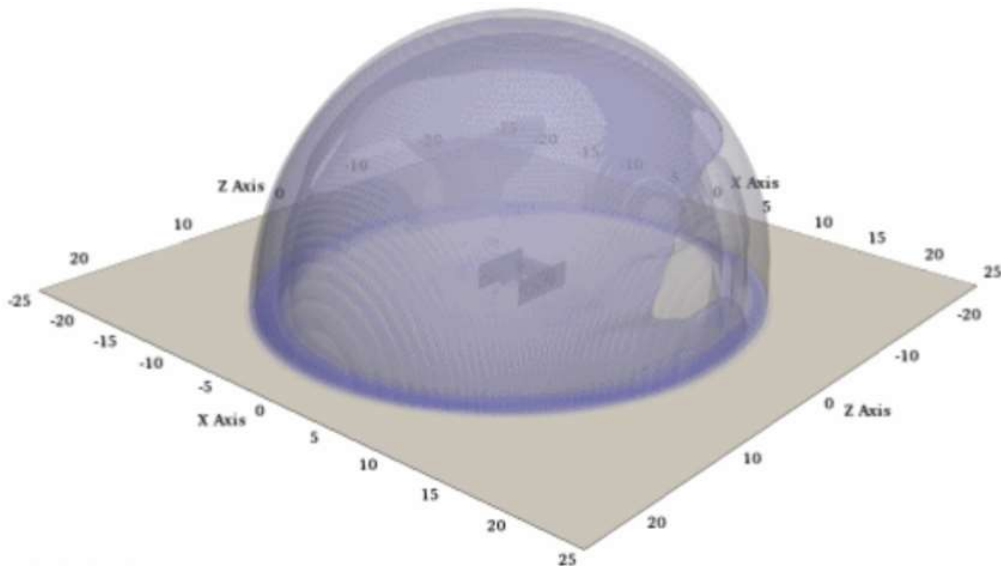


<b>연차보고서</b>	[√] 일반형		기술료
	[ ] 통합형(총괄)	[ ] 통합형(세부)	정수[ ] 비정수[ ]
중앙행정기관명	산업통상자원부	사업명	수소충전 인프라 안전관리 핵심기술 개발
전문기관명	한국에너지기술평가원		세부사업명
연구개발과제명(과제번호)	수소충전소 화재 폭발시 피해저감을 위한 방호벽 설계기술 및 안전기준 개발 (20215810100020)		

2025년 연차보고서 부록 :

고압수소 화재 폭발 해석코드(OpenFOAM: realGasReactingEsiFoam)

사용자 매뉴얼 v1.0



2025. 03. 17

오창보, 박예도

국립부경대학교

# 목 차

1. 서 론 .....	(2)
1-1. OpenFOAM 소개 및 설치 .....	(2)
1-2. realGasReactingEsiFoam 개요 및 설치 .....	(2)
2. realGasReactingEsiFoam 해석용 격자모델 생성방법 .....	(3)
2-1. 고압수소 화재폭발 실험 장치 격자모델 .....	(3)
2-2. blockMesh 프로그램 사용법 .....	(3)
3. realGasReactingEsiFoam 해석용 입력자료 .....	(7)
3-1. 물성치(constant 디렉토리) 입력 .....	(7)
3-2. 초기 및 경계조건(0 디렉토리) 입력 .....	(19)
3-3. 계산제어(system 디렉토리) 입력 .....	(31)
4. realGasReactingEsiFoam 실행방법 .....	(43)
5. 해석결과 분석방법 .....	(50)
6. 부록 .....	(58)
6-1. 격자모델 입력자료 .....	(58)
6-1. 고압 수소 초기조건 입력자료 .....	(58)

# 1. 서론

## 1-1. OpenFOAM 소개 및 설치

본 사용자 매뉴얼은 OpenFOAM-v7을 기반으로 개발된 고압수소 화재폭발 해석용 솔버인 realGasReactingEslFoam v1.0의 사용법을 다룬다. OpenFOAM은 “Open-source Field Operation And Manipulation” 의 약자로, C++ 언어로 구현된 오픈소스 전산유체역학 (Computational Fluid Dynamics, CFD) 소프트웨어 패키지이다. OpenFOAM은 다양한 물리적 문제를 해석할 수 있도록 설계된 다목적 도구로, 복잡한 유동 해석 뿐만 아니라 화학반응, 열 전달 등 여러 분야에 활용된다.

realGasReactingEslFoam을 설치하기 위해서는 Linux 운영체제(OS)에 OpenFOAM-v7을 먼저 설치하고 사용자 환경설정을 구성해야한다. 본 매뉴얼에서 사용한 리눅스 OS는 Rocky Linux 8.4이며, OpenFOAM-v7의 설치과정은 다음 링크에서 확인할 수 있다.

<https://openfoam.org/download/7-linux/>

## 1-2. realGasReactingEslFoam v1.0 개요 및 설치

본 절에서는 realGasReactingEslFoam 설치 방법에 대해 설명한다. realGasReactingEslFoam은 고압수소 화재폭발 해석을 위해 개발된 솔버로, 실제 조건을 모사하기 위한 열물리모델 (thermophysical Model), 난류모델(Turbulence Model) 및 연소모델 (Combustion Model)을 포함하고 있다. realGasReactingEslFoam 설치를 위해, 소스 파일을 Linux 컴퓨터에 다운로드 한 후, 각 모델을 컴파일해야한다, 그림 1는 이러한 모델 컴파일 과정을 단계별로 보여준다.

```
anode0-of7> pwd
/home/apps/opt/openfoam/7.0/opt/realGasEslFoam
anode0-of7> cd src/
anode0-of7> ls
Allclean Allwmake TurbulenceModels combustionModels thermophysicalModels
anode0-of7> cd thermophysicalModels/
anode0-of7> ls
Allwmake basic chemistryModel reactionThermo specie
anode0-of7> cd specie/
anode0-of7> wmake libso
anode0-of7> cd basic
anode0-of7> wmake libso
anode0-of7> cd reactionThermo/
anode0-of7> wmake libso
anode0-of7> cd chemistryModel/
anode0-of7> wmake libso

anode0-of7> cd TurbulenceModels/
anode0-of7> ls
Allwmake incompressible phaseIncompressible
compressible phaseCompressible turbulenceModels
anode0-of7> ./Allwmake

anode0-of7> cd combustionModels/
anode0-of7> ls
CombustionModel PaSR infinitelyFastChemistry radiationModels
EDC combustionModel laminar singleStepCombustion
FSD diffusion lnInclude zoneCombustion
Make functionObjects noCombustion
anode0-of7> wmake libso

anode0-of7> ls
Allclean Allwmake TurbulenceModels combustionModels thermophysicalModels
anode0-of7> ./Allwmake

anode0-of7> ls
Allwclean centralCourantNo.H realGasEslFoam.C setRDeltaT.H
Allwmake createFieldRefs.H rhoEEqn.H src
BCs createFields.H rhoEqn.H
'Dropped Text.txt' directionInterpolate.H rhoUEqn.H
Make readFluxScheme.H rhoYEqn.H
anode0-of7> ./Allwmake
```

그림 1. realGasReactingEslFoam 컴파일 프로세스

## 2. realGasReactingEslFoam 해석용 격자모델 생성방법

### 2-1. 고압수소 화재폭발 실험 격자모델 및 해석 대상

본 매뉴얼은 실제 규모의 고압수소 탱크 화재폭발 실험을 기반으로, 작성되었다. 실험에 사용된 조건을 바탕으로, 고압수소 탱크의 위치, 방호벽 등을 포함한 격자모델을 생성하였으며, 이를 통해 전산해석을 수행하는 방법을 설명한다. 그림 2는 실험에서 사용된 물리적 조건과 동일하게 수소용기, 방호벽의 배치를 나타내며 이는 격자 설계 시 기초 데이터로 활용되었다. 본 매뉴얼에서는 해당 실험 조건을 예제 파일로 사용하여 realGasReactingEslFoam 사용법에 대해 설명하고자 한다.

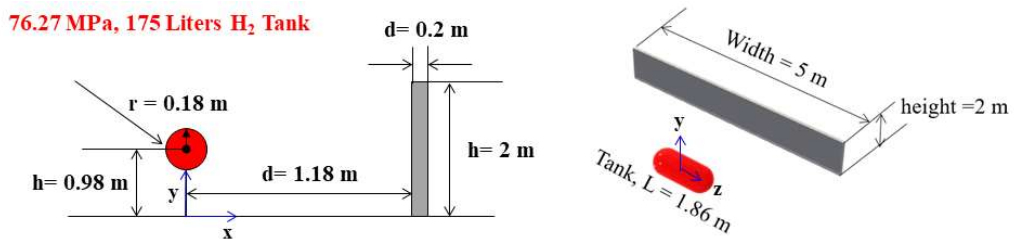


그림 2. 고압수소 화재폭발 실험 개략도

### 2-2. blockMesh 프로그램 사용법

본 절에서는 openFOAM-v7에 내장된 격자생성 프로그램인 blockMesh의 사용법을 설명하고자 한다. blockMesh 프로그램을 제어하는 입력파일은 “Case\system\blockMeshDict” 이다.

#### 2-2-1. 해석영역 지정용 블록좌표 생성방법

전산해석을 위해 계산영역을 설정하고 이를 각 직육면체 블록(block)으로 분할하였다. 각 블록의 꼭짓점에 해당하는 지점(point)의 좌표는 blockMeshDict 파일에 입력된다. 계산 영역은 (-25, 0, -25)에서 (25, 25, 25)까지 설정하였으며, 방호벽의 위치와 두께를 고려해 표0 에 따라 좌표를 입력하였다. 전체 blockMeshDict 파일은 부록0에 포함되어 있다.

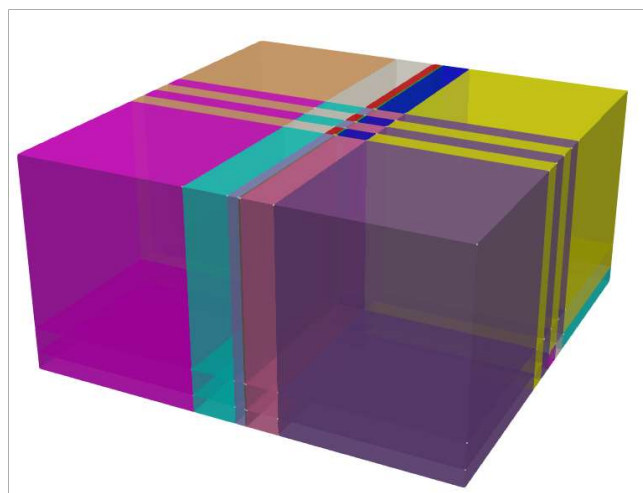


그림 3. 격자 생성을 위한 블록 구성도

표 1. 격자 생성을 위한 blockMeshDict Verrices (일부)

(-25.00 0.00 -25.00) // 0	(-25.00 0.00 -5.00) // 28	(-25.00 0.00 -2.50) // 56
(-5.00 0.00 -25.00) // 1	(-5.00 0.00 -5.00) // 29	(-5.00 0.00 -2.50) // 57
(0.00 0.00 -25.00) // 2	(0.00 0.00 -5.00) // 30	(0.00 0.00 -2.50) // 58
(1.20 0.00 -25.00) // 3	(1.20 0.00 -5.00) // 31	(1.20 0.00 -2.50) // 59
(1.40 0.00 -25.00) // 4	(1.40 0.00 -5.00) // 32	(1.40 0.00 -2.50) // 60
(5.00 0.00 -25.00) // 5	(5.00 0.00 -5.00) // 33	(5.00 0.00 -2.50) // 61
(25.00 0.00 -25.00) // 6	(25.00 0.00 -5.00) // 34	(25.00 0.00 -2.50) // 62
(-25.00 2.00 -25.00) // 7	(-25.00 2.00 -5.00) // 35	(-25.00 2.00 -2.50) // 63
(-5.00 2.00 -25.00) // 8	(-5.00 2.00 -5.00) // 36	(-5.00 2.00 -2.50) // 64
(0.00 2.00 -25.00) // 9	(0.00 2.00 -5.00) // 37	(0.00 2.00 -2.50) // 65
(1.20 2.00 -25.00) // 10	(1.20 2.00 -5.00) // 38	(1.20 2.00 -2.50) // 66
(1.40 2.00 -25.00) // 11	(1.40 2.00 -5.00) // 39	(1.40 2.00 -2.50) // 67
(5.00 2.00 -25.00) // 12	(5.00 2.00 -5.00) // 40	(5.00 2.00 -2.50) // 68
(25.00 2.00 -25.00) // 13	(25.00 2.00 -5.00) // 41	(25.00 2.00 -2.50) // 69
(-25.00 5.00 -25.00) // 14	(-25.00 5.00 -5.00) // 42	(-25.00 5.00 -2.50) // 70
(-5.00 5.00 -25.00) // 15	(-5.00 5.00 -5.00) // 43	(-5.00 5.00 -2.50) // 71
(0.00 5.00 -25.00) // 16	(0.00 5.00 -5.00) // 44	(0.00 5.00 -2.50) // 72
(1.20 5.00 -25.00) // 17	(1.20 5.00 -5.00) // 45	(1.20 5.00 -2.50) // 73
(1.40 5.00 -25.00) // 18	(1.40 5.00 -5.00) // 46	(1.40 5.00 -2.50) // 74
(5.00 5.00 -25.00) // 19	(5.00 5.00 -5.00) // 47	(5.00 5.00 -2.50) // 75
(25.00 5.00 -25.00) // 20	(25.00 5.00 -5.00) // 48	(25.00 5.00 -2.50) // 76
(-25.00 25.00 -25.00) // 21	(-25.00 25.00 -5.00) // 49	(-25.00 25.00 -2.50) // 77
(-5.00 25.00 -25.00) // 22	(-5.00 25.00 -5.00) // 50	(-5.00 25.00 -2.50) // 78
(0.00 25.00 -25.00) // 23	(0.00 25.00 -5.00) // 51	(0.00 25.00 -2.50) // 79
(1.20 25.00 -25.00) // 24	(1.20 25.00 -5.00) // 52	(1.20 25.00 -2.50) // 80
(1.40 25.00 -25.00) // 25	(1.40 25.00 -5.00) // 53	(1.40 25.00 -2.50) // 81
(5.00 25.00 -25.00) // 26	(5.00 25.00 -5.00) // 54	(5.00 25.00 -2.50) // 82
(25.00 25.00 -25.00) // 27	(25.00 25.00 -5.00) // 55	(25.00 25.00 -2.50) // 83

### 2-2-2. 블록 및 격자 셀 생성방법

blockMeshDict 파일에서는 총 195개의 좌표를 정의하며, 8개의 좌표를 묶어 직육면체 블록을 형성한다. hex (0 1 2 3 4 5 6 7)” 의 형식에서 숫자는 좌표의 번호를 의미한다. 표 2의 ①에서는 각 꼭짓점 좌표를 사용하여 생성된 직육면체 블록을 보여준다. 표 2의 ②는 각 블록의 셀(Cell)의 개수를, 표 2의 ③은 셀 크기의 비율을 나타낸다.

본 예제에서는 중심영역 (-5, 0, -5)부터 (5, 5, 5) 까지는 격자의 크기를 0.05m로 설정하였으며, 그 외부 영역에서는 simpleGrading을 이용해 격자의 크기가 점진적으로 증가하여 최대 0.5m에 이르도록 설정하였다. simpleGrading을 계산법은 다음 위치에서 참고할 수 있다.

[https://openfoamwiki.net/index.php/Scripts/blockMesh\\_grading\\_calculation](https://openfoamwiki.net/index.php/Scripts/blockMesh_grading_calculation)

표 2의 ④,⑤는 방호벽을 나타낸다. 방호벽 블록의 좌표는 // 기호를 사용해 제거하였으며, 이를 통해 격자 모델에서 방호벽이 생성된다.

표 2. 격자 생성을 위한 blockMeshDict blocks (일부)

①	②	③
hex ( 0 1 8 7 28 29 36 35)	( 102 40 102)	simpleGrading ( 0.1 1.0 0.1)
hex ( 1 2 9 8 29 30 37 36)	( 100 40 102)	simpleGrading ( 1.0 1.0 0.1)
hex ( 2 3 10 9 30 31 38 37)	( 24 40 102)	simpleGrading ( 1.0 1.0 0.1)
hex ( 3 4 11 10 31 32 39 38)	( 4 40 102)	simpleGrading ( 1.0 1.0 0.1)
hex ( 4 5 12 11 32 33 40 39)	( 72 40 102)	simpleGrading ( 1.0 1.0 0.1)
hex ( 5 6 13 12 33 34 41 40)	( 101 40 102)	simpleGrading ( 10.0 1.0 0.1)
hex ( 28 29 36 35 56 57 64 63)	( 102 40 50)	simpleGrading ( 0.1 1.0 1.0)
hex ( 29 30 37 36 57 58 65 64)	( 100 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 30 31 38 37 58 59 66 65)	( 24 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 31 32 39 38 59 60 67 66)	( 4 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 32 33 40 39 60 61 68 67)	( 72 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 33 34 41 40 61 62 69 68)	( 101 40 50)	simpleGrading ( 10.0 1.0 1.0)
hex ( 56 57 64 63 84 85 92 91)	( 102 40 50)	simpleGrading ( 0.1 1.0 1.0)
hex ( 57 58 65 64 85 86 93 92)	( 100 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 58 59 66 65 86 87 94 93)	( 24 40 50)	simpleGrading ( 1.0 1.0 1.0)
// hex ( 59 60 67 66 87 88 95 94)	( 4 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 60 61 68 67 88 89 96 95)	( 72 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 61 62 69 68 89 90 97 96)	( 101 40 50)	simpleGrading ( 10.0 1.0 1.0)
hex ( 84 85 92 91 112 113 120 119)	( 102 40 50)	simpleGrading ( 0.1 1.0 1.0)
hex ( 85 86 93 92 113 114 121 120)	( 100 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 86 87 94 93 114 115 122 121)	( 24 40 50)	simpleGrading ( 1.0 1.0 1.0)
// hex ( 87 88 95 94 115 116 123 122)	( 4 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 88 89 96 95 116 117 124 123)	( 72 40 50)	simpleGrading ( 1.0 1.0 1.0)
hex ( 89 90 97 96 117 118 125 124)	( 101 40 50)	simpleGrading ( 10.0 1.0 1.0)

### 2-2-3. 블록의 경계면 생성방법

블록이 생성되면, 블록의 각 면은 경계 조건(boundary)으로 정의된다. 경계면의 이름, 형식(type), 그리고 경계면을 구성하는 좌표번호를 blockMeshDict 파일에 입력한다. 표 3의 ①에서는 경계면 이름을 정의하고, 표 3의 ②에서는 경계면의 타입(예: wall, symmetry, patch 등)을 지정하였다. 표 3의 ③은 각 경계면의 꼭짓점 좌표번호를 나타낸다. 예를 들어, 방호벽의 경계면은 Blastwall\_로 시작하며, 이를 통해 각 방호벽의 위치를 명확히 나타낼 수 있다.

표 3. 격자 생성을 위한 blockMeshDict boundary (일부)

```

boundary
  ① base
  ② type wall;
    faces
  ③ ( ( 0 1 29 28)
      ( 1 2 30 29)
      ( 2 3 31 30)
      ( 3 4 32 31)
      ( 4 5 33 32)
      ( 5 6 34 33)
      ( 28 29 57 56)
      ( 29 30 58 57)
      ( 30 31 59 58)
      ( 31 32 60 59)
      ( 32 33 61 60)
      ( 33 34 62 61)
      ( 56 57 85 84)
      ( 57 58 86 85)
      ( 58 59 87 86)
  ④ // ( 59 60 88 87)
      ( 60 61 89 88)
      ( 61 62 90 89)
      ( 84 85 113 112)
      ( 85 86 114 113)
      ( 86 87 115 114)
      // ( 87 88 116 115)
      ( 88 89 117 116)
      ( 89 90 118 117)

outlet
{
  type patch;
  faces
  (
    ( 21 49 50 22)
    ( 22 50 51 23)
    ( 23 51 52 24)
    ( 24 52 53 25)
    ( 25 53 54 26)
    ( 26 54 55 27)
  )
}

Blastwall_Right_Face
{
  type patch;
  faces
  (
    (59 66 67 60)
  );
}

Blastwall_Left_Face
{
  type patch;
  faces
  (
    (116 123 122 115)
  );
}

Blastwall_Top_Face
{
  type patch;
  faces
  (
    (66 67 95 94)
    (94 95 123 122)
  );
}

Blastwall_Front_Face
{
  type patch;
  faces
  (
    (87 94 66 59)
    (115 122 94 87)
  );
}

Blastwall_Rear_Face
{
  type patch;
  faces
  (
    (60 67 95 88)
    (88 95 123 116)
  );
}

```

### 2-2-4. blockMesh 프로그램 실행방법

blockMeshDict 파일 작성이 완료되면, 터미널에서 blockMesh > log.blockMesh 명령어를 실행하여 격자를 생성한다. 명령어 뒤에 > log.blockMesh를 추가하면 실행 결과가 터미널이 아닌 log 파일에 저장된다. 이를 통해 생성 과정의 메시지, 경고 또는 오류를 확인하고, 문제 발생 시 디버깅 자료로 활용할 수 있다.

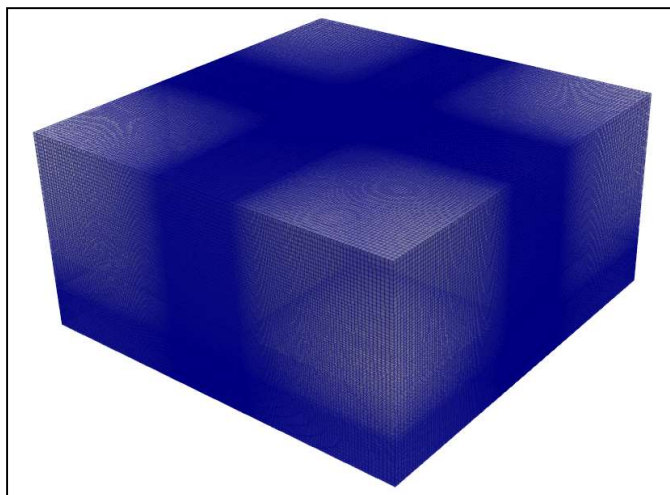


그림 4. blockMesh으로 생성된 realGasReactingEsiFoam 격자 모델

그림 5의 ①는 격자 모델 기본적인 정보를 제공한다. 내부 면(internal faces), 경계면(boundary faces), 정의되지 않은 경계 면(Undefined boundary faces)의 개수를 나타낸다. 그림 5의 ②는 각 블록에서 셀의 크기 분포를 나타낸다. 각 블록의 최소 및 최대 셀 크기를 확인하여 격자 세분화가 균일하게 이루어졌는지, 특정 영역에서 더 세밀하게 설정되었는지 파악할 수 있다. 그림 5의 ③는 격자가 차지하는 기하학적 영역의 범위를 나타낸다. 그림 5의 ④는 격자 경계면(patch) 목록과 이름, 면의 개수를 나타낸다.

```

Create time
Deleting polyMesh directory
"/home/users/yedo/WH_2_5/Case2/case2/constant/polyMesh"
Creating block mesh from
"/home/users/yedo/WH_2_5/Case2/case2/system/blockMeshDict"
Creating block edges
Creating block faces
No non-planar block faces defined
Creating topology blocks
Creating topology patches

Creating block mesh topology

Check topology
① Basic statistics
   Number of internal faces : 243
   Number of boundary faces : 150
   Number of defined boundary faces : 150
   Number of undefined boundary faces : 0
   Checking patch -> block consistency

Creating block offsets
Creating merge list .

Creating polyMesh from blockMesh
Creating patches
Creating cells
Creating points with scale 1
②
Block 0 cell size :
  i : 0.499637 .. 0.0499637
  j : 0.05 .. 0.05
  k : 0.499637 .. 0.0499637
Block 1 cell size :
  i : 0.05 .. 0.05
  j : 0.05 .. 0.05
  k : 0.499637 .. 0.0499637
Block 2 cell size :
  i : 0.05 .. 0.05
  j : 0.05 .. 0.05
  k : 0.499637 .. 0.0499637

Writing polyMesh
-----
Mesh Information
③
boundingBox: (-25 0 -25) (25 25 25)
nPoints: 33202980
nCells: 32872024
nFaces: 98946058
nInternalFaces: 98286086

Patches
-----
④
patch 0 (start: 98286086 size: 162412) name: base
patch 1 (start: 98448498 size: 488840) name: outlet
patch 2 (start: 98937338 size: 400) name: Blastwall_Top_Face
patch 3 (start: 98937738 size: 4000) name: Blastwall_Front_Face
patch 4 (start: 98941738 size: 4000) name: Blastwall_Rear_Face
patch 5 (start: 98945738 size: 160) name: Blastwall_Right_Face
patch 6 (start: 98945898 size: 160) name: Blastwall_Left_Face

```

그림 5. 격자 모델 생성 결과 (log.blockMesh)

### 2-2-5. 격자모델 정보 확인방법

격자 모델 생성 후, “constant/polyMesh” 디렉토리에 점(points), 면(faces), 셀(cells) 등 격자와 관련된 모든 정보가 저장된다(그림 6). 터미널에서 checkMesh 명령어를 실행하면 격자 품질과 일관성을 검증할 수 있다. 이 명령어는 격자 셀의 품질, 면 연결 상태 등을 분석하며, 생성된 격자가 정상적인지 확인하는 데 유용하다.

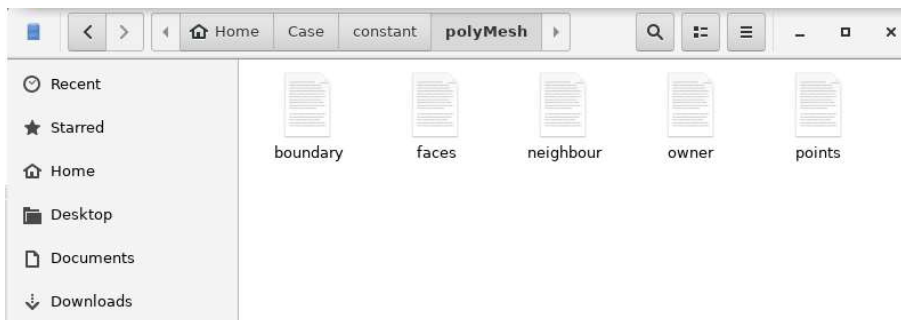


그림 6. polyMesh 디렉토리

### 3. Case directory 입력자료

본 장에서는 realGasReactingEsiFoam을 이용하여 전산해석을 수행하기 위해 필요한 디렉토리 구조와 설정 파일을 설명한다. 해석에 필요한 입력 파일은 총 3개의 디렉토리로 구성된다: constant, 0, system. 각 디렉토리에는 물성치, 초기 조건, 격자 정보 및 제어 설정이 포함된다.

#### 3-1. 물성치(constant 디렉토리) 입력

“constant” 디렉토리는 전산해석에 필요한 물리적 상수와 격자 데이터를 저장하며, 다음과 같은 파일 및 디렉토리를 포함한다: polyMesh, chemistryProperties, combustionProperties, thermophysicalProperties, turbulenceProperties 등(그림 7)

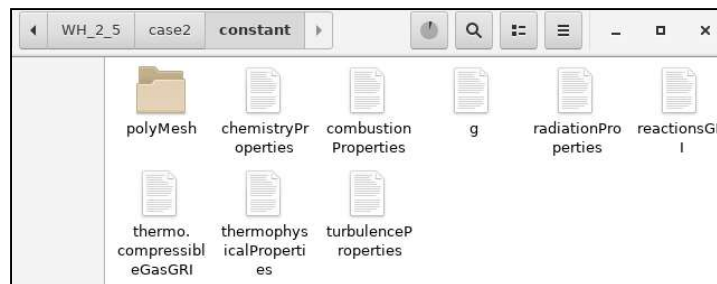


그림 7. 물성치(constant) 디렉토리

##### 3-1-1. polyMesh

이 디렉토리는 앞서 설명한 격자의 구조와 관련된 정보를 포함하며, 점(point), 면(face), 셀(cell) 데이터 및 경계면(boundary) 정보가 저장된다.

##### 3-1-2. chemistryProperties

화학 반응 모델과 관련된 설정 파일이다.

표 4. chemistryProperties

```

/*----- C++ -----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       chemistryProperties;
}
// ***** //

chemistryType
{
    solver       EulerImplicit;
    method       SRKchungTakaStandard;
}

chemistry       on;

initialChemicalTimeStep 1e-07;

EulerImplicitCoeffs
{
    cTauChem     1;
    equilibriumRateLimiter off;
}

odeCoeffs
{
    solver       Rosenbrock43;
    absTol       1e-12;
    relTol       0.01;
}
// ***** //

```

○ ChemistryType

- solver: 화학 반응 계산에 사용되는 수치적 방법을 정의한다. EulerImplicit는 유한차분법을 기반으로 하며, 큰 time step에서도 안정적으로 계산할 수 있다.
- method: 화학 반응 모델을 정의한다. SRKchungTakaStandard는 고압 조건에서의 정확한 반응 계산을 위한 모델이다.

○ Chemistry

이 항목은 화학 반응을 시뮬레이션에 포함할지 여부를 결정한다.

○ initialChemicalTimeStep

초기 화학 반응 time step을 설정한다. 이 값이 작을수록 화학 반응 계산의 정확도가 높아진다.

○ EulerImplicitCoeffs

시스템이 평형 상태에 도달했을 때, 반응 속도를 제한할지 여부를 설정한다.

- cTauChem: 화학 반응시간 척도를 조정한다. 값을 1로 설정했을 때 반응속도를 변경없이 사용하며, 값을 조정하여 반응속도를 빠르게 또는 느리게 조정한다.
- equilibriumRateLimiter: 시스템이 평형 상태에 도달했을 때, 화학 반응 속도를 제한할지 여부를 정의한다.



### 3-1-4. g (Acceleration of Gravity)

중력 가속도를 정의하는 파일로, 다음과 같은 항목을 포함한다.

표 6. g (Acceleration of Gravity)

```
/*-----*- C++ -*-----*/
=====
\\  /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
\\  /  O p e r a t i o n  | Website: https://openfoam.org
\\  /  A n d      | Version: 7
\\  /  M a n i p u l a t i o n  |
/*-----*/
FoamFile
{
  version      2.0;
  format       ascii;
  class        uniformDimensionedVectorField;
  location     "constant";
  object       g;
}
// ***** //

dimensions    [0 1 -2 0 0 0 0];
value         (0 -9.81 0);
// ***** //
```

○ dimension

물리적 차원의 지수를 정의한다. 숫자 값은 해당 차원의 지수를 의미한다. 중력가속도는 길이/시간<sup>2</sup> 이므로 다음과 같이 설정한다.

[질량(kg), 길이(m), 시간(s), 온도(K), 몰(mol), 전류(A), 광도(cd)]

○ value

본 입력인자는 (x, y, z) 축 방향으로의 중력가속도의 크기를 정의한다.

### 3-1-5. radiationProperties

복사 열전달 방정식을 해결하기 위한 매개변수를 정의한다.

표 7. radiationProperties

```

radiation off;

radiationModel none;

fvDOMCoeffs
{
    nPhi      2;      // azimuthal angles in PI/2 on X-Y.(from Y to X)
    nTheta    2;      // polar angles in PI (from Z to X-Y plane)
    tolerance 1e-1;   // convergence tolerance for radiation iteration
    maxIter   1;      // maximum number of iterations
}

// Number of flow iterations per radiation iteration
solverFreq 10;

absorptionEmissionModel greyMeanCombustion;

constantCoeffs
{
    absorptivity 0.01;
    emissivity   0.01;
    E            0;
}

greyMeanCombustionCoeffs
{
    lookupTableFileName none;

    EhrrCoeff      0.0;

    N2
    {
        Tcommon      200;
        invTemp       true;
        Tlow          0;
        Thigh         5000;

        loTcoeffs
        (
            0
            0
            0
            0
            0
            0
        );
        hiTcoeffs
        (
            0.1
            0
            0
            0
            0
            0
        );
    }
}

scatterModel none;

sootModel none;

```

○ radiation

복사 모델의 사용 여부를 설정하는 항목이다.

○ radiationModel

복사 모델의 유형을 정의한다.

○ fvDOMCoeffs

fvDOM (Finite Volume Discrete Ordinates Method) 설정으로, 복사가 전달되는 각도(nPhi, nTheta), 허용 오차(tolerance), 최대 반복 횟수(maxIter)등을 정의한다.

- nPhi: 복사 방정식을 풀 때, 복사가 전달되는 수평 방향을 몇 개의 구간으로 나누어 계산할지 설정한다. nPhi 값이 클수록 더 많은 구간으로 나누어 방정식을 풀기 때문에, 정확도가 높아지지만 계산 시간이 오래 소요된다.
- nTheta: 복사 방정식을 풀 때, 복사가 전달되는 수직 방향을 몇 개의 구간으로 나누어 계산할지를 설정한다. nTheta 값이 클수록 더 많은 구간으로 나누어 방정식을 풀기 때문에, 정확도가 높아지지만 계산 시간이 많이 소요된다.
- tolerance: 복사 방정식 계산과정에서 허용되는 수렴 오차 허용치를 의미한다. tolerance 값이 작을수록 정확한 해를 구할수 있지만 계산시간이 많이 소요된다.
- maxIter: 복사 방정식 계산과정에서 허용되는 최대 반복 횟수를 의미한다. 방정식 계산 반복 횟수가 maxIter에 설정된 값에 도달하면 계산이 중단된다. 값이 작으면 계산이 빨리 중단될 수 있지만, 수렴하지 않은 상태에서 계산이 중단될수 있기 때문에, 복잡한 열전달 상황에서는 maxIter 값을 크게 설정해야 한다.

○ solverFreq

복사 방정식을 얼마나 자주 해결할지를 나타낸다. 시뮬레이션에서 복사 열전달 방정식을 풀기 위한 빈도를 측정한다.

○ absorptionEmissionModel

흡수 및 방출을 처리하는 모델을 설정하는 항목이다. 복사 방정식에서 물질의 흡수 및 방출 특성을 고려할 수 있다.

○ constantCoeffs

흡수율과 방출율에 관련된 상수를 설정하는 항목이다.

- absorptivity: 흡수율을 나타낸다. 물질이 얼마나 복사에너지를 흡수하는지를 나타낸다.
- emissivity: 방출율을 나타낸다. 물질이 복사에너지를 얼마나 방출하는지를 나타낸다.
- E: 복사에너지에 관련된 계수로, 복사에너지의 에너지 전이에 영향을 미친다.

○ greyMeanCombustionCoeffs

grey body 방사에 대한 매개변수를 설정하는 부분이다.

- lookUpTableFileName: grey body 복사 특성을 저장한 테이블 파일의 이름을 지정한다.
- EhrCoeff: 화학 반응에 의해 방출되는 복사 에너지를 조절하는 계수이다.
- E: 복사에너지를 나타내는 계수로, 열 방출에 영향을 미친다.
- Tcommon: 공통온도를 나타낸다.

- invTemp: 온도의 역수를 의미하는 값으로, 온도에 대한 복사 특성을 정의할 때 사용된다.
- Tlow: 낮은 온도구간을 설정한다.
- Thigh: 높은 온도구간을 설정한다.
- loTcoeffs: 낮은 온도에서의 계수를 설정한다.
- hiTcoeffs: 높은 온도에서의 계수를 설정한다.

○ scatterModel

산란 모델을 정의하는 항목이다. 복사 열전달 시 산란 현상을 고려할 때 사용된다.

○ sootModel

soot 모델을 정의하는 항목이다. 연소 시 soot 생성과 관련된 복사 모델을 사용할 때 설정한다.

### 3-1-6. reactionGRI

이 항목은 시뮬레이션 내 화학반응을 정의하는 파일이다. 화학종(Species)와 반응(reaction) 특성을 정의한다.

표 8. reactionGri

```

/*----- C++ -----*/
=====
// Field      | OpenFOAM: The Open Source CFD Toolbox
// Operation  | Website: https://openfoam.org
// And       | Version: 7
// Manipulation

/*-----*/
//
//          SETUP
// Chemkin format uses these units:
//   mol, cm^3, s, K; cal
// OpenFOAM native format uses:
//   kmol, m^3, s, K; K
// The native format is what this file is using.
// *****

FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       reactions;
}
// *****

species
4
(
  O2
  H2
  H2O
  N2
);

reactions
{
  hydrogenReaction
  {
    type      irreversibleArrheniusReaction;
    reaction  "2H2 + O2 = 2H2O";
              A      9.87e08;
              beta   0;
              Ta     3728.42;
  }
}
// *****

```

○ Species

전산해석에 사용되는 화학종을 정의한다.

○ reactions

화학 반응에 대한 세부 정보를 정의하는 부분이다.

- type: 반응의 유형을 지정하는 부분이다. irreversibleArrheniusReaction은 아레니우스 방정식에 기반한 불가역적 화학 반응으로, 한 방향으로만 진행되는 반응을 모델링할 때 사용된다. 반응 속도를 예측하는 데 매우 유용하며, 연소와 같은 불가역 반응을 시뮬레이션할 때 자주 사용된다.
- reaction: 구체적인 반응식을 정의한다. 화학 반응에서 반응물과 생성물의 관계를 나타낸다.
- A: 반응 속도 상수의 전제곱계수(pre-exponential factor)를 나타낸다.
- beta: 온도 의존성을 나타내는 지수로, 반응 속도가 온도에 따라 어떻게 변화하는지를 결정한다.
- Ta: 활성화 에너지에 해당하는 온도 계수로, 반응이 일어나기 위해 필요한 최소 에너지를 설정한다.

### 3-1-7. thermocompressibleGasGRI

전산해석에서 기체의 열역학적 및 물리적 특징을 정의한다.

표 9. thermocompressibleGasGRI

```
02
{
  specie
  {
    molWeight 31.9988;
  }
  thermodynamics
  {
    Tlow 0;
    Thigh 5000;
    Tcommon 1000;
    highCpCoeffs ( 3.28253784 0.00148308754 -7.57966609e-07 2.09470555e-10 -2.16717794e-14 -1088.45772 5.45323129 );
    lowCpCoeffs ( 3.78245636 -0.00299673416 9.84730201e-06 -9.68129509e-09 3.24372837e-12 -1063.94356 3.65767573 );
  }
  transport
  {
    As 1.512e-06;
    Ts 120;
  }
  elements
  {
    O 2;
  }
  equationOfState
  {
    Tc 154.6;
    Vc 73.4e-03;
    Zc 0.288;
    Pc 50.43e05;
    omega 0.022;
  }
}
```

○ species

기체를 구성하는 화학 종에 대한 특성을 정의한다.

- molweight: 각 화학종의 분자량을 정의한다.

○ thermodynamics

기체의 열역학적 특성을 정의한다.

- Tlow: 열역학 계산에 적용할 수 있는 가장 낮은 온도이다.

- Thigh: 열역학 계산에 적용할 수 있는 가장 높은 온도이다.

- Tcommon: 중간 온도 범위를 나타낸다.

- highCpCoeffs: 높은 온도에서 비열을 계산하기 위한 계수이다.

- lowCpCoeffs: 낮은 온도에서 비열을 계산하기 위한 계수이다.

○ transport

Transport Equation (수송 방정식)에 사용되는 상수를 정의한다.

- As: viscosity constant (점성 상수)

- Ts: Sutherland constant (서덜랜드 상수)

○ elements

기체 내 포함된 화학 원소를 정의한다.

○ equationOfState

상태 방정식의 압력, 부피, 온도 사이의 관계를 정의한다.

- Tc: critical temperature (임계 온도)

- Vc: critical volume (임계 부피)

- Zc: compressibility factor (압축인자)

- Pc: critical pressure (임계 압력)

- omega: acentric factor (비대칭성 인자)

### 3-1-8. thermophysicalProperties

열역학적 특성과 수송 모델을 정의한다.



- equationOfState: 상태방정식의 모델을 정의한다. realGas는 실제 기체 상태방정식을 사용하여 유체의 물리적 상태(압력, 부피, 온도)를 계산한다.

- specie: 화학 종의 특성을 정의한다. rfSpecie는 rfJanaf 모델과 함께 사용되는 화학 종의 특성을 정의한다.

○ inertSpecie

비활성 화학 종을 정의한다. 비활성 화학 종은 화학 반응에는 참여하지 않지만, 열역학적 또는 수송 특성에 영향을 줄 수 있는 물질이다.

○ ChemistryReader

화학 반응 정보를 읽어드리는 방식을 정의한다. foamChemistryReader은 OpenFOAM의 기본 화학 반응 정보 읽기 메커니즘이다.

○ foamChemistryFile

화학 반응 메커니즘을 정의한 파일의 경로를 지정한다.

○ foamChemistryThermoFile

화학 종의 열역학적 데이터를 포함한 파일의 경로를 지정한다.

### 3-1-9. turbulenceProperties

난류 모델링과 관련된 설정을 정의한다.

표 11. turbulenceProperties

```

/*-----* C++ *-----*/
=====
// \ / F i e l d           | OpenFOAM: The Open Source CFD Toolbox
//  \| O p e r a t i o n   | Website:  https://openfoam.org
//   \| A n d              | Version:   7
//    \| M a n i p u l a t i o n
/*-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "constant";
    object       turbulenceProperties;
}
// *****

simulationType RAS;

RAS
{
    RASModel     kOmegaSST;

    turbulence   on;

    printCoeffs on;
}

// *****

```

○ simulationType

난류 모델링 방식을 정의한다. RAS(Reynolds-Averaged Simulation, RANS)의 줄임말로, 난류

흐름을 평균하여 계산하는 방식이다. 이 모델은 공학적 유체 흐름 시뮬레이션에서 널리 사용되며, 복잡한 난류 흐름을 다루는 데 적합하다.

○ RAS

- RASModel: KOmegaSST 모델은  $k-\omega$  SST(Shear Stress Transport) 모델을 의미한다. 이 모델은 난류 경계층 근처의 흐름을 정확하게 모델링할 수 있고, 특히 유동 박리와 같은 복잡한 현상에서 자주 사용된다. 이 모델은  $k-\epsilon$  와  $k-\omega$  모델의 장점을 결합한 방식으로, 경계층 안쪽에서는  $k-\omega$ 를 사용하고, 바깥쪽에서는  $k-\epsilon$ 을 사용하는 방식이다.
- turbulence: 난류 모델링을 사용할지 여부를 결정한다.
- printCoeffs: 난류 모델 계수를 출력할지 여부를 결정한다.

### 3-2. 초기 및 경계조건(0 디렉토리) 입력

“0” 디렉토리는 전산해석에 필요한 입력인자 또는 필드의 초기 조건(initial condition)과 경계조건(boundary condition)을 정의하는 파일이 저장되는 디렉토리이다.

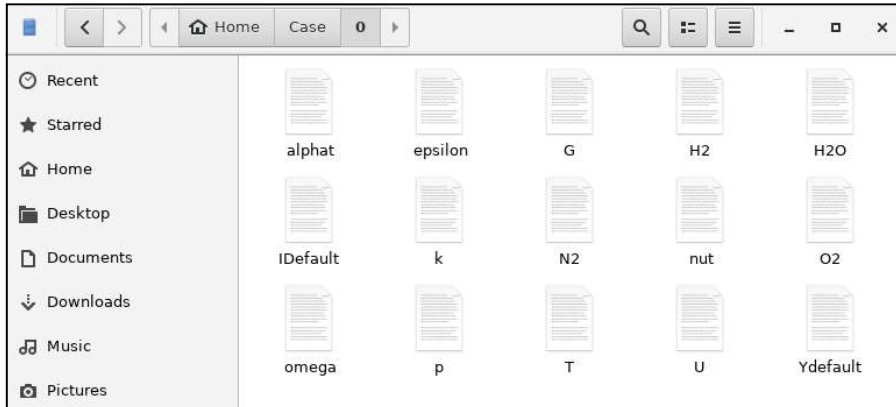


그림 8. 초기 및 경계조건(0) 디렉토리

#### 3-2-1. p (Pressure)

본 입력 인자는 표 12와 같이 계산 영역 내 압력 필드에 대한 초기 조건을 지정한다.

표 12. p (pressure) 초기 및 경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       p;
}
// *****

dimensions    [1 -1 -2 0 0 0];
internalField uniform 101325;

boundaryField
{
  base
  {
    type       zeroGradient;
  }
  outlet
  {
    type       waveTransmissive;
    gamma      1.4;
    fieldInf   101325;
    lInf       100;
    value      uniform 101325;
  }
  symmetry
  {
    type       symmetryPlane;
  }
  Blastwall_Front_Face
  {
    type       zeroGradient;
  }
}
    
```

#### ○ dimension

본 입력인자는 물리적 차원을 아래와 같은 순서로 지정하며, 숫자 값은 해당 차원의 지수를 의미한다. 압력의 물리적 차원은  $[Kg/m \cdot s^2]$  ( $=[Pa]$ ) 이므로  $[1 -1 -2 0 0 0]$ 으로 입력한다.

- [질량(kg), 길이(m), 시간(s), 온도(K), 몰(mol), 전류(A), 광도(cd)]

○ internalField

본 입력인자는 계산 영역 내 초기조건을 지정한다. uniform 101325는 모든 유동장에 동일하게 101325 Pa의 압력 값을 지정한다. “system/setFieldsDict” 파일을 사용하여 비균일하게 압력을 설정할 수 있다. 본 해석에서는 SetFieldsDict 파일에서 비균일 압력 분포를 설정하여 계산 영역 내 고압 수소탱크를 정의하였다.

○ boundaryField

본 입력인자는 계산 영역 내 열·유동장의 각 경계에서 압력을 어떻게 처리할지 정의한다. zeroGradient 경계 조건은 경계면에서 압력의 기울기가 0임을 의미한다. 즉, 경계에서 압력값이 경계 내부 값과 동일하며, 압력 변화가 없다. symmetryPlane 경계조건은 경계면을 대칭면으로 취급한다. 경계에서 유체의 흐름과 물리적 특성이 대칭적으로 처리되며, 경계 자체에서는 흐름이 없는 것으로 간주한다.

### 3-2-2. T (Temperature)

본 입력 인자는 표 13과 같이 계산 영역 열·유동장에서 온도 필드에 대한 초기조건과 경계조건을 정의한다. 물리적 차원은 [K] 이며, 초기 온도 조건은 290K(약 16°C)이다. 비균일한 온도분포의 경우에 “system/setFieldsDict” 파일을 사용하여 설정할 수 있다. 본 해석에서는 SetFieldsDict 파일에서 비균일 온도 분포를 설정하여 계산 영역 내 고압 수소탱크를 정의하였다.

표 13. T(Temperature) 초기 및 경계조건

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       T;
}
// *****

dimensions    [0 0 0 1 0 0 0];
internalField uniform 290;
boundaryField
{
  base
  {
    type       zeroGradient;
  }
  outlet
  {
    type       zeroGradient;
  }
  symmetry
  {
    type       symmetryPlane;
  }
  Blastwall_Front_Face
  {
    type       zeroGradient;
  }
}
```

### 3-2-3. U (Velocity)

본 입력 인자는 표 14와 같이 계산 영역 내 속도 필드를 정의한다. 물리적 차원은 [m/s]

이므로 다음과 같이 설정한다. 초기 계산 영역 내부의 속도는 0이다. noSlip 경계 조건은 경계면에서의 유체의 속도가 경계와 동일하다는 것을 의미한다. 즉, 벽면에서 유체는 정지 상태가 된다. waveTransmissive 경계 조건은 파동이 경계를 통과할 수 있게 하며, 속도 필드에서 속도가 경계를 자유롭게 통과할 수 있도록 한다.

gamma는 기체의 비열비를 나타낸다. lInf는 파동이 경계를 넘어가면서 감쇠되는 길이를 나타낸다. 값이 클수록 파동이 경계를 넘어가면서 더 천천히 감쇠된다. fieldInf는 경계 바깥쪽의 속도값을 나타낸다. (0 0 0)은 경계 바깥에서 유체가 정지되어 있다고 가정한다.

표 14. U(Velocity) 초기 및 경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volVectorField;
  object       U;
}
// ***** //

dimensions    [0 1 -1 0 0 0];

internalField uniform (0 0 0);

boundaryField
{
  base
  {
    type        noSlip;
  }
  outlet
  {
    type        waveTransmissive;
    value       uniform (0 0 0);
    field       U;
    gamma       1.40; // Air is 1.4, H2 is 1.41
    phi         phi;
    psi         thermo:psi;
    lInf        100;
    fieldInf    (0 0 0);
  }
  symmetry
  {
    type        symmetryPlane;
  }

  Blastwall_Front_Face
  {
    type        noSlip;
  }
}

```

### 3-2-4. H2 (Mass Fraction of Hydrogen)

본 입력 인자는 표 15와 같이 계산 영역 내 수소 기체의 초기조건과 경계조건을 정의한다. 수소 기체의 질량 비율을 설정하기 때문에, 물리적 차원을 정의하지 않는다. internalField 값이 0이므로 대기 중에 수소 기체가 존재하지 않는다고 가정한다. “system/setFieldsDict” 파일을 사용하여 비균일하게 압력을 설정할 수 있다. 본 해석에서는 SetFieldsDict 파일에서 비균일 수소 기체의 분포를 설정하여 계산 영역 내 고압 수소탱크를 정의하였다.

표 15. H2(Mass fraction of Hydrogen) 초기 및  
경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       H2;
}
// *****

dimensions     [0 0 0 0 0 0 0];

internalField  uniform 0.0; // from Marcantoni

boundaryField
{
  base
  {
    type        fixedValue;
    value       uniform 0.0;
  }
  outlet
  {
    type        zeroGradient;
  }
  symmetry
  {
    type        symmetryPlane;
  }
  Blastwall_Front_Face
  {
    type        fixedValue;
    value       uniform 0.0;
  }
}

```

### 3-2-5. N2 (Mass Fraction of Nitrogen)

본 입력 인자는 표 16과 같이 계산 영역 내 질소 기체의 초기조건과 경계조건을 정의한다. 질소 기체의 질량 비율을 설정하기 때문에, 물리적 차원을 정의하지 않는다. internalField 값은 0.767으로 대기 중에 질소 기체가 차지하는 비율을 의미한다.

표 16. N2(Mass fraction of Nitrogen) 초기 및  
경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       N2;
}
// *****

dimensions     [0 0 0 0 0 0 0];

internalField  uniform 0.767;

boundaryField
{
  base
  {
    type        fixedValue;
    value       uniform 0.0;
  }
  outlet
  {
    type        zeroGradient;
  }
  symmetry
  {
    type        symmetryPlane;
  }
  Blastwall_Front_Face
  {
    type        fixedValue;
    value       uniform 0.0;
  }
}

```

### 3-2-6. O2 (Mass Fraction of Oxygen)

본 입력 인자는 표 17과 같이 계산 영역 내 질소 기체의 초기 조건과 경계조건을 정의한다. 산소 기체의 질량 비율을 설정하기 때문에, 물리적 차원을 정의하지 않는다. internalField 값은 0.233으로 대기 중에 산소 기체가 차지하는 비율을 의미한다.

표 17. O2(Mass fraction of Oxygen) 초기 및 경계조건

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       O2;
}
// ***** //
dimensions    [0 0 0 0 0 0 0];
//internalField  uniform 0.22781;
internalField  uniform 0.233; // from Marcantoni

boundaryField
{
  base
  {
    type        fixedValue;
    value       uniform 0.0;
  }
  outlet
  {
    type        zeroGradient;
  }
  symmetry
  {
    type        symmetryPlane;
  }
}
Blastwall_Front_Face
{
  type         fixedValue;
  value        uniform 0.0;
}
```

### 3-2-7. H2O (Mass Fraction of water)

본 입력 인자는 표 18와 같이 계산 영역 내 물의 초기조건과 경계조건을 정의한다. 물의 질량 비율을 설정하기 때문에, 물리적 차원을 정의하지 않는다. internalField 값이 0이므로 대기 중에 물이 존재하지 않는다고 가정한다.

표 18. H20(Mass fraction of H20) 초기 및 경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       H20;
}
// *****

dimensions    [0 0 0 0 0 0];
internalField uniform 0;

boundaryField
{
  base
  {
    type        fixedValue;
    value       uniform 0.0;
  }
  outlet
  {
    type        zeroGradient;
  }
  symmetry
  {
    type        symmetryPlane;
  }
  Blastwall_Front_Face
  {
    type        fixedValue;
    value       uniform 0.0;
  }
}

```

### 3-2-8. alphas (Turbulent Thermal Diffusivity)

본 입력 인자는 표 19와 같이 계산 영역 내 난류 열 확산 계수를 나타내며, 물리적 차원은  $[m^2/s]$ 이다. 초기값은 0으로 outlet value의 \$internalField는 outlet경계에서 alphas 값이 0임을 나타낸다.

compressible::alphaWallFunction 경계조건은 벽 근처에서 난류 열 확산을 계산하는 함수이다. alphaWallFunction은 벽 근처의 난류 효과를 고려하여 난류 열 확산 계수를 계산하는데 사용된다. Prt는 Prandtl number로 동점성계수( $\nu$ )와 열확산계수( $\alpha$ )의 비율을 의미한다.

표 19. alphas(Turbulent Thermal Diffusivity) 초기 및 경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       alphas;
}
// *****

dimensions    [1 -1 -1 0 0 0 0];

internalField uniform 0;

boundaryField
{
  outlet
  {
    type        calculated;
    value       $internalField;
  }

  symmetry
  {
    type        symmetryPlane;
  }

  base
  {
    type        compressible::alphasWallFunction;
    Prt         0.85;
    value       $internalField;
  }

  Blastwall_Front_Face
  {
    type        compressible::alphasWallFunction;
    Prt         0.85;
    value       $internalField;
  }
}

```

### 3-2-9. epsilon (Turbulent dissipation rate)

본 입력 인자는 표 20과 같이 계산 영역 내 난류 소산율, 즉 난류 운동에너지가 시간에 따라 소산되는 속도를 의미한다. 물리적 차원은  $[m^2/s^3]$ 로 초기 값은 200이다. epsilonWallFunction 경계조건은 벽 근처에서 난류 소산율을 계산하는 데 사용되는 함수이다.

$C_{\mu}(C_{\mu})$ 는 난류 모형에서 자주 사용되는 상수로,  $k-\omega$  모형에서 난류 점성 계수와 난류 소산율을 계산할 때 사용된다.  $\kappa(\kappa)$ 는 카르만 상수로, 벽 근처의 속도 profile을 설명할 때 사용된다.  $E$ 는 벽 함수에서 사용되는 상수로, 벽 근처에서의 유동 변화를 보정하기 위한 값이다.

표 20. epsilon(Turbulent dissipation rate) 초기 및 경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       epsilon;
}
// *****

dimensions     [0 2 -3 0 0 0];

internalField  uniform 200;

boundaryField
{
  base
  {
    type        epsilonWallFunction;
    Cmu         0.09;
    kappa       0.41;
    E           9.8;
    value       uniform 200;
  }
  outlet
  {
    type        zeroGradient;
  }

  symmetry
  {
    type        symmetryPlane;
  }

  Blastwall_Front_Face
  {
    type        zeroGradient;
  }
}

```

### 3-2-10. G (Radiation Energy Density)

본 입력인자는 표 21같이 계산 영역 내 복사 에너지 밀도를 설정하는데 사용되며, 물리적 차원은  $[kg/s^3]$ 로, 초기값은 0으로 설정되어 있다.

표 21. G(Radiation Energy Density) 초기 및 경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  object       G;
}
// *****

dimensions     [1 0 -3 0 0 0];

internalField  uniform 0;

boundaryField
{
  "*"
  {
    type        MarshakRadiation;
    emissivityMode  lookup;
    emissivity  uniform 1.0;
    value       uniform 0;
  }

  symmetry
  {
    type        symmetryPlane;
  }
}
}

```

### 3-2-11. IDefault (Default Radiation Intensity)

본 입력 인자는 표 22과 같이 계산 영역 내 복사열 강도를 설정하는데 사용되며, 복사열의 에너지를 의미한다. 물리적 차원은  $[\text{kg}/\text{s}^3]$ 로, 초기값은 0으로 설정되어 있다.

표 22. IDefault (Default Radiation Intensity) 초기 및 경계조건

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  object       IDefault;
}
// ***** //
dimensions    [1 0 -3 0 0 0 0];
internalField uniform 0;
boundaryField
{
  "*"
  {
    type          greyDiffusiveRadiation;
    emissivityMode lookup;
    emissivity    uniform 1;
    value         uniform 0;
  }
  symmetry
  {
    type          symmetryPlane;
  }
}
}
```

### 3-2-12. k (Turbulent Kinetic Energy)

본 입력 인자는 표 23과 같이 계산 영역 내 난류 운동에너지를 의미한다. 난류 운동에너지는 난류 흐름에서 유체의 속도 변동에 의해 발생하는 운동에너지를 설명하는 물리적 변수이다. 물리적 차원은  $[m^2/s^2]$ 이며 초기 값은 1로 설정되어있다.

kqRWallFunction 경계조건은 벽 근처에서 난류 운동에너지를 계산하기 위한 경계조건이다. 벽 근처에서는 유동이 느려지고, 에너지가 소산되거나 감쇠하는 경향이 있기 때문에, 이를 정확하게 모델링하기 위해 kqRWallFunction이 사용된다.

표 23. k(Turbulent kinetic energy) 초기 및 경계 조건

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       k;
}
// *****

dimensions    [0 2 -2 0 0 0 0];
internalField uniform 1;
boundaryField
{
  base
  {
    type        kqRWallFunction;
    value       uniform 1;
  }
  outlet
  {
    type        zeroGradient;
  }
  symmetry
  {
    type        symmetryPlane;
  }
  Blastwall_Front_Face
  {
    type        zeroGradient;
  }
}
```

### 3-2-13. nut (Turbulence Viscosity)

본 입력 인자는 표 24와 같이 계산 영역 내 난류 점성계수를 의미한다. 물리적 차원은  $[m/s^2]$  이고 초기 값은 0이다. 난류 점성계수는 유체의 난류 상태에서 점성이 얼마나 커지는 지를 나타내며, 난류로 인해 유체의 혼합, 에너지 전달, 열 전달 등이 더 활발하게 일어나는 현상을 설명하는 변수이다. nutkWallFunction 경계조건은 경계면에서 난류 점성계수를 계산하기 위한 함수이다. 벽 근처의 난류 운동에너지( $\kappa$ )와 난류 소산율(epsilon)을 이용하여 난류로 인한 점성 증가를 계산하며, 이를 통해 경계 근처의 난류 흐름을 모델링한다.

표 24. nut(Turbulence viscosity) 초기 및 경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       nut;
}
// *****

dimensions    [0 2 -1 0 0 0 0];

internalField uniform 0;

boundaryField
{
  base
  {
    type          nutkWallFunction;
    value         uniform 0;
  }
  outlet
  {
    type          calculated;
    value         uniform 0;
  }
  symmetry
  {
    type          symmetryPlane;
  }

  Blastwall_Front_Face
  {
    type          calculated;
    value         uniform 0;
  }
}

```

### 3-2-14. omega (Specific Dissipation Rate)

본 입력 인자는 표 25와 같이 계산 영역 내 비소산율을 나타낸다. 물리적 차원은  $s^{-1}$  이고 초기 값은 18이다. omega는 k- $\omega$  모델에서 사용되는 변수로 난류 운동에너지가 소산되는 속도를 나타낸다.

표 25. omega(Specific Dissipation Rate) 초기 및 경계조건

```

FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  object       omega;
}
// *****

dimensions    [0 0 -1 0 0 0 0];

internalField uniform 18;

boundaryField
{
  base
  {
    type          omegaWallFunction;
    value         uniform 18;
  }
  outlet
  {
    type          zeroGradient;
  }
  symmetry
  {
    type          symmetryPlane;
  }

  Blastwall_Front_Face
  {
    type          zeroGradient;
  }
}

```

### 3-2-15. Ydefault (Default Chemical Mass Fraction)

본 입력 인자는 표 26과 같이 각 경계조건에서 화학종의 질량 분율을 초기화하는 데 사용된다.

표 26. Ydefault (Default Chemical Mass fraction)

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  location     "0";
  object       Ydefault;
}
// ***** //

dimensions    [0 0 0 0 0 0];
internalField uniform 0.0;

boundaryField
{
  base
  {
    type      fixedValue;
    value     uniform 0.0;
  }
  outlet
  {
    type      zeroGradient;
  }
  symmetry
  {
    type      symmetryPlane;
  }
  Blastwall_Front_Face
  {
    type      fixedValue;
    value     uniform 0.0;
  }
}
```

### 3-3. 계산제어(system 디렉토리) 입력

“system” 디렉토리에는 전산 해석에 대한 설정을 제어하고, 계산 방법 및 경계조건에 대한 설정파일을 포함하는 디렉토리이다.

#### 3-3-1. controlDict

본 파일의 내용은 전산해석 제어 설정을 정의하는 파일이다. 전산 해석을 어떻게 시작하고, 진행되며. 결과를 기록할지에 대한 설정을 포함한다.

표 27. controlDict

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "system";
  object       controlDict;
}
// ***** //

application    realGasEslFoam;

startFrom      latestTime;

startTime      0;

stopAt         endTime;

endTime        1.0;

deltaT         1e-10;

writeControl   adjustableRunTime;

writeInterval  0.0001 ;

purgeWrite     0;

writeFormat    binary;

writePrecision 6;

writeCompression off;

timeFormat     general;

timePrecision  6;

runTimeModifiable true;

adjustTimeStep yes;

maxCo          0.1;

maxDeltaT      0.1;

maxAcousticCo 0.1;

functions
{
  #include "surfaces"

  MachNumber
  {
    type          MachNo;
    libs          ("libfieldFunctionObjects.so");
    executeControl  timeStep;
    writeControl   writeTime;
  }

  probes_Pressure
  {
    type          probes;
    libs          ("libsampling.so");
    writeControl  adjustableRunTime;
    writeInterval 0.00001;
    probeLocations
    (
      ( 0.0000 0.98 0.0000)
      ( 0.1000 0.98 0.0000)
      ( 0.2000 0.98 0.0000)
      ( 0.3000 0.98 0.0000)
      ( 0.4000 0.98 0.0000)
      ( 0.5000 0.98 0.0000)
      ( 0.6000 0.98 0.0000)
      ( 0.7000 0.98 0.0000)
      ( 0.8000 0.98 0.0000)
    )
  }
}
```

○ application

전산 해석을 실행할 solver의 이름을 지정하는 항목이다,

○ startFrom

전산 해석을 이전 단계에서 시작하거나 새로운 시점에서 시작할지를 결정하는 항목이다.

○ startTime

전산 해석에서 시작 시간을 지정하는 항목이다.

○ stopAt 및 endTime

stopAt는 전산 해석을 종료할 시간을 지정하는 항목이다. endTime은 0.1으로 설정되어있다.

○ deltaT

전산 해석에서 계산시간의 간격을 지정하는 명령어로, 계산의 정확도와 계산 시간을 결정하는 중요한 변수이다. 본 전산 해석에서는 deataT =  $1e-10$  으로 설정되어있다.

○ writeControl 및 writeInterval

“writeControl” 은 전산 해석 결과를 어떤 기준으로 저장할지를 정의하는 항목이다. 본 전산 해석에서는 “AdjustableRunTime” 으로 설정되어 전산해석 내 시간을 기준으로 결과를 저장한다. “writeInterval” 은 “writeControl” 에서 설정한 기준에 따라 결과를 저장할 주기를 정의하는 항목이다. 본 전산 해석에서는 0.0001간격으로 결과를 저장한다.

○ purgeWrite 및 writeFormat

“purgeWrite” 는 저장된 파일을 유지 또는 삭제할 지를 결정하는 항목이다. “0” 으로 설정된 경우, 모든 출력 파일을 저장한다.

“writeFormat” 은 출력 파일을 이진수 형식(binary) 또는 텍스트 형식(ASCII)로 저장할지 여부를 정의하는 항목이다. “binary” 는 데이터를 이진수 형태로 저장하여 데이터를 바르게 처리할 수 있으며, 저장 공간을 효율적으로 사용할 수 있다.

○ writePrecision 및 writeCompression

“writePrecision” 은 출력 파일에 기록될 소수점 자릿수를 정의하는 항목이다. 본 전산 해석에서는 “6” 으로 설정되어 소수점 6자리까지 값이 저장된다. “writeCompression” 은 출력 파일을 압축하여 저장할지를 결정하는 항목이다.

○ timeFormat 및 timePrecision

“timeFormat” 은 시간 값의 형식을 정의하는 항목이다. “general” 로 설정된 경우 지수 형식으로 저장된다. “timePrecision” 은 시간 값을 저장할 때 소수점 이하 자릿수를 정의하는 항목이다.

○ runTimeModifiable 및 adjustTimeStep

“runTimeModifiable” 은 전산 해석이 실행되는 중에 “controlDict” 파일의 일부를 수정할 수 있는지를 결정하는 항목이다. “adjustTimeStep” 은 “deltaT” 를 자동으로 조정할 지를 설정하는 항목이다. “yes” 로 설정되면, Courant number 같은 안정성 조건을 기반으로

“deltaT” 를 조정하여 수치적 안정성을 유지한다.

○ maxCo 및 maxDeltaT

“macCo” 는 Courant Number의 최댓값을 설정하는 항목이다. Courant Number는 시간 스텝의 크기, 유체의 속도 및 공간 격자 크기 간의 관계를 나타내는 수치적 안정성 지표이며 다음과 같은 식에 의해 결정된다.

$$- Co = u \frac{\Delta t}{\Delta x} \quad (u: \text{유체의 속도}, \Delta t: \text{시간 스텝의 크기}, \Delta x: \text{공간 격자의 크기})$$

“maxDeltaT” 는 deltaT의 최대 값을 지정하는 항목이다. 전산 해석 중 “deltaT” 가 커지면, 세밀한 계산을 할 수 없기 때문에, 전산 해석 결과의 정확도를 보장하기 위해 정의된다.

○ maxAcousticCo

“maxAcousticCo” 는 Acoustic Courant Number의 최댓값을 설정하는 항목이다. Acoustic Courant Number는 음속을 고려하여 유동의 수치적 안정성을 유지한다. “maxAcousticCo” 를 작게 설정하면 시간 스텝이 더 작아져 수치적 안정성이 증가하지만, 계산 시간이 길어진다.

$$- \text{Acoustic Co} = c \frac{\Delta t}{\Delta x} \quad (c: \text{음속}, \Delta t: \text{시간 스텝의 크기}, \Delta x: \text{공간 격자의 크기})$$

○ functions

전산 해석 중에 특정 계산 또는 후처리 작업을 자동으로 수행하도록 정의하는 항목이다.

- type: 함수 객체의 종류를 정의한다. 본 전산 해석에서는 Mach Number, Pressure 등을 계산한다.

- libs: 함수 객체가 사용하는 라이브러리를 지정한다. type에서 정한 함수 객체가 필요한 기능을 제공하는 OpenFOAM 내부 또는 외부의 라이브러리이다.

- executeControl: 함수 객체가 언제 실행될지를 제어한다. “timeStep” 으로 설정되면, 전산 해석의 각 시간 스텝이 진행될 때마다 함수가 실행된다.

- writeControl: 전산 해석 결과를 어떤 기준으로 저장할지를 정의하는 항목이다. 본 “AdjustableRunTime” 는 전산 해석 내 시간을 기준으로 결과를 저장한다.

- writeInterval: “writeControl” 에서 설정한 기준에 따라 결과를 저장할 주기를 정의하는 항목이다. 본 전산 해석에서는 0.0001간격으로 결과를 저장한다.

- probeLocation: 특정 위치에서 데이터를 추출할 때 사용하는 항목이다. (x,y,z) 좌표를 입력하여 해당 지점에서의 시간에 따른 물리량을 저장한다.

### 3-3-2. createPatchDict

본 파일은 경계 patch를 생성하거나 수정하는 데 사용되는 설정 파일이다.

표 28. createPatchDict

```
pointSync false;
// Optional: Write cyclic matches into .obj format; defaults to false.
writeCyclicMatch false;

// Patches to create.
patches
(
  {
    // Name of new patch
    name inlet;

    // Type of new patch
    patchInfo
    {
      type patch;
    }

    // How to construct: either from 'patches' or 'set'
    constructFrom set;

    // If constructFrom = patches : names of patches. Wildcards allowed.
    patches ("periodic.*");

    // If constructFrom = set : name of faceSet
    set f0;
  }
);
```

#### ○ pointSync

주기적 경계(cyclic boundary)의 점을 동기화할지 여부를 설정한다. true로 설정하면 경계의 점들이 동기화 되어 주기적 조건이 정확히 적용된다. 본 코드에서는 False로 설정하였다.

#### ○ writeCyclicMatch

주기적 경계를 “.obj” 형식으로 저장할지 여부를 설정한다. true로 설정하면 주기적 경계 매칭정보가 “.obj” 파일로 저장된다. 본 코드에서는 False로 설정하였다.

#### ○ patches

생성하거나 수정할 patch의 목록을 정의하는 섹션이다. 이 섹션에서는 각 patch의 이름, 유형, 구성 방법 등을 설명한다.

- name: 새로 생성될 patch의 이름을 지정한다.

- patchInfo

- type: patch의 유형을 정의한다.
- constructFrom: patch를 생성하는 방법을 지정한다.
- patches: 기존의 경계면을 사용하여 새로운 patch를 만들 때 사용된다.  
( “periodic.\*” )으로 설정하면 이름이 periodic으로 시작하는 모든 patch가 선택된다.
- set: faceSet 명령어를 사용해 생성된 face set을 기반으로 새로운 패치를 만들 때 사용된다. 본 코드에서는 f0로 설정되어 f0라는 이름의 face set에 포함된 면들이 새로운 패치로 정의된다.

### 3-3-3. decomposeParDict

본 파일은 병렬 계산을 위해 격자를 여러 하위 영역(sub-domains)으로 분할하는 설정을 정의한다.

표 29. decomposeParDict

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "system";
  object       decomposeParDict;
}
// *****

numberOfSubdomains 64;

method            hierarchical;

simpleCoeffs
{
  n               (8 4 7);
  delta           0.001;
}

hierarchicalCoeffs
{
  n               (4 4 4);
  delta           0.001;
  order           xyz;
}

manualCoeffs
{
  dataFile       "";
}

distributed      no;

roots            ( );
```

○ numberOfSubdomains

전산해석을 위해 만들 sub-domains의 개수를 지정한다. 본 코드에서는 64개의 sub-domains을 이용한다. 사용자의 환경에 맞추어 조절하여 사용한다.

○ method

분할 방법을 지정한다. 본 코드에서는 격자를 계층적으로 분할하는 방식인 hierarchical 방법을 사용한다. 이 외에도 simple, scotch, metis 등의 방법이 존재한다.

○ simpleCoeffs

simple method로 분할을 할 때 사용되는 설정이다.

○ hierarchicalCoeffs

hierarchical method로 분할을 할 때 사용되는 설정이다.

- n: 각 방향 (x,y,z)으로 분할 수를 지정한다.

- delta: 각 분할 경계의 공차(tolerance)를 설정한다. 값이 작을수록 더 정확한 분할을 시도한다.

- order: 분할 순서를 지정한다. 본 코드에서는 x, y, z 순서로 분할을 수행한다.

○ manualCoeffs

수동으로 분할을 지정할 때 사용하는 설정이다.

- deltaFile: 수동 분할 정보를 저장한 파일 경로를 지정한다.

○ distributed

병렬 분할을 여러 물리적 노드에 분산할지 여부를 지정한다. yes로 설정하면 분산 환경에서 사용할 수 있다. 본 코드에서는 no로 설정하여, 단일 노드에서 분할이 이루어진다.

○ roots

각 하위 영역의 root 디렉토리를 지정한다. 이 옵션은 distributed가 yes일 때만 필요하며, 여러 노드에 걸쳐 작업을 분산할 때 각 노드의 경로를 설정한다.

### 3-3-4. fvSchemes

본 파일은 유한체적(Finite Volume)방법으로 미분 방정식을 수치적으로 해석할 때, 각 항목의 공간적, 시간적 차분 방식을 정의하는 설정파일이다.

표 30. fvSchemes

```

FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    location     "system";
    object       fvSchemes;
}
// *****

fluxScheme      Kurganov;

ddtSchemes
{
    default      Euler;
}

gradSchemes
{
    default      Gauss linear;
    grad(U)      cellMDLimited Gauss linear 1.0;
    grad(p)      cellMDLimited Gauss linear 1.0;
}

divSchemes
{
    default      none;
    div(tauMC)   Gauss linear;
    div(phi,K)   Gauss limitedLinear 1;
    div(phi,omega) Gauss limitedLinear 1;
    div(phi,epsilon) Gauss limitedLinear 1;
    div(phi,Yi_h) Gauss limitedLinear 1;
    div(Ji,Ii_h) Gauss upwind;
    div(((rho*nuEff)*dev2(T(grad(U)))) Gauss linear;
}

laplacianSchemes
{
    default      Gauss linear corrected;
}

interpolationSchemes
{
    default      linear;
    reconstruct(rho) vanLeer;
    reconstruct(U) vanLeerV;
    reconstruct(T) vanLeer;
    reconstruct(Yi) vanLeer;
}

snGradSchemes
{
    default      corrected;
}

wallDist
{
    method       meshWave;
}

```

○ fluxScheme

유체 흐름의 보존방정식에서 사용하는 flux 계산 방식을 지정한다. 본 코드에서는 kurganov 방식이 사용되었다. Kurganov는 수치 해석에서 고차원 보존 방정식의 해를 계산하는 데 사용되는 중심 차분 방식의 일종이다.

○ ddtSchemes

시간 미분 항의 차분 방식을 정의한다. default 항목에서 모든 시간 미분에 대해 Euler method를 사용한다.

○ gradSchemes

공간 미분에서 구배(Gradient,  $\nabla$ )를 계산하는 방식을 정의한다.

- grad(U): 속도 벡터의 구배를 계산할 때 사용하는 차분방식을 설정한다. 본 코드에서는 cellMDLlimited Gauss linear 1.0으로 설정되어 있다. cellMDLlimited Gauss linear은 구배 계산 시 가우스 적분(Gauss integration) 방식을 사용하여 셀 내부의 구배를 면 중심에서 셀 중심으로 적분하여 계산하고, 선형 보간(linear interpolation)을 통해 셀 중심에서 면 중심으로 값을 보간한다.
- grad(p): 압력의 구배를 계산할 때 사용하는 차분방식을 설정한다. 본 코드에서는 cellMDLlimited Gauss linear 1.0으로 설정되어 있다.

○ divSchemes

발산(Divergence) 연산에 대한 차분 방식을 지정한다.

- div(tauMC): 응력 텐서의 발산을 계산할 때 사용하는 방법을 설정한다. 본 코드에서는 Gauss linear 방법을 사용하여 가우스 적분과 선형보간을 조합하여 발산 연산을 계산한다.
- div(phi, k): 난류 운동에너지 k의 수송 항에서 발산을 계산하는 방법을 설정한다. 본 코드에서는 Gauss limitedLinear 1 방법을 사용한다. 이 방법은 Gauss linear 방법에 수치적 진동을 억제하기 위한 제한자(limiter)를 추가한 방식이다. 1은 수치적 진동에 대한 제한 강도를 의미한다.
- div(phi, omega): 난류 특성  $\omega$ 의 발산을 계산하는 방법을 설정한다. 본 코드에서는 Gauss limitedLinear 1 방법을 사용한다.
- div(phi, epsilon): 난류 소산을  $\epsilon$ 의 발산을 계산하는 방법을 설정한다. 본 코드에서는 Gauss limitedLinear 1 방법을 사용한다.
- div(phi, Yi\_h): 화학종의 질량 분율의 발산을 계산하는 방법을 설정한다. 본 코드에서는 Gauss limitedLinear 1 방법을 사용한다.
- div(Ju, li\_h): 반응 속도를 포함한 화학 종의 발산을 계산하는 방법을 설정한다. 본 코드에서는 Gauss limitedLinear 1 방법을 사용한다.

-  $\text{div}(\text{((rho*nuEff)*dev2(T(grad(U))))})$

: 점성력에 의한 발산 항을 계산하는 방법을 설정한다. 본 코드에서는 Gauss upwind 방법을 사용한다, Gauss upwind는 상류 차분(upwind differencing)을 사용하여 발산을 계산하는 방식으로 안정성을 우선하는 방법이다.

#### ○ laplacianSchemes

라플라시안(Laplacian,  $\nabla^2$ )연산에 대한 차분 방식을 정의한다. default 항목은 모든 라플라시안 연산에 대하여 Gauss linear corrected 방식을 사용한다,

#### ○ interpolationSchemes

격자 내에서 데이터를 보간하는 방식을 정의한다.

- default: 기본 보간 방식을 정의한다. 본 코드에서는 linear 방식으로 설정한다.

- reconstruct(rho): 밀도 값을 셀 중심에서 면 중심으로 보간하여 계산하는 방식이다. 유동장에서 밀도의 분포를 보다 정확하게 모델링하기 위해 사용된다. 본 코드에서는 van Leer 방법을 사용한다. van Leer 방법은 고속 유동이나 압축성 유동에서 높은 정확도를 유지하면서 수치적 진동을 억제하는 방식이다.

- reconstruct(U): 속도 벡터 값을 보간하는 방식으로 유체가 흐르는 방향과 크기에 대한 정보를 정확히 계산하기 위해 사용된다.

- reconstruct(T): 온도 필드를 셀에서 면으로 보간하여 면 중심의 온도 값을 계산한다.

- reconstruct(Yi): 화학 종의 질량 분율을 면 중심에서 보간하여, 화학 반응에 필요한 정확한 농도를 계산하는 데 사용된다.

#### ○ snGradSchemes

surface-normal gradient를 계산하는 방식을 정의한다. surface-normal gradient는 셀 면의 법선 방향으로 스칼라 또는 벡터 필드의 변화율을 계산하는 데 사용된다. 본 코드에서는 correctd로 설정하여, 표면의 법선 방향에 따라 경계면에서 더 정확한 값을 계산한다.

#### ○ wallDist

wall distance를 계산하는 방식을 설정한다. wall distance는 경계면에서 각 셀까지의 거리를 나타내며 특히 경계층이나 벽 근처의 유동 특성을 정확히 반영하기 위해 사용된다. 본 코드에서는 meshWave 방식을 사용한다. meshWave는 격자를 따라 물결처럼 퍼져나가면서 벽으로부터의 거리를 계산하는 방법이다. 이는 격자 내의 셀이 벽과 얼마나 떨어져 있는지를 격자 구조를 따라 파악하는 방식이다.

### 3-3-5. fvSolution

이 파일은 전산해석에 사용되는 수치 solver와 관련된 설정을 정의한다. 각 변수에 대한 솔버의 종류, 수렴 기준, 허용 오차 등을 지정하여, 계산 중에 각 방정식을 어떻게 풀 것인지 구체적으로 제어할 수 있다.

표 31. fvSolution

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "system";
  object       fvSolution;
}
// ..... //
solvers
{
  "(rho|rhoU|rhoE|rhoYi)"
  {
    solver      diagonal;
  }
  U
  {
    solver      smoothSolver;
    smoother    GaussSeidel;
    nSweeps     2;
    tolerance   1e-15;
    relTol     0.01;
  }
  h
  {
    $U;
    tolerance   1e-15;
    relTol     0;
  }
  "(O2|H2|H2O|N2|Yi)"
  {
    $U;
    tolerance   1e-15;
    relTol     0;
  }
  e
  {
    $U;
    tolerance   1e-15;
    relTol     0;
  }
  "(k|epsilon|omega)"
  {
    $U;
    tolerance   1e-15;
    relTol     0;
  }
  "Ii.*"
  {
    solver      GAMG;
    tolerance   1e-4;
    relTol     0;
    smoother    symGaussSeidel;
    maxIter     1;
    nPostSweeps 1;
  }
  "G.*"
  {
    solver      PCG;
    preconditioner DIC;
    tolerance   1e-04;
    relTol     0;
  }
}
CENTRAL
{
}
```

#### ○ Solver

각 변수(rho, U, P 등)에 대해 선형 방정식을 풀기 위해 사용되는 수치적 해석 방법을 설정하는 항목이다.

- diagonal: 이 solver는 대각 행렬을 풀 때 사용되며, 계산이 효율적이고 속도가 빠르다.
- smootherSolver: 이 solver는 수렴성을 높이기 위해 여러 번 반복하여 계산의 정확도를 높이는 방식으로, 비정렬 격자나 경계가 복잡한 영역에서 수렴성을 개선한다.

- GAMG: Generalized Algebraic Multi-Grid을 사용하여 계산의 반복 횟수를 줄이고, 대규모 연립 방정식을 풀 때 효율적이다.

- PCG: Preconditioned Conjugate Gradient solver로 전처리 기법을 통해 계산의 수렴성을 크게 개선하며, 큰 값의 변수를 다루는 문제에 적합하다.

○ smoother

연립 방정식을 풀 때 남아있는 오차를 줄이기 위해 사용하는 항목이다. 주로 다중 그리드 (Multi-grid)솔버에서 사용된다.

- GaussSeidel: 연립 방정식을 반복적으로 풀어 나가며, 이전 값에 대한 정보를 사용하여 차례로 계산하는 방식이다. 수렴 속도가 빠르고 수렴 안정성이 높은 편이다.

○ nSweeps

각 반복에서 얼마나 많은 smoothing 과정을 수행할지 결정하는 항목이다.. 값을 높게 설정할 수록 smoothing이 더 많이 수행되어 오차가 줄어들지만. 계산 시간이 증가한다.

○ tolerance

solver의 절대 허용 오차를 설정하는 항목이다. 계산 결과가 설정된 값보다 작아지면, 수렴했다고 판단하여 반복계산을 중지한다.

○ relTol

solver의 상대 허용 오차를 나타내며, tolerance와 유사하지만, 현재 계산 단계의 상대적인 오차를 기준으로 수렴을 판단하는 항목이다.

○ maxIter

solver가 한 단계에서 수행할 수 있는 최대 반복 횟수를 설정하는 항목이다. 한 단계에서 계산의 최대 반복 횟수에 도달하기 전에 수렴하지 않으면, solver는 해당 단계에서 반복을 중지하고 다음 단계로 넘어간다.

○ nPostSweeps

Multi-grid 방식에서 한 레벨 계산 후 남아있는 오차를 줄이기 위해 추가로 수행하는 smoothing의 횟수를 설정하는 항목이다.

○ preconditioner

연립 방정식을 풀 때, 계산 효율성을 높이고 수렴성을 개선하기 위해 사용하는 전처리 기법을 설정하는 항목이다.

- DIC: Diagonal Incomplete Cholesky 방식으로, 대각 요소만을 사용하여 전처리를 수행하여 계산량이 줄어들고, 수렴 속도를 높이는 데 도움을 준다.

### 3-3-6. setFieldsDict

표 32. setFieldsDict

```

=====
\\ \ / F i e l d
\\ \ / O p e r a t i o n
\\ \ / A n d
\\ \ / M a n i p u l a t i o n
=====
OpenFOAM: The Open Source CFD Toolbox
Website: https://openfoam.org
Version: 7
=====
FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "system";
  object       setFieldsDict;
}
// *****

defaultFieldValues
(
  volScalarFieldValue T 300
  volScalarFieldValue p 101325
);

regions
(
  cylinderToCell
  {
    p1 (0 0.98 -0.75);
    p2 (0 0.98 0.75);
    radius 0.18;

    fieldValues
    (
      volScalarFieldValue p 76269000
      volScalarFieldValue H2 1
      volScalarFieldValue O2 0
      volScalarFieldValue N2 0
      volScalarFieldValue H2O 0
      volScalarFieldValue T 318
    );
  }
  sphereToCell
  {
    centre (0 0.98 -0.75);
    radius 0.18;

    fieldValues
    (
      volScalarFieldValue p 76269000
      volScalarFieldValue H2 1
      volScalarFieldValue O2 0
      volScalarFieldValue N2 0
      volScalarFieldValue H2O 0
      volScalarFieldValue T 318
    );
  }
  sphereToCell
  {
    centre (0 0.98 0.75);
    radius 0.18;

    fieldValues
    (
      volScalarFieldValue p 76269000
      volScalarFieldValue H2 1
      volScalarFieldValue O2 0
      volScalarFieldValue N2 0
      volScalarFieldValue H2O 0
      volScalarFieldValue T 318
    );
  }
);

```

본 파일은 전산해석 전체 계산 영역에 대해 초기 필드값을 설정하는 파일이다. 지정된 영역에서 원하는 물리적 상태를 정의할 수 있다. 본 코드에서는 setFieldsDict 파일을 통해 고압 수소 용기의 압력과 체적을 설정하였다.

#### ○ defaultFieldValues

전산해석의 계산 영역 전체에 적용될 기본 초기 필드값을 설정하는 항목이다. defaultFieldValues에 지정된 값은 특정 영역에 별도의 값을 설정하지 않을 경우, 전체 계산 영역에 걸쳐 적용된다.

- volScalarFieldValue: 특정 체적(vol)에서 스칼라값을 설정한다.

○ regions

계산 영역 내 특정 영역을 정의하고, 해당 영역에 대해 기본 값과 다른 초기 조건을 설정하는 항목이다.

- cylinderToCell: 원통형(Cylinder) 영역을 정의한다.
  - P1, P2: 원통의 시작점과 끝점을 설정한다.
  - radius: 원통의 반지름을 설정한다.
  - fieldValue: 원통형 영역에 적용할 필드 값을 설정한다.
- sphereToCell: 구형(Sphere) 영역을 정의한다.
  - centre: 구의 반지름을 설정한다.
  - radius: 구의 반지름을 설정한다.

### 3-3-7. surfaces

본 파일은 특정 표면에 대해 데이터를 출력하거나 샘플링하는 설정을 정의하는 파일이다.

표 33. surface

```
surfaces
{
  type          surfaces;
  libs          ("libsampling.so");
  writeControl  adjustableRunTime;
  writeInterval 0.001;

  surfaceFormat vtk;
  fields        (p U T);

  interpolationScheme cell;

  surfaces
  {
    blastblock1_top_face
    {
      type          patch;
      patches       ("blastblock1_top_face.*");
      interpolate   true;
    }

    blastblock1_rear_face
    {
      type          patch;
      patches       ("blastblock1_rear_face.*");
      interpolate   true;
    }
  }
}
```

○ type

표면 샘플링의 타입을 지정하는 항목이다.

○ libs

샘플링에 필요한 라이브러리를 지정하는 항목이다. 본 코드에서는 “libsampling.so” 로 지정하여 공유 라이브러리를 불러온다.

○ writeControl

데이터를 출력할 주기를 설정하는 항목이다. 본 코드에서는 “adjustableRunTime” 으로 설정되어, 설정된 writeInterval에 따라 실행 시간의 변화에 맞춰 데이터를 출력한다.

○ writeInterval

데이터를 출력하는 간격을 설정하는 항목이다. 본 코드에서는 0.001 초마다 데이터를 출력한

다.

○ surfaceFormat

출력할 파일의 형식을 지정하는 항목이다. 본 코드에서는 “vtk” 로 설정되어 있어, 표면 데이터가 VTK 형식으로 출력된다. VTK 파일 형식은 ParaView와 같은 소프트웨어에서 쉽게 읽을 수 있어 시각화에 유용하다.

○ fields

샘플링을 할 물리적 필드를 정의하는 항목이다. 본 코드에서는 “(p U T)” 로 설정되어 있어, 압력, 속도, 온도 필드를 샘플링 하도록 지정되어 있다.

○ interpolationScheme

샘플링 할 값의 보간 방식을 설정하는 항목이다. 본 코드에서는 “cell” 로 설정되어 있어, 셀 중심에서 샘플링된 값을 사용한다.

#### 4. realGasReactingEsiFoam 실행방법

본 장에서는 제2장, 제3장에서 설명한 입력파일을 작성한 후, realGasReactingEsiFoam을 이용하여 고압수소 화재폭발 해석을 진행하는 과정을 설명한다.

##### 4-1. 실행 옵션

###### 4-1-1. OpenFOAM 환경설정 및 realGasReactingEsiFoam 입력 생성

터미널을 열고, 다음 명령어를 순서대로 입력하여 realGasReactingEsiFoam의 실행에 필요한 입력 데이터를 생성한다(그림 9).

- (1) of7e blockMesh > log.blockMesh
- (2) of7e setFields > log.setFieldsDict

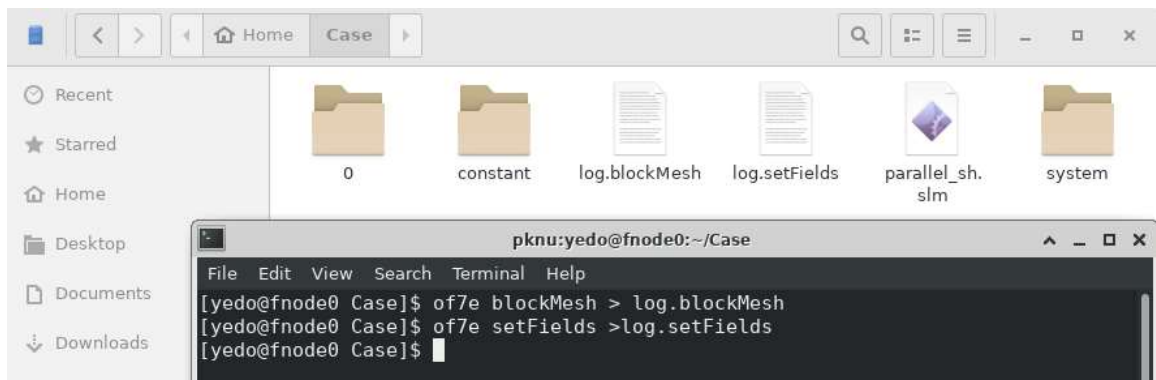


그림 9. IHEsiFoam 실행을 위한 파일 구조와 blockMesh 및 setFields 명령어 실행 화면

###### 4-1-2. 병렬계산 실행방법

○ 격자모델 decompose 실행

병렬 계산을 위해, 위 명령어 (blockMesh, setFields)를 입력한 후, 생성된 격자 모델을 병

렬계산용으로 세분화 하기 위해서 “decomposePar” 명령어를 실행한다. 명령어가 정상적으로 작동되면 “systemWdecomposeParDict” 파일에서 지정한 개수만큼 계산 영역이 분할되고 Case 디렉토리 내에 분할 한 개수만큼 “processor[Num]” 디렉토리가 생성된다.

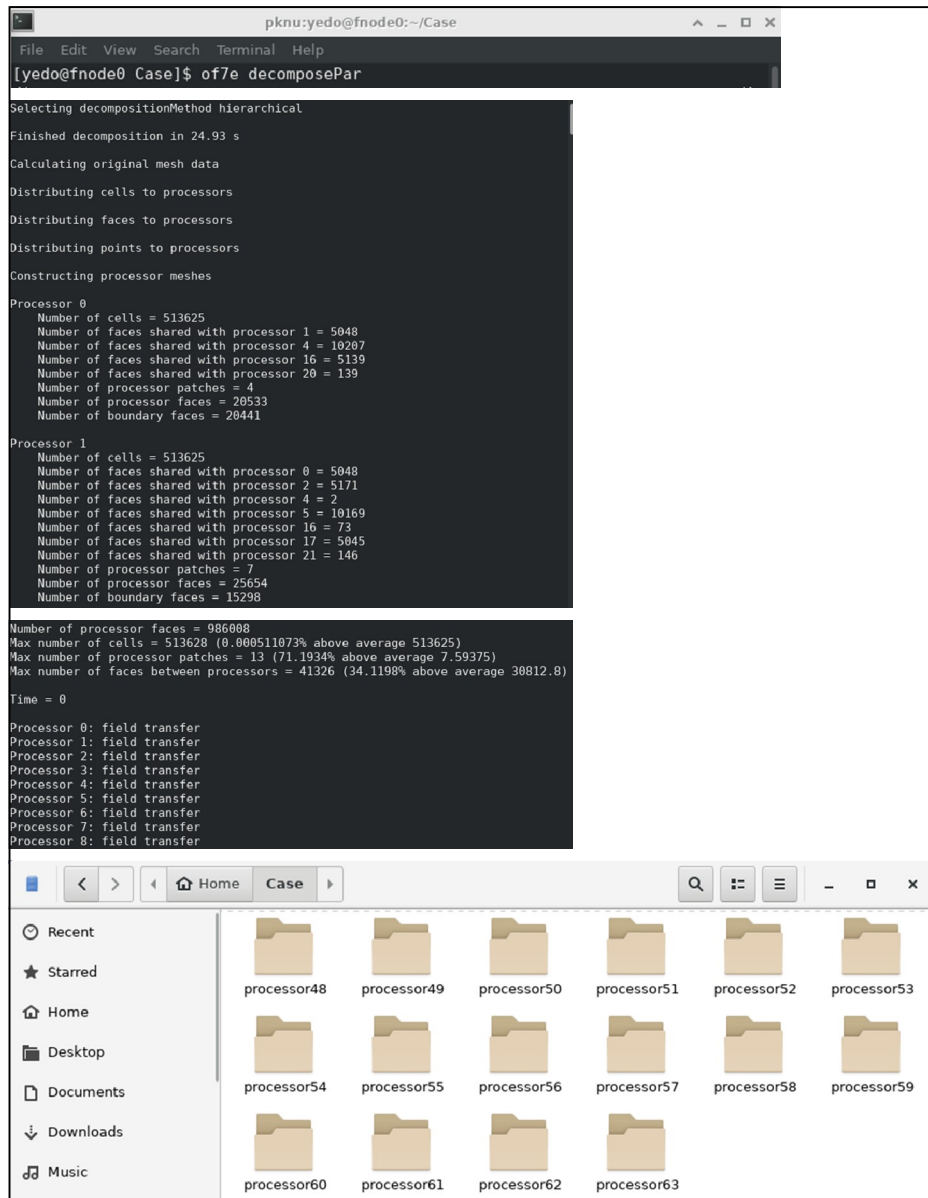


그림 10. 병렬 계산을 위한 도메인 분할 결과 및 각 processor 별 할당 정보

### ○ 병렬계산 실행

decomposePar 명령어를 입력하여 격자 모델에 대한 분할 작업이 완료되면, “CaseWparallel\_sh.slm” 파일을 이용하여 계산을 수행한다. 이 파일은 클러스터에서 병렬 계산을 실행할 때 “qsub parallel\_sh.slm” 명령어를 통해 사용되며, 클러스터 환경에서 계산 작업을 제출하고 실행하기 위한 스크립트이다. parallel\_sh.slm 파일은 작업의 리소스 할당, 계산 환경 설정, 그리고 병렬 계산을 위한 명령어들을 포함하고 있다. 각 항목에 대한 설명은 다음과 같다.

표 34. parallel\_sh.slm

```
#!/bin/bash
#SBATCH -J test                # Job name
###SBATCH -N 1                 # Num of total nodes
#SBATCH -n 64                  # Num of total processes
#SBATCH -p long                # Partition name(It is same with properties of Torque)
#SBATCH -o %x.o.%j            # %x : Job name, %j : JobID
#SBATCH -e %x.e.%j
###SBATCH --time 00:30:00      # Max time for running
###SBATCH --gres=gpu:2

cd $SLURM_SUBMIT_DIR

hostfile=$(mktemp hostfile.XXXX)
srun hostname | sort > $hostfile

echo Running on host `hostname`
echo Time is `date`
echo Directory is `pwd`
echo This jobs runs on the following processors:
echo `cat $hostfile`

NPROCS=`wc -l < $hostfile`
echo This job has allocated $NPROCS cores

source /etc/profile.d/modules.sh
module purge >&/dev/null
module load sgl

of7m -np $NPROCS -machinefile $hostfile realGasEslFoam -parallel >& log

#/opt/mpi/4.1.1-it/bin/mpirun -np $NPROCS -machinefile $hostfile /opt/vasp/bin/vasp_std >& log

[ -n "$hostfile" -a -e $hostfile ] && rm -f $hostfile

exit 0
```

- #!/bin/bash: 스크립트가 실행될 때 Bash를 사용하도록 지정한다.
- #SBATCH -J test: 작업의 이름을 지정한다. 본 코드에서는 “test”로 설정하였다. 클러스터 작업 목록에서 test라는 이름으로 나타나며, 작업을 구분하는데 도움을 준다.
- #SBATCH -n 64: 병렬 계산에서 사용할 프로세스 수를 지정한다. 본 코드에서는 64개로 지정했으며, 이 설정값은 “decomposeParDict”에서의 계산 영역을 분할한 개수와 일치하여야 한다.
- #SBATCH -p long: “long”이라는 파티션에서 작업을 실행하도록 지정한다. 파티션은 클러스터에서 자원 그룹을 나타내며, 특정 자원설정이나 작업의 우선순위에 맞게 파티션을 선택할 수 있다.
- #SBATCH -o %x.o.%j: 작업의 표준 출력을 저장할 파일 이름을 지정한다. %x는 작업 이름을, %J는 Job ID를 나타낸다. “test.o.1234” 같은 형식으로 생성된다.
- #SBATCH -e %x.o.%j: 작업의 표준 에러 출력을 저장할 파일 이름을 지정한다. %x는 작업 이름을, %J는 Job ID를 나타낸다. “test.e.1234” 같은 형식으로 생성된다.
- cd \$SLURM\_SUBMIT\_DIR: 스크립트를 제출한 디렉토리로 이동한다. “\$SLURM\_SUBMIT\_DIR”는 작업이 제출된 디렉토리 경로를 나타내며, 계산이 해당 디렉토리에서 수행되도록 설정한다.

- hostfile=\$(mktemp hostfile.XXXX)
- : 임시 파일을 생성하여 “hostfile” 이라는 파일 이름으로 저장되어, 클러스터에서 사용할 호스트 목록을 저장하는 데 사용된다.
  
- srun hostname | sort > \$hostfile
- : “srun” 명령어를 사용하여 할당된 노드의 호스트 이름을 “hostfile” 에 저장한다. “sort” 명령어로 정렬하여 호스트 파일을 구성한다.
  
- echo Running on host hostname: 현재 실행 중인 호스트 이름을 출력한다.
  
- echo Time is date: 작업 실행시간을 출력한다.
  
- echo Directory is pwd: 현재 작업 디렉토리를 출력한다.
  
- echo cat \$hostfile: 호스트 파일에 저장된 노트 목록을 출력한다.
  
- echo This job runs on the following processors  
: 이 작업이 실행되는 프로세서 목록을 출력하기 위한 안내메세지이다.
  
- NPROCS=wc -l < \$hostfile: 할당된 프로세서 수를 계산하여 NPROCS 변수에 저장한다.
  
- echo This job has allocated \$NPROCS cores: 할당된 총 프로세서 수를 출력한다.
  
- source /etc/profile.d/modules.sh  
: 클러스터에서 모듈 환경을 초기화한다. “/etc/profile.d/modules.sh” 스크립트를 로드하여 사용 가능한 모듈 명령어를 활성화한다.
  
- module purge >&/dev/null: 현재 로드된 모든 모듈을 언로드한다.
  
- module load sgl: “sgl” 모듈을 로드하여 계산에 필요한 소프트웨어 환경을 설정한다.
  
- of7m -np \$NPROCS -machinefile \$hostfile realGasReactingEslFoam -parallel >& log  
: realGasReactingEslFoam solver를 병렬 모드로 실행한다. “-np \$NPROCS” 는 병렬 프로세스 수를 지정하며, “-machinefile \$hostfile” 을 통해 사용할 호스트 목록을 지정한다. 결과는 “log” 파일에 저장된다.
  
- [ -n "\$hostfile" -a -e \$hostfile ] && rm -f \$hostfile  
: “hostfile” 이 비어있지 않고 존재하는 경우, “hostfile” 을 삭제하여 임시 파일을 정리한다. 이는 작업이 종료될 때, 임시 파일을 삭제하여 시스템 리소스를 관리하기 위한 부분이다.

```
pknu:yedo@fnnode0:~/Case
File Edit View Search Terminal Help
[yedo@fnode0 Case]$ qsub parallel_sh.slm
463
[yedo@fnode0 Case]$ qstat
Job id          Name          Username      Time Use S Queue
-----
463             test          yedo         00:00:00 R Long
```

그림 11. paralle\_sh을 이용한 realGasReactingEslFoam 실행 및 작업 상태 확인

## 4-2. 실행 화면 및 출력정보 확인

본 절에서는 realGasReactingEslFoam solver을 정상적으로 실행시킨 후, “log” 파일에서 확인할 수 있는 정보에 대해 설명한다. “log” 파일이란 전산해석 중에 발생하는 모든 과정과 정보를 기록한 텍스트 파일로, 사용되는 모델, 계산 진행 상태, 오류메세지 등을 포함하여 전산해석의 전체 흐름을 한 눈에 확인할 수 있는 파일이다.

### 4-2-1. 초기 설정 및 Model 로딩

```

Create time

Create mesh for time = 0

Reading thermophysical properties

Selecting thermodynamics package
{
    type            hePsiThermo;
    mixture         SRKchungTakaReactingMixture;
    transport       chungTaka;
    thermo          rfJanaf;
    energy          sensibleEnthalpy;
    equationOfState realGas;
    specie          rfSpecie;
}

Selecting chemistryReader foamChemistryReader
    elements not defined in "/home/users/yedo/Case/constant/reactionsGRI"
Reading field U

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;
}

Creating reaction model

Selecting combustion model EDC
Selecting chemistry solver
{
    solver           EulerImplicit;
    method           SRKchungTakaStandard;
}

SRKchungTakaStandardChemistryModel: Number of species = 4 and reactions = 1
    using integrated reaction rate
Creating field kinetic energy K

No MRF models present

No finite volume options present
fluxScheme: Kurganov
Starting time loop

Reading surface description:
    blastblock1_top_face
    blastblock1_rear_face
    blastblock1_left_face
    blastblock1_right_face
    blastblock1_front_face
    block1_top_face
    block2_top_face

```

그림 12. realGasReactingFoam 실행 시 초기화 및 난류/연소 모델 설정 로그 (log 파일)

○ 기본 설정 및 초기화

Solver의 시작과 기본 격자 생성을 위한 초기화 단계이다.

○ 열역학 및 물리모델 정보

전산해석에 사용된 유체의 특성과 열역학 모델을 보여준다. “constant/thermophysicalProperties” 파일에서 지정한 설정이 사용된다.

○ 화학 반응 모델 정보

전산해석에 사용된 화학 반응 모델을 보여준다. “constant/reactionGRI, chemistryProperties, combustionProperties” 파일에서 지정한 설정이 사용된다.

○ 난류 모델 정보

전산해석에 사용된 난류 모델을 보여준다. “constant/turbulenceProperties” 파일에서 지정한 설정이 사용된다.

○ Surface 정보

전산해석에 사용된 Surface 정보를 보여준다. “constant/surface” 파일에서 지정한 설정이 사용된다.

4-2-2. 수치 계산 및 변수 풀이 과정

그림 13은 “Case/log” 파일에서 확인 할 수 있는 계산 진행 상황을 보여준다.

```

deltaT = 1.19999e-10
Mean and max Courant Numbers = 5.05618e-07 4.44219e-06 ①
Time = 1.19999e-10

diagonal: Solving for rho, Initial residual = 0, Final residual = 0, No Iterations 0
diagonal: Solving for rhoUx, Initial residual = 0, Final residual = 0, No Iterations 0 ②
diagonal: Solving for rhoUy, Initial residual = 0, Final residual = 0, No Iterations 0
diagonal: Solving for rhoUz, Initial residual = 0, Final residual = 0, No Iterations 0
smoothSolver: Solving for Ux, Initial residual = 7.17339e-08, Final residual = 4.58983e-17, No Iterations 2
smoothSolver: Solving for Uy, Initial residual = 7.28336e-08, Final residual = 5.41434e-17, No Iterations 2
smoothSolver: Solving for Uz, Initial residual = 7.90114e-08, Final residual = 5.43979e-17, No Iterations 2

min / max mag(U) ; 1.09659e-39 / 0.134966 ③

smoothSolver: Solving for O2, Initial residual = 4.86305e-07, Final residual = 4.79357e-13, No Iterations 1000
smoothSolver: Solving for H2, Initial residual = 4.85099e-07, Final residual = 7.12211e-17, No Iterations 2
smoothSolver: Solving for H2O, Initial residual = 1, Final residual = 2.04714e-16, No Iterations 4
diagonal: Solving for rhoE, Initial residual = 0, Final residual = 0, No Iterations 0
smoothSolver: Solving for h, Initial residual = 1.09009e-08, Final residual = 2.79695e-15, No Iterations 1000
min/max(T) = 300, 318
smoothSolver: Solving for omega, Initial residual = 2.71559e-09, Final residual = 8.97704e-16, No Iterations 2
smoothSolver: Solving for k, Initial residual = 1, Final residual = 2.09069e-07, No Iterations 1000
ExecutionTime = 397.4 s ClockTime = 407 s ④
    
```

그림 13. 수치해석 진행 상황 과 주요 변수의 잔차 및 계산 정보 (log 파일)

○ Time step 및 Courant 수 확인

그림 13의 ①전산해석의 Time step과 Courant 수를 확인하는 항목이다.

- deltaT: 전산해석에서 사용되는 시간 간격(Time Step)을 보여준다. 각 계산 단계에서 시간 적으로 얼마나 이동하는 지를 결정하는 값으로 계산 정확성과 안정성에 있어 매우 중요하다.
- Mean and max Courant Numbers: 평균 및 최대 Courant 수를 보여준다. Courant 수가 낮을 수록 계산의 안정성이 높아진다.
- Time: 현재 전산해석 중인 시간을 보여준다.

○ 변수 풀이 및 수렴 상태

그림 13의 ②는 각 변수(rho, U)에 대한 풀이 결과를 보여주는 항목이다. 각 변수에 대해 초

기 잔차(Initial residual), 최종 잔차(Final residual), 반복 횟수(No Iterations)가 기록되어 있어. 수치 풀이가 얼마나 수렴했는지 평가할 수 있다.

○ 속도 및 온도 분포 분석

그림 14의 ③는 전산해석 내 속도 및 온도의 최솟값과 최댓값을 나타내는 항목이다.

○ 전산해석 실행 시간

그림 13의 ④는 전산해석의 실제 실행 시간(Execution Time)과 전체 소요 시간(Clock Time)을 나타내는 항목이다. Execution Time은 순수 계산 시간만을 나타내며, Clock Time은 실제 경과 된 wall clock time으로, 전산해석의 효율성을 평가할 수 있다.

## 5. 해석결과 분석방법

본 장에서는 사용자가 제4장과 같이 realGasReactingEsiFoam 코드로 예제 파일에 대한 해석을 수행하여 생성된 해석결과를 분석하는 방법에 대해서 설명하고자 한다.

### 5-1. postProcessing 디렉토리

본 절에서는 “case/system/controlDict” 파일의 functions에서 지정한 특정 지점에서 계산된 압력, 온도, 유속 등의 데이터가 생성된 위치와 그 내용에 관해 설명한다.

#### 5-1-1. 디렉토리 위치

controlDict의 functions를 통해 생성된 물리량은 “case/postProcessing” 내의 “probes\_[pressure]” 등의 디렉토리 내부에 저장된다. 디렉토리의 이름은 functions에서 지정한 이름과 동일하게 생성된다. 예를 들어, 압력 데이터는 “ case/postProcessing/probes\_pressure/0” 에 위치한다(그림 14).

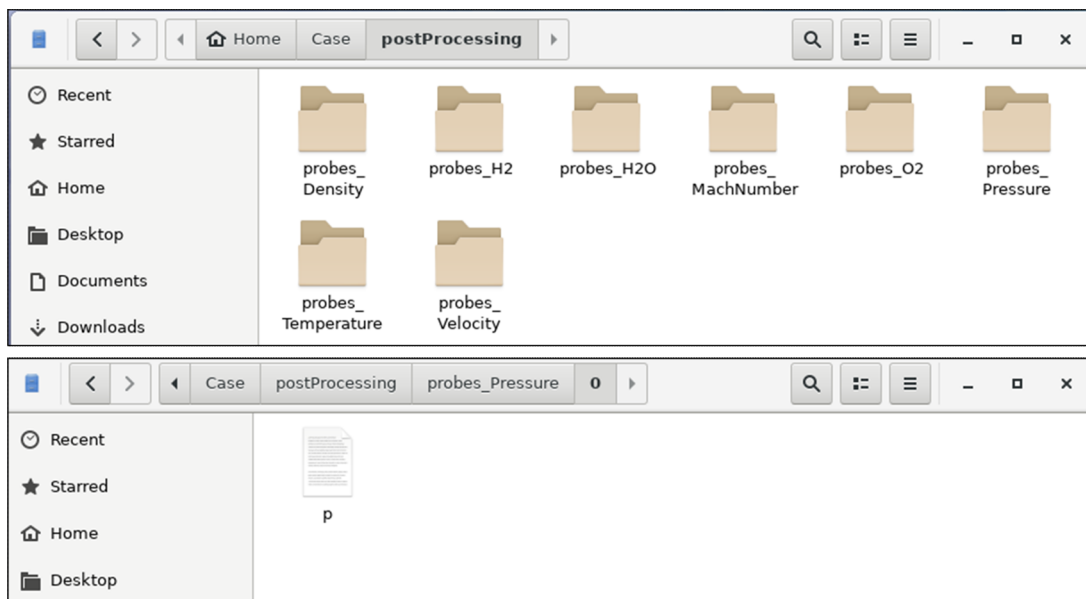


그림 14. postProcessing 디렉토리 구조 및 압력 계산 데이터 위치

## 5-1-2. 데이터 내용

압력 데이터가 저장된 “p” 파일을 열면, 그림 0와 같이 나타난다. “controlDict”의 functions에서 probes 위치를 지정한 순서에 따라서 probes 번호가 0부터 차례대로 부여된다. 처음에는 각 probes 번호에 따른 probes의 좌표가 출력되고, 이후에는 각 probe 위치에서의 시간에 따른 값이 출력된다. 시간 간격은 “controlDict”에서 지정한 “writeInterval”에 따라 출력되며 단위는 [s] 이다. 압력은 절대압력(absolute pressure)이고 단위는 [Pa] 이다.

## 5-2. ParaView 사용 가시화 방법

본 절에서는 realGasReactingEslFoam으로 해석한 결과를 3차원으로 분석할 수 있는 ParaView 소프트웨어의 사용법을 설명한다. ParaView는 제1장에서 설명한 OpenFOAM-v7 설치 과정에서 함께 설치된다. 만약 이 과정에서 정상적으로 설치가 되지 않았다면, ParaView 공식 웹사이트에서 Linux용 binary 파일로 다운로드하여 설치할 수 있다. 본 매뉴얼에서는 ParaView 5.6.0 버전을 기준으로 설명한다.

### 5-2-1. ParaView 설치방법

ParaView 5.6.0 버전은 ParaView 공식 웹사이트에서 Linux용 Binary 파일로 다운로드 할 수 있다. 다운로드 한 파일을 임의의 폴더에서 압축 해제하여 사용한다.

### 5-2-2. ParaView 실행방법

- ParaView 실행 및 realGasReactingEslFoam 해석결과 파일 불러오기  
적절한 위치에서 터미널을 열고 “of7e paraFoam”을 입력한다. Case 디렉토리가 아닌 위치에서 명령어를 입력할 경우, 안내 문구가 생성되는데 “y”를 입력하면 ParaView가 실행된다. 명령어가 정상적으로 실행되면, 그림16 과 같이 ParaView 창이 생성된다. realGasReactingEslFoam로 해석한 데이터를 불러오기 위해서, 그림 16의 ①을 선택한다. 전산 해석 데이터가 있는 디렉토리로 이동한 후(②), ③의 File of type을 All files(\*)로 설정한다.

그림 17에서 “log” 파일을 선택하고 ③의 OpenFOAMReader를 선택한다. 그림 17의 properties에서 “Skip Zero Time”을 해제하고, Case Type을 “Reconstructed Case”에서 “Decomposed Case”로 변경한 후, Mesh Regions을 모두 선택하고 Apply를 선택한다,

```

File Edit View Terminal Tabs Help
[yedo@fnode0 Desktop] $ cd
[yedo@fnode0 ~] $ of7e paraFoam
WARN file does not exist:
./system/controlDict
./system/fvSchemes
./system/fvSolution
Cannot locate OpenFOAM-format case files
Would you like to open ParaView anyway <Y|n>:y

```

그림 15. ParaView 실행 명령 화면

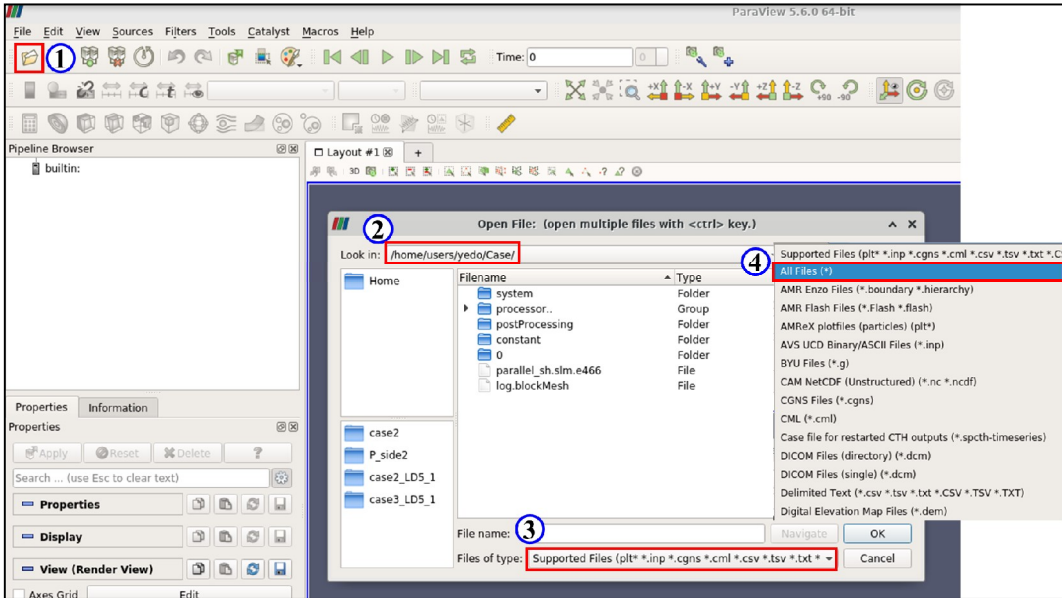


그림 16. ParaView에서 전산해석 결과 파일 불러오기.

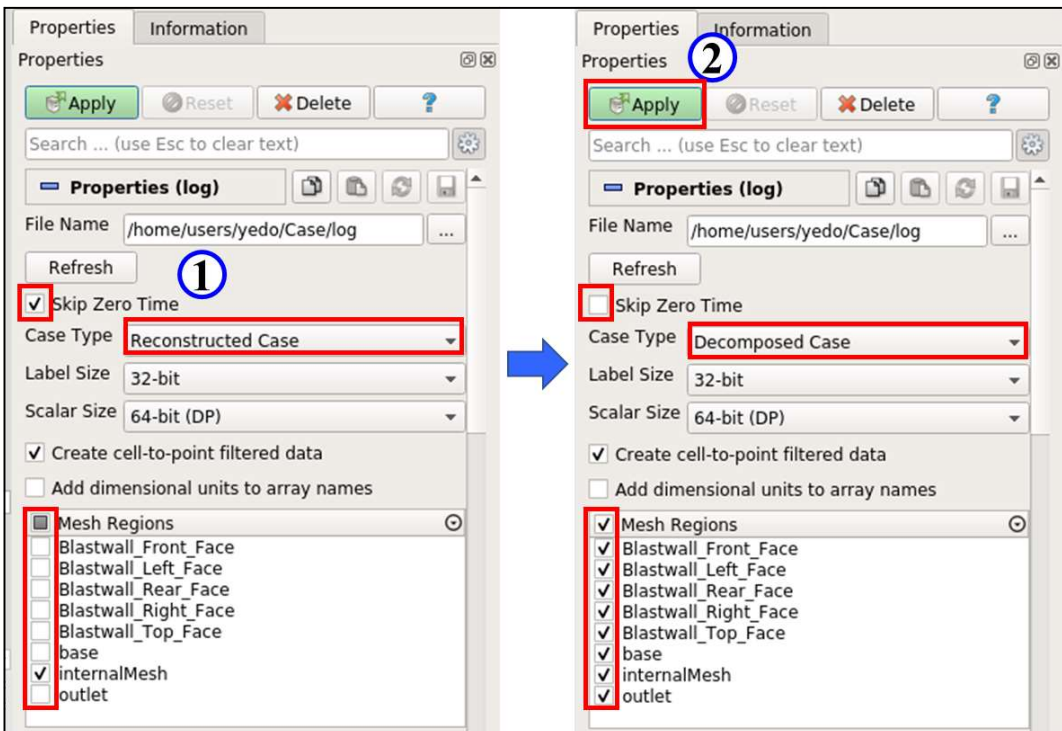


그림 17. Case Type 변경 및 로드 데이터 선택.

○ realGasReactingEsIFoam 해석결과 가시화 및 해석시간 선정방법

realGasReactingEslFoam으로 해석한 결과 중에서 ParaView를 통해 가시화하고자 하는 변수를 그림 18의 ①에서 변경할 수 있다. 특정 시간에서의 해석 결과를 가시화할 때는 ②에서 특정 시간을 선택할 수 있다.

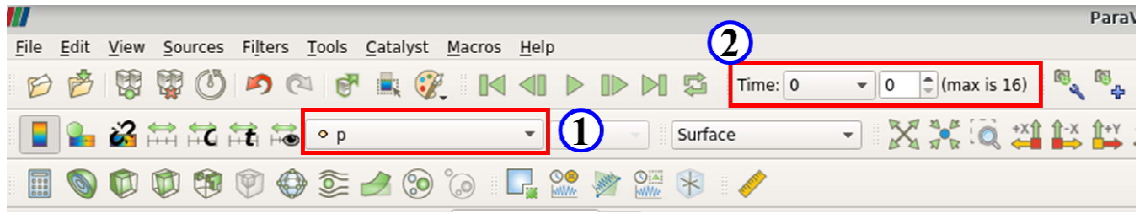


그림 18. 데이터 필터링 및 시간 단계 (Time step) 설정 화면

#### ○ ParaView Layout 화면 조작방법

- 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

본 절에서는 ParaView에서 사용할 수 있는 주요 “Filter”에 대해 설명한다. 특히 본 고압 수소 화재폭발 예제파일의 해석 결과를 가시화 하는데 필요한 Filter를 위주로 설명한다.

#### 5-3-1. Clip 및 Slice

##### ○ Clip

Clip Filter은 전산해석으로 계산한 3D 데이터의 일부분을 잘라내어 내부의 데이터를 드러내는 데 사용된다. 사용자가 지정한 평면을 기준으로 데이터를 잘라내어 원래의 데이터에서 선택한 영역을 제거할 수 있다(그림 19).

- Clip Type: 데이터를 자를 평면의 형태를 지정한다. 기본적으로 평면(Plane)을 사용하고, 구형(Sphere), 상자(Box) 등의 형태로 Clip 할 수 있다.
- Origin: Clip 평면의 기준점(중심점)을 지정한다.
- Normal: Clip 평면의 방향을 지정한다.

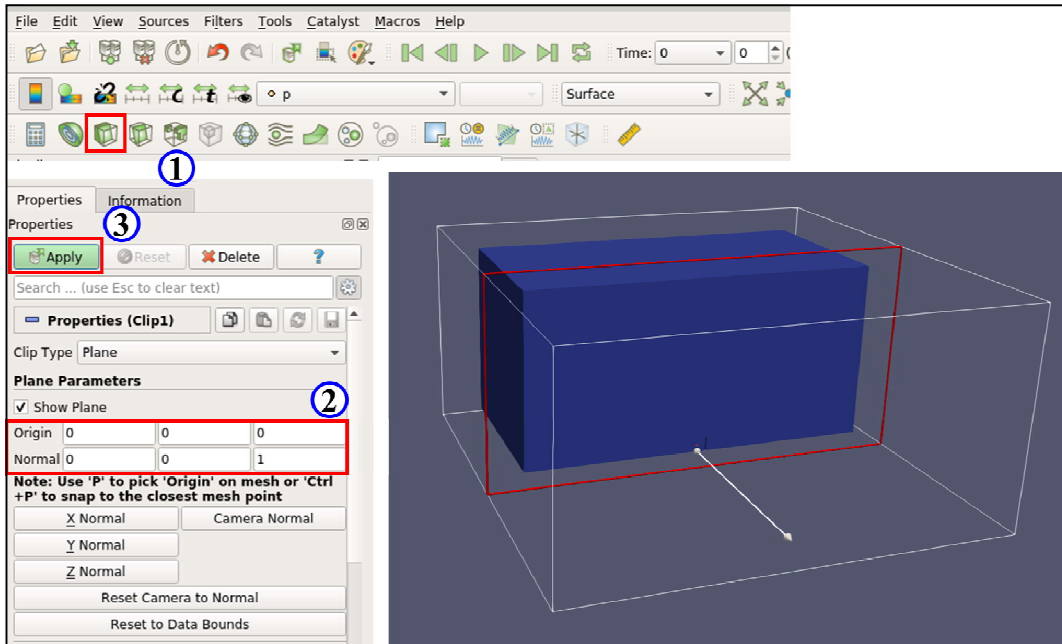


그림 19. ParaView Clip 필터 사용법

○ Slice

Slice Filter는 전산해석으로 계산한 3D 데이터의 일부분을 잘라내어 내부 데이터를 시각화 하는데 사용된다. 이 필터는 사용자가 지정한 방향과 위치에서 Slice를 생성하여 데이터의 단면을 확인할 수 있다.

- Origin: Clip 평면의 기준점(중심점)을 지정한다.
- Normal: Clip 평면의 방향을 지정한다.

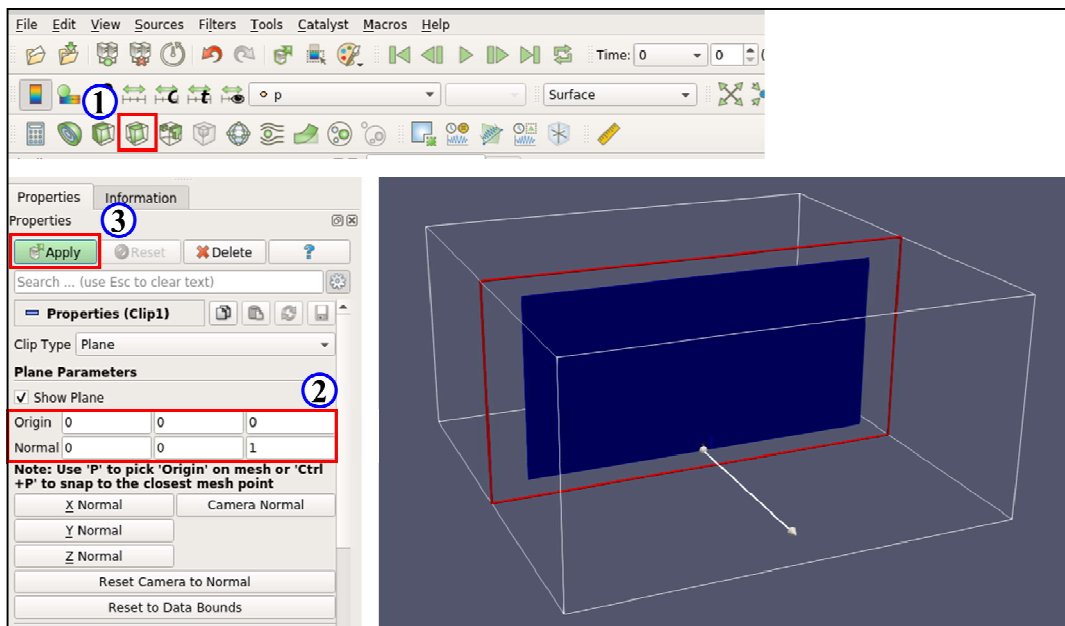


그림 20. ParaView Slice 필터 사용법

5-3-3. Annotate Time Filter

Annotate Time Filter는 전산해석 데이터의 시간정보를 화면에 텍스트로 표시하는 데 사용된다. 사용자가 시각적으로 시간 경과를 이해하는데 도움을 주고, 동영상이나 이미지 시퀀스로 시각화할 때, 각 프레임에 대한 시간 정보를 명확히 표현할 수 있다.

상단의 tool bar에서 filter를 선택하고 “Alphabetical”, “Annoatate Time Filter”을 차례로 선택한다. 여러 옵션을 통해 형식을 설정할 수 있다.

- Format: 소수점 자릿수나 텍스트를 조정하여 원하는 형식으로 설정할 수 있다. 본 예제에서는 “ms”를 입력하여 단위를 [ms]로 표시하도록 설정하였다.
- Shift: 기본 시간 값에서 시간을 더하거나 빼서 시간값을 조정할 수 있다. 예를 들어, 5로 설정한다면, “0.1 s” 일 때 “5.1 s”로 표현된다.
- Scale: 기본 시간 값에서 곱해지는 비율을 설정할 수 있다. 본 예제에서는 [ms] 단위를 사용하기 위해서 “1000”을 입력하였다.
- Font Properties: 글꼴의 스타일, 크기 등을 설정할 수 있다.
- Text Position: 화면에서 텍스트가 표시될 위치를 설정할 수 있다.

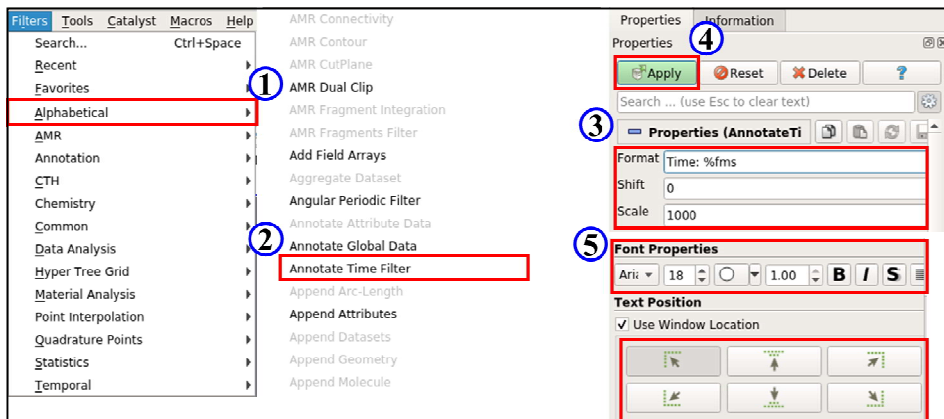


그림 21. ParaView Annoate Time fileter 사용법

#### 5-4. ParaView 시각화 도구 사용법

본 절에서는 ParaView의 시각화 도구를 사용하여 데이터를 직관적으로 분석하고, 원하는 시각적 효과를 적용할 수 있는 방법을 설명한다.

##### 5-4-1. Color Map 편집

본 기능은 전산해석 데이터의 시각화를 돕는 Color Map을 설정할 수 있는 기능이다. 그림 22의 ①를 선택하여 Color Map Editor 창을 생성한다. ②를 선택하면 ParaView에서 제공하는 Color bar의 Preset을 확인할 수 있다. 시각화하고자 하는 물리량과 값의 범위에 따라 적절한 Preset을 선택한다. 본 예제에서는 jet Preset을 선택하였다. ⑤를 선택하면 Color Legend Properties를 수정할 수 있는 창이 생성된다. Color Legend의 생성 위치, 제목, 글꼴 크기 등을 설정할 수 있다.

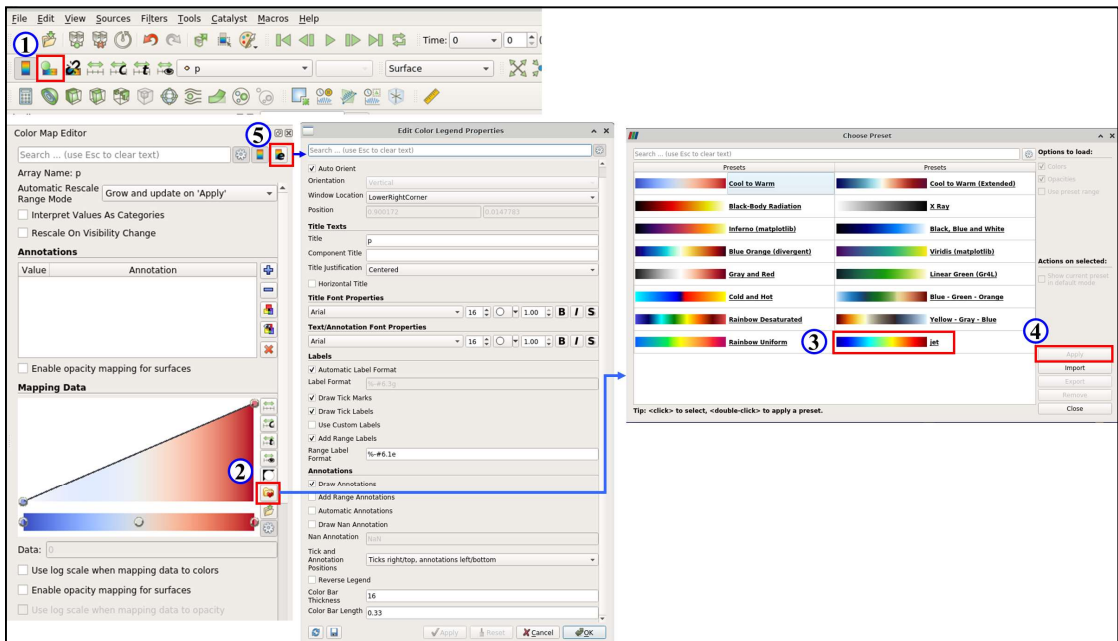


그림 22. ParaView Color Map 사용법

#### 5-4-2. Data Range 편집

본 기능은 전산해석 데이터를 시각적으로 나타낼 때 사용되는 Color Map의 최솟값과 최댓값을 설정하는 기능이다. 그림 23의 ①을 선택하여 Set Range 창을 생성하고 ②에서 최솟값과 최댓값을 입력한다.

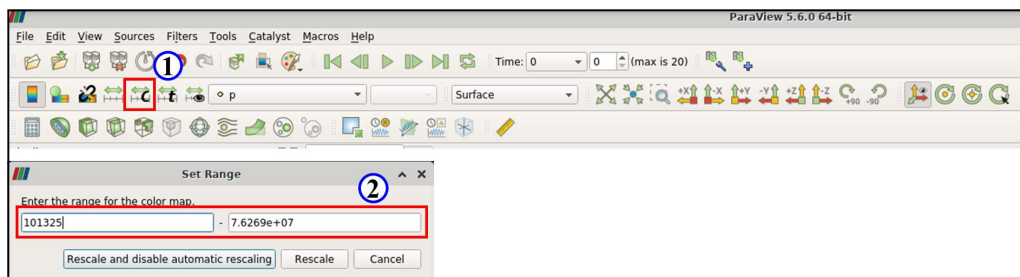


그림 23. ParaView 데이터 범위 설정

## 6. 부록

### 6-1. 격자모델 입력자료

#### ○ Case 격자모델 입력자료 (blockMesh)

입력자료는 본 매뉴얼에서 설명한 예제 파일과 동일하다. (단, 가독성을 위하여 일부 데이터는 단을 나누어 작성됨)

```
convertToMeters 1;
```

```
vertices
(
  (-25.00 0.00 -25.00) // 0
  (-5.00 0.00 -25.00) // 1
  (0.00 0.00 -25.00) // 2
  (1.20 0.00 -25.00) // 3
  (1.40 0.00 -25.00) // 4
  (5.00 0.00 -25.00) // 5
  (25.00 0.00 -25.00) // 6

  (-25.00 2.00 -25.00) // 7
  (-5.00 2.00 -25.00) // 8
  (0.00 2.00 -25.00) // 9
  (1.20 2.00 -25.00) // 10
  (1.40 2.00 -25.00) // 11
  (5.00 2.00 -25.00) // 12
  (25.00 2.00 -25.00) // 13

  (-25.00 5.00 -25.00) // 14
  (-5.00 5.00 -25.00) // 15
  (0.00 5.00 -25.00) // 16
  (1.20 5.00 -25.00) // 17
  (1.40 5.00 -25.00) // 18
  (5.00 5.00 -25.00) // 19
  (25.00 5.00 -25.00) // 20

  (-25.00 25.00 -25.00) // 21
  (-5.00 25.00 -25.00) // 22
  (0.00 25.00 -25.00) // 23
  (1.20 25.00 -25.00) // 24
  (1.40 25.00 -25.00) // 25
  (5.00 25.00 -25.00) // 26
  (25.00 25.00 -25.00) // 27

  (-25.00 0.00 -5.00) // 28
  (-5.00 0.00 -5.00) // 29
  (0.00 0.00 -5.00) // 30
  (1.20 0.00 -5.00) // 31
  (1.40 0.00 -5.00) // 32
  (5.00 0.00 -5.00) // 33
  (25.00 0.00 -5.00) // 34

  (-25.00 2.00 -5.00) // 35
  (-5.00 2.00 -5.00) // 36
  (0.00 2.00 -5.00) // 37
  (1.20 2.00 -5.00) // 38
  (1.40 2.00 -5.00) // 39

  (5.00 2.00 -5.00) // 40
  (25.00 2.00 -5.00) // 41
  (-25.00 5.00 -5.00) // 42
  (-5.00 5.00 -5.00) // 43
  (0.00 5.00 -5.00) // 44
  (1.20 5.00 -5.00) // 45
  (1.40 5.00 -5.00) // 46
  (5.00 5.00 -5.00) // 47
  (25.00 5.00 -5.00) // 48

  (-25.00 25.00 -5.00) // 49
  (-5.00 25.00 -5.00) // 50
  (0.00 25.00 -5.00) // 51
  (1.20 25.00 -5.00) // 52
  (1.40 25.00 -5.00) // 53
  (5.00 25.00 -5.00) // 54
  (25.00 25.00 -5.00) // 55

  (-25.00 0.00 -2.50) // 56
  (-5.00 0.00 -2.50) // 57
  (0.00 0.00 -2.50) // 58
  (1.20 0.00 -2.50) // 59
  (1.40 0.00 -2.50) // 60
  (5.00 0.00 -2.50) // 61
  (25.00 0.00 -2.50) // 62

  (-25.00 2.00 -2.50) // 63
  (-5.00 2.00 -2.50) // 64
  (0.00 2.00 -2.50) // 65
  (1.20 2.00 -2.50) // 66
  (1.40 2.00 -2.50) // 67
  (5.00 2.00 -2.50) // 68
  (25.00 2.00 -2.50) // 69

  (-25.00 5.00 -2.50) // 70
  (-5.00 5.00 -2.50) // 71
  (0.00 5.00 -2.50) // 72
  (1.20 5.00 -2.50) // 73
  (1.40 5.00 -2.50) // 74
  (5.00 5.00 -2.50) // 75
  (25.00 5.00 -2.50) // 76

  (-25.00 25.00 -2.50) // 77
  (-5.00 25.00 -2.50) // 78
  (0.00 25.00 -2.50) // 79
  (1.20 25.00 -2.50) // 80

  (1.40 25.00 -2.50) // 81
  (5.00 25.00 -2.50) // 82
  (25.00 25.00 -2.50) // 83

  (-25.00 0.00 0.00) // 84
  (-5.00 0.00 0.00) // 85
  (0.00 0.00 0.00) // 86
  (1.20 0.00 0.00) // 87
  (1.40 0.00 0.00) // 88
  (5.00 0.00 0.00) // 89
  (25.00 0.00 0.00) // 90

  (-25.00 2.00 0.00) // 91
  (-5.00 2.00 0.00) // 92
  (0.00 2.00 0.00) // 93
  (1.20 2.00 0.00) // 94
  (1.40 2.00 0.00) // 95
  (5.00 2.00 0.00) // 96
  (25.00 2.00 0.00) // 97

  (-25.00 5.00 0.00) // 98
  (-5.00 5.00 0.00) // 99
  (0.00 5.00 0.00) // 100
  (1.20 5.00 0.00) // 101
  (1.40 5.00 0.00) // 102
  (5.00 5.00 0.00) // 103
  (25.00 5.00 0.00) // 104

  (-25.00 25.00 0.00) // 105
  (-5.00 25.00 0.00) // 106
  (0.00 25.00 0.00) // 107
  (1.20 25.00 0.00) // 108
  (1.40 25.00 0.00) // 109
  (5.00 25.00 0.00) // 110
  (25.00 25.00 0.00) // 111

  (-25.00 0.00 2.50) // 112
  (-5.00 0.00 2.50) // 113
  (0.00 0.00 2.50) // 114
  (1.20 0.00 2.50) // 115
  (1.40 0.00 2.50) // 116
  (5.00 0.00 2.50) // 117
  (25.00 0.00 2.50) // 118

  (-25.00 2.00 2.50) // 119
  (-5.00 2.00 2.50) // 120
```

```

( 0.00 2.00 2.50) // 121
( 1.20 2.00 2.50) // 122
( 1.40 2.00 2.50) // 123
( 5.00 2.00 2.50) // 124
( 25.00 2.00 2.50) // 125

(-25.00 5.00 2.50) // 126
(-5.00 5.00 2.50) // 127
( 0.00 5.00 2.50) // 128
( 1.20 5.00 2.50) // 129
( 1.40 5.00 2.50) // 130
( 5.00 5.00 2.50) // 131
( 25.00 5.00 2.50) // 132

(-25.00 25.00 2.50) // 133
(-5.00 25.00 2.50) // 134
( 0.00 25.00 2.50) // 135
( 1.20 25.00 2.50) // 136
( 1.40 25.00 2.50) // 137
( 5.00 25.00 2.50) // 138
( 25.00 25.00 2.50) // 139

(-25.00 0.00 5.00) // 140
(-5.00 0.00 5.00) // 141
( 0.00 0.00 5.00) // 142
( 1.20 0.00 5.00) // 143
( 1.40 0.00 5.00) // 144
( 5.00 0.00 5.00) // 145
( 25.00 0.00 5.00) // 146

(-25.00 2.00 5.00) // 147
(-5.00 2.00 5.00) // 148
( 0.00 2.00 5.00) // 149
( 1.20 2.00 5.00) // 150
( 1.40 2.00 5.00) // 151
( 5.00 2.00 5.00) // 152
( 25.00 2.00 5.00) // 153

(-25.00 5.00 5.00) // 154
(-5.00 5.00 5.00) // 155
( 0.00 5.00 5.00) // 156
( 1.20 5.00 5.00) // 157
( 1.40 5.00 5.00) // 158
( 5.00 5.00 5.00) // 159
( 25.00 5.00 5.00) // 160

(-25.00 25.00 5.00) // 161
(-5.00 25.00 5.00) // 162
( 0.00 25.00 5.00) // 163
( 1.20 25.00 5.00) // 164
( 1.40 25.00 5.00) // 165
( 5.00 25.00 5.00) // 166
( 25.00 25.00 5.00) // 167

(-25.00 0.00 25.00) // 168
(-5.00 0.00 25.00) // 169
( 0.00 0.00 25.00) // 170
( 1.20 0.00 25.00) // 171

( 1.40 0.00 25.00) // 172
( 5.00 0.00 25.00) // 173
( 25.00 0.00 25.00) // 174

(-25.00 2.00 25.00) // 175
(-5.00 2.00 25.00) // 176
( 0.00 2.00 25.00) // 177
( 1.20 2.00 25.00) // 178
( 1.40 2.00 25.00) // 179
( 5.00 2.00 25.00) // 180
( 25.00 2.00 25.00) // 181

(-25.00 5.00 25.00) // 182
(-5.00 5.00 25.00) // 183
( 0.00 5.00 25.00) // 184
( 1.20 5.00 25.00) // 185
( 1.40 5.00 25.00) // 186
( 5.00 5.00 25.00) // 187
( 25.00 5.00 25.00) // 188

(-25.00 25.00 25.00) // 189
(-5.00 25.00 25.00) // 190
( 0.00 25.00 25.00) // 191
( 1.20 25.00 25.00) // 192
( 1.40 25.00 25.00) // 193
( 5.00 25.00 25.00) // 194
( 25.00 25.00 25.00) // 195

```

);

blocks

```

( hex ( 0 1 8 7 28 29 36 35) ( 102 40 102) simpleGrading ( 0.1 1.0 0.1)
  hex ( 1 2 9 8 29 30 37 36) ( 100 40 102) simpleGrading ( 1.0 1.0 0.1)
  hex ( 2 3 10 9 30 31 38 37) ( 24 40 102) simpleGrading ( 1.0 1.0 0.1)
  hex ( 3 4 11 10 31 32 39 38) ( 4 40 102) simpleGrading ( 1.0 1.0 0.1)
  hex ( 4 5 12 11 32 33 40 39) ( 72 40 102) simpleGrading ( 1.0 1.0 0.1)
  hex ( 5 6 13 12 33 34 41 40) ( 101 40 102) simpleGrading ( 10.0 1.0 0.1)

  hex ( 28 29 36 35 56 57 64 63) ( 102 40 50) simpleGrading ( 0.1 1.0 1.0)
  hex ( 29 30 37 36 57 58 65 64) ( 100 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 30 31 38 37 58 59 66 65) ( 24 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 31 32 39 38 59 60 67 66) ( 4 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 32 33 40 39 60 61 68 67) ( 72 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 33 34 41 40 61 62 69 68) ( 101 40 50) simpleGrading ( 10.0 1.0 1.0)

  hex ( 56 57 64 63 84 85 92 91) ( 102 40 50) simpleGrading ( 0.1 1.0 1.0)
  hex ( 57 58 65 64 85 86 93 92) ( 100 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 58 59 66 65 86 87 94 93) ( 24 40 50) simpleGrading ( 1.0 1.0 1.0)
  // hex ( 59 60 67 66 87 88 95 94) ( 4 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 60 61 68 67 88 89 96 95) ( 72 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 61 62 69 68 89 90 97 96) ( 101 40 50) simpleGrading ( 10.0 1.0 1.0)

  hex ( 84 85 92 91 112 113 120 119) ( 102 40 50) simpleGrading ( 0.1 1.0 1.0)
  hex ( 85 86 93 92 113 114 121 120) ( 100 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 86 87 94 93 114 115 122 121) ( 24 40 50) simpleGrading ( 1.0 1.0 1.0)
  // hex ( 87 88 95 94 115 116 123 122) ( 4 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 88 89 96 95 116 117 124 123) ( 72 40 50) simpleGrading ( 1.0 1.0 1.0)
  hex ( 89 90 97 96 117 118 125 124) ( 101 40 50) simpleGrading ( 10.0 1.0 1.0)

```

```

hex ( 112 113 120 119 140 141 148 147) ( 102 40 50) simpleGrading ( 0.1 1.0 1.0)
hex ( 113 114 121 120 141 142 149 148) ( 100 40 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 114 115 122 121 142 143 150 149) ( 24 40 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 115 116 123 122 143 144 151 150) ( 4 40 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 116 117 124 123 144 145 152 151) ( 72 40 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 117 118 125 124 145 146 153 152) ( 101 40 50) simpleGrading ( 10.0 1.0 1.0)

hex ( 140 141 148 147 168 169 176 175) ( 102 40 102) simpleGrading ( 0.1 1.0 10.0)
hex ( 141 142 149 148 169 170 177 176) ( 100 40 102) simpleGrading ( 1.0 1.0 10.0)
hex ( 142 143 150 149 170 171 178 177) ( 24 40 102) simpleGrading ( 1.0 1.0 10.0)
hex ( 143 144 151 150 171 172 179 178) ( 4 40 102) simpleGrading ( 1.0 1.0 10.0)
hex ( 144 145 152 151 172 173 180 179) ( 72 40 102) simpleGrading ( 1.0 1.0 10.0)
hex ( 145 146 153 152 173 174 181 180) ( 101 40 102) simpleGrading ( 10.0 1.0 10.0)

hex ( 7 8 15 14 35 36 43 42) ( 102 60 102) simpleGrading ( 0.1 1.0 0.1)
hex ( 8 9 16 15 36 37 44 43) ( 100 60 102) simpleGrading ( 1.0 1.0 0.1)
hex ( 9 10 17 16 37 38 45 44) ( 24 60 102) simpleGrading ( 1.0 1.0 0.1)
hex ( 10 11 18 17 38 39 46 45) ( 4 60 102) simpleGrading ( 1.0 1.0 0.1)
hex ( 11 12 19 18 39 40 47 46) ( 72 60 102) simpleGrading ( 1.0 1.0 0.1)
hex ( 12 13 20 19 40 41 48 47) ( 101 60 102) simpleGrading ( 10.0 1.0 0.1)

hex ( 35 36 43 42 63 64 71 70) ( 102 60 50) simpleGrading ( 0.1 1.0 1.0)
hex ( 36 37 44 43 64 65 72 71) ( 100 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 37 38 45 44 65 66 73 72) ( 24 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 38 39 46 45 66 67 74 73) ( 4 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 39 40 47 46 67 68 75 74) ( 72 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 40 41 48 47 68 69 76 75) ( 101 60 50) simpleGrading ( 10.0 1.0 1.0)

hex ( 63 64 71 70 91 92 99 98) ( 102 60 50) simpleGrading ( 0.1 1.0 1.0)
hex ( 64 65 72 71 92 93 100 99) ( 100 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 65 66 73 72 93 94 101 100) ( 24 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 66 67 74 73 94 95 102 101) ( 4 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 67 68 75 74 95 96 103 102) ( 72 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 68 69 76 75 96 97 104 103) ( 101 60 50) simpleGrading ( 10.0 1.0 1.0)

hex ( 91 92 99 98 119 120 127 126) ( 102 60 50) simpleGrading ( 0.1 1.0 1.0)
hex ( 92 93 100 99 120 121 128 127) ( 100 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 93 94 101 100 121 122 129 128) ( 24 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 94 95 102 101 122 123 130 129) ( 4 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 95 96 103 102 123 124 131 130) ( 72 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 96 97 104 103 124 125 132 131) ( 101 60 50) simpleGrading ( 10.0 1.0 1.0)

hex ( 119 120 127 126 147 148 155 154) ( 102 60 50) simpleGrading ( 0.1 1.0 1.0)
hex ( 120 121 128 127 148 149 156 155) ( 100 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 121 122 129 128 149 150 157 156) ( 24 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 122 123 130 129 150 151 158 157) ( 4 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 123 124 131 130 151 152 159 158) ( 72 60 50) simpleGrading ( 1.0 1.0 1.0)
hex ( 124 125 132 131 152 153 160 159) ( 101 60 50) simpleGrading ( 10.0 1.0 1.0)

hex ( 147 148 155 154 175 176 183 182) ( 102 60 102) simpleGrading ( 0.1 1.0 10.0)
hex ( 148 149 156 155 176 177 184 183) ( 100 60 102) simpleGrading ( 1.0 1.0 10.0)
hex ( 149 150 157 156 177 178 185 184) ( 24 60 102) simpleGrading ( 1.0 1.0 10.0)
hex ( 150 151 158 157 178 179 186 185) ( 4 60 102) simpleGrading ( 1.0 1.0 10.0)
hex ( 151 152 159 158 179 180 187 186) ( 72 60 102) simpleGrading ( 1.0 1.0 10.0)
hex ( 152 153 160 159 180 181 188 187) ( 101 60 102) simpleGrading ( 10.0 1.0 10.0)

hex ( 14 15 22 21 42 43 50 49) ( 102 102 102) simpleGrading ( 0.1 10.0 0.1)
hex ( 15 16 23 22 43 44 51 50) ( 100 102 102) simpleGrading ( 1.0 10.0 0.1)

```

```

hex ( 16 17 24 23 44 45 52 51) ( 24 102 102) simpleGrading ( 1.0 10.0 0.1)
hex ( 17 18 25 24 45 46 53 52) ( 4 102 102) simpleGrading ( 1.0 10.0 0.1)
hex ( 18 19 26 25 46 47 54 53) ( 72 102 102) simpleGrading ( 1.0 10.0 0.1)
hex ( 19 20 27 26 47 48 55 54) ( 101 102 102) simpleGrading ( 10.0 10.0 0.1)

hex ( 42 43 50 49 70 71 78 77) ( 102 102 50) simpleGrading ( 0.1 10.0 1.0)
hex ( 43 44 51 50 71 72 79 78) ( 100 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 44 45 52 51 72 73 80 79) ( 24 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 45 46 53 52 73 74 81 80) ( 4 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 46 47 54 53 74 75 82 81) ( 72 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 47 48 55 54 75 76 83 82) ( 101 102 50) simpleGrading ( 10.0 10.0 1.0)

hex ( 70 71 78 77 98 99 106 105) ( 102 102 50) simpleGrading ( 0.1 10.0 1.0)
hex ( 71 72 79 78 99 100 107 106) ( 100 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 72 73 80 79 100 101 108 107) ( 24 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 73 74 81 80 101 102 109 108) ( 4 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 74 75 82 81 102 103 110 109) ( 72 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 75 76 83 82 103 104 111 110) ( 101 102 50) simpleGrading ( 10.0 10.0 1.0)

hex ( 98 99 106 105 126 127 134 133) ( 102 102 50) simpleGrading ( 0.1 10.0 1.0)
hex ( 99 100 107 106 127 128 135 134) ( 100 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 100 101 108 107 128 129 136 135) ( 24 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 101 102 109 108 129 130 137 136) ( 4 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 102 103 110 109 130 131 138 137) ( 72 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 103 104 111 110 131 132 139 138) ( 101 102 50) simpleGrading ( 10.0 10.0 1.0)

hex ( 126 127 134 133 154 155 162 161) ( 102 102 50) simpleGrading ( 0.1 10.0 1.0)
hex ( 127 128 135 134 155 156 163 162) ( 100 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 128 129 136 135 156 157 164 163) ( 24 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 129 130 137 136 157 158 165 164) ( 4 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 130 131 138 137 158 159 166 165) ( 72 102 50) simpleGrading ( 1.0 10.0 1.0)
hex ( 131 132 139 138 159 160 167 166) ( 101 102 50) simpleGrading ( 10.0 10.0 1.0)

hex ( 154 155 162 161 182 183 190 189) ( 102 102 102) simpleGrading ( 0.1 10.0 10.0)
hex ( 155 156 163 162 183 184 191 190) ( 100 102 102) simpleGrading ( 1.0 10.0 10.0)
hex ( 156 157 164 163 184 185 192 191) ( 24 102 102) simpleGrading ( 1.0 10.0 10.0)
hex ( 157 158 165 164 185 186 193 192) ( 4 102 102) simpleGrading ( 1.0 10.0 10.0)
hex ( 158 159 166 165 186 187 194 193) ( 72 102 102) simpleGrading ( 1.0 10.0 10.0)
hex ( 159 160 167 166 187 188 195 194) ( 101 102 102) simpleGrading ( 10.0 10.0 10.0)

```

```
);
```

```
edges
```

```
();
```

```

boundary
( base
{
    type wall;
    faces
    (
    ( 0 1 29 28)
    ( 1 2 30 29)
    ( 2 3 31 30)
    ( 3 4 32 31)
    ( 4 5 33 32)
    ( 5 6 34 33)

    ( 28 29 57 56)
    ( 29 30 58 57)
    ( 30 31 59 58)
    ( 31 32 60 59)
    ( 32 33 61 60)
    ( 33 34 62 61)

    ( 56 57 85 84)
    ( 57 58 86 85)
    ( 58 59 87 86)
    // ( 59 60 88 87)
    ( 60 61 89 88)
    ( 61 62 90 89)

    ( 84 85 113 112)
    ( 85 86 114 113)
    ( 86 87 115 114)
    // ( 87 88 116 115)
    ( 88 89 117 116)

    ( 89 90 118 117)
    ( 112 113 141 140)
    ( 113 114 142 141)
    ( 114 115 143 142)
    ( 115 116 144 143)
    ( 116 117 145 144)
    ( 117 118 146 145)

    ( 140 141 169 168)
    ( 141 142 170 169)
    ( 142 143 171 170)
    ( 143 144 172 171)
    ( 144 145 173 172)
    ( 145 146 174 173)
    );
}

```

```

}
outlet
{
    type patch;
    faces
    (
( 21 49 50 22)
( 22 50 51 23)
( 23 51 52 24)
( 24 52 53 25)
( 25 53 54 26)
( 26 54 55 27)

( 49 77 78 50)
( 50 78 79 51)
( 51 79 80 52)
( 52 80 81 53)
( 53 81 82 54)
( 54 82 83 55)

( 77 105 106 78)
( 78 106 107 79)
( 79 107 108 80)
( 80 108 109 81)
( 81 109 110 82)
( 82 110 111 83)

( 105 133 134 106)
( 106 134 135 107)
( 107 135 136 108)
( 108 136 137 109)
( 109 137 138 110)
( 110 138 139 111)

( 133 161 162 134)
( 134 162 163 135)
( 135 163 164 136)
( 136 164 165 137)
( 137 165 166 138)
( 138 166 167 139)

( 161 189 190 162)
( 162 190 191 163)
( 163 191 192 164)
( 164 192 193 165)
( 165 193 194 166)
( 166 194 195 167)

( 0 7 8 1)
( 1 8 9 2)
( 2 9 10 3)
( 3 10 11 4)
( 4 11 12 5)
( 5 12 13 6)

( 7 14 15 8)
( 8 15 16 9)
( 9 16 17 10)
( 10 17 18 11)

( 11 18 19 12)
( 12 19 20 13)

( 14 21 22 15)
( 15 22 23 16)
( 16 23 24 17)
( 17 24 25 18)
( 18 25 26 19)
( 19 26 27 20)

( 6 13 41 34)
( 34 41 69 62)
( 62 69 97 90)
( 90 97 125 118)
( 118 125 153 146)
( 146 153 181 174)

( 13 20 48 41)
( 41 48 76 69)
( 69 76 104 97)
( 97 104 132 125)
( 125 132 160 153)
( 153 160 188 181)

( 20 27 55 48)
( 48 55 83 76)
( 76 83 111 104)
( 104 111 139 132)
( 132 139 167 160)
( 160 167 195 188)

( 169 176 175 168)
( 170 177 176 169)
( 171 178 177 170)
( 172 179 178 171)
( 173 180 179 172)
( 174 181 180 173)

( 176 183 182 175)
( 177 184 183 176)
( 178 185 184 177)
( 179 186 185 178)
( 180 187 186 179)
( 181 188 187 180)

( 183 190 189 182)
( 184 191 190 183)
( 185 192 191 184)
( 186 193 192 185)
( 187 194 193 186)
( 188 195 194 187)

( 28 35 7 0)
( 56 63 35 28)
( 84 91 63 56)
( 112 119 91 84)
( 140 147 119 112)
( 168 175 147 140)

( 35 42 14 7)
( 63 70 42 35)
( 91 98 70 63)
( 119 126 98 91)
( 147 154 126 119)
( 175 182 154 147)

( 42 49 21 14)
( 70 77 49 42)
( 98 105 77 70)
( 126 133 105 98)
( 154 161 133 126)
( 182 189 161 154)
    );
}
Blastwall_Top_Face
{
    type patch;
    faces
    (
(66 67 95 94)
(94 95 123 122)
    );
}
Blastwall_Front_Face
{
    type patch;
    faces
    (
(87 94 66 59)
(115 122 94 87)
    );
}
Blastwall_Rear_Face
{
    type patch;
    faces
    (
(60 67 95 88)
(88 95 123 116)
    );
}
Blastwall_Right_Face
{
    type patch;
    faces
    (
(59 66 67 60)
    );
}
Blastwall_Left_Face
{
    type patch;
    faces
    (
(116 123 122 115)
    );
}
);

```

## 6-2. 고압 수소 초기조건 입력자료

### ○ 고압 수소 초기조건 입력자료 (setFieldsDict)

```
mergePatchPairs
(
);defaultFieldValues
(
    volScalarFieldValue T 300
    volScalarFieldValue p 101325
);
regions
(cylinderToCell
{
    p1 (0 0.98 -0.75);
    p2 (0 0.98 0.75);
    radius 0.18;
    fieldValues
    (
        volScalarFieldValue p 76269000
        volScalarFieldValue H2 1
        volScalarFieldValue T 318
    );
}
sphereToCell
{
    centre (0 0.98 -0.75);
    radius 0.18;
    fieldValues
    (
        volScalarFieldValue p 76269000
        volScalarFieldValue H2 1
        volScalarFieldValue T 318
    );
}
sphereToCell
{
    centre (0 0.98 0.75);
    radius 0.18;
    fieldValues
    (
        volScalarFieldValue p 76269000
        volScalarFieldValue H2 1
        volScalarFieldValue T 318
    );
}
```

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.