
    decayControl    false;
    kInf            0;
    omegaInf        0;
}

Creating field Xi

Creating the unstrained laminar flame speed

Selecting laminar flame speed correlation powerLaw
Reading strained laminar flame speed field Su

Calculating turbulent flame speed field St

Creating field dpdt

Creating field kinetic energy K

No MRF models present

Selecting radiationModel P1
Selecting absorptionEmissionModel vaporGreyMeanAbsorptionEmission
Selecting scatterModel none
Selecting sootModel none
No finite volume options present
Courant Number mean: 0 max: 0

Starting time loop

Courant Number mean: 0 max: 0
deltaT = 1.20482e-05
Time = 1.20482e-05

```

그림 16. radXiFoam 계산을 위한 초기화 과정 화면

- PIMPLE loop corrector 횟수
“case/system/fvSolution” 에서 입력한 정보와 동일한 지 확인한다.
- ignition model 사용
“case/constnat/combustionProperties” 에서 입력한 정보와 동일한 지 확인한다.
- thermodynamics package 정보
“case/constant/themophysicalProperties” 에서 입력한 정보와 동일한 지 확인한다.
- turbulence model 정보
“case/constnat/turbulenceProperties” 에서 입력한 정보와 동일한 지 확인한다. 난류모델에 적용된 계수들은 실험과 이론을 통해서 결정된 값이므로 OpenFOAM에서 제공하는 기본 값을 사용할 것을 추천한다.
- combustion 계산 정보
“case/constnat/combustionProperties” 에서 입력한 정보와 동일한 지 확인한다.
- radiative heat transfer 정보
“case/constnat/radiationProperties” 에서 입력한 정보와 동일한 지 확인한다.

4-2-2. 해석 진행 중 출력 정보

본 절에서는 radXiFoam 천이과정으로 진행되기 때문에 각 time에서 iteration이 진행되는 과정이 화면에 그림 17과 같이 출력된다.

- 운동량, 화학종 차분방정식 해석 진행과정
PIMPLE loop 안에서 운동량 및 화학종 차분방정식의 해를 구하기 위해서 iteration이 진행되면서 계산된 residual과 오차 등이 화면에 출력된다. 각 차분방정식에 적용된 Solver, Corrector 및 Loop 수 등은 “case/system/fvSolution” 에서 지정된 것이다.
- radiative heat transfer 계산관련 warning 메시지
radXiFoam은 대기 중에 존재하는 수증기에 의한 복사열전달을 계산하는 알고리즘이 개발되어 삽입되었다. 따라서 대기 중의 수증기는 수소증기운 폭발과정에서 수소-산소 연소반응 같은 화학반응을 일으키지 않으므로 그림 17과 같이 Heat Source인 “ Q_{dot} field” 가 발견되지 않는다는 경고메시지가 출력된다. 그러나 이것은 비물리적인 현상이 아니므로 무시한다.
- 압력, 난류유동 차분방정식 해석 진행과정
PIMPLE loop 안에서 압력 및 난류유동 차분방정식의 해를 구하기 위해서 iteration이 진행되면서 계산된 residual과 오차 등이 화면에 출력된다.

○ 해석에 소요된 하드웨어 시간정보

해석이 진행되면서 컴퓨터 CPU가 사용된 시간과 소요된 물리적 시간정보가 출력된다.

○ 천이과정 계산정보

각 time step에서 천이과정 계산정보인 계산시간 간격크기(deltaT), 현재 계산시간 (Time) 및 계산된 Courant 수가 출력된다.

```
diagonal: Solving for rho, Initial residual = 0, Final residual = 0, No Iterations 0
PIMPLE: iteration 1
DILUPBiCGStab: Solving for Ux, Initial residual = 0.0444981, Final residual = 1.17837e-05, No Iterations 1
DILUPBiCGStab: Solving for Uy, Initial residual = 0.0362185, Final residual = 5.55443e-06, No Iterations 1
DILUPBiCGStab: Solving for Uz, Initial residual = 0.0424185, Final residual = 8.11744e-06, No Iterations 1
DILUPBiCGStab: Solving for ft, Initial residual = 0.00160561, Final residual = 5.89658e-07, No Iterations 1
DILUPBiCGStab: Solving for wv, Initial residual = 0.00305759, Final residual = 7.20431e-07, No Iterations 1
DILUPBiCGStab: Solving for n2, Initial residual = 0.000605209, Final residual = 3.75509e-07, No Iterations 1
StCorr = 1
Max St-Courant Number = 0.0189866
DILUPBiCGStab: Solving for b, Initial residual = 7.22803e-06, Final residual = 7.22803e-06, No Iterations 0
min(b) = 3.09928e-06
DILUPBiCGStab: Solving for Xi, Initial residual = 0.00225357, Final residual = 1.33675e-08, No Iterations 1
max(Xi) = 6.98634
max(XiEq) = 39.8829
Combustion progress = 0.000222346%
FOAM Warning :
From virtual Foam::tmp<Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh> > Foam::radiation::
vaporGreyMeanAbsorptionEmission::ECont(Foam::label) const
in file submodels/absorptionEmissionModel/vaporGreyMeanAbsorptionEmission/vaporGreyMeanAbsorptionEmission.
C at line 343
Qdot field not found in mesh
DILUPBiCGStab: Solving for hau, Initial residual = 0.00354425, Final residual = 8.30977e-07, No Iterations 1
--> FOAM Warning :
From virtual Foam::tmp<Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh> > Foam::radiation::
vaporGreyMeanAbsorptionEmission::ECont(Foam::label) const
in file submodels/absorptionEmissionModel/vaporGreyMeanAbsorptionEmission/vaporGreyMeanAbsorptionEmission.
C at line 343
Qdot field not found in mesh
DILUPBiCGStab: Solving for ha, Initial residual = 0.00282378, Final residual = 5.13239e-07, No Iterations 1
DICPCG: Solving for p, Initial residual = 0.0319295, Final residual = 8.97777e-05, No Iterations 1
DICPCG: Solving for p, Initial residual = 8.77902e-05, Final residual = 2.75798e-08, No Iterations 2
diagonal: Solving for rho, Initial residual = 0, Final residual = 0, No Iterations 0
time step continuity errors : sum local = 6.18211e-12, global = 8.26343e-13, cumulative = 6.45675e-11

PIMPLE: iteration 2
DILUPBiCGStab: Solving for Ux, Initial residual = 0.0110105, Final residual = 4.09814e-06, No Iterations 1
DILUPBiCGStab: Solving for Uy, Initial residual = 0.00876576, Final residual = 1.89999e-06, No Iterations 1
DILUPBiCGStab: Solving for Uz, Initial residual = 0.0103703, Final residual = 2.75393e-06, No Iterations 1
DILUPBiCGStab: Solving for ft, Initial residual = 0.000238275, Final residual = 5.99978e-09, No Iterations 1
DILUPBiCGStab: Solving for wv, Initial residual = 0.000536102, Final residual = 1.0918e-08, No Iterations 1
DILUPBiCGStab: Solving for n2, Initial residual = 2.69512e-05, Final residual = 3.14296e-09, No Iterations 1
StCorr = 1
Max St-Courant Number = 0.0194186
DILUPBiCGStab: Solving for b, Initial residual = 8.85391e-06, Final residual = 8.85391e-06, No Iterations 0
min(b) = 3.09928e-06
DILUPBiCGStab: Solving for Xi, Initial residual = 0.000575016, Final residual = 4.50767e-09, No Iterations 1
max(Xi) = 7.02714
max(XiEq) = 39.6784
Combustion progress = 0.000222346%
--> FOAM Warning :
From virtual Foam::tmp<Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh> > Foam::radiation::
vaporGreyMeanAbsorptionEmission::ECont(Foam::label) const
in file submodels/absorptionEmissionModel/vaporGreyMeanAbsorptionEmission/vaporGreyMeanAbsorptionEmission.
C at line 343
Qdot field not found in mesh
DILUPBiCGStab: Solving for hau, Initial residual = 0.000650167, Final residual = 4.0106e-08, No Iterations 1
--> FOAM Warning :
From virtual Foam::tmp<Foam::GeometricField<double, Foam::fvPatchField, Foam::volMesh> > Foam::radiation::
vaporGreyMeanAbsorptionEmission::ECont(Foam::label) const
in file submodels/absorptionEmissionModel/vaporGreyMeanAbsorptionEmission/vaporGreyMeanAbsorptionEmission.
C at line 343
Qdot field not found in mesh
DILUPBiCGStab: Solving for ha, Initial residual = 0.00108479, Final residual = 1.1664e-07, No Iterations 1
DICPCG: Solving for p, Initial residual = 0.0130602, Final residual = 5.74877e-05, No Iterations 1
DICPCG: Solving for p, Initial residual = 5.60853e-05, Final residual = 9.64354e-07, No Iterations 1
diagonal: Solving for rho, Initial residual = 0, Final residual = 0, No Iterations 0
time step continuity errors : sum local = 2.24024e-10, global = 7.14864e-11, cumulative = 1.36054e-10
smoothSolver: Solving for omega, Initial residual = 0.00347668, Final residual = 3.76304e-06, No Iterations 1
smoothSolver: Solving for k, Initial residual = 0.00282802, Final residual = 8.39213e-06, No Iterations 1
ExecutionTime = 16.04 s ClockTime = 17 s
Courant Number mean: 1.33829e-05 max: 0.0434063
deltaT = 5e-05
Time = 0.00135
```

그림 17. radXiFoam 해석 진행과정

5. 해석결과 분석방법

본 장에서는 사용자가 제4장과 같이 radXiFoam 코드로 SRI Test 4-02에 대한 해석을 수행하여 생성된 해석결과를 분석하는 방법에 대해서 설명하고자 한다.

5-1. postProcessing 디렉토리

본 절에서는 “case\system\controlDict” 파일의 functions에서 지정한 국부위치에서의 압력과 온도 데이터가 생성된 디렉토리 위치와 그 내용에 대해서 설명하고자 한다.

5-1-1. 디렉토리 위치

국부지점의 압력과 온도 데이터인 p, T 파일이 생성된 곳은 그림 40과 같이 “case\postProcessing” 하부의 sensorP, sensorT 디렉토리 내부이다.

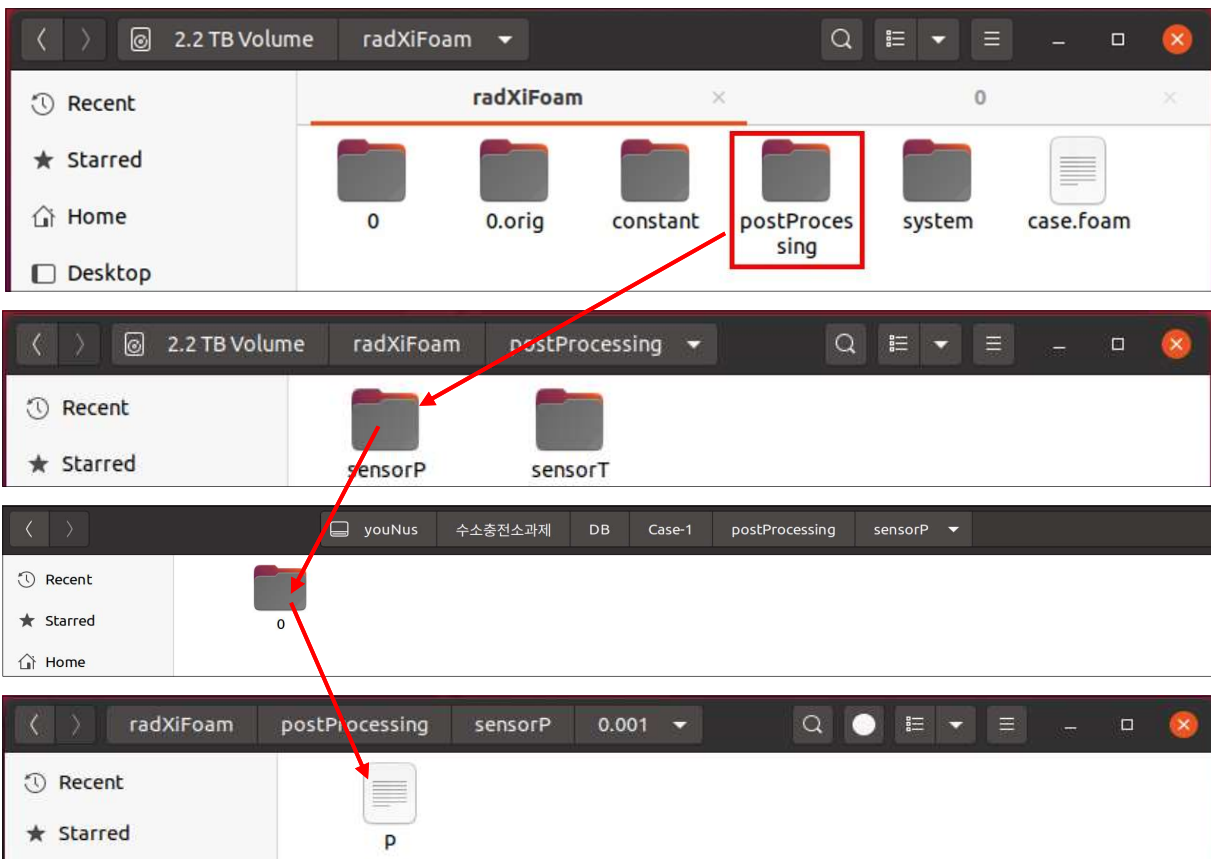


그림 18. controlDict의 functions에 의해 생성된 postProcessing 디렉토리

5-1-2. 데이터 내용

압력 데이터 파일인 “p” 를 문서 편집기로 열어보면 그림 19와 같이 나타나며, “controlDict” 의 functions에서 probe 위치를 지정한 순서에 따라서 probe 번호가 “0” 부터 부여되며, “라인번호 9” 줄에 전체 probe가 번호순서 대로 출력된다. 그 아래 줄에는 문자열 “time” 이 출력되며, “라인번호 11” 부터는 time, pressure 데이터가 probe 번호 아래에서 각각 출력된다. 여기서, 시간은 “controlDict” 에서 지정된 write interval에 의해서 출력되며 단위는 [s] 이다. 압력은 절대압력(absolute pressure)이고 단위는 [Pa] 이다. 그런데 증기운 폭발사고 해석결과 분석에서는 게이지 압력인 과압(overpressure)과 [kPa] 단위를 주로 사용한다. 따라서 본 압력 데이터를 사용해서 [kPa] 단위의 과압 그래프로 그려주는 파이썬 프로그램 “overPressurePlot.py” 를 개발하였고, 그 소스 내용은 부록 3에 나타나 있다.

9 #	Probe	0	1	2	3	4	5	6	7
10 #	Time								
11	0.0001	101325	98322.8	101325	101325	101325	101325	101325	101325
12	0.0002	101325	98736.5	101325	101325	101325	101325	101325	101325
13	0.0003	101325	100272	101325	101325	101325	101325	101325	101325
14	0.0004	101325	102841	101325	101325	101325	101325	101325	101325
15	0.0005	101325	104845	101325	101325	101325	101325	101325	101325
16	0.0006	101325	105158	101325	101325	101325	101325	101325	101325
17	0.0007	101325	104015	101325	101325	101325	101325	101325	101325
18	0.0008	101325	102250	101325	101325	101325	101325	101325	101325
19	0.0009	101325	100725	101325	101325	101325	101325	101325	101325
20	0.001	101325	99775.5	101325	101325	101325	101325	101325	101325
21	0.0011	101325	99342.1	101325	101325	101325	101325	101325	101325
22	0.0012	101325	99272.7	101325	101325	101325	101325	101325	101325
23	0.0013	101325	99426	101325	101325	101325	101325	101325	101325
24	0.0014	101325	99640.3	101325	101325	101325	101325	101325	101325
25	0.0015	101325	99791.4	101325	101325	101325	101325	101325	101325
26	0.0016	101325	99842.7	101325	101325	101325	101325	101325	101325
27	0.0017	101325	99833	101325	101325	101325	101325	101325	101325
28	0.0018	101325	99839.2	101325	101325	101325	101325	101325	101325
29	0.0019	101325	99885.7	101325	101325	101325	101325	101325	101325
30	0.002	101325	100325	101325	101325	101325	101325	101325	101325
31	0.0021	101325	100881	101325	101325	101325	101325	101325	101325
32	0.0022	101325	101636	101325	101325	101325	101325	101325	101325
33	0.0023	101325	102525	101325	101325	101325	101325	101325	101325
34	0.0024	101325	103446	101325	101325	101325	101325	101325	101325
35	0.0025	101325	104289	101325	101325	101325	101325	101325	101325
36	0.0026	101325	104929	101325	101325	101325	101325	101325	101325
37	0.0027	101325	105299	101325	101325	101325	101325	101325	101325
38	0.0028	101325	105375	101325	101325	101325	101325	101325	101325
39	0.0029	101325	105178	101325	101325	101325	101325	101325	101325
40	0.003	101325	104764	101325	101325	101325	101325	101325	101325

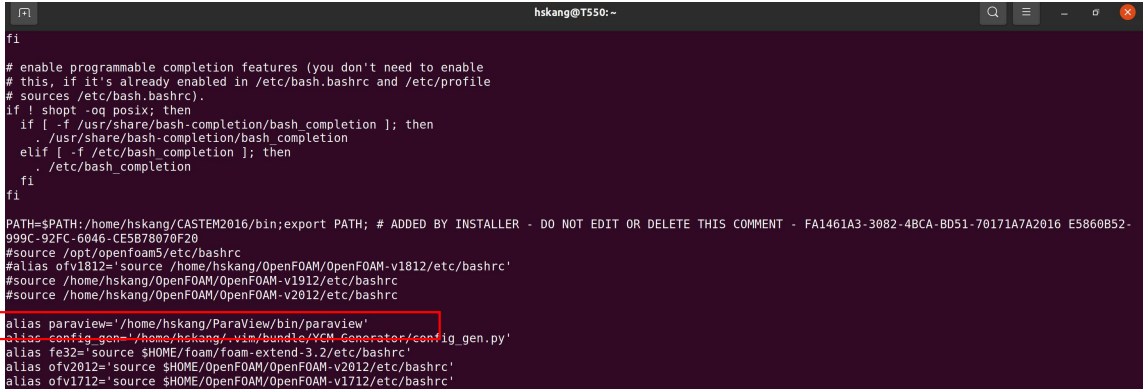
그림 19. 국부지점 압력 데이터 파일 내용

5-2. ParaView 사용 가시화 방법

천이과정으로 진행된 radXiFoam의 해석결과는 “case\system\controlDict” 에서 지정된 “writeInterval” 에 의해서 생성된 여러 time directory 안에 각각 저장된다. 본 절에서는 그 해석결과를 3차원으로 분석할 수 있는 ParaView 소프트웨어[21]의 사용법을 설명하고자 한다. ParaView는 제1장에서 설명한 OpenFOAM-v2112 설치과정의 하나인 ThirdParty-v2112.tgz 파일을 압축해제한 후 수행하는 과정에서 설치되거나, 만약 이 과정이 정상적으로 이루어지지 않으면 ParaView 홈페이지[22]에서 “binary 실행파일” 을 다운로드해서 설치할 수도 있다. 본 매뉴얼에서는 “binary 실행파일” 을 이용하여 설치된 ParaView-5.6을 기준으로 설명하고자 한다.

5-2-1. ParaView 설치방법

ParaView를 다운로드하기 위해서는 ParaView 홈페이지 “www.paraview.org/download”에서 적합한 Version을 선택한 후 다운로드용 아이콘을 누르면 된다. 다운로드한 파일을 적절한 곳에서 압축해제한 후, 실행파일이 있는 곳의 위치를 리눅스 환경설정 파일 (“/Home/.bashrc”)에 그림 20과 같이 alias 명령어를 사용하여 연결하고, 리눅스 터미널 창에서 “paraview” 명령어를 입력하면 ParaView 소프트웨어의 GUI 화면이 나타난다.



```
hskang@T550: ~  
fi  
# enable programmable completion features (you don't need to enable  
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile  
# sources /etc/bash.bashrc).  
# sources /etc/bash.bashrc).  
if ! shopt -oq posix; then  
if [ -f /usr/share/bash-completion/bash_completion ]; then  
./usr/share/bash-completion/bash_completion  
elif [ -f /etc/bash_completion ]; then  
./etc/bash_completion  
fi  
fi  
PATH=$PATH:/home/hskang/CASTEM2016/bin;export PATH; # ADDED BY INSTALLER - DO NOT EDIT OR DELETE THIS COMMENT - FA1461A3-3082-4BCA-BD51-70171A7A2016 E5860B52-999C-92FC-6046-CE5B78070F20  
#source /opt/openfoam5/etc/bashrc  
#alias ofv1812='source /home/hskang/OpenFOAM/OpenFOAM-v1812/etc/bashrc'  
#source /home/hskang/OpenFOAM/OpenFOAM-v1912/etc/bashrc  
#source /home/hskang/OpenFOAM/OpenFOAM-v2012/etc/bashrc  
alias paraview="/home/hskang/ParaView/bin/paraview"  
alias config_gen="/home/hskang/vim/bundle/YCM-Generator/config_gen.py"  
alias fe32='source $HOME/foam/foam-extend-3.2/etc/bashrc'  
alias ofv2012='source $HOME/OpenFOAM/OpenFOAM-v2012/etc/bashrc'  
alias ofv1712='source $HOME/OpenFOAM/OpenFOAM-v1712/etc/bashrc'
```

그림 20. Paraview 소프트웨어 환경설정 내용 화면

5-2-2. ParaView 실행방법

설치완료된 ParaView를 실행한 후, SRI Test 4-02 해석결과 파일을 불러오고 주요 변수의 결과를 가시화하는 방법은 다음과 같다.

○ ParaView 실행 및 radXiFoam 해석결과 파일 불러오기

ParaView를 실행하기 전에 먼저 해석결과가 있는 디렉토리에서 “/case/touch case.foam” 명령어를 입력하여 case.foam 파일을 생성한 후, “paraview case.foam” 명령어를 입력하면 ParaView 소프트웨어의 GUI 화면이 그림 21과 같이 나타난다. 그림 21의 “Case Type”을 Reconstructed Case에서 Decomposed Case로 변경한 후 “Apply” 버튼을 누르고, 가시화 변수로 Pressure(p), Mesh Regions을 “internalMesh”로 선택하면 압력분포가 그림 22처럼 나타난다.

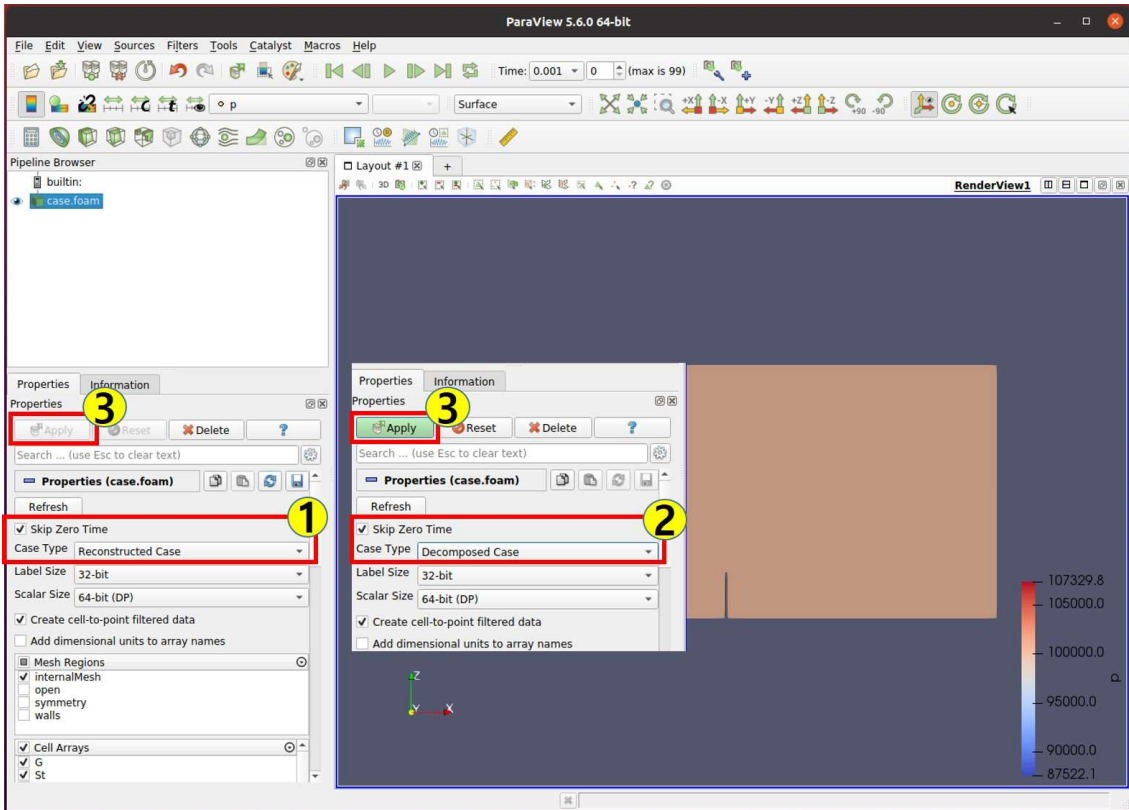


그림 21. Paraview를 사용하여 해석결과 불러오기 화면

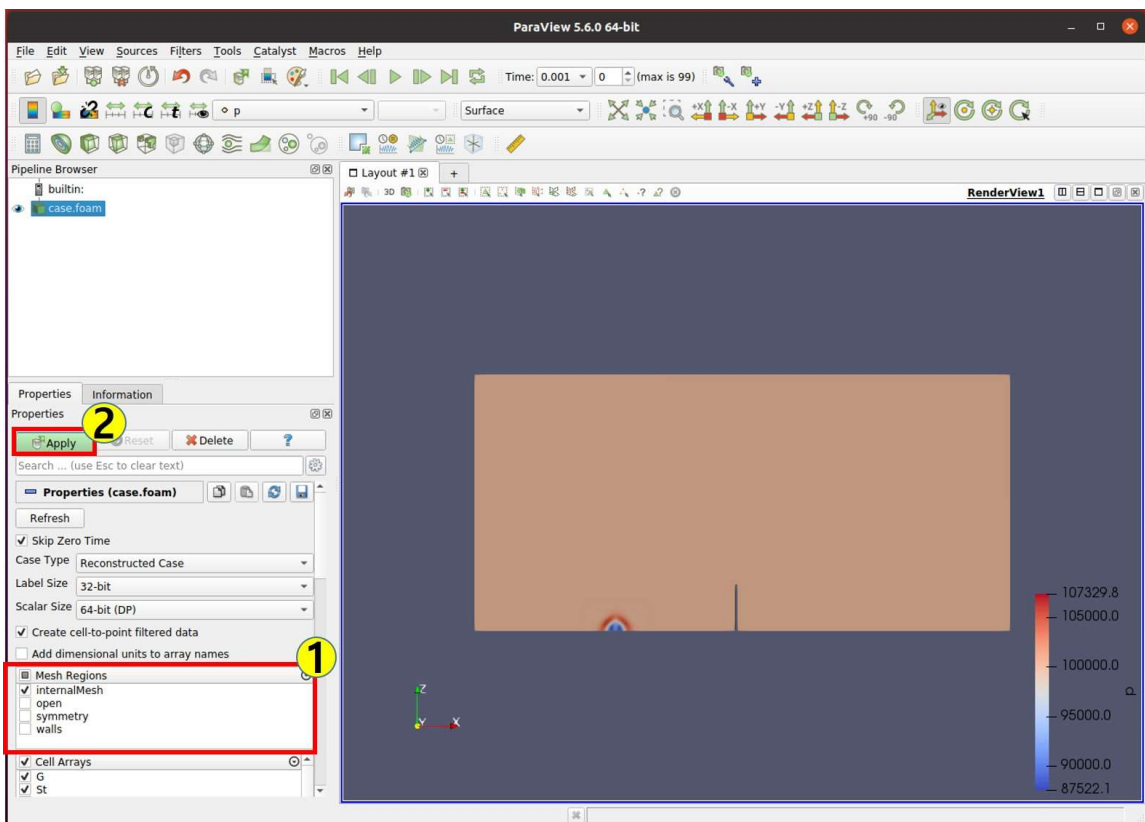


그림 21. Paraview를 사용하여 해석결과 압력분포 가시화 화면

○ radXiFoam 해석결과 가시화 및 해석시간 선정방법

ParaView 초기화면에서 radXiFoam 해석결과 중에서 가시화 하고자는 하는 변수 (field)를 변경하거나, 천이과정 해석결과 중에서 특정 시간을 먼저 가시화 할 때는 그림 23의 적색 박스 내의 아이콘을 누른 후에 선택하면 된다.

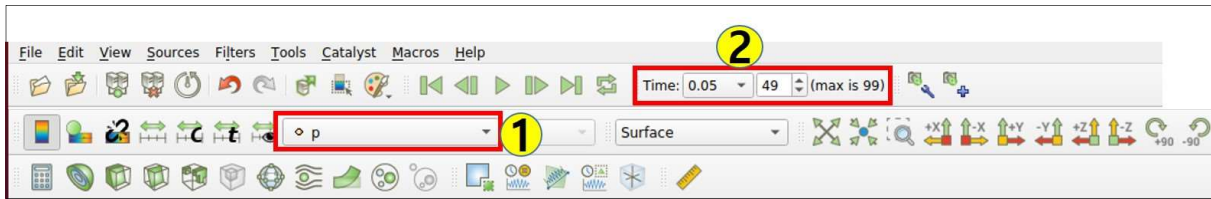


그림 23. Paraview 가시화용 변수 및 해석시간 선택 화면

○ ParaView Layout 화면 조작방법

그림 21에서 압력분포가 가시화된 Layout 화면을 조작하는 방법은 아래와 같고, 더 상세한 방법은 참고문헌[21]에 기술되어 있다.

- 화면 확대 및 축소: 마우스 스크롤 휠 사용하여 조작한다.
- 화면 회전: 마우스 왼쪽 버튼을 누른 후 회전한다.
- 화면 수평이동: 키보드 왼쪽의 shift key와 마우스 오른쪽 버튼을 동시에 누른 후 이동한다.
- 화면 수평회전: 키보드 왼쪽의 shift key와 마우스 왼쪽 버튼을 동시에 누른 후 회전한다.

5-3. ParaView 주요 Filter 사용법

ParaView에는 해석결과의 가시화 작업을 도와주는 많은 Filter가 그림 24와 같이 있는데, 이 중에서 주로 사용되는 Filter를 소개하고자 한다.

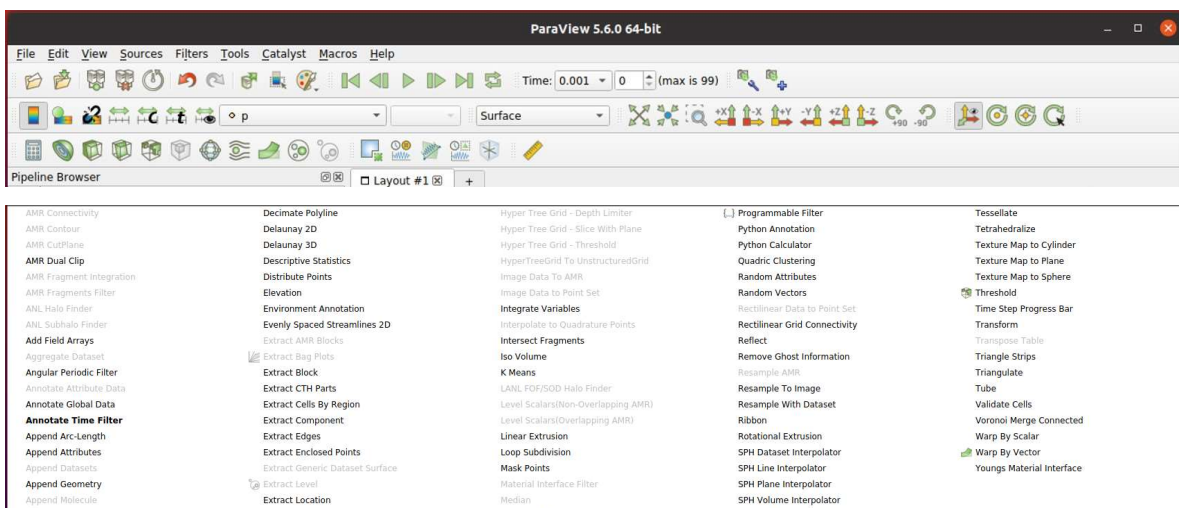


그림 23. Paraview 가시화용 Filter 선택 화면

5-3-1. Clip 및 Slice

사용자가 해석결과의 전체 영역 중에서 특정한 부분을 가시화할 때 사용하는 Filter가 Clip 및 Slice 이며, 그림 24와 같이 ParaView 초기 화면에서 Clip 또는 Slice Filter를 선택한 후, 필요한 좌표면 데이터를 입력하면 특정 부분만 가시화할 수 있다.

○ Clip

Clip은 전체 해석영역을 특정한 위치에서의 Plane으로 절단한 후, 남은 부분을 가시화해주는 Filter 이다.

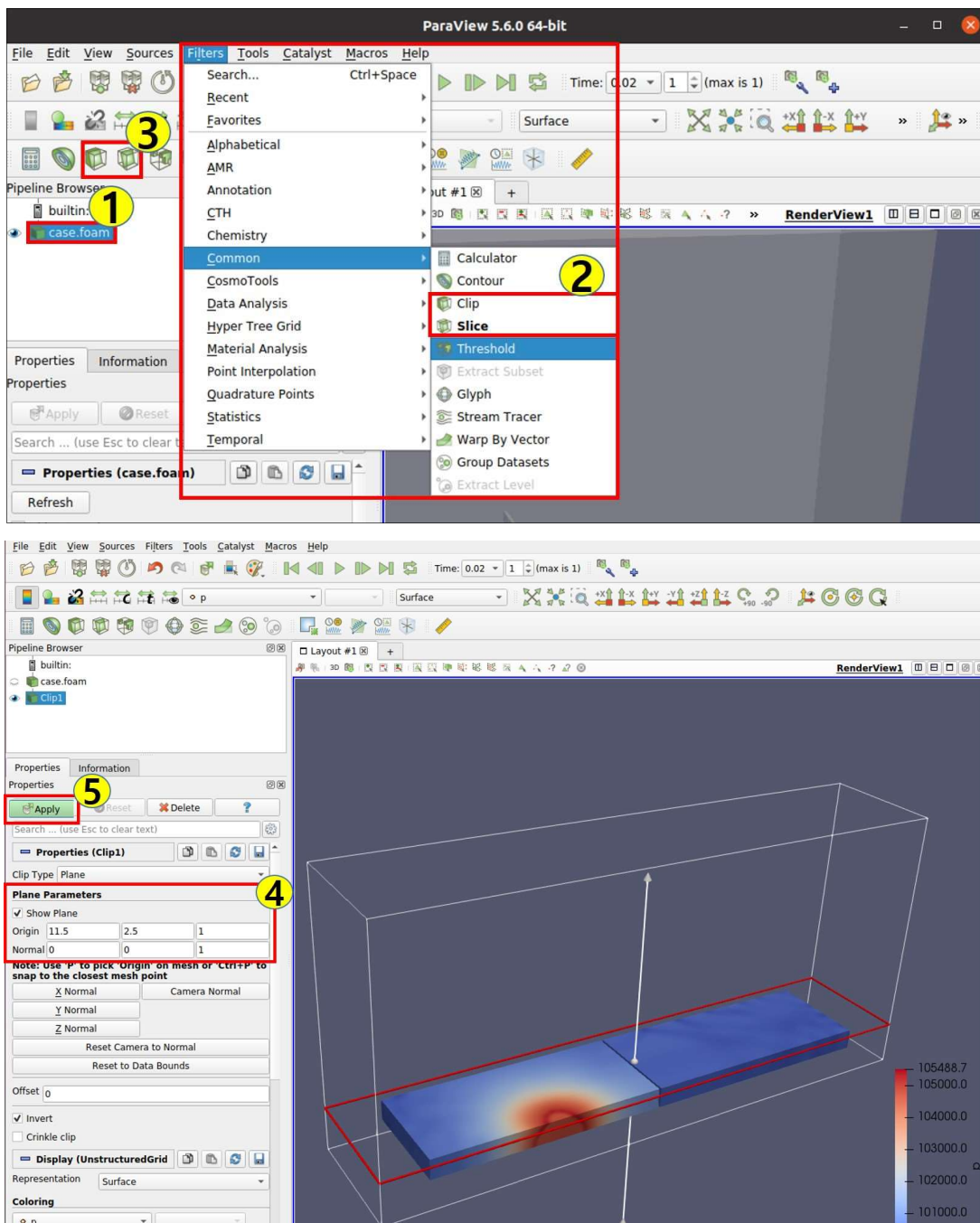


그림 24. Clip 및 Slice Filters 사용방법

○ Slice

Slice는 전체 해석영역 중에서 특정한 위치에서의 Plane으로 절단 후, 그 Plane만 가시화해주는 Filter 이다. 상세한 사용절차는 아래와 같다.

- Filter를 적용할 해석결과 파일(case.foam)을 선택한다.
- 마우스로 화면메뉴 Filters / Common으로 이동한다.
- 마우스로 Clip 또는 Slice를 선택한다.
- 절단면 Plane을 정의할 수 있는 좌표계 원점과 수직방향 벡터를 입력한다.
- Apply 버튼을 누른다.

5-3-2. Threshold

사용자가 해석결과 중에서 특정 변수를 가시화할 때, 그 변수의 전체 수치범위 중에서 특정 범위만 가시화할 때 사용하는 Filter 이다. 구체적인 사용절차는 그림 25와 같이 Clip/Slice 사용법과 거의 유사하고, 추가로 변수의 최소 및 최대 값을 지정해야 한다.

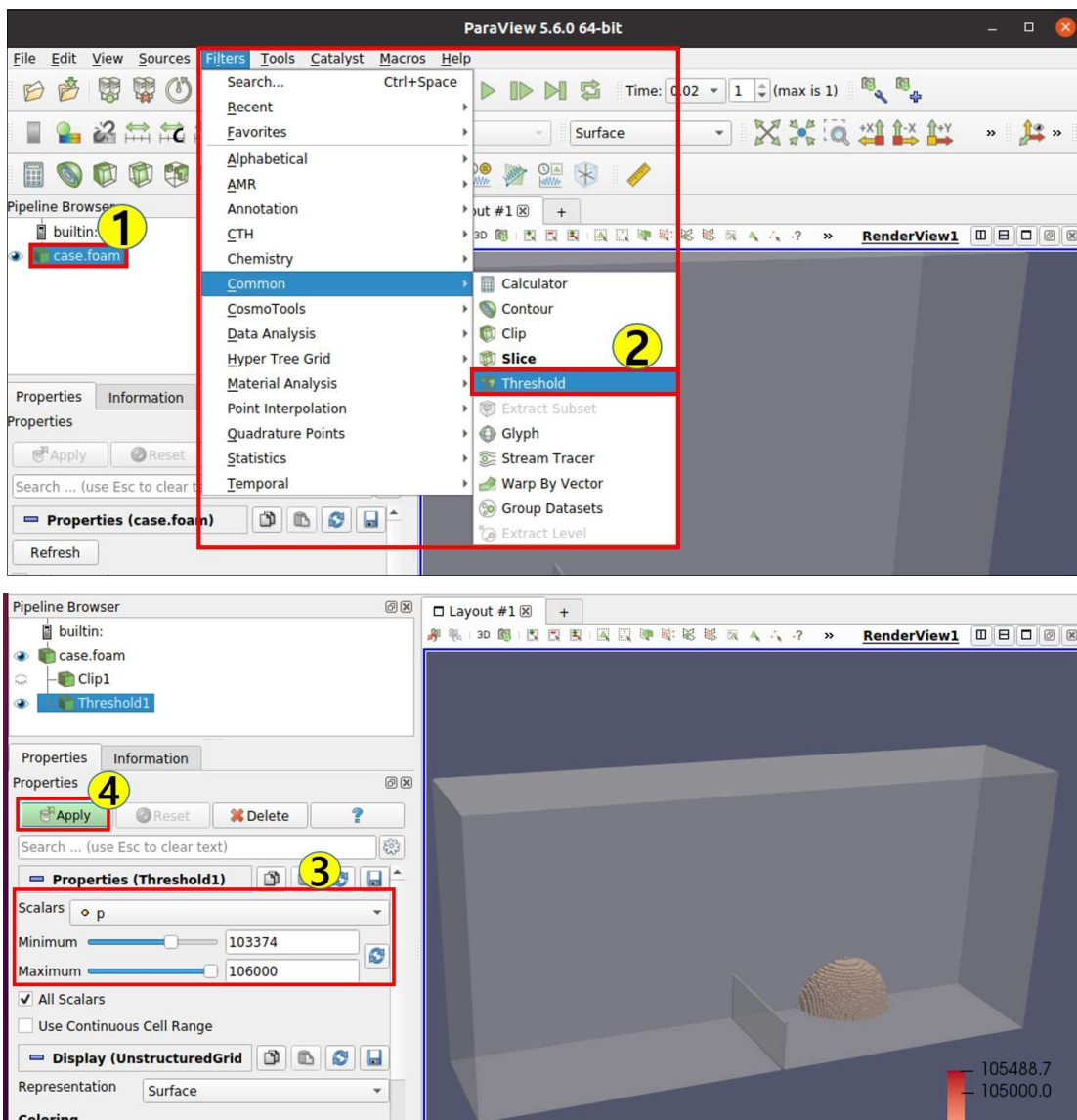


그림 25. Threshold Filter 사용방법

5-3-3. Annotate Time Filter

천이과정으로 계산이 진행된 해석결과를 가시화할 때, Layout 화면에 해석결과에 해당하는 계산시간을 표시해주는 Filter 이며, 사용법은 그림 26과 같다. 사용 시 주의할 점은 시간표시 형식과 Scale을 지정해야 하며, Scale이 1이면 시간단위가 [s]이고, Scale이 1000이면 [ms] 이다. 더 자세한 사용법은 참고문헌[21]에 기술되어 있다.

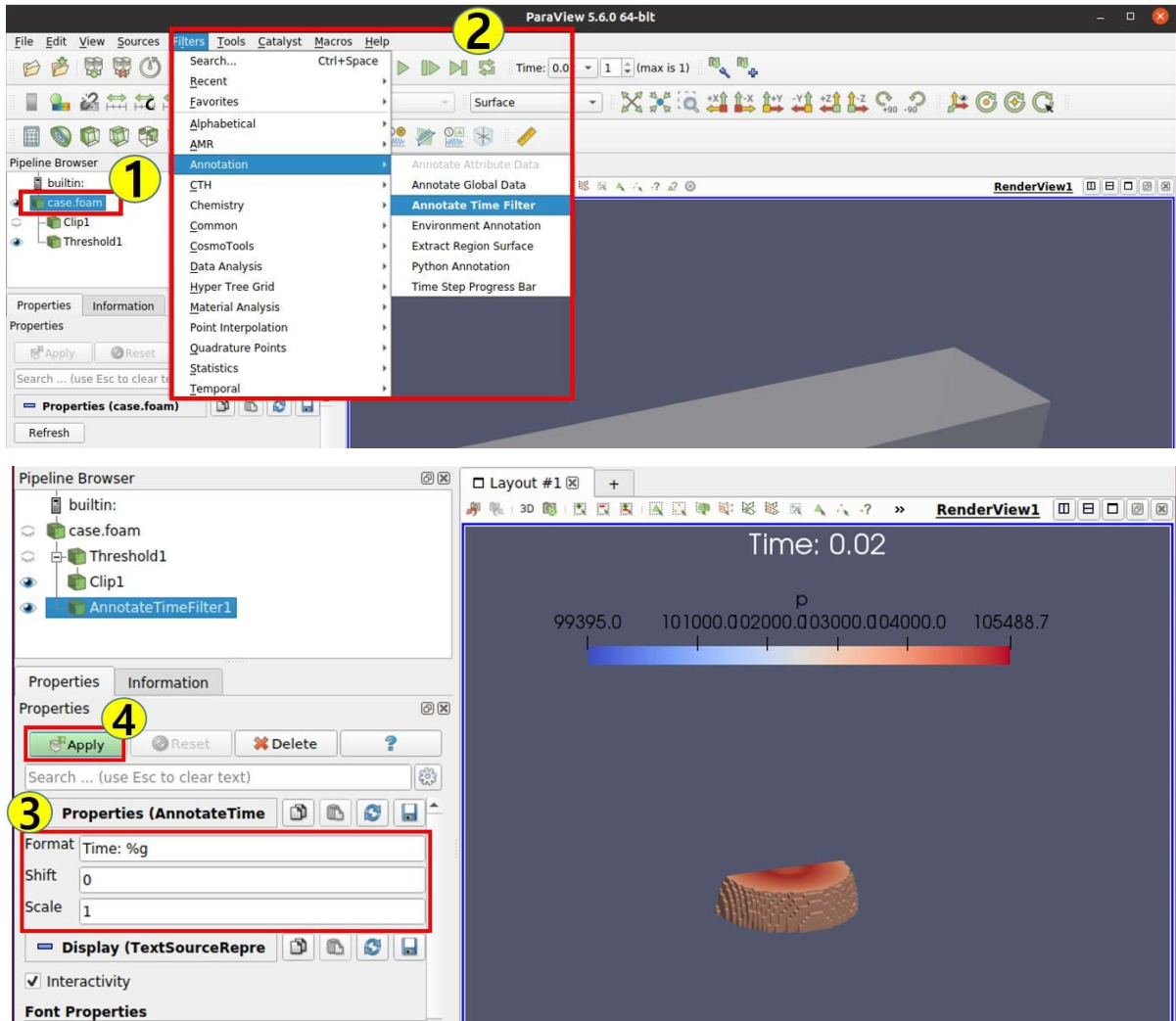


그림 26. Annotate Time Filter 사용방법

5-3-4. Plot Over Line

사용자가 해석결과물의 특정한 두 지점 사이에서 계산된 변수 값의 분포를 알고 싶을 때, 두지점 사이의 직선을 정의한 후 그 직선을 따라서 변수 값을 그래프로 그리고 해당 데이터를 따로 저장할 수 있도록 해주는 Filter 이다.

○ Plot Over Line 사용방법

2차원 그래프 그리기, 데이터 저장방법의 구체적인 사용 절차는 아래와 같다.

- Plot Over Line Filter 적용하여 해석결과를 가시화할 case.foam를 선택한다.
- Plot Over Line을 적용할 Mesh Regions(예: walls)을 선택한다.
- 마우스를 사용하여 “Filters / Data Analysis / Plot Over Line” 을 선택한다.
- 두 지점의 각 좌표계를 입력한다.
- Apply 버튼을 클릭한다.
- 2차원 그래프로 가시화할 변수를 선택한다.
- Apply 버튼을 클릭한다.
- 2차원 그래프가 그려진 “LineChartView1” 선택한다.
- 마우스로 File / Save Data 또는 Save Screenshot를 저장한다.

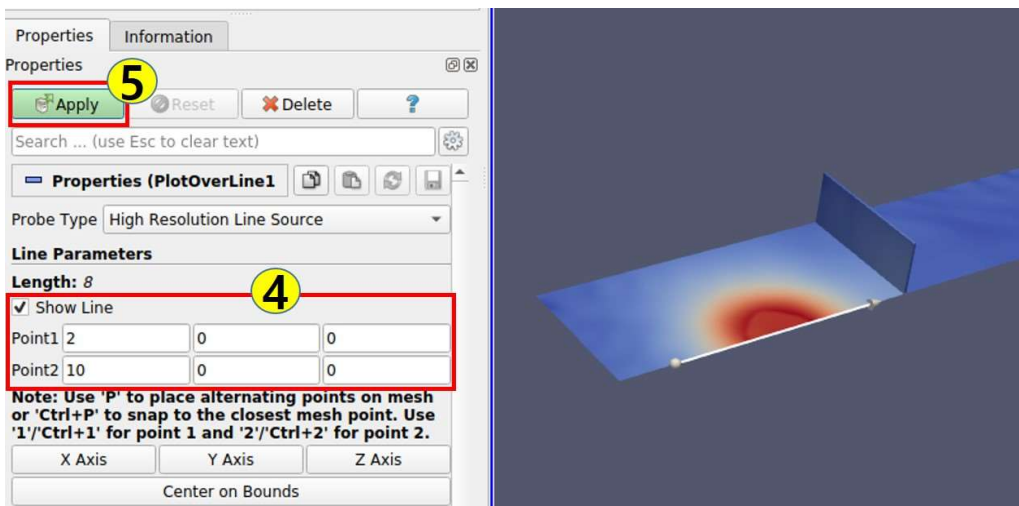
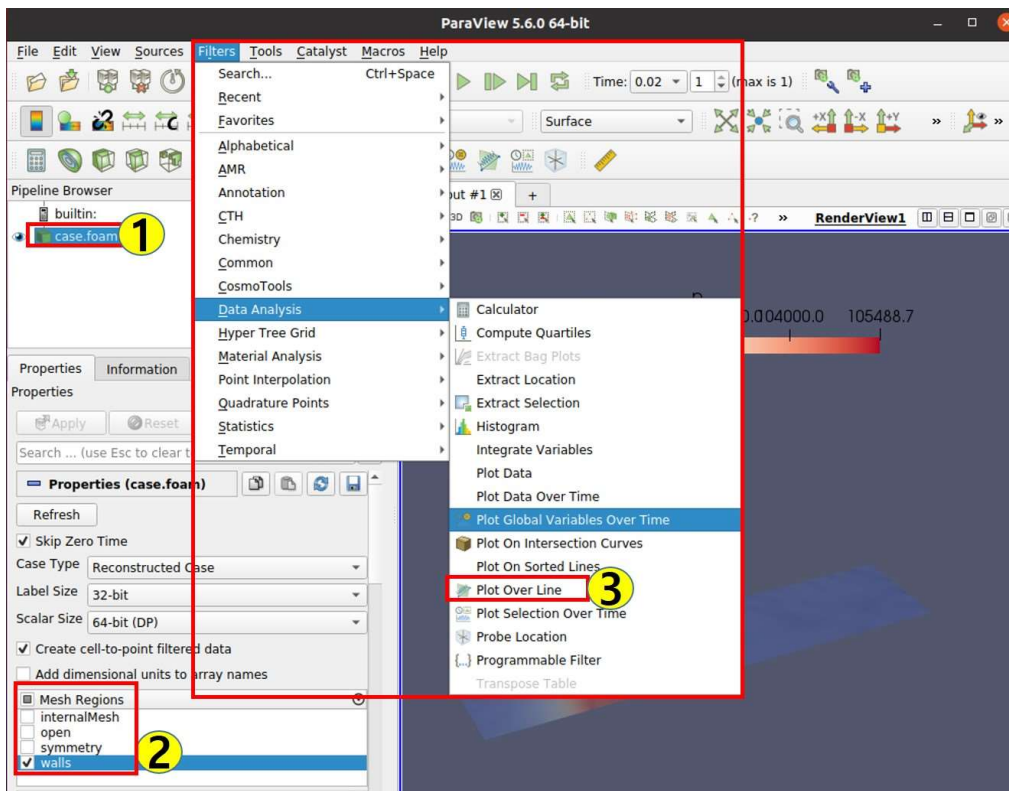


그림 26. Plot Over Line Filter 설정방법

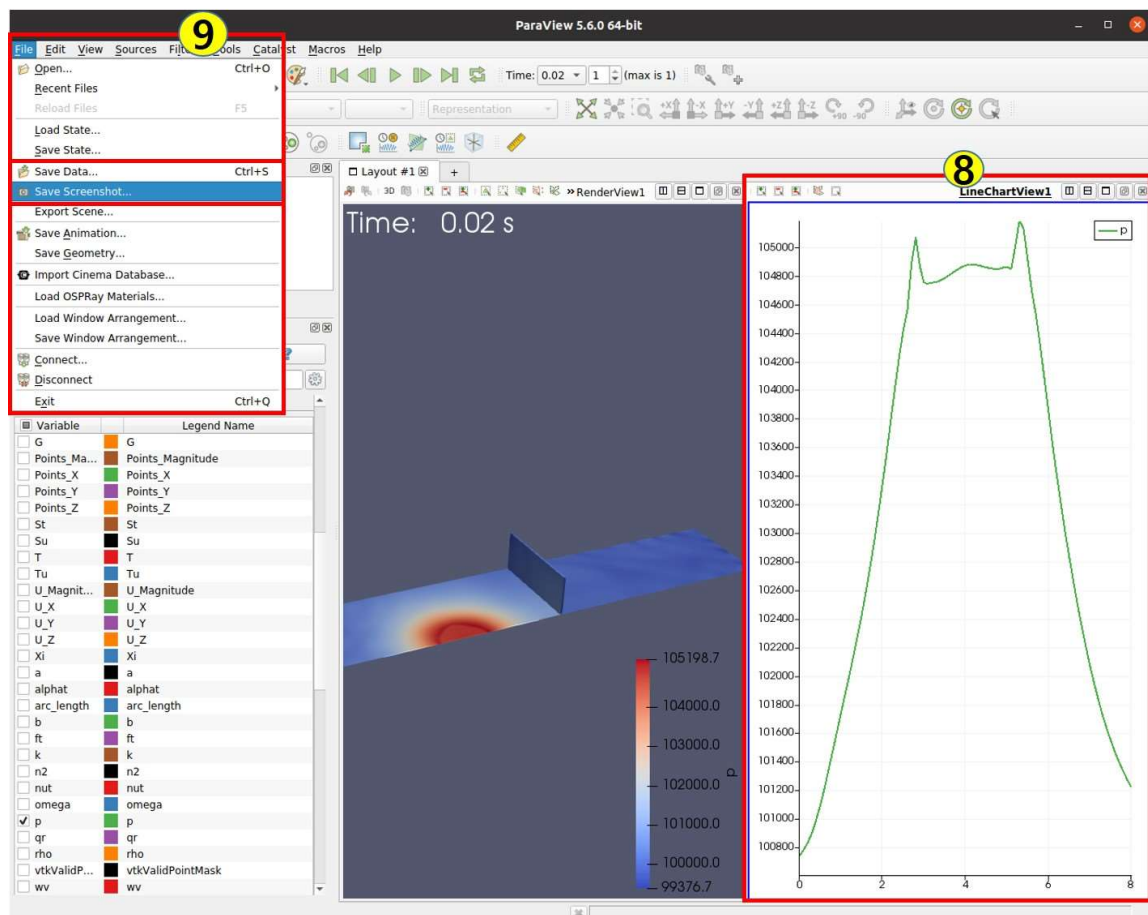
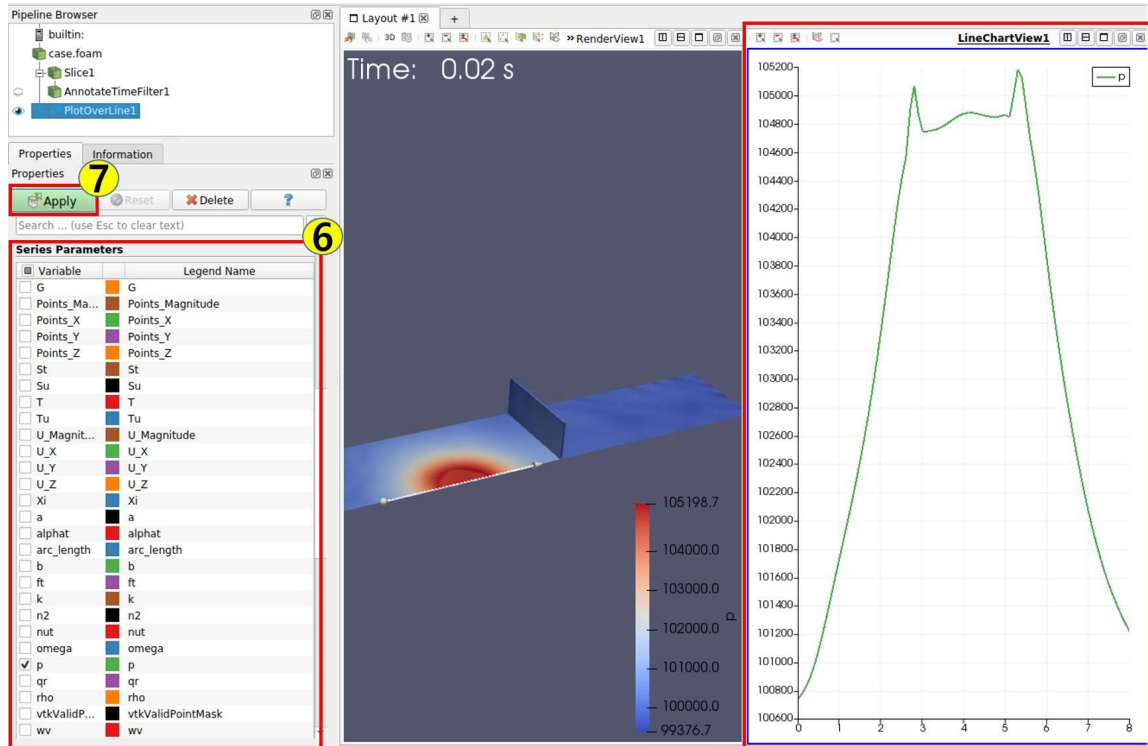


그림 27. 2차원 그래프 및 데이터 저장방법

6. 참고문헌

1. 김상민, 강형석, 최근상, “수소증기운 폭발사고 해석코드(radXiFoam) 사용자 매뉴얼 v1.0,” 기술보고서, KAERI/UM-55/2023, 한국원자력연구원, 2023.
2. L. C. Shirvill et al., “Safety Studies on High-Pressure Vehicle Refueling Stations: Releases into a Simulated High-Pressure Dispensing Area “, *Int. J. Hydrogen Energy* 2012, **37**, 6949–6964.
3. H. S. Kang et al., “CFD Analysis of the Effects of a Barrier in Hydrogen Refueling Station Mock-Up Facility during a Vapor Cloud Explosion Using the radXiFoam v2.0 Code” , *Processes*, 2024, **11**, 3025.
4. EDF-CEA, SALOME, Internet:salome-platform.org
5. ESI, OpenFOAM-v2112, Internet:www.openfoam.com
6. Canonical Ltd, Ubuntu 20.04 LTS, Internet:release.ubuntu.com/2024
7. M. A. Groethe and J.D. Colton, “FY01 Annual Report on Hydrogen Safety in the World Energy Network,” Stanford Research Institute, SRI Project 10367, 2002.
8. H. S. Kang et al, “Safety Issues of a Hydrogen Refueling Station and a Prediction for an Overpressure Reduction by a Barrier Using OpenFOAM Software for an SRI Explosion Test in an Open Space” , *Energies*, 2022, **15**, 7556.
9. H. S. Kang et al, A Database Extension for a Safety Evaluation of a Hydrogen Refueling Station with a Barrier Using a CFD Analysis and a Machine Learning Method” , *Processes*, 2023, **11**, 3025.

7. 부록

7-1. radXiFoam v2.0 계산 순서도

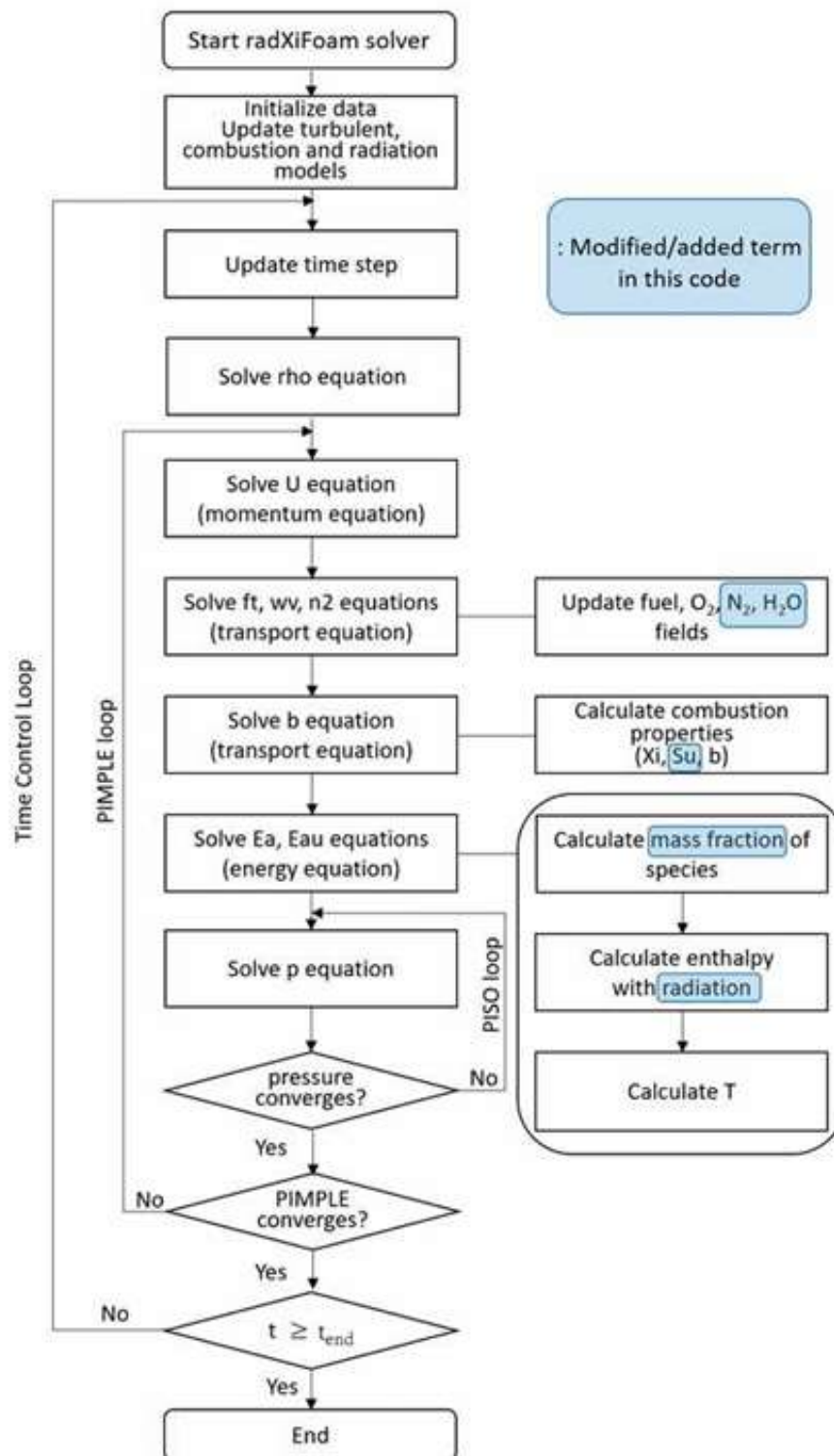


그림 1. radXiFoam v2.0 계산 순서도

7-2. alphaBetaCalc.py

```
# Order: H2, O2, N2, H2O
H2_vf = float(input("H2 volume fraction: "))
H2O_vf = float(input("H2O volume fraction: "))
O2_vf = (1-(H2_vf+H2O_vf))*0.21
N2_vf = (1-(H2_vf+H2O_vf))*0.79
mm = [2, 32, 28, 18]
vf = [H2_vf, O2_vf, N2_vf, H2O_vf]
mf = []
mTot = 0
for i in range(4):
    mTot = mTot + (mm[i] * vf[i])
for i in range(4):
    mf.append((mm[i] * vf[i]) / mTot)
H2_ratio = mf[0] / mf[1]
H2_eq_ratio = H2_ratio / 0.125 # 0.125 = stoichiometry H2/O2 mass ratio
H2_alpha = 2.18 - (0.8 * (H2_eq_ratio - 1))
H2_beta = -0.16 + 0.22 * (H2_eq_ratio - 1)
print("=====")
print("H2 mass fraction: ", mf[0])
print("H2O mass fraction: ", mf[3])
print("N2 mass fraction: ", mf[2])
print("equivalence ratio: ", H2_eq_ratio)
print("H2 alpha: ", H2_alpha)
print("H2 beta: ", H2_beta)
print("=====")
```

7-3. overPressurePlot.py

```
# Import modules
import pandas as pd
import matplotlib.pyplot as plt

# Convert original file to dataframe
df_orig = pd.read_csv('p')
df_temp = df_orig[9:]
cols = list(df_temp.columns)
df_temp.rename(columns = {cols[0]:'text'}, inplace = True)
df_temp2 = pd.DataFrame(df_temp.text.str.split('\s+',9).tolist(), columns = ['Null', 'Time', 'Probe 0',
'Probe 1', 'Probe 2', 'Probe 3', 'Probe 4', 'Probe 5', 'Probe 6', 'Probe 7'])
df_temp3 = df_temp2.loc[:,['Time', 'Probe 0', 'Probe 1', 'Probe 2', 'Probe 3', 'Probe 4', 'Probe 5',
'Probe 6', 'Probe 7']]
df_temp3 = df_temp3.astype('float')
df_abs_pres = df_temp3.loc[:,['Probe 0', 'Probe 1', 'Probe 2', 'Probe 3', 'Probe 4', 'Probe 5', 'Probe
6', 'Probe 7']]
# Convert absolute pressure dataframe to gauge pressure dataframe
df_gauge_pres = (df_abs_pres - 101325)/1000

df_time = df_temp3.loc[:, ['Time']]

df_final = df_time.join(df_gauge_pres)

# Export .xlsx file
with pd.ExcelWriter('p_modified.xlsx') as writer:
    df_final.to_excel(writer)

# Export graph (.png) files
for i in range(8):
    df_final.plot(x='Time', y='Probe {}'.format(i))
    plt.xlabel('Time [s]')
    plt.ylabel('Overpressure [kPa]')
    plt.ylim(-8.0,8.0)
    plt.grid(True, linestyle='--')
    plt.savefig('Probe_{}.png'.format(i))
```

7-4. radXiFoam v2.0 설치방법

- 다운로드한 압축파일을 풀고 디렉토리 위치를 확인한 후 아래 그림 1과 같이 컴파일한다.
- 1. Terminal 실행 후, OpenFOAM-v2112 환경설정을 위한 명령어 입력 (ofv2112).
- 2. SRC 폴더 내 라이브러리 및 솔버 폴더로 이동 (터미널 창에 cd “폴더명” 입력).
- 3. wmake 입력 및 실행.
- 4. SRC 폴더 내 모든 폴더에서 2, 3 반복 (단, radXiFoam은 가장 나중에 컴파일 해야함).

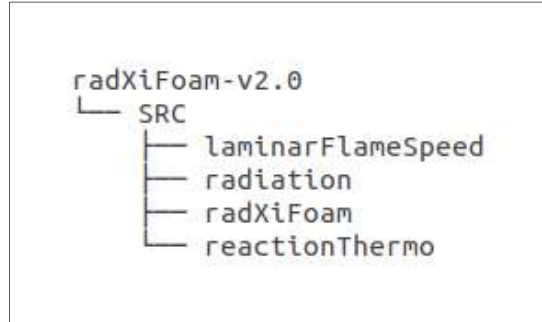


그림 1. 설치용 압축파일 구성내용 및 컴파일 명령어

- 컴파일이 완료되면 그림 2와 같이 컴파일된 파일들의 생성 위치가 표시된다.



그림 2. 라이브러리 및 솔버 컴파일 완료 메시지

7-5. HSL 수소충전소 모형 실험장치 CAD 방식 격자모델링 기법

- 본 매뉴얼에서는 영국 Health and Safety Laboratory(HSL)에서 수행한 수소충전소 모형 실험장치(Mock-Up Facility)에서 수행한 수소증기운 폭발실험[1]을 해석하기 위해서 blockMesh Utility Software로 생성한 격자모델[2]을 CAD 방식인 SALOME 프로그램[3]으로 생성하는 방법을 설명하고자 한다. HSL의 수소충전소 모형 실험장치는 그림 1과 같고, 이 장치를 모의하는 격자모델을 blockMesh로 생성한 것은 그림 2에 나타나 있다. 그 격자모델은 수소-공기 혼합기체가 위치한 Tent 바닥의 두 dispenser 사이 중앙지점에 있는 전기스파크 설치지점을 기준으로 1/2 대칭 조건이 적용되어 생성되었다.



그림 1. 영국 HSL의 수소충전소 모형 실험장치

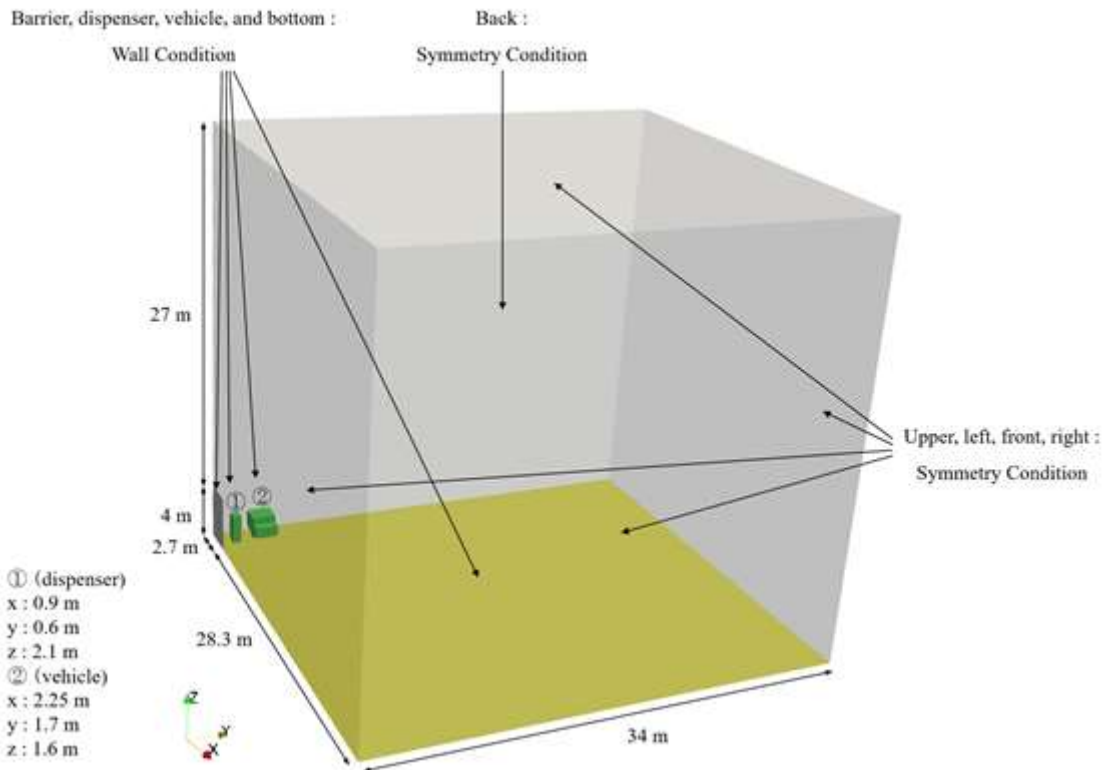


그림 2. BlockMesh Utility Software로 생성한 격자모델

1. 형상 생성

형상 생성 과정에서는 계산이 이루어지는 영역을 생성하기 위해 구조물(디스펜서, 차량, 방호벽)과 계산이 이루어지는 영역을 포함하는 직육면체 영역을 각각 생성한 후 전체 직육면체 영역에서 구조물이 차지하는 체적을 제거하여 계산이 이루어지는 영역의 형상을 생성할 것이다.

가. 구조물 생성

(1) 디스펜서

도형을 생성하기 위해 SALOME를 실행한 후 그림 3의 ①과 같이 “Shaper”를 실행한다. 디스펜서는 xy 평면에 사각형을 생성하여 z 방향으로 늘여 생성할 것이다. “Shaper” 내 메뉴에서 ②와 같이 “Sketch”를 선택한다.



그림 3. Shaper 및 Sketch 실행 화면

“Sketch”를 실행하면 어느 평면에 스케치할 것인지 선택하는 화면으로 바뀐다. 그림 4와 같이 좌측 메뉴의 “construction”에서 선택하거나 3D 도면 화면에서 선택할 수 있다. 여기서는 XOY 평면을 선택한다.

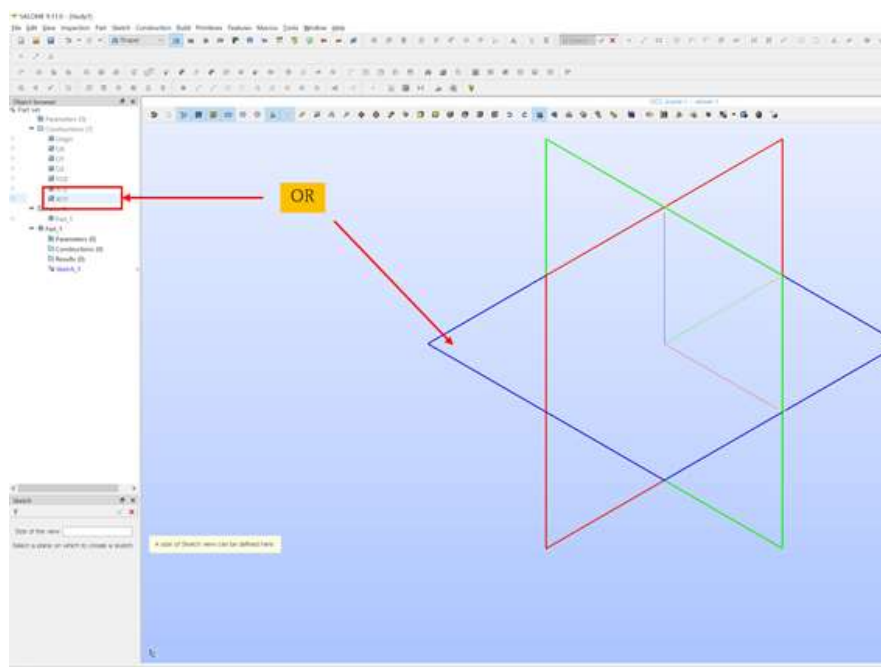


그림 4. Sketch 작업을 수행할 평면 선택 방법

가. 구조물 생성, (1) 디스펜서 (계속)

다음 단계는 사각형을 그리고 제약조건을 모두 정의하는 과정이다. 그림 5의 ①과 같이 직사각형을 선택하여 평면 상에 직사각형을 그린다. 그리고 ②의 “Length”를 선택하여 사각형의 수직한 두 변의 길이를 정의한다. ③의 “Parallel”을 선택하여 x 방향 또는 y방향의 변을 선택한 뒤 x축 또는 y축과 평행 제약조건을 설정한다. ④의 “Vertical distance”를 선택하여 원점으로부터의 수직(y 방향) 거리를 정의한다. ⑤의 “Horizontal distance”를 선택하여 원점으로부터의 수평(x 방향) 거리를 정의한다. ① ~ ⑤의 과정을 거치면 빨간색이었던 사각형의 네 변이 초록색으로 변한다. 이것은 가능한 모든 제약조건이 정의되었음을 의미한다. 제약조건이 모두 정의되었으면 그림 6과 같이 왼쪽 메뉴에서 초록색 V 표시를 선택하여 스케치를 마친다.

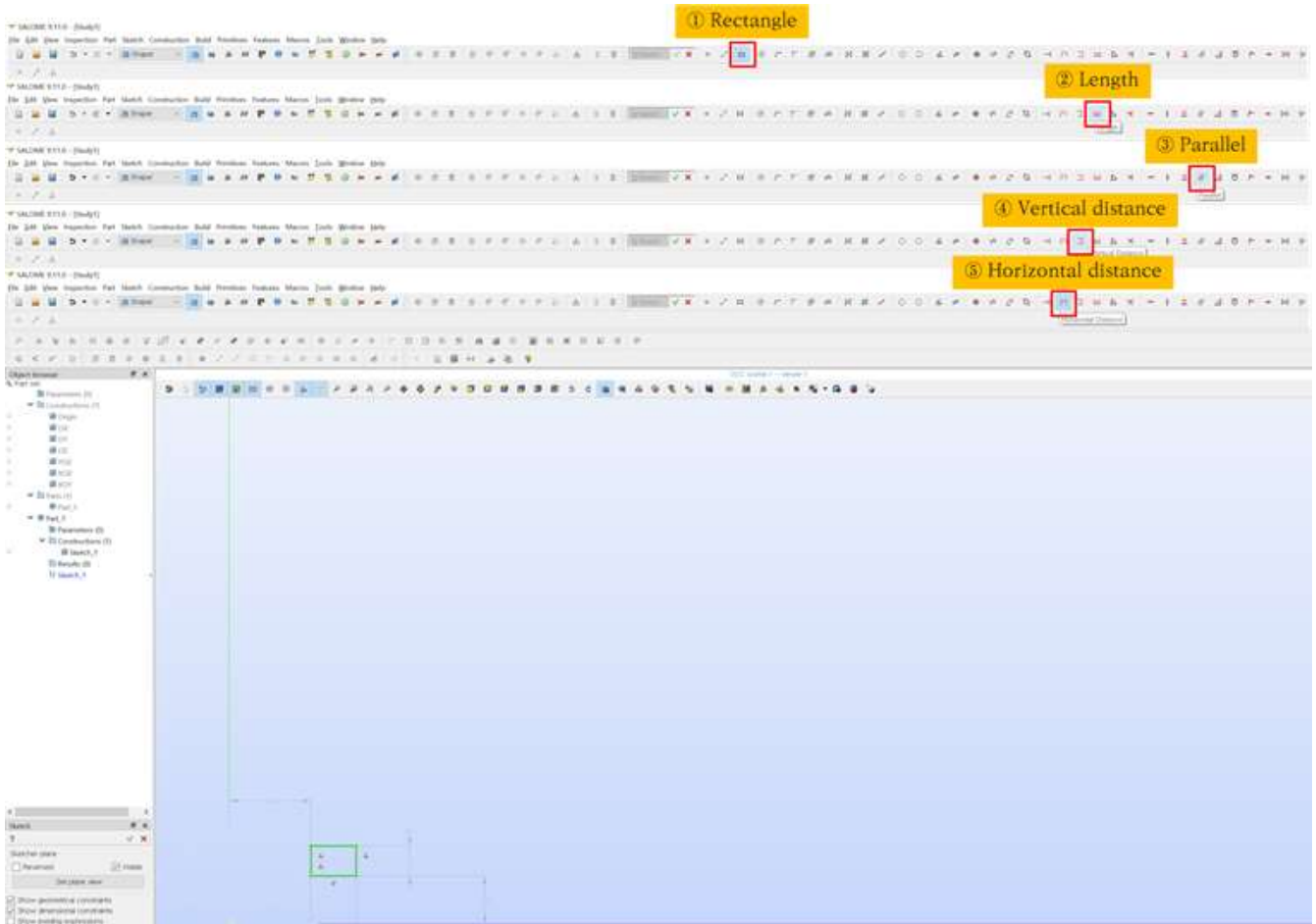


그림 5. SALOME에서 직사각형을 생성하는 방법

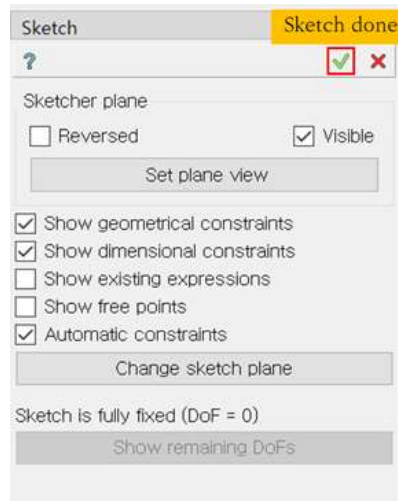


그림 6. Sketch 작업 완료 화면

가. 구조물 생성, (1) 디스펜서 (계속)

스케치를 완료하면 완료된 스케치를 기반으로 면을 생성해야 한다. 그림 7의 ①과 같이 “Face”를 선택한 후 방금 생성한 스케치를 왼쪽 메뉴 혹은 3D 도면상에서 선택한다. 그 뒤 직육면체 형태의 디스펜서를 생성하기 위해 그림 8의 ①과 같이 “Extrusion”을 선택한 후 앞서 생성한 면을 왼쪽 메뉴 혹은 3D 도면상에서 선택한다. 그 결과로 직육면체 형태의 디스펜서가 생성된다.

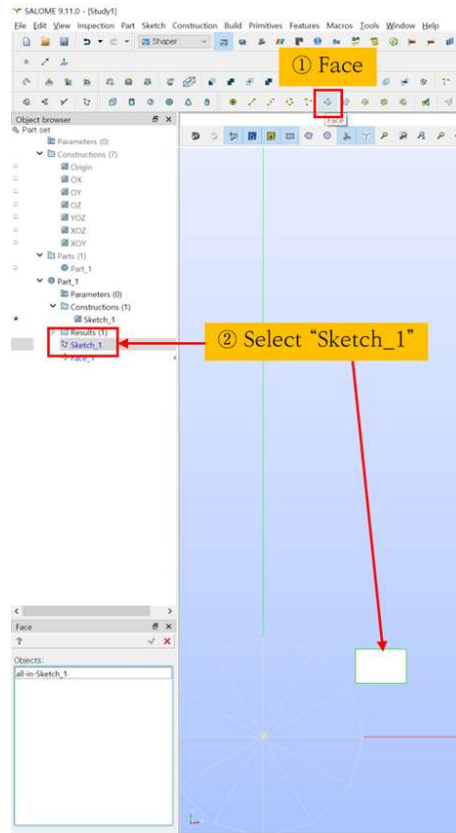


그림 7. Face 생성 방법

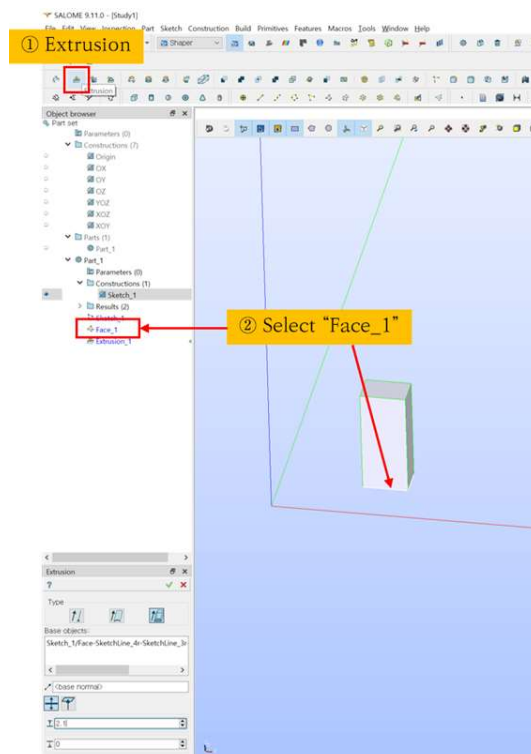


그림 8. 직육면체 생성 방법

가. 구조물 생성
 (2) 차량 - 바퀴

격자모델에서 차량(Vehicle)은 엔진룸, 승객석, 바퀴로 이루어져 있다. 먼저 두 바퀴를 생성할 것이다. (1) 디스펜서와 같이 스케치를 통해 사각형을 생성한다. 이때 바퀴의 경우 y 방향으로 두 개가 놓이기 때문에 이번에는 xz 평면에서 y 방향으로 늘이는 방식으로 생성할 것이다. 따라서 xz 평면에 사각형 스케치를 생성한다. 그 결과 xz 평면(y = 0)상에 사각형이 생성된다. 이를 (1)과 같이 면을 생성하고, 원하는 위치로 이동시켜야 한다. 면을 이동시키기 위해 그림 9의 ①과 같이 “Translation”을 선택하고 “Main objects”에서 ②와 같이 방금 생성한 면을 선택한다. 그 뒤 그림 10과 같이 방향 벡터 OY를 선택하고 이동할 거리를 입력한다. 그리고 (1)과 같이 y 방향으로 확장한다(Extrusion).

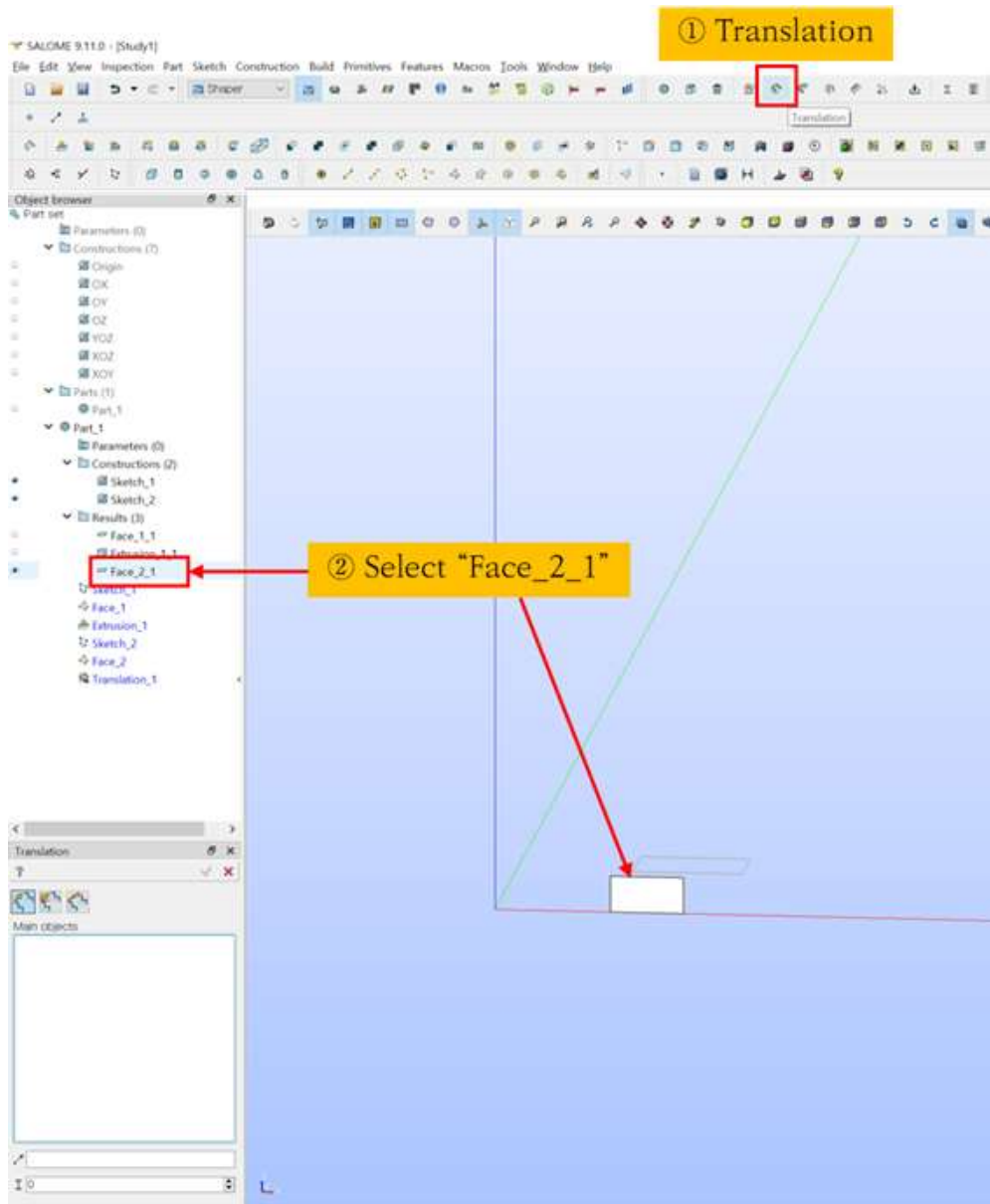


그림 9. Face 생성 및 이동

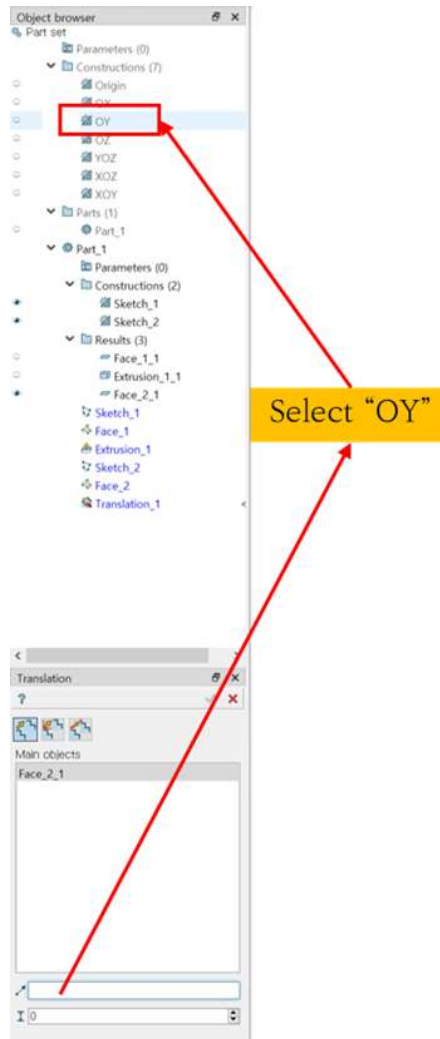


그림 10. Transition 실행 방법

본 격자 모델에서는 바퀴 부분을 하나의 연속된 직육면체로 정의하지 않고 두 바퀴를 따로 정의했으므로, y 방향으로 똑같은 형태의 바퀴가 두 개 있어야 한다. 앞서 생성하여 이동 작업을 수행한 바퀴를 “Linear copy” 기능을 이용하여 y 방향으로 복사한다.



그림 11. Linear copy 버튼 위치

(2) 차량 - 본체

앞선 바퀴 생성 과정과 마찬가지로 xz 평면에 모의할 차량 본체를 스케치한 후 y 방향으로 늘이는 방식으로 형상을 제작할 것이다. 사용하는 기능은 아래 그림 12와 같이 순서대로 “Sketch”, “Face”, “Translation”, “Extrusion”이다.

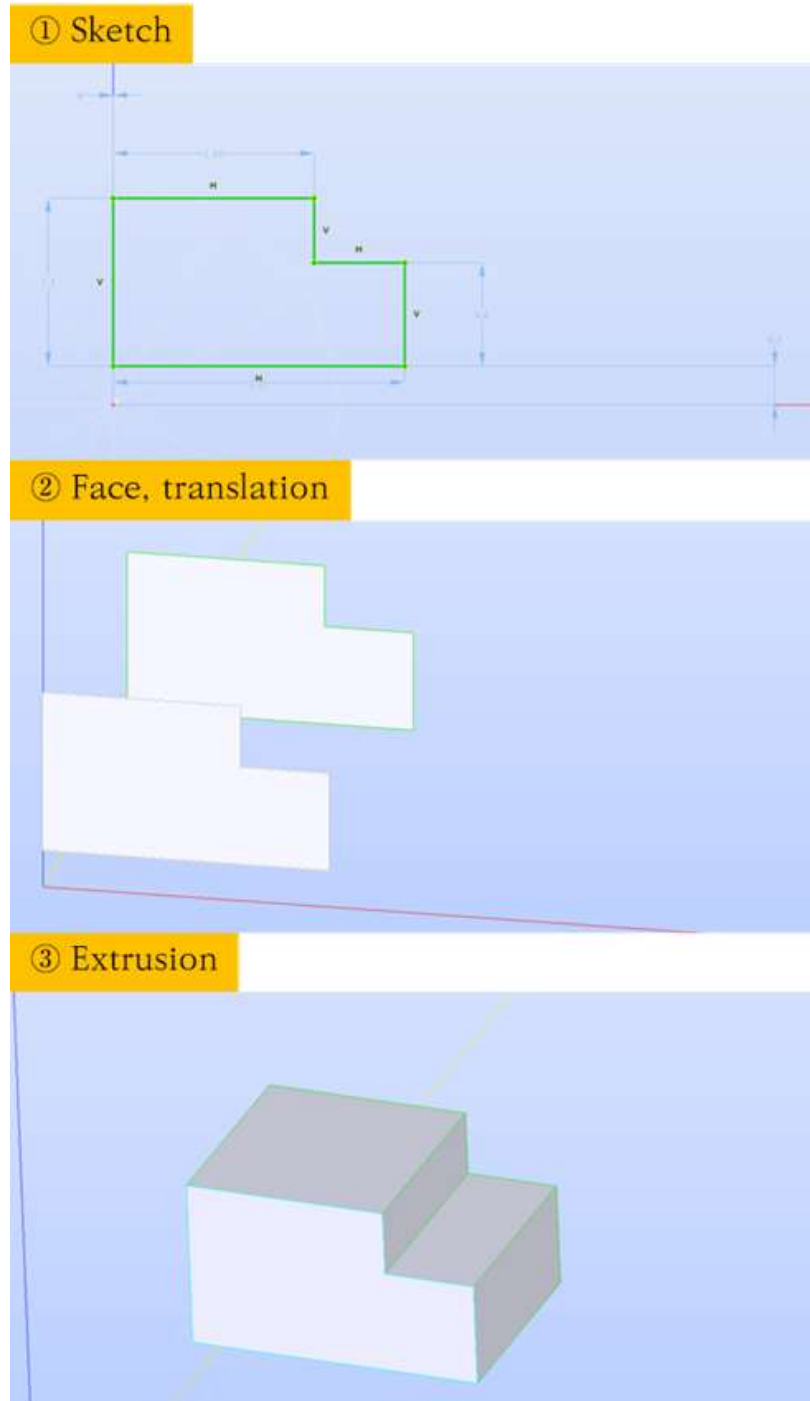


그림 12. 차량 본체 생성 과정

(2) 차량 - 방호벽

방호벽은 (1), (2)와 마찬가지로 그림 13과 같이 xz 평면에 스케치 및 면을 생성한 후 원하는 y 방향으로 이동한다.

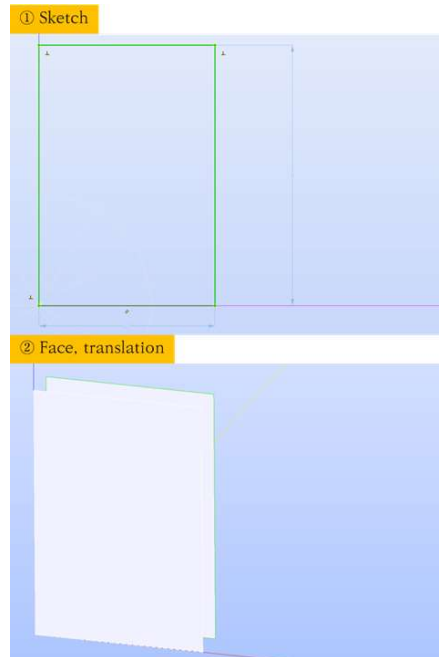


그림 13. 방호벽 생성 과정

나. 계산영역 생성

먼저 전체 영역을 포괄하는 직육면체 형상을 생성한다. 앞의 “가”에 기술된 내용들과 같이 xz 평면에 사각형 스케치 및 면을 생성한 후 y 방향으로 늘여 직육면체를 생성한다. 과정은 그림 14와 같다. 형상을 생성한 뒤 그림 15와 같이 각 형상별 이름(Name)을 지정하면 혼동을 막을 수 있다.

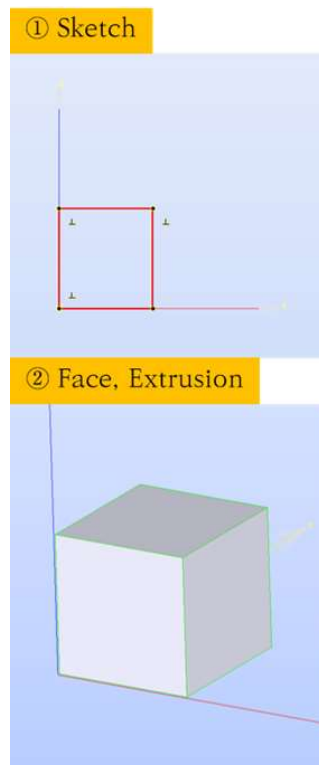


그림 14. 직육면체 형상 생성

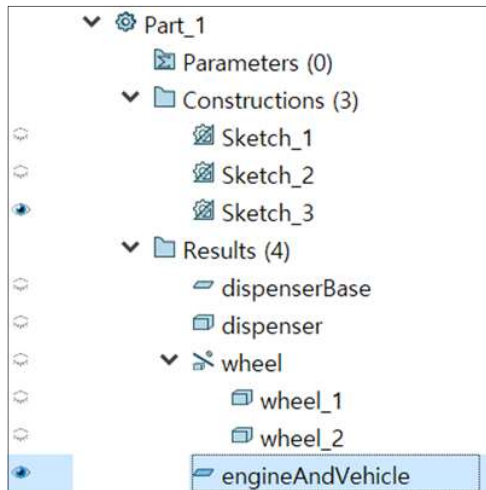


그림 15. 생성한 형상의 이름 지정

다음 단계로 방호벽(barrier)를 주 계산 영역에서 분리하는 작업을 한다. 방호벽은 나중에 계산 결과를 후처리 및 시각화할 때 명확히 분리되어 있어야 하므로 이 작업은 필수적이다. 그림 16의 ①과 같이 Split을 선택하여 ②와 같이 “Main objects”에서 “mainDomain”을 선택하고 “Tool objects”에서 “barrier”를 선택한다. 그리고 디스펜서와 차량 형상을 주 계산 영역에서 제외하는 작업을 한다. 앞선 “Split”과는 달리, “Cut” 기능을 이용해야 하는데, 그 이유는 디스펜서와 차량의 내부 공간은 CFD 계산이 이루어지는 유동 공간이 아니기 때문이고, 따라서 해당 구조물들의 외부 경계면들만 계산 영역에 남아있어야 한다. 그림 17의 ①과 같이 “Cut”을 선택하여 ②와 같이 “Main objects”에 “mainDomain”을 넣고, ③과 같이 “Tool objects”에 구조물들을 넣고 실행한다.

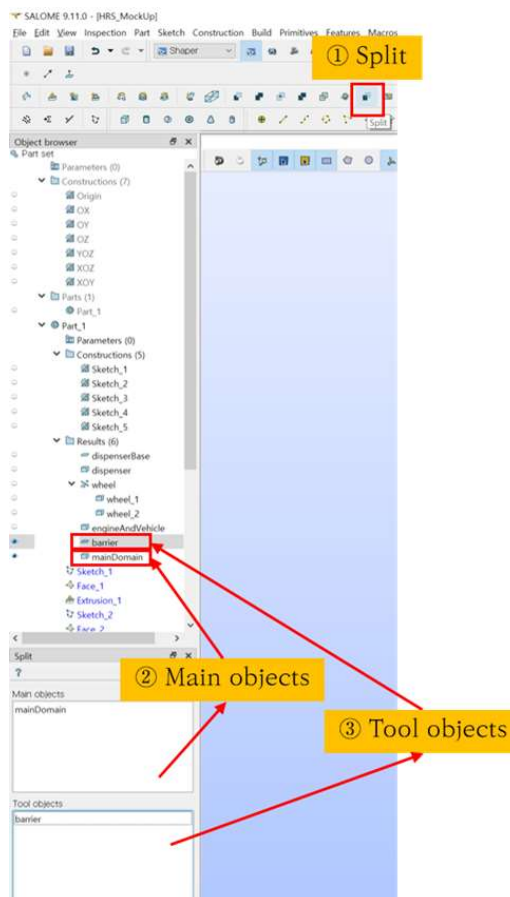


그림 16. 방호벽 분리 과정

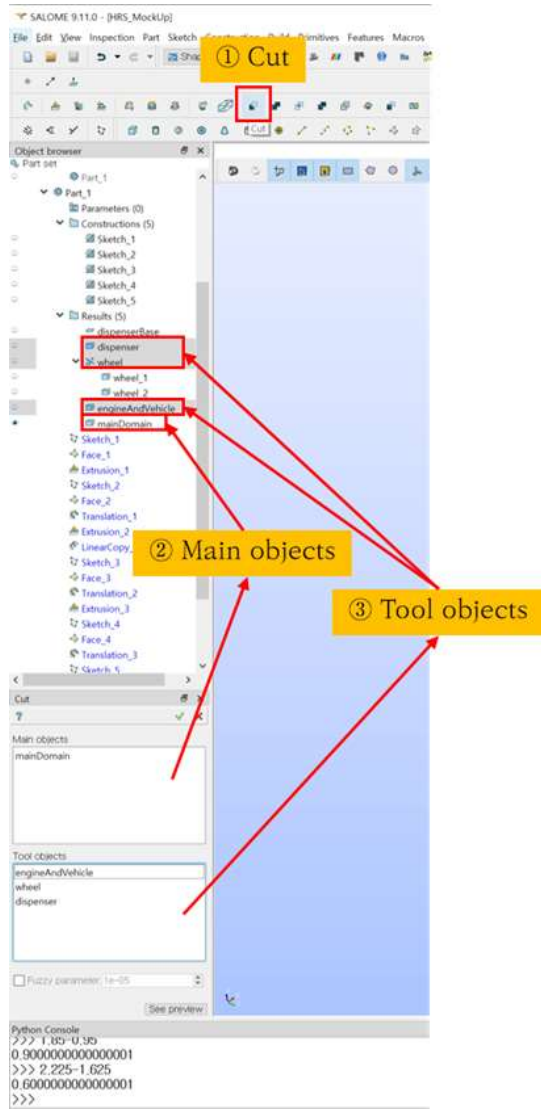


그림 17. 계산영역에서 구조물을 제외하는 과정

격자 생성을 위한 형상 전처리를 위해 “Shaper”에서 완성한 형상을 SALOME의 내부 프로그램인 “Geometry”로 이동시켜야 한다. 그림 18과 같이 “Export to GEOM” 버튼을 클릭하면 “Shaper”에서 생성한 모든 형상이 “Geometry”로 복사된다.

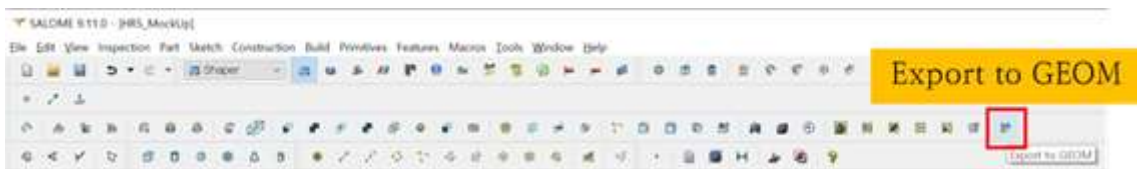


그림 18. Sharper의 형상을 Geometry 모듈로 추출하는 과정

2. 격자 생성을 위한 형상 전처리

그림 19와 같이 메뉴 바에서 “Shaper” 우측에 있는 “Geometry”를 선택한다.



그림 19. Geometry 모듈로 전환하는 과정

가. 계산영역 형상 내 경계면 정의

“Shaper”에서 “Geometry”로 복사한 형상을 그림 20과 같이 오른쪽 클릭하여 “Create Group”을 선택한다.

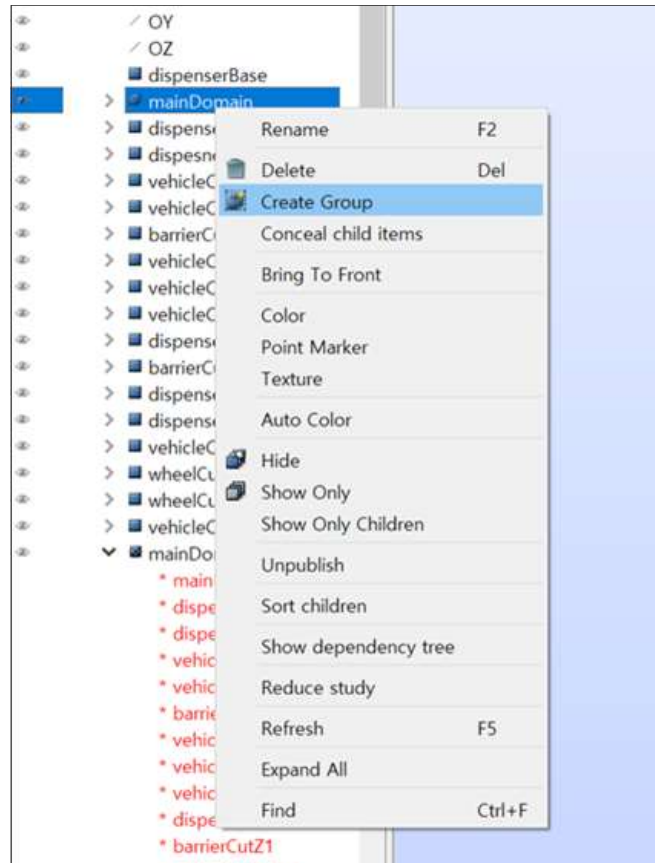


그림 20. Face 그룹 생성 과정

우선 방호벽을 먼저 지정해야 한다. 그림 21의 ①과 같이 면 선택 모드로 전환 후 ②와 같이 경계면에 적절한 이름을 부여한다. 그리고 ③의 부분을 클릭한 후 ④와 같이 방호벽에 해당하는 면을 선택한다. 그리고 ⑤와 같이 “Add”를 클릭하고, ⑥과 같이 “Apply”를 클릭한다. 그림 22와 같이 다른 경계면에 대해서도 같은 작업을 반복한다.

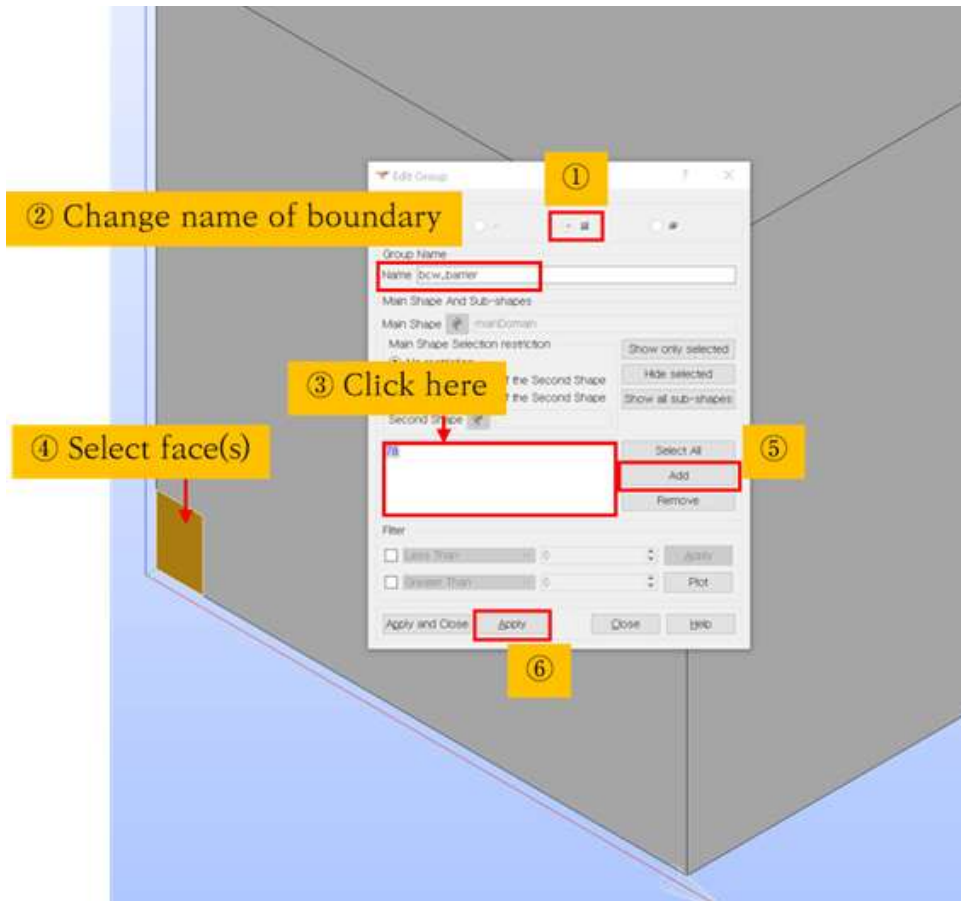


그림 21. 경계면(Boundary) 지정 과정

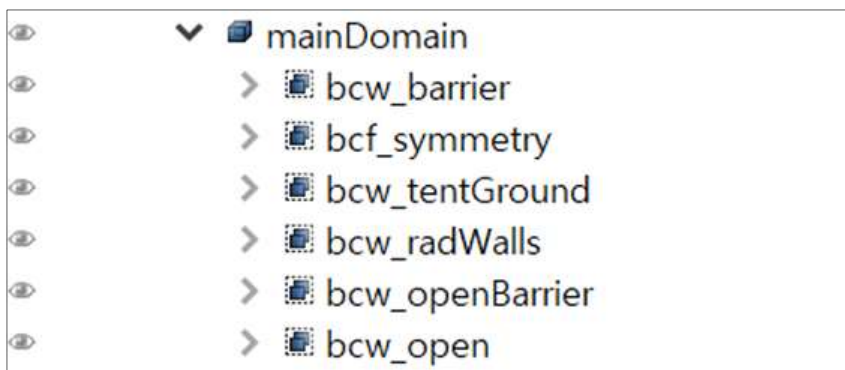


그림 22. 생성한 경계면의 목록

나. 격자영역 나누기

직육면체 셀만으로 이루어진 격자를 생성하기 위해서는 구조물의 각 지점별로 격자가 정의되는 영역을 세분화할 필요가 있다. 앞서 지정한 방호벽, 디스펜서 및 차량(디스펜서와 차량은 본 예제에서는 “radWalls”로 통합되어 있음) 좌측의 눈 표시를 클릭하고 나머지 구성요소들의 눈 표시를 비활성화하여 방호벽, 디스펜서 및 차량만 도면상에 보이게 한다. 그 뒤 그림 23의 ①과 같이 메뉴 바의 “Create a plane” 기능을 선택한 후 ②와 같이 “Point”에 평면을 생성할 구조물의 한 점을, ③과 같이 “Vector”에 평면이 필요한 방향의 법선 벡터(OX, OY, OZ)를 선택한다. 본 예제에서는 그림 24와 같이 17개의 평면을 생성하였다.

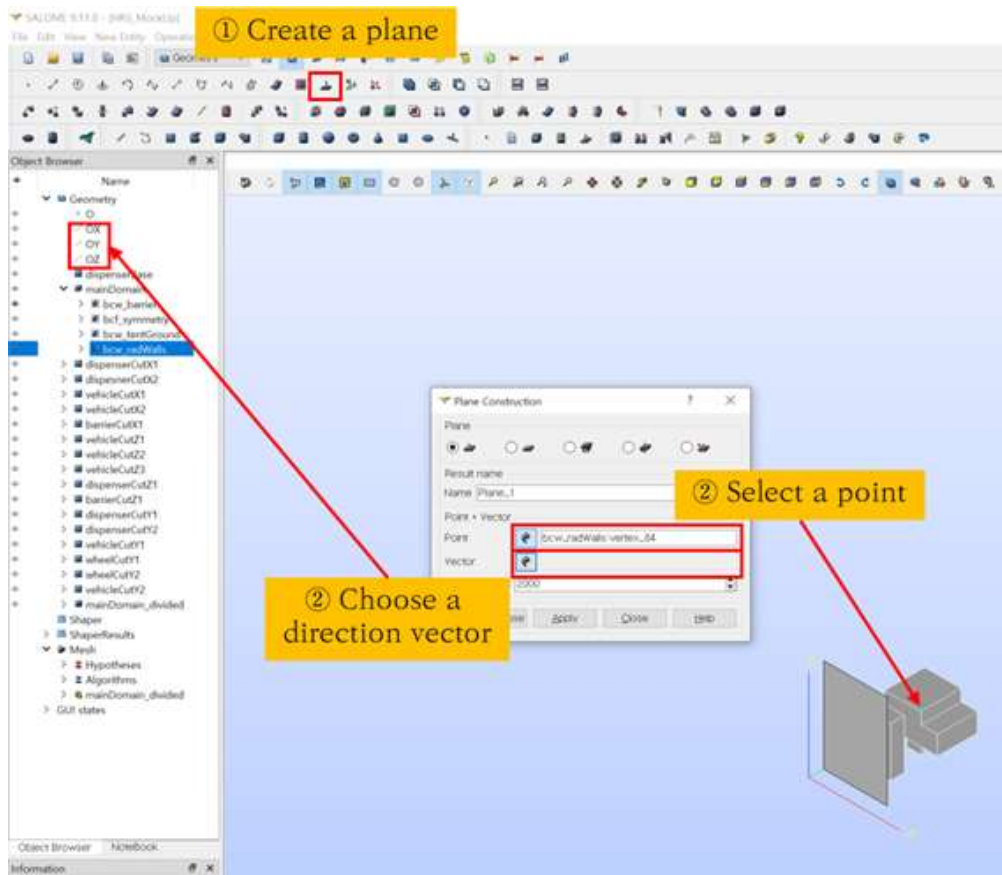


그림 23. 평면 생성 과정

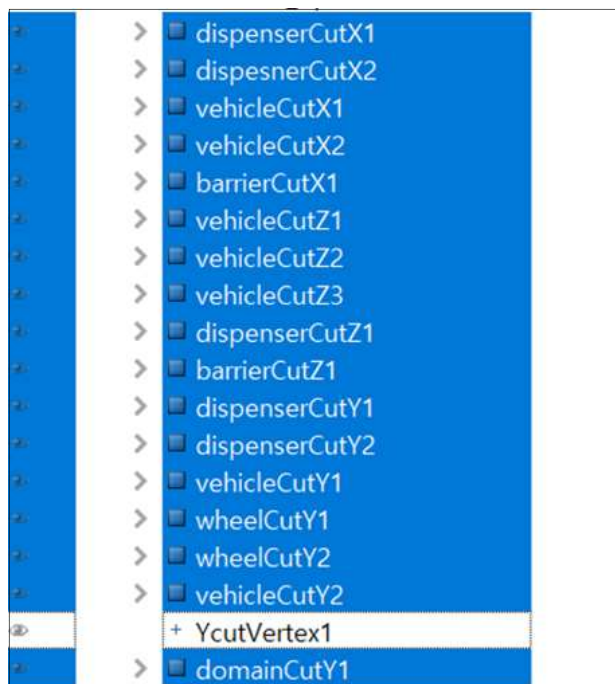


그림 24. 생성된 평면 목록

격자 생성을 위해 앞서 생성한 평면을 이용하여 계산 영역을 나누어 줄 것이다. 그림 25의 ①과 같이 메뉴 바의 “Partition” 기능을 선택한 후 ②와 같이 “Objects”에 계산 영역 형상을, ③과 같이 “Tool objects”에 생성한 평면을 선택한 후 “Apply” 버튼을 눌러 나누어 준다.

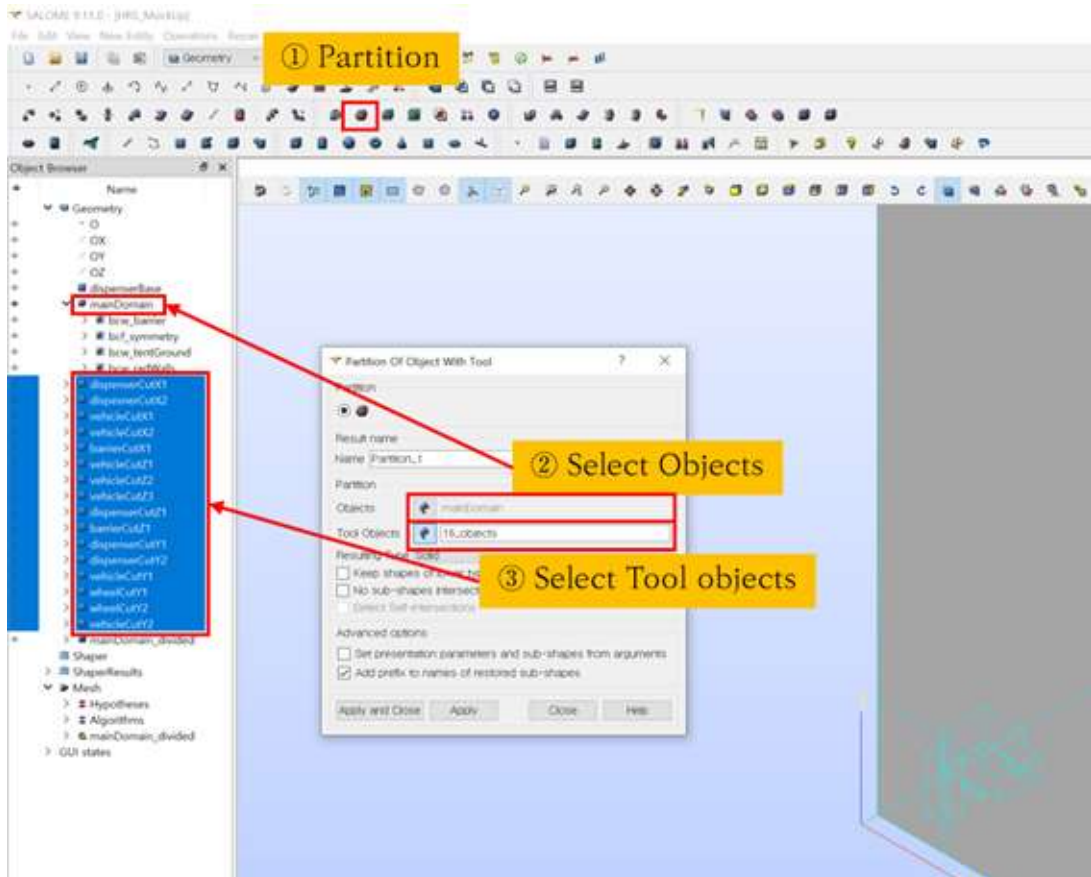


그림 25. Geometry 모듈의 Partition 기능 설명

계산 영역의 형상을 “Partition” 기능을 이용하여 나누면 새로운 형상이 생성된다. 새로 생성된 형상에서 경계면을 다시 정의해주어야 한다. 새로 생성된 형상을 오른쪽 클릭하여 “Create Groups”를 선택한다. 그 뒤 그림 26의 ①과 같이 면 선택 모드로 전환한 후 ②와 같이 “Geometrical parts of the Second Shape”를 선택한다. ③과 같이 “Partition” 기능을 이용해 나누기 전의 형상에서 정의할 면을 선택한다. ④의 “Select All”을 클릭하고 ⑤의 “Add”를 클릭한 뒤 ⑥의 “Apply”를 클릭한다. 이 과정을 나누기 전의 형상에서 정의된 면에 대해 모두 수행한다.

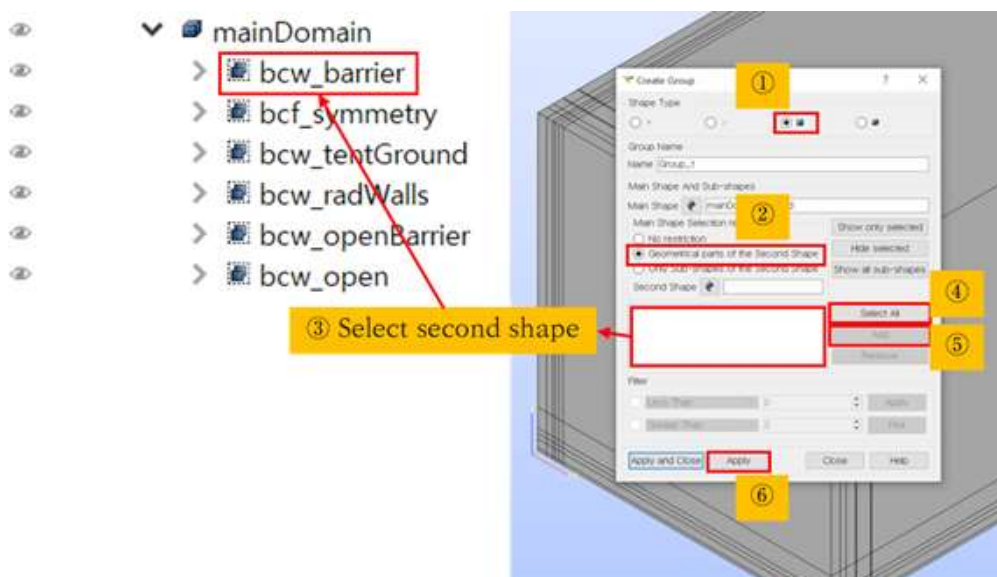


그림 26. 생성된 형상에서 경계면 재정의 과정

다. 나뉜 격자영역의 라인(Line) 그룹화

앞에서 설명한 “Partition” 기능을 이용해 형상을 17개 평면으로 나눈 것은 평면으로 나누어 생성된 부 형상들을 각각 원하는 크기로 잘라 전체 격자를 생성하고자 함이었다. 이를 위해서는 생성된 각각의 선분을 원하는 개수로 나누어야 한다. 다행히도 현재 예제의 전체 계산 영역 형상은 직육면체 형태이기 때문에 x, y, z 각각의 방향을 대표하는 하나의 변에 해당하는 라인들만 n 개로 나누어주면 된다. “Partition” 기능을 이용해 평면으로 나눈 형상을 오른쪽 클릭한 후 “Create Groups”를 선택한다. 그림 27의 ①과 같이 선 선택 모드로 변경한 후 ②, ③, ④와 같이 라인을 선택한다. 여기서 라인의 이름을 정의할 때 방향(x, y, z)과 나눌 개수(2, 4, 10, ...)를 명시하면 좋다(예: x1_32, y3_25, z2_16). 본 예제에서는 그림 27의 왼쪽과 같이 총 20개의 라인을 정의하였다.

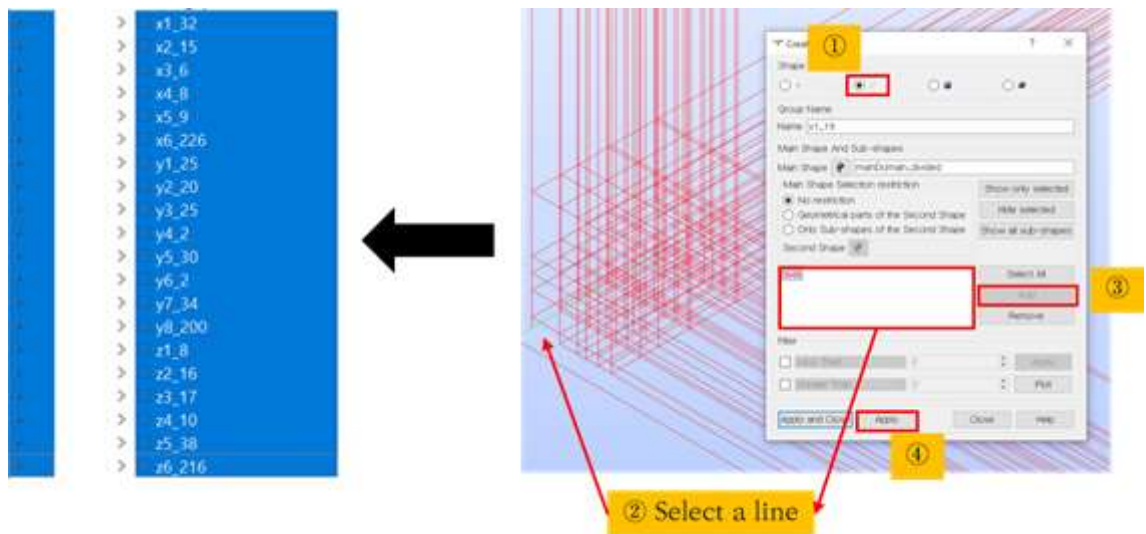


그림 27. 생성된 형상에서 경계면 재정의 과정

3. 격자 생성

그림 28의 메뉴 바에서 “Geometry” 우측에 있는 “Mesh” 모듈을 실행한다.



그림 28. Mesh 모듈로 전환 과정

가. 주 격자(Main Mesh) 생성

그림 29의 ①과 같이 메뉴 바에서 “Create a mesh”를 선택한다. ②와 같이 “Geometry” 모듈에서 작업한 형상을 선택한다. 그림 30은 3D, 2D, 1D 격자 생성 알고리즘을 지정하는 과정을 나타낸다. 3D 알고리즘에 “Hexahedron(i,j,k)”를, “Hypothesis”에 “<Default>”를 선택한다. 생성된 1D, 2D 격자를 기반으로 3D 격자를 육면체 격자로 생성한다는 것을 나타낸다.

나. 부 격자(Sub-mesh) 생성

그림 31과 같이 앞 절의 “2, 다”. 에서 정의한 선분들에 1차원 부 격자(Sub-mesh)를 할당한다. 이때 ④와 같이 “Add. Hypothesis”에 “Propagation of Node Distribution on Opposite Edges”를 선택해 주어야 한다. 이 옵션은 부 격자가 정의되는 선분과 평행한 방향에 놓인 모든 선분을 해당 부 격자와 똑같이 나눠주는 옵션이다. 만약 1차원 부 격자를 등 간격으로 나눌 것이라면 ③-①과 같이, 선분이 뻗어나가는 방향에 따라(예: 원점에서 x방향으로 1 m) 격자 크기가 작아지거나 커지게 하고 싶다면 ③-②와 같이 입력하면 된다. ③-②의 “Scale Factor”는 첫 번째 격자 대비 마지막 격자의 크기를 나타낸다.

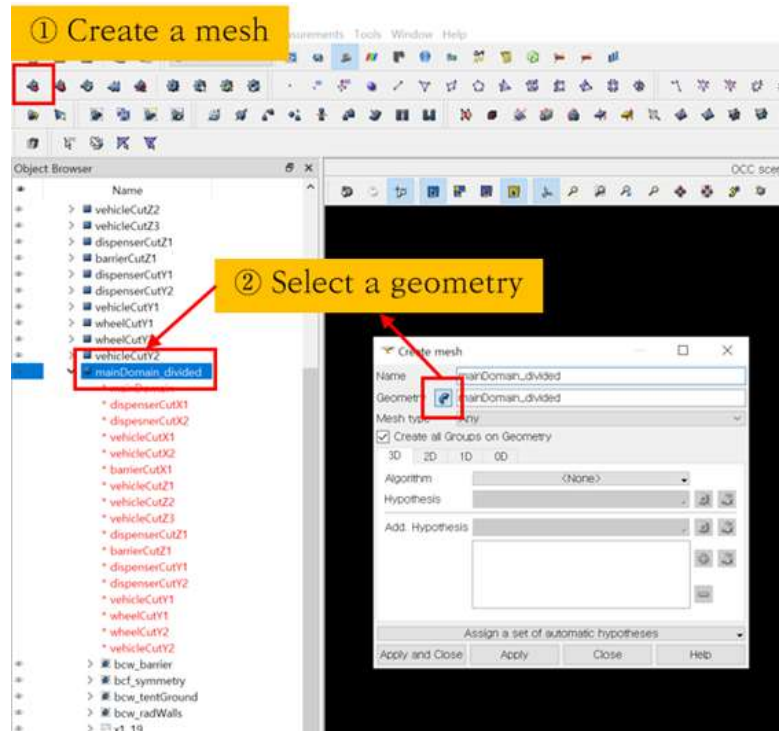


그림 29. 주 격자 생성-1

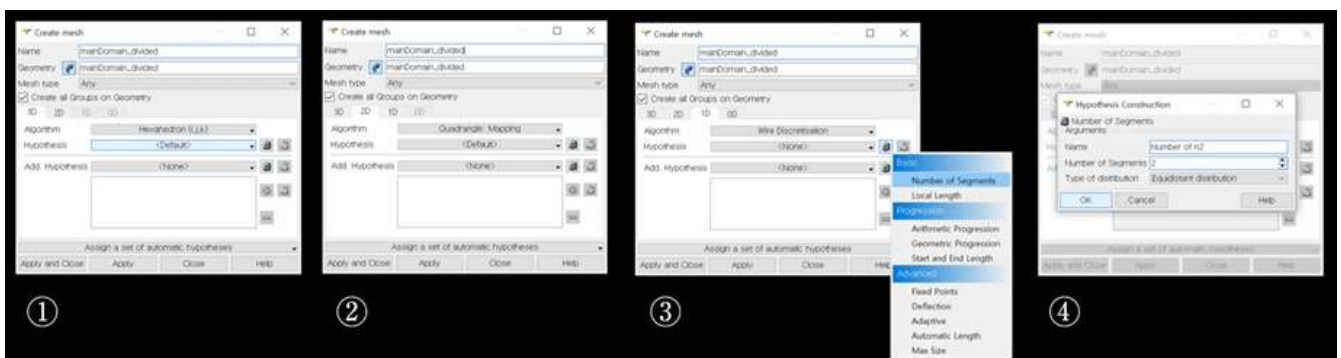


그림 30. 주 격자 생성-2



그림 31. 부 격자 생성

다. 격자 생성 및 오류 해결 방법

부 격자 정의를 완료한 후 격자 생성 이전에 격자의 최대 셀 수를 설정해야 한다. 본 예제의 경우 격자의 셀 수가 3,000만 개 이상으로 많아 기본 설정으로는 생성이 되지 않을 수도 있다. 그림 32와 같은 절차로 격자의 최대 셀 수를 설정할 수 있다. 설정을 마친 후 작업한 격자를 오른쪽 클릭한 후 “Compute”를 선택하면 격자가 생성된다. 이때 그림 33과 같이 격자 일부분이 깨지는 것을 확인할 수 있다. 이 경우 다시 “Geometry” 모듈로 돌아가 “2-다” 와 같이 그림 34에 나타나 있는 대로 격자가 깨진 영역을 의미하는 선분을 모두 선택한다. 이후 다시 “Mesh” 모듈로 돌아와 해당 선분들에 대한 부 격자를 추가로 생성한 후 다시 “Compute”를 실행한다.

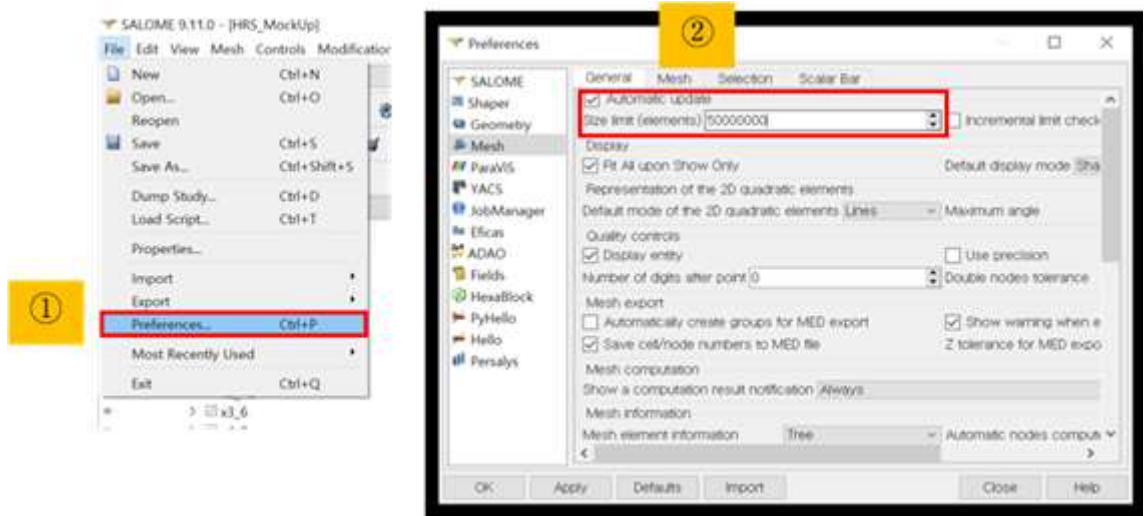


그림 32. 격자의 최대 셀 수 입력화면

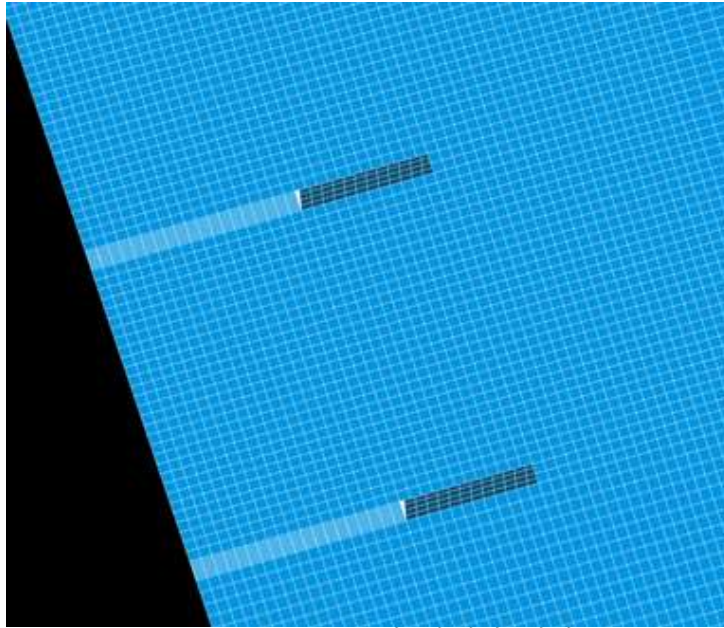


그림 33. 오류가 발생한 격자

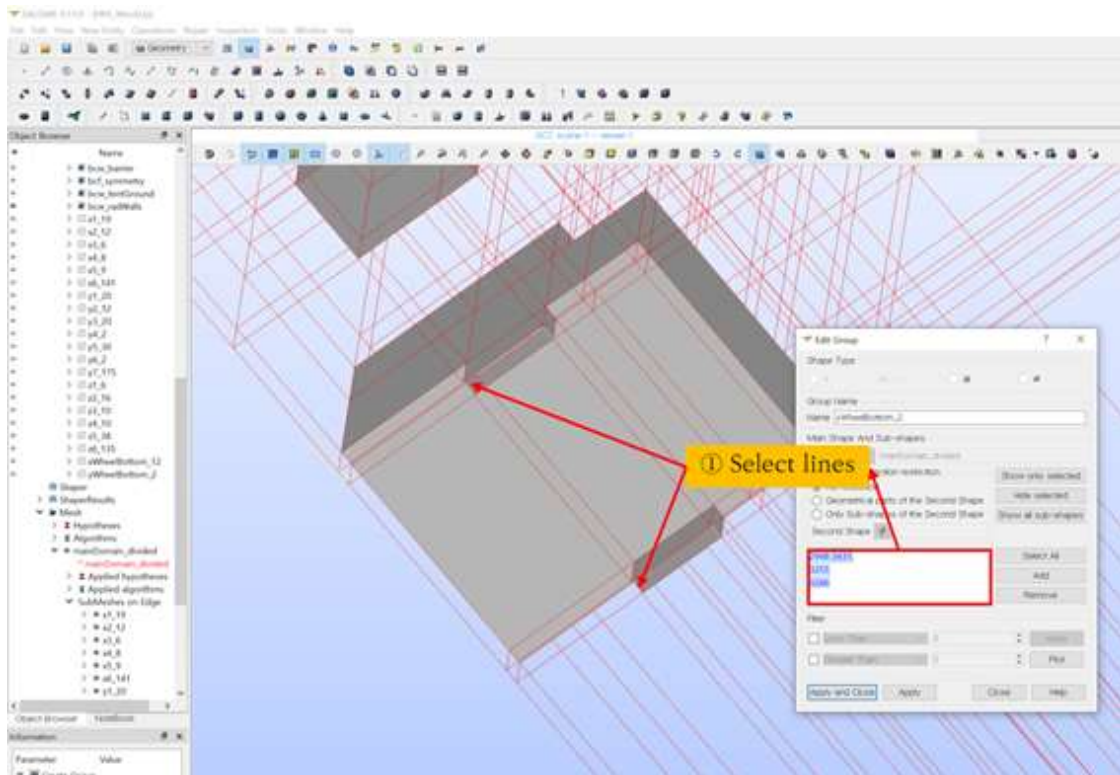


그림 34. 격자 오류 해결을 위한 Geometry 모듈에서의 추가 라인그룹 정의

라. 생성된 격자를 OpenFOAM 해석용 격자로 추출 방법

격자를 추출하기 전 생성된 격자의 “Groups of Edges”내에 정의된 선들을 모두 삭제해주어야 한다. 그 이유는 후술할 OpenFOAM의 “ideasUnvToFoam” 유틸리티를 사용할 때 추출한 UNV 파일에 선이 경계로 정의되어 있으면 오류가 발생하기 때문이다. 그림 35와 같은 격자에 존재하는 선들을 모두 삭제한다.

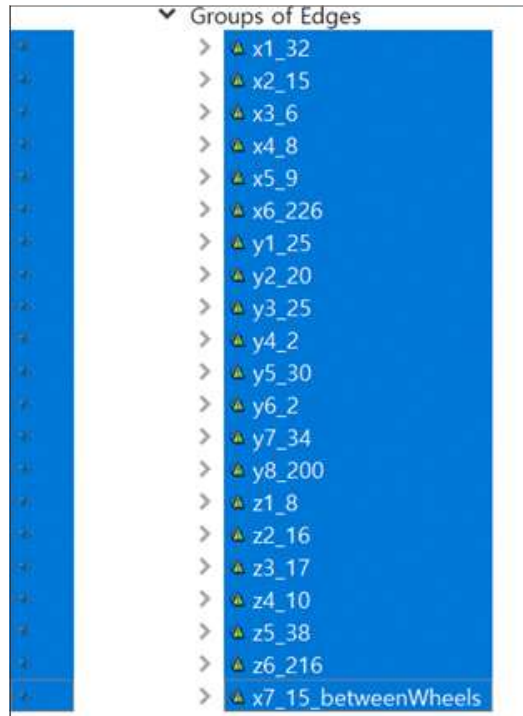


그림 35. 격자에 정의된 라인 그룹

격자 추출을 위하여 생성된 격자를 오른쪽 클릭한 후 그림 36과 같이 Export-UNV File을 선택한다. 추출한 파일을 OpenFOAM 격자로 바꾸주기 위해 OpenFOAM case 폴더로 UNV 파일을 복사한다. 그리고 case 폴더에서 터미널을 실행하여 OpenFOAM-v2112 환경설정을 위한 명령어를 입력한 후 “ideasUnvFoam 파일명.unv”를 터미널 창에 입력하여 UNV 파일을 OpenFOAM 격자로 변환한다. 격자 변환이 완료되면 그림 37과 같은 메시지가 출력된다. 이때 그림 37에서 모든 경계면이 patch로 정의되는데, 적절한 종류의 경계면으로 바꿔줄 필요가 있다. 그림 39와 같이 “constant/boundary” 파일을 직접 변경할 수 있다.

모든 변환과정을 완료한 후에 터미널 창에서 “checkMesh”를 입력하여 그림 40과 같이 생성된 격자 정보를 확인한다.

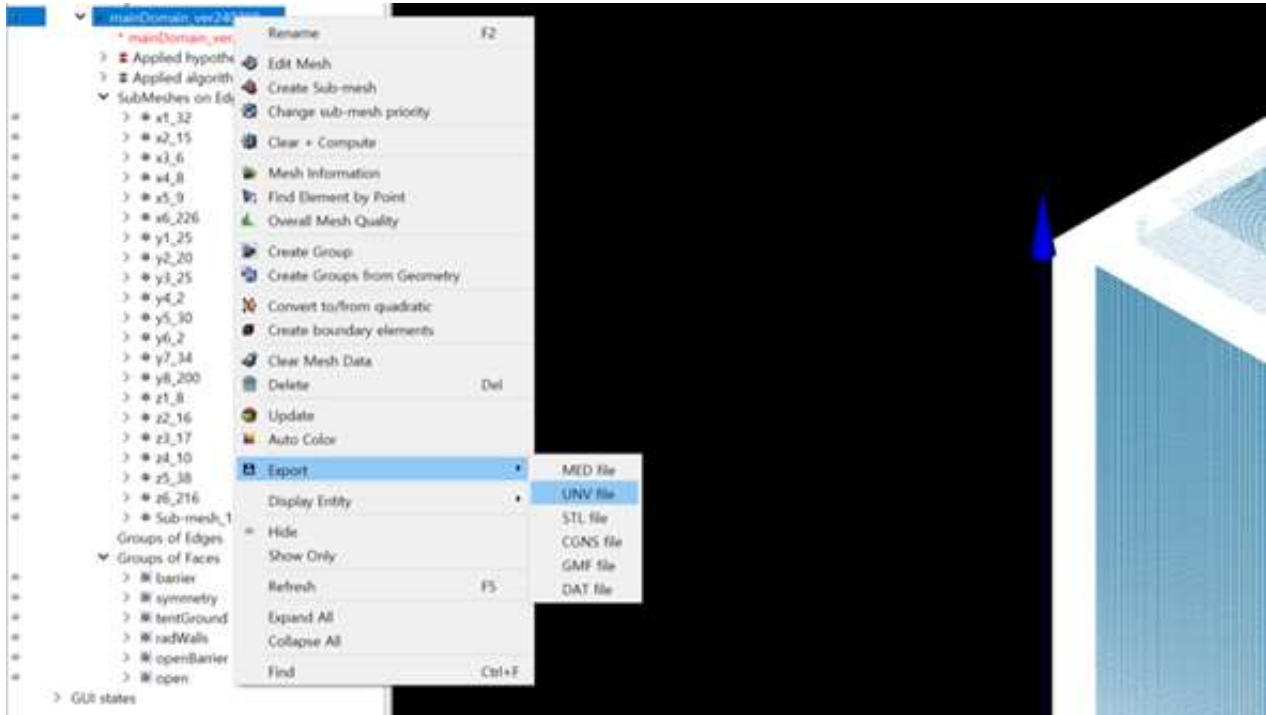


그림 36. 생성한 격자를 UNV 파일로 Export 하는 과정



그림 37. ideasUNVToFoam Utility 명령어 입력화면

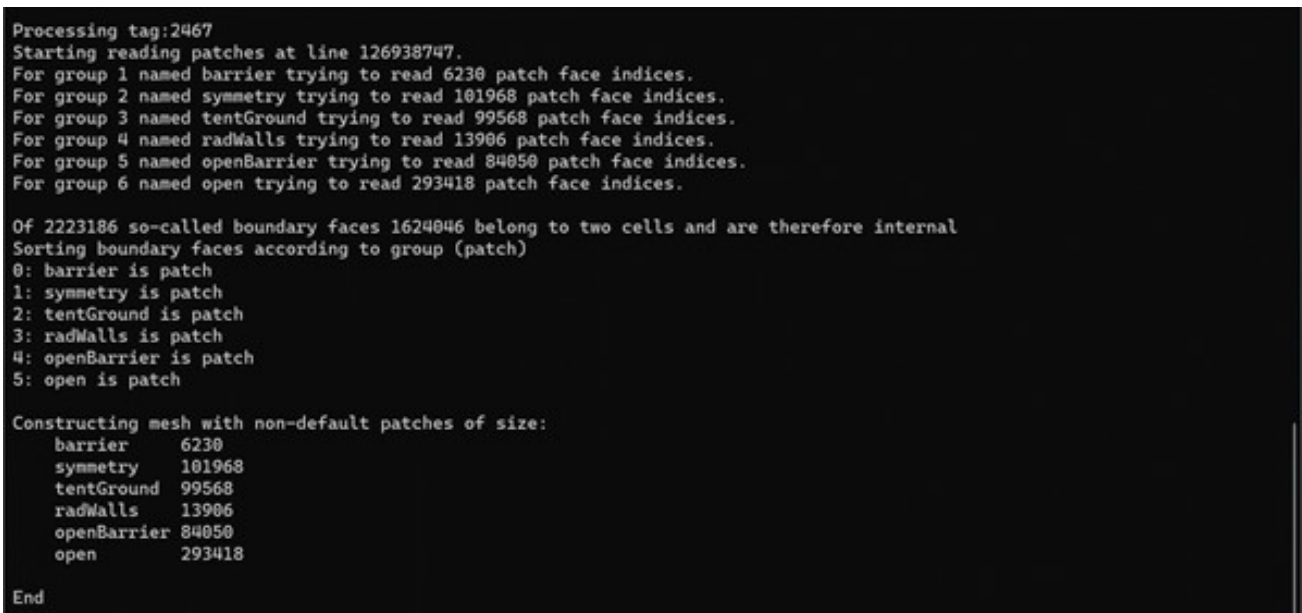


그림 38. OpenFOAM 해석용 격자변환 완료 메시지 화면

Before modification

After modification

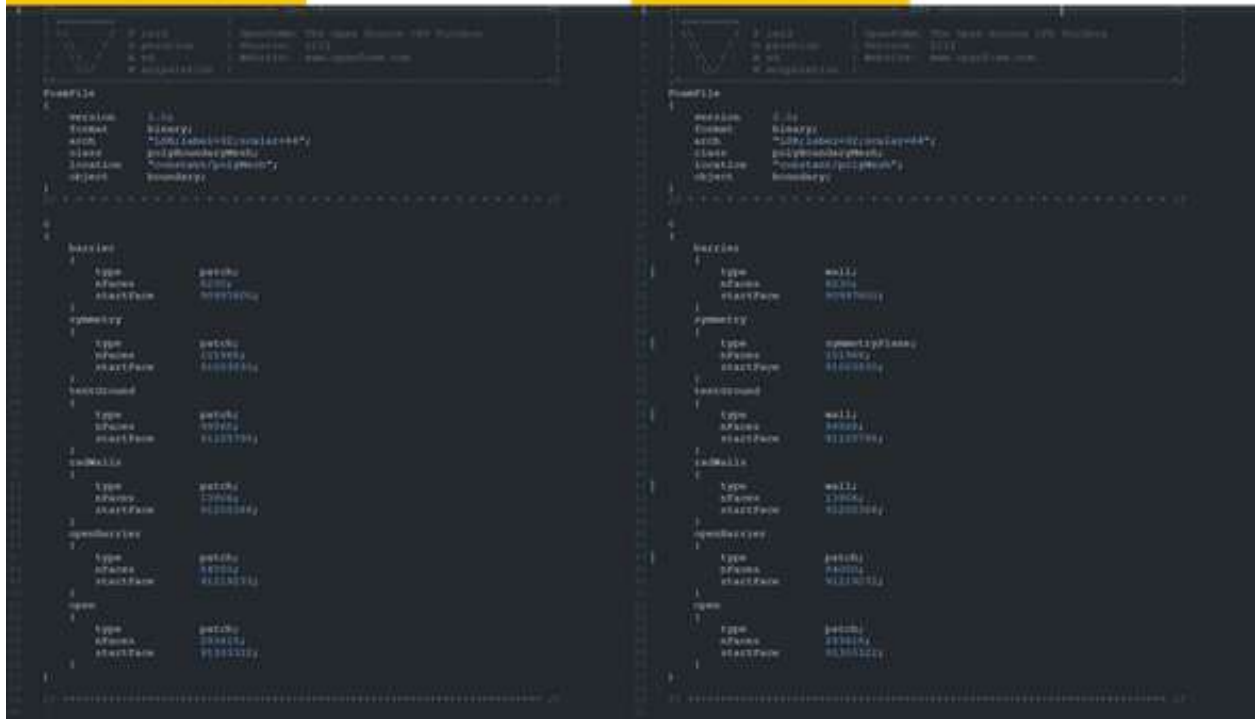


그림 39. Boundary 파일의 경계면 종류 변경 방법

```

Time = constant

Mesh stats
  points:          30732804
  faces:           91596740
  internal faces:  90997600
  cells:           30432390
  faces per cell:  6
  boundary patches: 6
  point zones:    0
  face zones:     0
  cell zones:     0

Overall number of cells of each type:
  hexahedra:      30432390
  prisms:         0
  wedges:         0
  pyramids:       0
  tet wedges:     0
  tetrahedra:    0
  polyhedra:     0

Checking topology...
Boundary definition OK.
Cell to face addressing OK.
Point usage OK.
Upper triangular ordering OK.
Face vertices OK.
Number of regions: 1 (OK).

Checking patch topology for multiply connected surfaces...
Patch      Faces   Points  Surface topology
barrier    6230   6390   ok (non-closed singly connected)
symmetry   101968 102678 ok (non-closed singly connected)
tentGround 99568  100275 ok (non-closed singly connected)
radWalls  13906  14848  ok (non-closed singly connected)
openBarrier 84650  84652  ok (non-closed singly connected)
open       293418 294358 ok (non-closed singly connected)

Checking faceZone topology for multiply connected surfaces...
No faceZones found.

Checking basic cellZone addressing...
No cellZones found.

Checking geometry...
Overall domain bounding box (-1.73467e-14 0.625 -1.02365e-22) (31.34.625 31)
Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
Mesh has 3 solution (non-empty) directions (1 1 1)
Boundary openness (-2.6252e-14 -1.76802e-15 -9.04043e-15) OK.
Max cell openness = 4.35999e-16 OK.
Max aspect ratio = 8.79149 OK.
Minimum face area = 0.800873162. Maximum face area = 0.05998076. Face area magnitudes OK.
Min volume = 2.61949e-05. Max volume = 0.0138085. Total volume = 32668.5. Cell volumes OK.
Mesh non-orthogonality Max: 0 average: 0
Non-orthogonality check OK.
Face pyramids OK.
Max skewness = 1.19538e-11 OK.
Coupled point location match (average 0) OK.

Mesh OK.

End
  
```

그림 40. checkMesh 명령어 실행결과 화면

7-6. OF-Interface 프로그램 사용방법

○ OpenFOAM은 해석을 위해 입력해야 하는 변수들을 문자 편집기(Text Editor)를 사용해서 작성해야 하며, 그 입력변수들의 특성에 따라서 총 3개의 디렉토리로 그룹화되어 있다. 따라서 radXiFoam 코드도 이러한 사용법을 적용하여 개발되었기 때문에 상용 CFD 코드에 익숙한 사용자들은 이런 방법에 매우 불편함을 느낄 수 있고, 입력작성 과정에서도 실수가 잘 유발될 수 있다. 현재 단계에서는 Microsoft Excel과 유사한 기능을 가진 리눅스의 스프레드시트 프로그램인 LibreOffice Calc에서 OpenFOAM 입력파일을 생성할 수 있는 OF-Interface를 개발하였다.

○ OF-Interface 사용 방법

1. 제공된 OF-Interface 파일을 열면 그림 1과 같은 스프레드시트 화면이 나타난다. 이 파일은 ①의 4개 Sheet로 구성되어 있고, Main Sheet를 제외한 나머지 세 개 시트는 radXiFoam case의 초기 및 경계조건(0), 물성치(constant), 계산제어(system) 디렉토리를 나타낸다. ②에서 radXiFoam case를 생성할 경로를 입력할 수 있고, ③을 클릭하여 radXiFoam case를 생성할 수 있다.

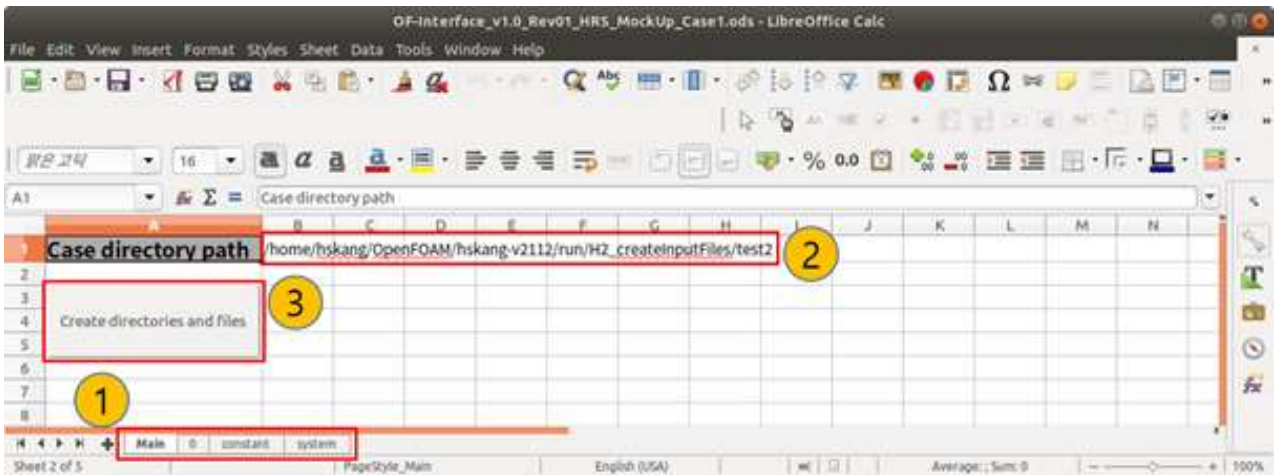


그림 1. OF-Interface 프로그램의 Main Sheet 화면

2. 그림 2의 0 Sheet에서 “제3장, 제2절”에 기술된 초기 및 경계조건을 입력할 수 있다. ①은 물리량, ②는 경계면 이름, ③은 경계조건 종류를 나타낸다. 필요한 물리량, 경계면의 수 만큼 셀을 추가/삭제하여 경계조건을 정의할 수 있다. 만약 ③의 버튼을 클릭했을 때 Java runtime environment 오류가 발생한다면 다음 명령어를 터미널에서 실행시키면 된다.

```
sudo apt install default-jre
```

```
sudo apt install libreoffice-java-common
```


4. 그림 4의 system Sheet는 radXiFoam case의 system 폴더 내 계산 제어와 관련된 내용을 입력하는 시트이다. 해당 시트에 입력하는 값들의 의미는 “제3장, 제3절”에 기술되어 있다.

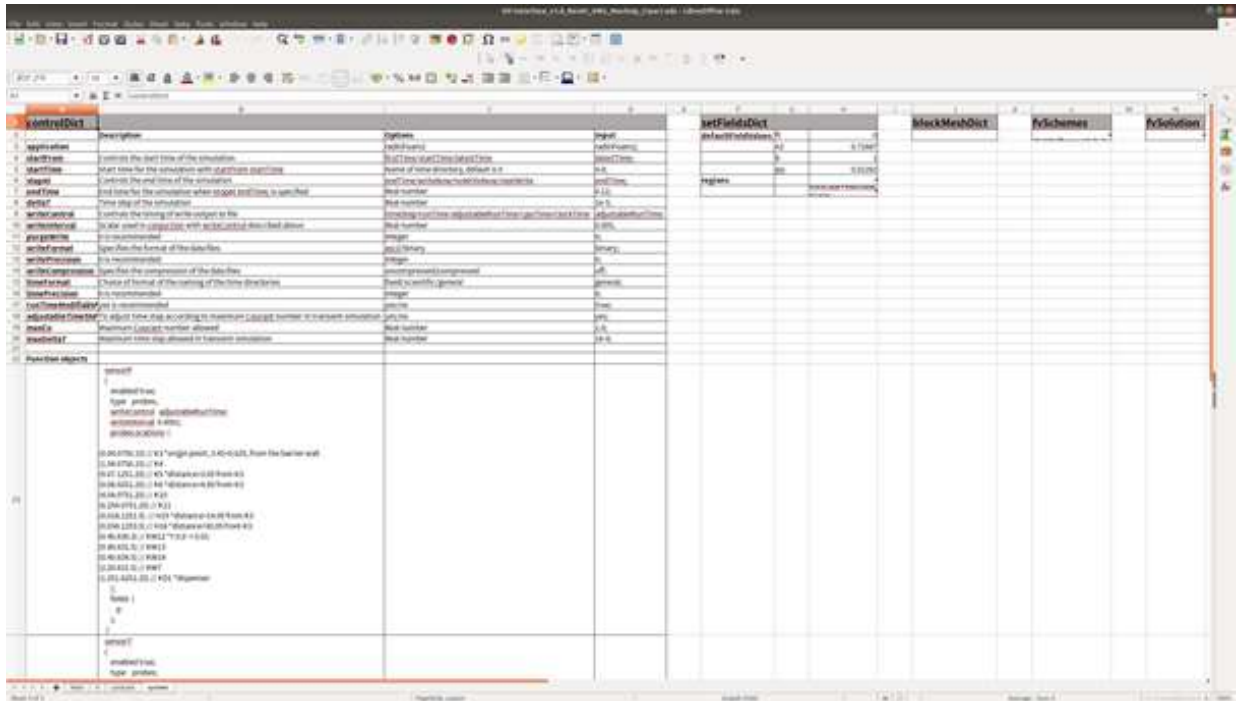


그림 4. OF-Interface 프로그램의 system Sheet 화면

[뒷면지]

주 의

1. 이 보고서는 산업통상자원부에서 시행한 산업기술혁신사업 연구개발과제 최종보고서이다.
2. 이 연구개발내용을 대외적으로 발표할 때에는 반드시 산업통상자원부(한국에너지기술평가원)에서 시행한 산업기술혁신사업의 결과임을 밝혀야 한다.
3. 국가과학기술 기밀 유지에 필요한 내용은 대외적으로 발표 또는 공개하여서는 안 된다.