

# Foam-extend를 이용한 가스터빈 유동해석

한국항공우주연구원  
항공추진연구부  
강승환





# Contents

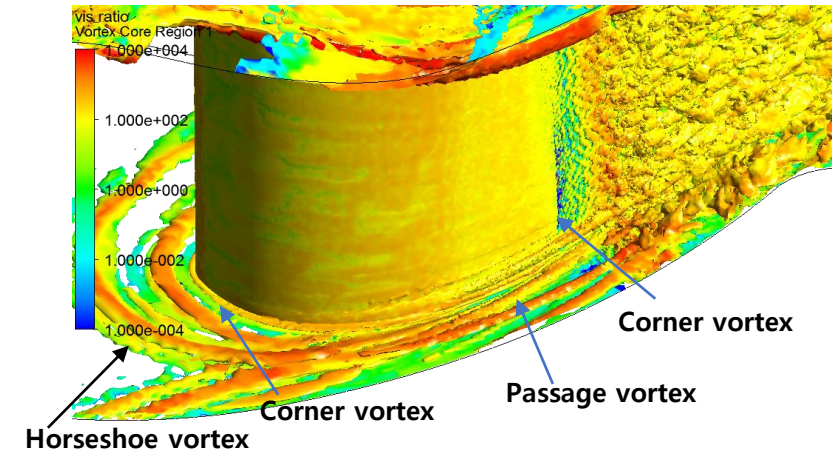
- I Introduction
- II mixingPlane Issue
- III Wall Temperature Issue
- IV The Other Issues
- V Summary and Conclusion

# **1. Introduction**

# 1. Introduction

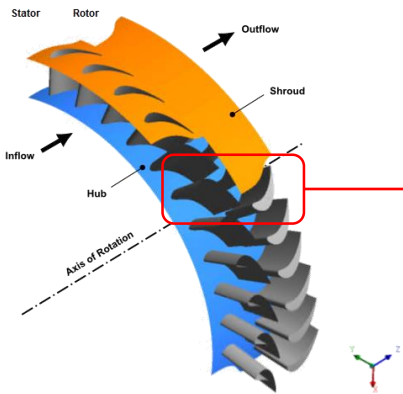
## Background

- (Gas turbine 포함) CFD 영역에서 고정밀도 해석 요구됨.  
→ 따라서 더 많은 격자 및 계산 능력 필요.  
→ 대규모 병렬 해석 선호됨.
- 라이선스 비용의 부담 없는 오픈 소스 CFD 코드는 대규모 병렬 해석을 위한 하나의 대안. → **OpenFOAM**

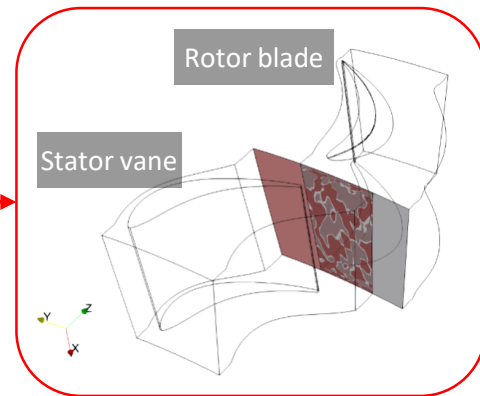


A Large Eddy Simulation of Axial Turbine using ANSYS CFX

## Turbomachinery analysis and interface



A turbine stage: stator(nozzle) vane + rotor(turbine) blade (CFX tutorial)



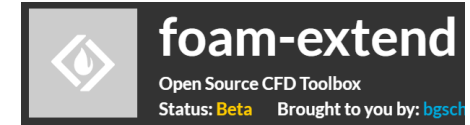
One passage geometry and Unmatched interface

- (가스터빈) 터보유체기계(압축기 & 터빈)를 풀기 위해
  - 압축성 솔버,
  - MRF(Multiple rotation reference frame) 솔버,
  - Interface 유틸리티 필요.
- 기존 OpenFOAM 버전의 *cyclicAMI*으로선 한계  
→ 다양한 Interface 유틸리티 필요: *ggi*, *overlapGgi* (frozen rotor), *mixingPlane* (stage)  
→ **foam-extend** (community-driven version) 에서 제공

# 1. Introduction

## steadyUniversalMRFFoam(SUMF) Solver

~/foam/foam-extend-4.1/applications/solvers/compressible/steadyUniversalMRFFoam/



- PIMPLE solver 기반
  - 회전 도메인의 에너지 방정식을 풀기 위해 Enthalpy 대신 Rothalpy (“iEqn.H”)로 계산.
  - 정지 및 회전 도메인 사이 Interface에 Rothalpy jump 적용.
- *overlapGgi* 또는 *mixingPlane* 가능.

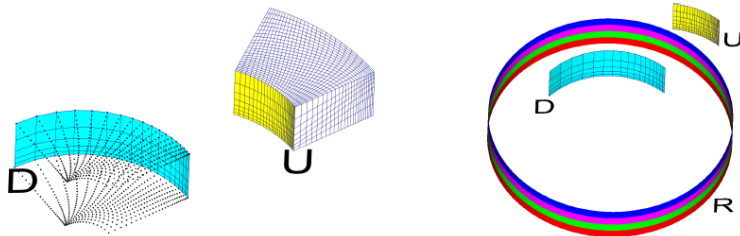


Figure 2. Illustration of a simple mixingPlane interface configuration.

(M Beaudoin *et al*, 2014, “Evaluation of an improved mixing plane interface for OpenFOAM,” *IOP Conf. Ser.: Earth Environ. Sci.* **22** 022004)

### mixingPlane

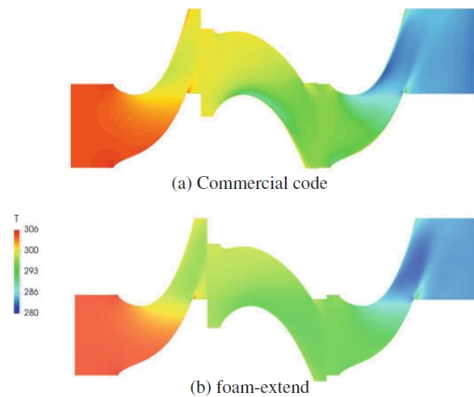


Fig. 6 Temperature field comparison

(I. De Dominicis *et al*, “Enhanced Turbomachinery Capabilities for Foam-Extend: Development and Validation,” OpenFOAM®: selected Papers of the 11<sup>th</sup> OpenFOAM Workshop, pp. 145-155, 2019.)

### A calculation of 1.5 axial turbine with foam-extend

```

foam-extend: Open Source CFD
Version: 4.1
Web: http://www.foam-extend.org
For copyright notice see file Copyright

Build : 4.1-f2c57318fe7
Exec  : steadyUniversalMRFFoam1013
Date  : Oct 28 2022
Time  : 16:02:21
Host  : "DESKTOP-OL3ODLI"
PID   : 728
CtrlDict : "/mnt/c/openfoam_2022/E3LPT/E3LPT_221017_80um_k-e/system/controlDict"
Case   : "/mnt/c/openfoam_2022/E3LPT/E3LPT_221017_80um_k-e"
nProc  : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
allowSystemOperations : Disallowing user-supplied system call operations

// *****

Create time

Create mesh for time = 0

Initializing the GGI interpolator between master/shadow patches: R1-TO-R1-INTERNAL-SIDE-1/R1-TO-R1-INTERNAL-SIDE-2
Initializing the GGI interpolator between master/shadow patches: R1-TO-R1-PERIODIC-1-SIDE-1/R1-TO-R1-PERIODIC-1-SIDE-2
Initializing the GGI interpolator between master/shadow patches: R2-TO-R2-INTERNAL-SIDE-1/R2-TO-R2-INTERNAL-SIDE-2

Creating MRF for cell zone R1. rpm = -3208.7002
Creating MRF for cell zone R2. rpm = -3208.7002

Starting time loop

Creating MachNumber for field U
Creating minMaxField for field U
Creating minMaxField for field p
Creating minMaxField for field rho
Creating minMaxField for field T
Time = 1

smoothSolver: Solving for Ux, Initial residual = 1, Final residual = 0.038212446, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 1, Final residual = 0.071812734, No Iterations 2
smoothSolver: Solving for Uz, Initial residual = 1, Final residual = 0.027832545, No Iterations 4
smoothSolver: Solving for p, Initial residual = 1, Final residual = 0.25017412, No Iterations 1000
smoothSolver: Solving for p, Initial residual = 0.14842559, Final residual = 0.085921163, No Iterations 1000
time step continuity errors : sum local = 2236.0458, global = 1916.1807, cumulative = 1916.1807
smoothSolver: Solving for p, Initial residual = 0.31784944, Final residual = 0.085931858, No Iterations 1000
smoothSolver: Solving for p, Initial residual = 0.087681165, Final residual = 0.067266582, No Iterations 1000
time step continuity errors : sum local = 1563.3826, global = 1497.7787, cumulative = 3413.9593
smoothSolver: Solving for i, Initial residual = 0.99860139, Final residual = 0.056989679, No Iterations 4
smoothSolver: Solving for epsilon, Initial residual = 0.9857257, Final residual = 0.099239388, No Iterations 68
smoothSolver: Solving for k, Initial residual = 1, Final residual = 0.021788685, No Iterations 4
ExecutionTime = 58.83 s  ClockTime = 64 s

Mach number min = 0.00155111 max = 0.89548717
Field U magnitude min = 0 (0.63371076) max = 340.56255 (340.56255)
Field p min = 100000 (100000) max = 400000 (400000)
Field rho min = 1.318262 (1.3196661) max = 3.1730009 (3.1730009)
Field T min = 360.94405 (360.94405) max = 434.26873 (434.26873)
Time = 2

smoothSolver: Solving for Ux, Initial residual = 0.26004993, Final residual = 0.0085877531, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 0.26044107, Final residual = 0.020881107, No Iterations 2
    
```

### An Execution of steadyUniversalMRFFoam

# 1. Introduction

## Default condition

- Viscous heating term 생략.
- KISTI(Korea Institute of Science and Technology Information) supercomputer
  - 한 노드에서 최대 68 코어 사용. (foam-extend-4.1에서 병렬 노드 계산 불가.)
- 세 가지 터빈 형상으로 테스트.
  - CFX tutorial에서 제공하는 터빈
  - 항우연 무냉각 터빈
  - NASA/GE E3 Low Pressure Turbine
- foam-extend의 압축성 MRF solver 사용 시 여러 이슈(문제) 발생.  
→ 대표적으로 *mixingPlane* 및 **벽 온도** 이슈



KISTI 5<sup>th</sup> national supercomputer, Cray CS500

<https://www.ksc.re.kr/eng/resources/nurion>

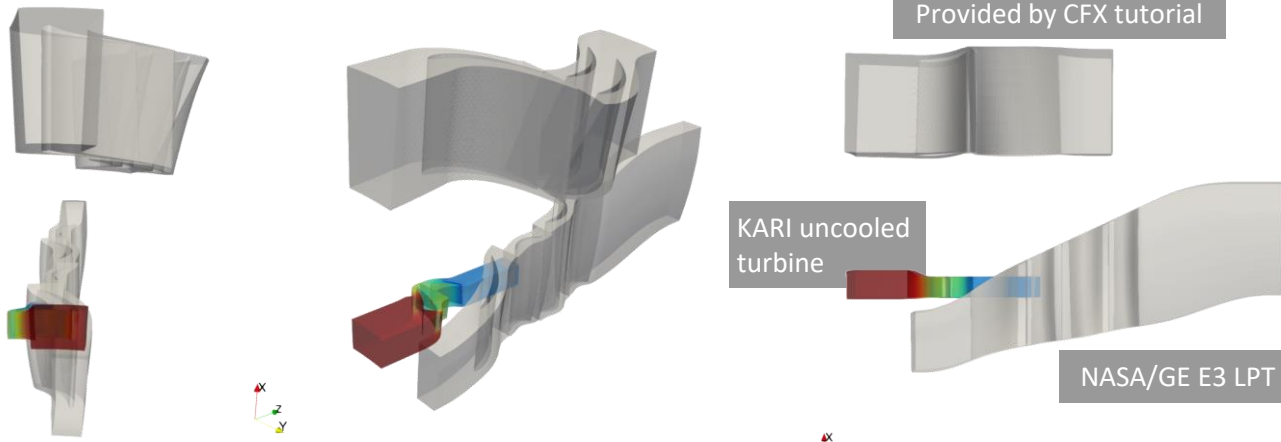
### fvSolution

```
solver      smoothSolver;

PIMPLE
{
  nOuterCorrectors      1;
  nCorrectors            2;
  nNonOrthogonalCorrectors 1;
  compressible           yes;
  convergence            1e-5;
}

relaxationFactors
{
  // p                0.4;
  // rho              0.01;
  "(U|i|k|omega|p|rho)" 0.5;
}
```

fvSolution default setting



Three different turbine geometries in various perspectives

## **2. mixingPlane Issue**

# 2. mixingPlane Issue

## Rothalpy

- 작동유체가 터보기계를 지날 때의 축 A에 작용하는 토크 ( $\vec{\tau} = \vec{r} \times \vec{F}$ )

$$\tau = \dot{m}(r_2 C_{\theta 2} - r_1 C_{\theta 1})$$

여기서,  $C$ 는 유체의 절대 속도,  $C_{\theta}$ 는 유체의 접선 속도

- 일률(Power)

$$P = \tau \Omega = \dot{m}(U_{rot,2} C_{\theta 2} - U_{rot,1} C_{\theta 1})$$

여기서, blade speed,  $U_{rot} = \Omega r$

- Specific work

$$\Delta w = \frac{P}{\dot{m}} = U_{rot,2} C_{\theta 2} - U_{rot,1} C_{\theta 1} = h_{02} - h_{01}$$

- $\Delta w > 0$ : 압축기 또는 펌프,  $\Delta w < 0$ : 터빈

- Rothalpy

$$h_{01} - U_{rot,1} C_{\theta 1} = h_{02} - U_{rot,2} C_{\theta 2} = Const.$$

$$\therefore i = h_0 - U_{rot} C_{\theta} \quad \text{Rothalpy jump}$$

- 회전이 없다면 “Rothalpy = Total Enthalpy”

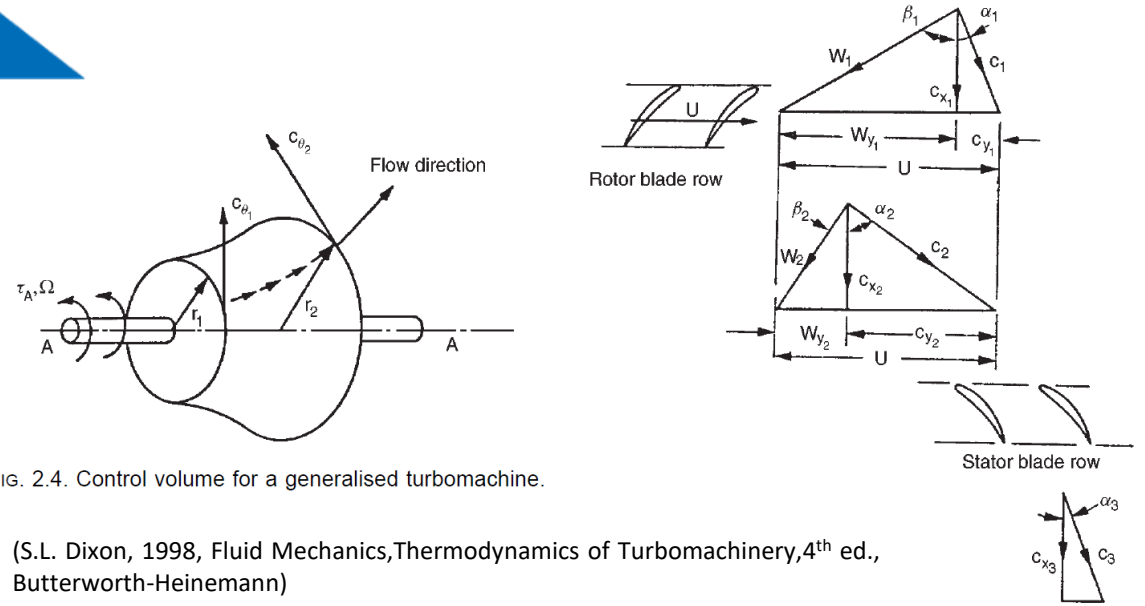


Fig. 2.4. Control volume for a generalised turbomachine.

(S.L. Dixon, 1998, Fluid Mechanics, Thermodynamics of Turbomachinery, 4th ed., Butterworth-Heinemann)

Fig. 5.2. Velocity diagrams for a compressor stage.

※ 속도 삼각형으로 Rothalpy ( $W$ 는 유체의 상대속도)

$$\begin{aligned} h_0 &= h + \frac{1}{2} C^2 \\ i &= h + \frac{1}{2} C^2 - U_{rot} C_{\theta} = h + \frac{1}{2} (C_x^2 + C_{\theta}^2 - 2U_{rot} C_{\theta}) \\ &= h + \frac{1}{2} (C_x^2 + (C_{\theta} - U_{rot})^2 - U_{rot}^2) \\ &= h + \frac{1}{2} (C_x^2 + W_{\theta}^2 - U_{rot}^2) = h + \frac{1}{2} (W^2 - U_{rot}^2) \\ &= h + \frac{1}{2} (U_{rel}^2 - U_{rot}^2) \end{aligned}$$



# 2. mixingPlane Issue

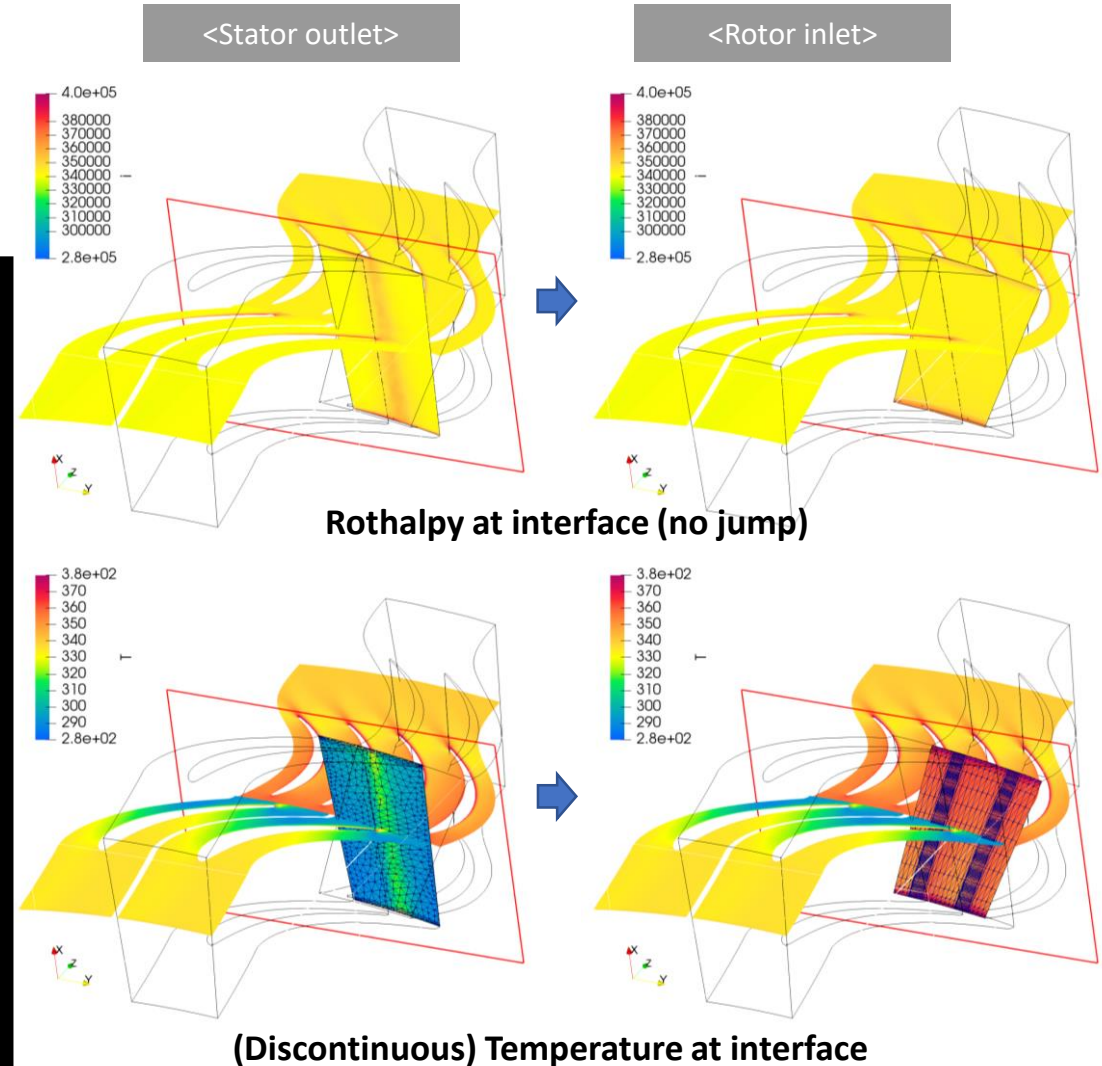
## MixingPlane error

- 경고 메시지
  - 솔버와 라이브러리에서 정의된 변수의 대소문자가 달라서 발생하는 문제.
  - Rothalpy jump 미작동 → 온도 불연속

```
Time = 1
smoothSolver: Solving for Ux, Initial residual = 1, Final residual = 0.0366622, No Iterations 2
smoothSolver: Solving for Uy, Initial residual = 1, Final residual = 0.0738305, No Iterations 2
smoothSolver: Solving for Uz, Initial residual = 1, Final residual = 0.0224172, No Iterations 2
BiCGStab: Solving for p, Initial residual = 1, Final residual = 0.000349875, No Iterations 112
time step continuity errors : sum local = 0.816323, global = 0.120505, cumulative = 0.120505
BiCGStab: Solving for p, Initial residual = 0.434237, Final residual = 0.000235576, No Iterations 81
time step continuity errors : sum local = 1.27011, global = -0.525613, cumulative = -0.405108

From function void gradientEnthalpyFvPatchScalarField::updateCoeffs(const vectorField& Up)
in file derivedFvPatchFields/gradientEnthalpy/gradientEnthalpyFvPatchScalarField.C at line 141
Velocity fields U or Urot or Utheta not found. Performing enthalpy value update for field i and patch 0
objects
46
(
(interpolate(alphaEff)*magSf)
-(devRhoReff&&grad(Urel))
K
MRFZones
RASProperties
S
T
TPrevIter
U
Urel
Urot
V
alpha
alphaEff
.....
```

Warning message in log at first run of compressible MRF solver



# 2. mixingPlane Issue

## MixingPlane error correction

- 참고,  
<https://www.cfd-online.com/Forums/openfoam/210386-unphysical-temperature-mixing-plane.html#post779039>

```

July 29, 2020, 03:55 #13
WhiteW
Member
Join Date: Dec 2015
Posts: 74
Rep Power: 9

Thanks to Nicola we also identified a solution for the case when UTheta is defined as volScalarField in:
src/thermophysicalModels/basic/derivedFvPatchFields/mixingPlaneEnthalpyJump/mixingPlaneEnthalpyJumpFvPatchFields.C

Furthermore, running the tutorials of the steadyCompressibleMRFfoam, for Fe4.1, I also encountered the following warning:

From function void gradientEnthalpyFvPatchScalarField::updateCoeffs(c onst vectorField& Up)
in file derivedFvPatchFields/fixeEnthalpyFvPatchScalarField.C at line 135
Velocity fields U or URot or UTheta not found. Performing enthalpy value update for field i and patch 0

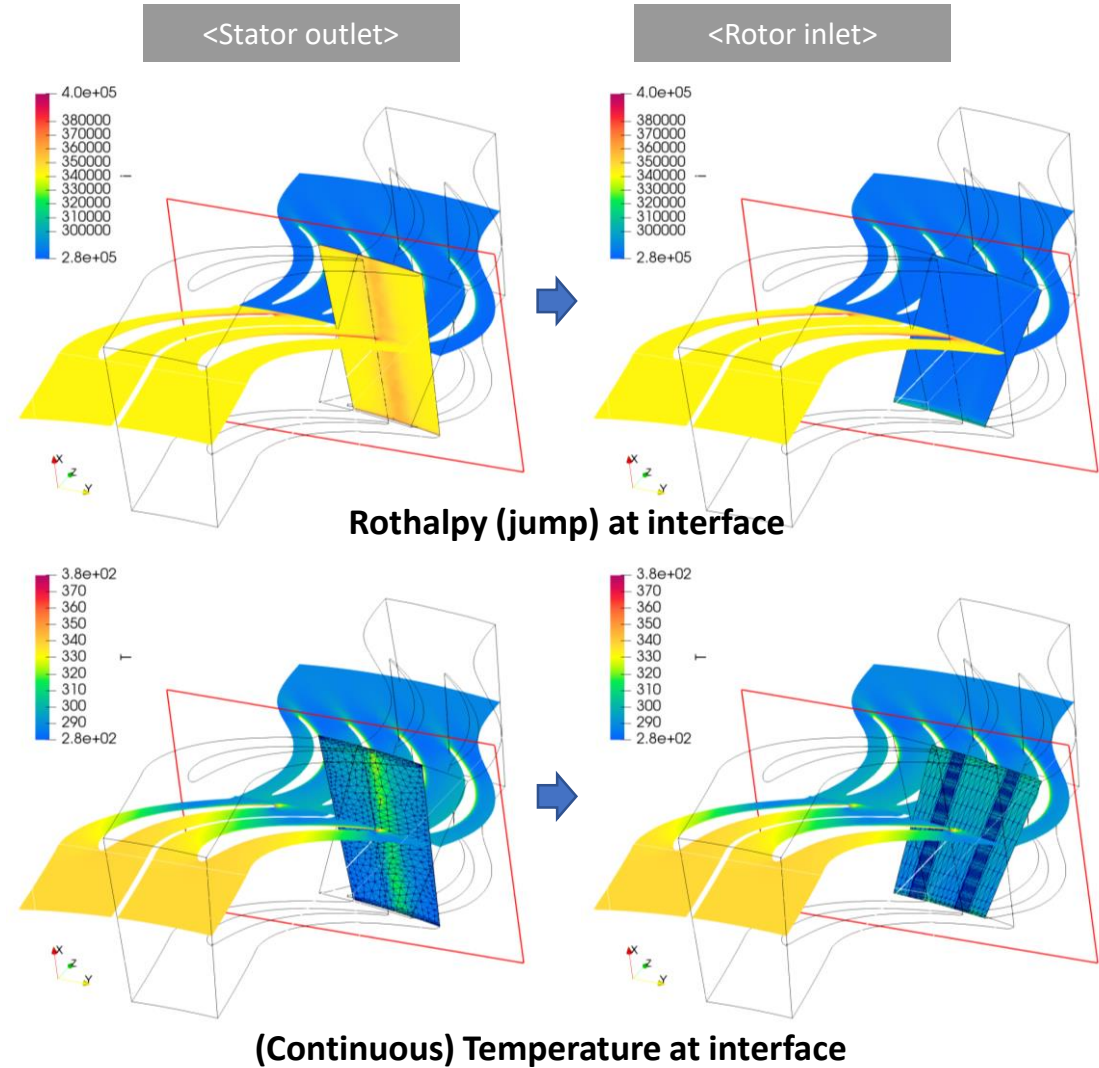
From function void gradientEnthalpyFvPatchScalarField::updateCoeffs(c onst vectorField& Up)
in file derivedFvPatchFields/gradientEnthalpyFvPatchScalarField.C at line 141
Velocity fields U or URot or UTheta not found. Performing enthalpy value update for field i and patch 1

This solution also calculates URot as a copy of Urot to remove the previous warning.

1) The first thing needed is to add 3 fields in createFields.H:
- tangential velocity vector UThetaV
- tangential velocity scalar UTheta
- rotational velocity vector URot
This can be done for example with:

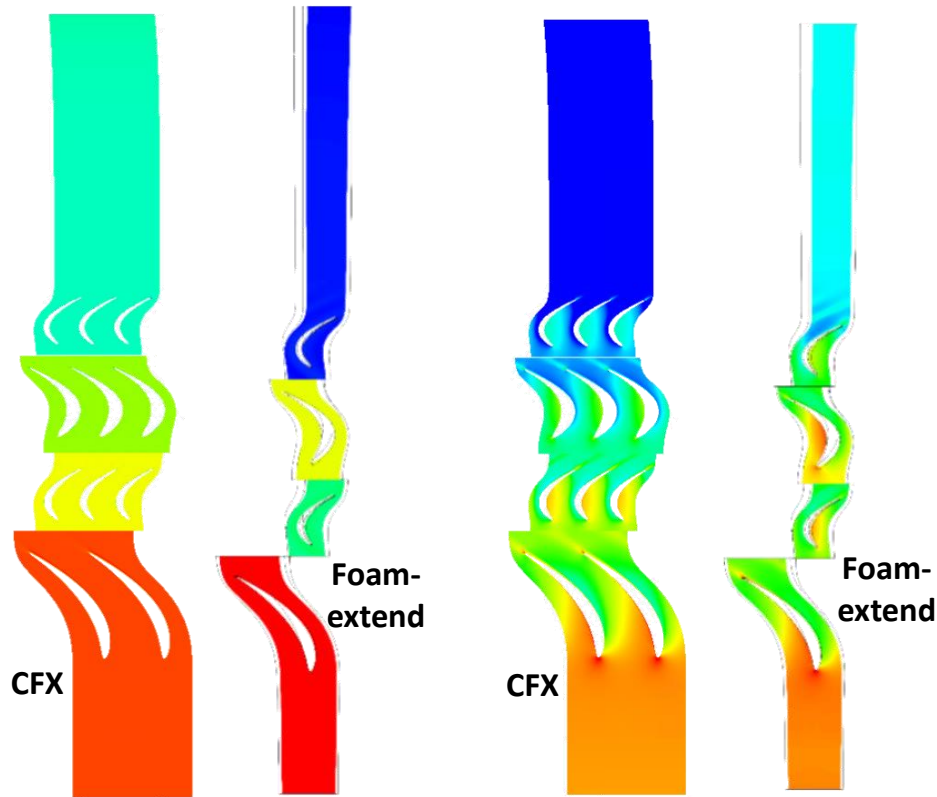
Code:
volVectorField UThetaV
(
    "UThetaV",
    U
);
    
```

- 관련된 MRFZone files에 UTheta function code 추가.
- createFields.H 및 iEqn.H에 URot 및 UTheta 변수 추가.
- Rothalpy jump 작동 → 온도 연속



# 2. mixingPlane Issue

## Multi-stage error



Comparison of rothalpy

Comparison of temperature

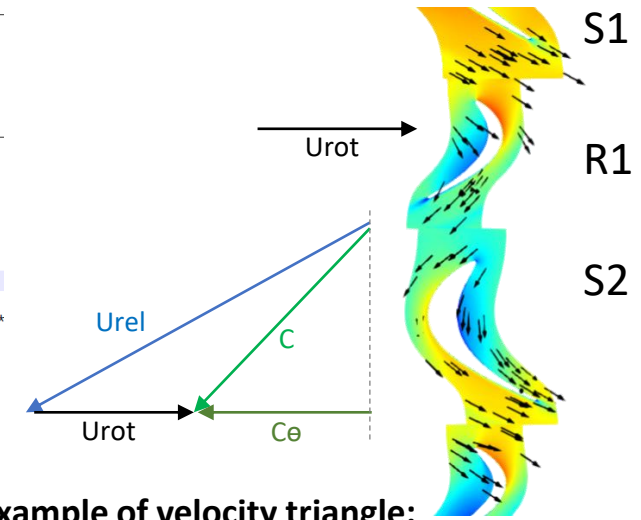
- 단과 단 사이의 온도 불연속 다시 발생.
- $U_{rot} = U - U_{rel}$
- Rothalpy:  $i = h_0 - U_{rot}C_\theta$
- $C_\theta$ 와  $U_{rot}$ 는 같은 벡터 평면에 위치하지만, 반대 방향일 수 있음.
- OpenFOAM에서  $U_{rot}$ 는 벡터 data.
- 따라서,  $U_{rot}C_\theta$ 는 내적이어야 하고, 그러면 각 속도 방향이 반대일 때 음수해를 산출할 수 있음.

```

1  /*----- C++ -----*/
2
3  \ \ \ \ \ Field      | foam-extend: Open Source CFD
4  \ \ \ \ \ Operation  | Version:      4.1
5  \ \ \ \ \ And        | Web:         http://www.foam-extend.org
6  \ \ \ \ \ Manipulation |
7  /*-----*/
8  FoamFile
9  {
10     version      2.0;
11     format       ascii;
12     class        volVectorField;
13     location     "2000";
14     object       Urot;
15 }
16 // *****
17
18 dimensions     [0 1 -1 0 0 0];
19
20 internalField   nonuniform List<vector>
21 704520
22 (
23  (-1.343462 -41.384598 0)
24  (-1.2631974 -41.387237 0)
25  (-1.1831929 -41.389721 0)
26  (-1.1029095 -41.392048 0)
27  (-1.0222642 -41.394203 0)

```

An example of  $U_{rot}$



An example of velocity triangle: different direction between  $C_\theta$  and  $U_{rot}$

# 2. mixingPlane Issue

## Original code

```
createFields.H
68
69
70 // corrections
71 volVectorField URot //corrected(1008)
72 (
73     "URot",
74     U - Urel
75 );
76 /* volVectorField URot
77 (
78     IObject
79     (
80         "URot",
81         runtime.timeName(),
82         mesh,
83         IObject::NO_READ,
84         IObject::AUTO_WRITE
85     ),
86     U - Urel
87 );*/
88
89 // create U_thetaV
90 volVectorField UThetaV
91 (
92     "UThetaV",
93     U
94 );
95
96 // Create UTheta
97 volScalarField UTheta
98 (
99     "UTheta",
100     mag(U)
101 );
102 /* volScalarField UTheta
103 (
104     IObject
105     (
106         "UTheta",
107         runtime.timeName(),
108         mesh,
109         IObject::NO_READ,
110         IObject::AUTO_WRITE
111     ),
112     mag(U)
113 );
114 mrfZones.UTheta(UThetaV);
115 UTheta = mag(UThetaV); */
116
117
118

mixingPlaneEnthalpyJumpFvPatchFields.C
67
68 if
69 (
70     !this->db().objectRegistry::found(URotName)
71     || !this->db().objectRegistry::found(UThetaName)
72 )
73 {
74     // Velocities not available, do not update
75     InfoIn
76     (
77         "void gradientEnthalpyFvPatchScalarField::"
78         "updateCoeffs(const vectorField& Up)"
79     ) << "Velocity fields " << URotName << " or "
80     << UThetaName << " not found. "
81     << "Performing enthalpy value update" << endl;
82
83     jump_ = 0;
84 }
85 else
86 {
87     const fvPatchVectorField& URotp =
88         lookupPatchField<volVectorField, vector>(URotName);
89
90     const fvPatchScalarField& UThetap =
91         lookupPatchField<volScalarField, scalar>(UThetaName);
92
93     if (rotating)
94     {
95         // We can either make jump_ on neighbour field and interpolate (in
96         // jumpOverlapGgi) or interpolate first, then add jump_ for
97         // internalField
98         const scalarField UThetaIn = UThetap.patchInternalField();
99
100         jump_ = - (
101             mag(UThetaIn)
102             *mag(URotp.patchInternalField())
103         );
104     }
105     else
106     {
107         const scalarField UThetaIn = UThetap.patchNeighbourField();
108
109         jump_ = (
110             mag(UThetaIn)
111             *mag(URotp.patchNeighbourField())
112         );
113     }
114 }
115
116 jumpMixingPlaneFvPatchField<scalar>::updateCoeffs();
117
```

UTheta defined as scalar

UTheta transformed to magnitude

Scalar product with magnitude value of UTheta and URot

createFields.H

mixingPlaneEnthalpyJumpFvPatchFields.C

# 2. mixingPlane Issue

## Corrected code

```
rtg@DESKTOP-OL3ODLI: ~
38 #include "surfaceFields.H"
39
40 // * * * * * Static Data Members * * * * * //
41
42 namespace Foam
43 {
44     makeTemplatePatchTypeField
45     (
46         fvPatchScalarField,
47         mixingPlaneEnthalpyJumpFvPatchScalarField
48     );
49 }
50
51 // * * * * * Member Functions * * * * * //
52
53 template<
54 void Foam::mixingPlaneEnthalpyJumpFvPatchField<Foam::scalar>::updateCoeffs()
55 {
56     if (updated())
57     {
58         return;
59     }
60
61     // Get access to relative and rotational velocity
62     const word URotName("URot");
63     /*const word UThetaName("UTheta");*/
64     const word UThetaName("UThetaV");
65
66     jump_ = 0;
67
68     if
69     (
70         !this->db().objectRegistry::found(URotName)
71         || !this->db().objectRegistry::found(UThetaName)
72     )
73     {
74         // Velocities not available, do not update
75         InfoIn
76         (
77             "void gradientEnthalpyFvPatchScalarField::"
78             "updateCoeffs(const vectorField& Up)"
79         ) << "Velocity fields " << URotName << " or "
80         << UThetaName << " not found. "
81         << "Performing enthalpy value update" << endl;
```

Vector value, UThetaV

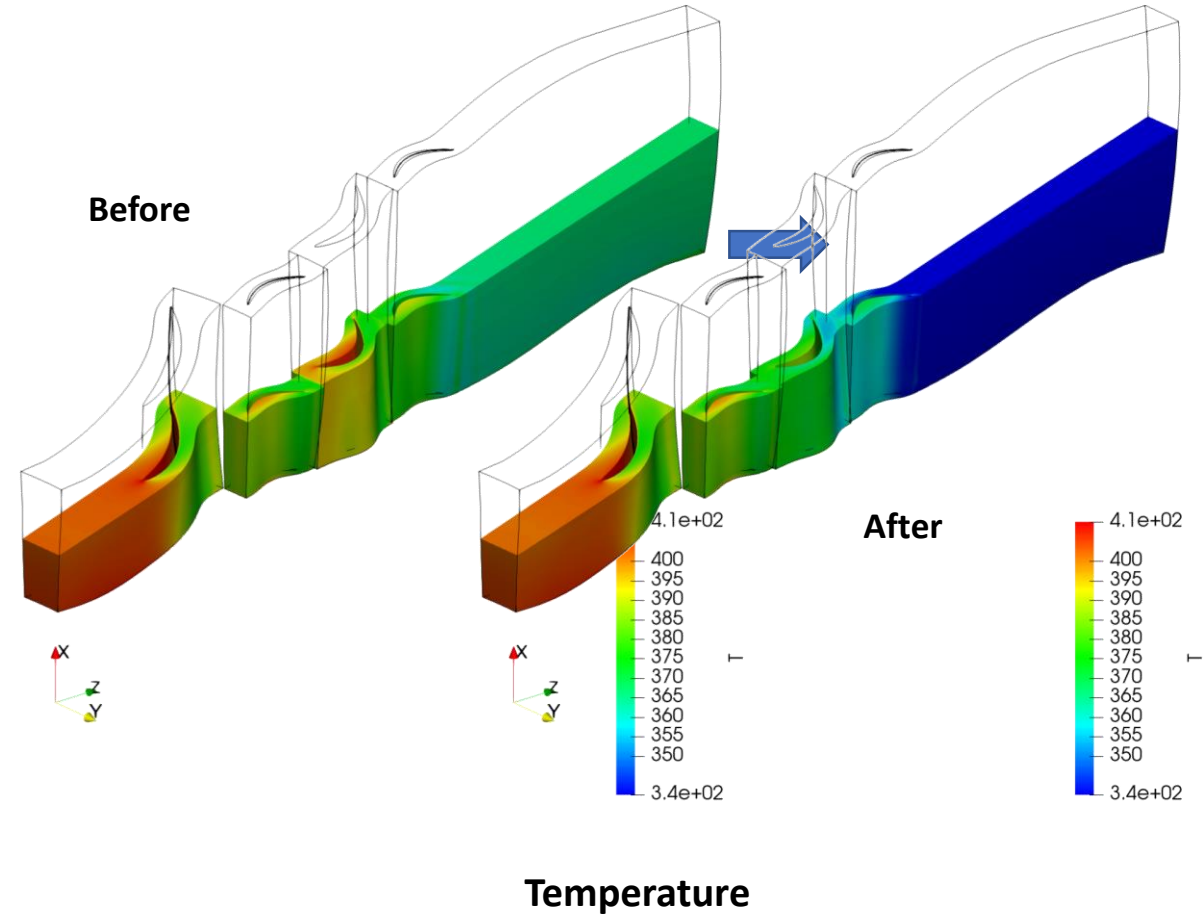
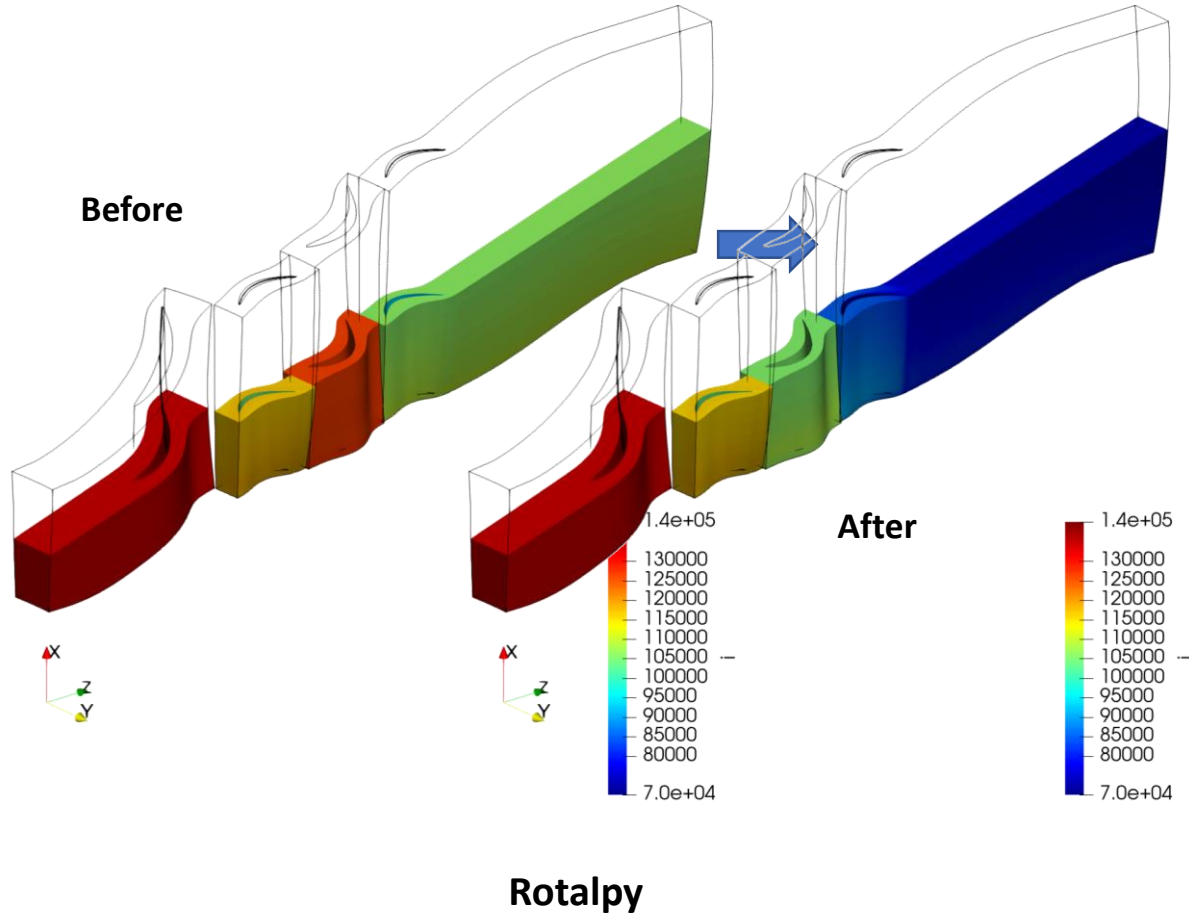
```
82
83     jump_ = 0;
84 }
85 else
86 {
87     const fvPatchVectorField& URotp =
88         lookupPatchField<volVectorField, vector>(URotName);
89
90     /*
91     const fvPatchScalarField& UThetap =
92     lookupPatchField<volScalarField, scalar>(UThetaName);*/
93     const fvPatchVectorField& UThetap =
94     lookupPatchField<volVectorField, vector>(UThetaName);
95
96     if (rotating_)
97     {
98         // We can either make jump_ on neighbour field and interpolate (in
99         // jumpOverlapGgi) or interpolate first, then add jump_ for
100         // internalField
101         /*const scalarField UThetaIn = UThetap.patchInternalField();*/
102
103         jump_ = - (
104             // mag(UThetaIn)
105             /*mag(URotp.patchInternalField()
106             UThetap.patchInternalField()
107             &URotp.patchInternalField()
108             );
109         }
110     else
111     {
112         /*const scalarField UThetaIn = UThetap.patchNeighbourField();*/
113
114         jump_ = (
115             // mag(UThetaIn)
116             /*mag(URotp.patchNeighbourField()
117             UThetap.patchNeighbourField()
118             &URotp.patchNeighbourField()
119             );
120         }
121     }
122     jumpMixingPlaneFvPatchField<scalar>::updateCoeffs();
123 }
124
125 // *****
126
```

Inner product with vector value of UthetaV and URot

## 2. mixingPlane Issue

### Corrected result of E3 LPT 2 stages

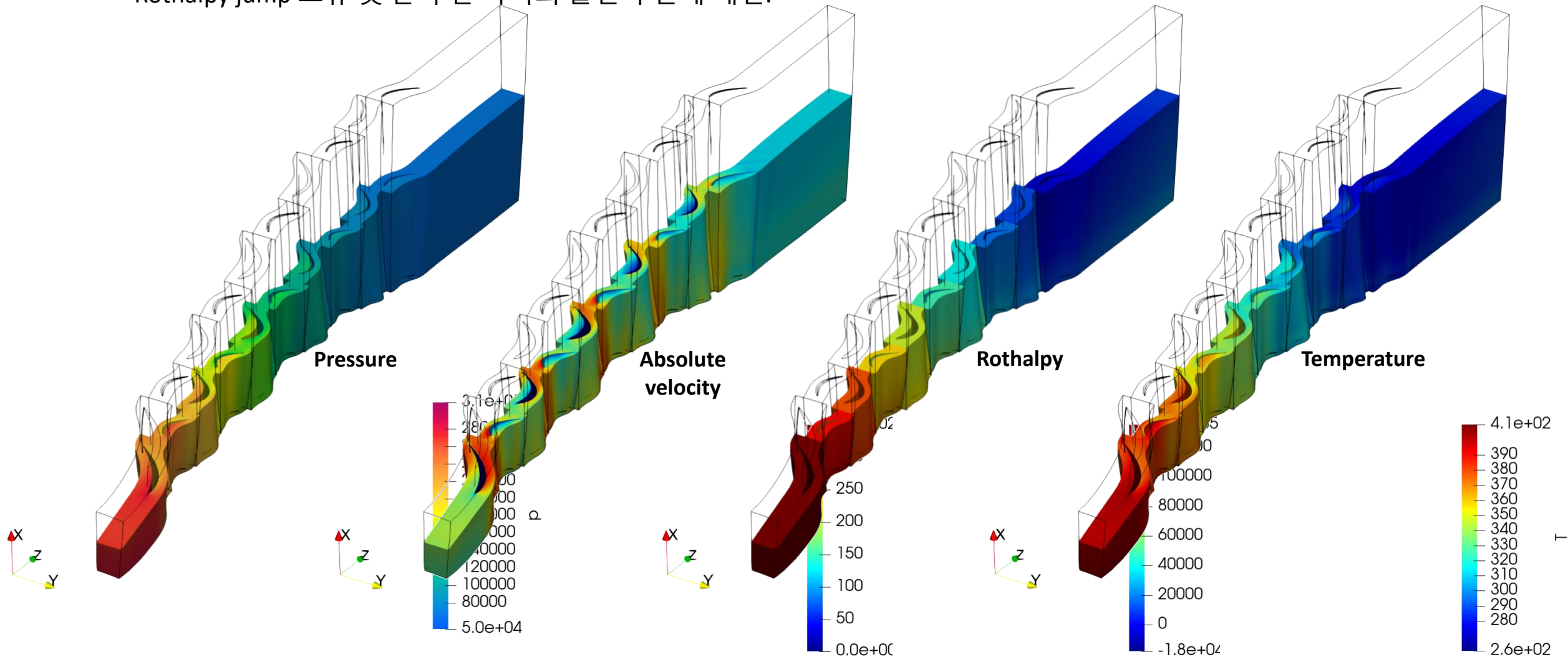
- Rothalpy jump 오류 및 단과 단 사이의 불연속 문제 해결.



## 2. mixingPlane Issue

### Corrected result of E3 LPT 5 stages

- Rothalpy jump 오류 및 단과 단 사이의 불연속 문제 해결.



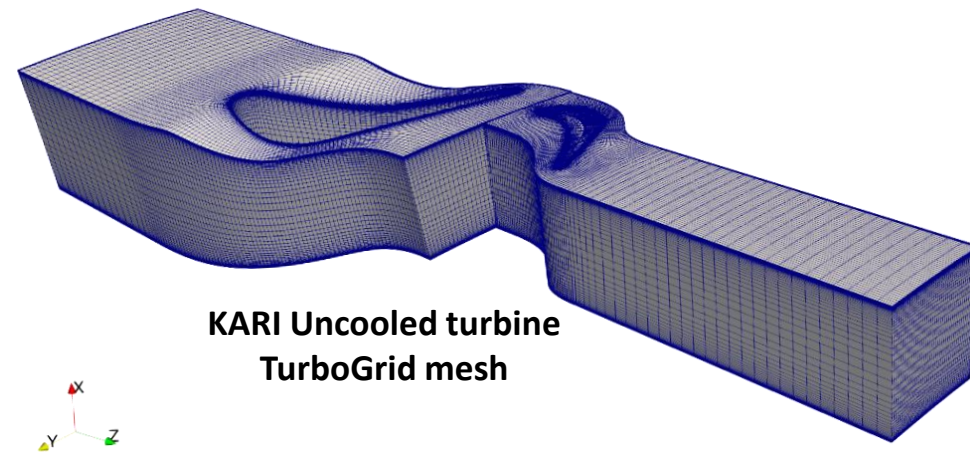
# **3. Wall Temperature Issue**



# 3. Wall Temperature Issue

## KARI turbine

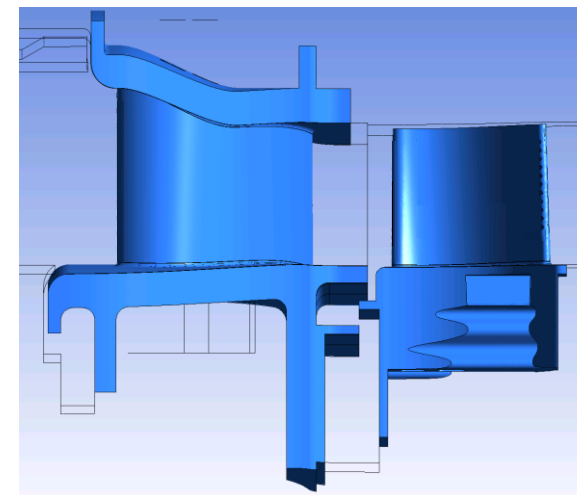
- 항우연에서 설계한 (un)cooled turbine
  - 56 Nozzle vanes
  - 104 Turbine blades
- ANSYS TurboGrid → OpenFOAM grid로 변환
  - (Fine grid)
  - Wall distance =  $2e-6$  m
  - Number of element of stator: 643,376
  - Number of element of rotor: 1,102,895



KARI Uncooled turbine  
TurboGrid mesh

### Boundary conditions of design point

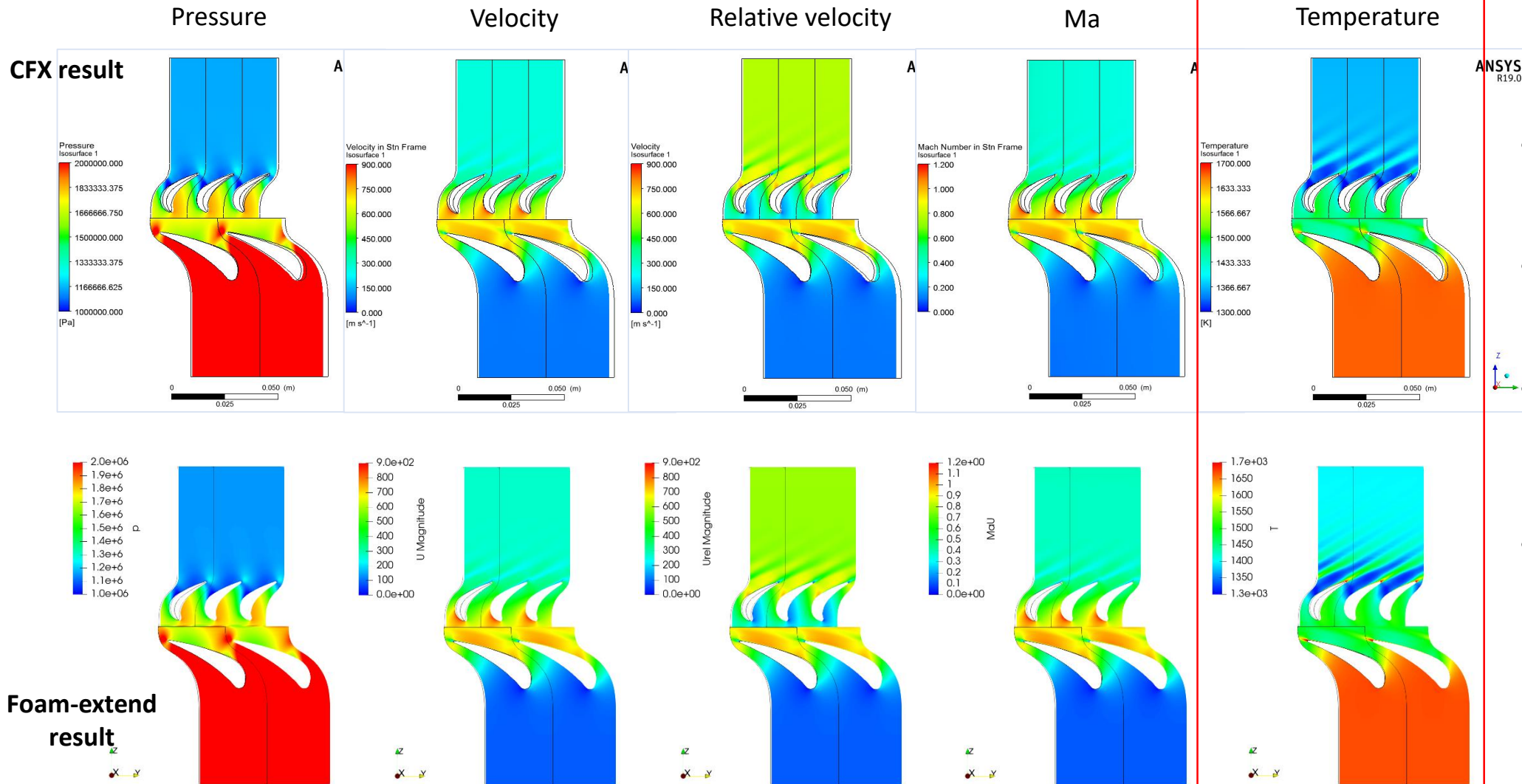
|                | CFX   | OpenFoam  |
|----------------|---|---|
| Inlet          | Total Pressure 30.685 [bar]<br>Total Temperature 1400 [C] | totalPressure 3068500 //Pa<br>totalTemperature 1673 //K |
| Outlet         | Average Static Pressure 11.5 [bar]                        | uniform 1150000 // Pa                                   |
| Rotating / MRF | -17000 [rev min <sup>-1</sup> ]                           | -1780.236 // rad/s                                      |
| Fluid          | Calibrated Ideal Gas                                      | sutherlandTransport<br>janafThermo                      |
| Turbulence     | SST   | kOmegaSST   |



Side view of 1<sup>st</sup> turbine stage

# 3. Wall Temperature Issue

## Calculation results at span 0.5 of KARI turbine



- 오픈폼 결과의 온도 상승 → 벽 온도 상승에서 기인
- Viscous heating term에 상관 없이 벽 온도 상승.
- *gradientEnthalpy* 또는 *zeroGradient*에 상관 없이 벽 온도 상승.
- Rotating domain 뿐 아니라 stationary domain에서도 벽 온도 상승.

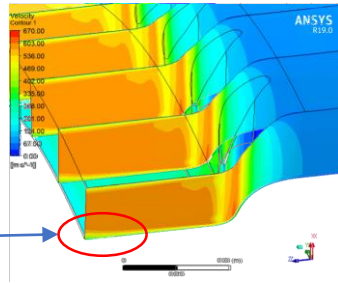
# 3. Wall Temperature Issue

## Wall temperature rising, analysis

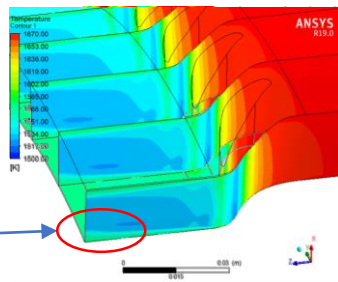
- 모든 솔버가 속도 예측은 잘 해주는 한편, OpenFOAM v.7 버전에서만 상용 코드와 유사한 온도장 산출.
- foam-extend 버전의 솔버들은 온도장 및 벽 온도 예측 실패. → 에너지 방정식 검토 필요.

$$h_{stag} = h + \frac{1}{2}U^2$$

Velocity



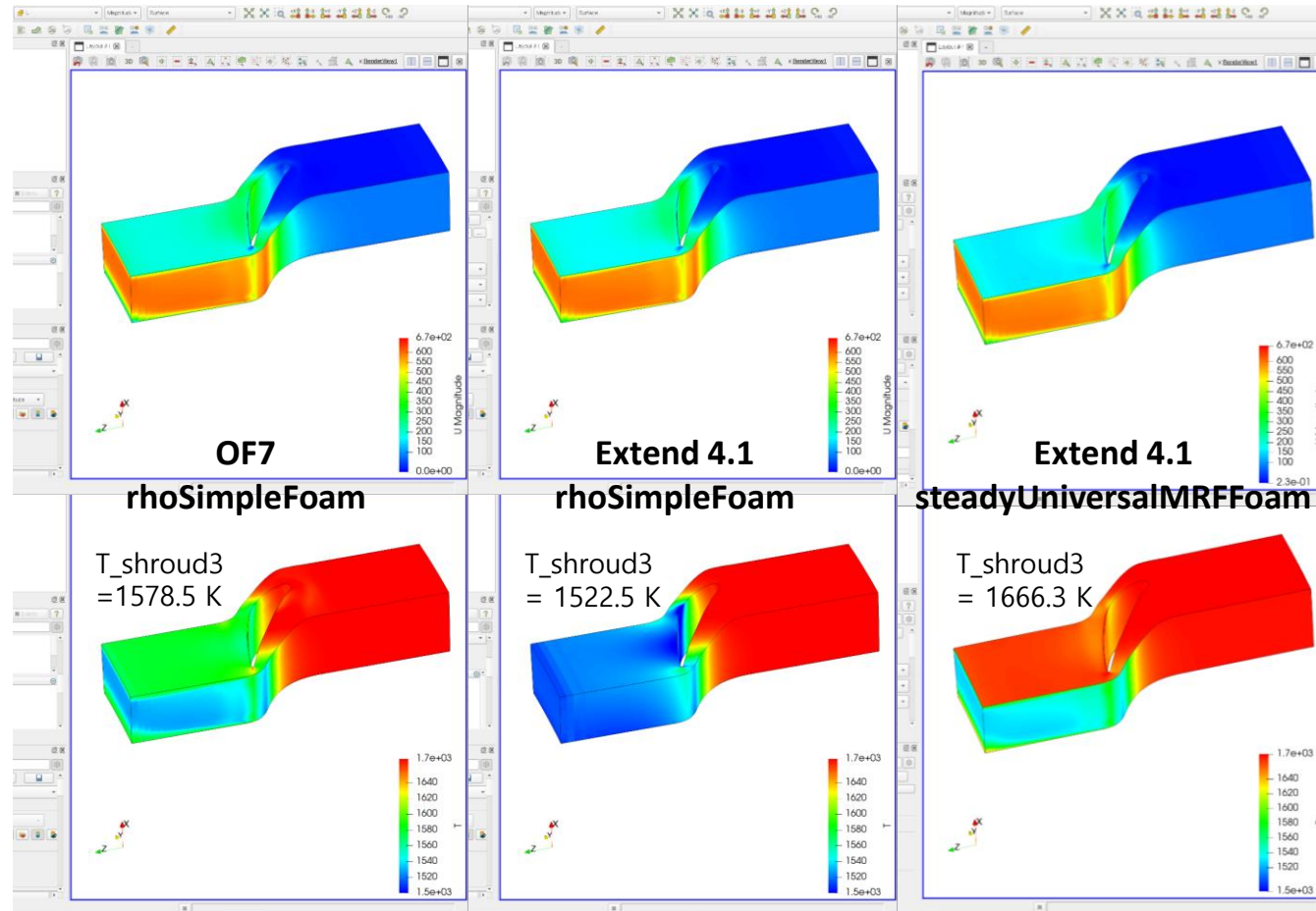
CFX



Temperature

Velocity boundary layer develops toward outlet

Temperature boundary layer develops along with velocity boundary layer



### 3. Wall Temperature Issue

#### Energy Equation (1), Intergral equation

$$\frac{dE}{dt} = \frac{d}{dt} \left( \int_{CV} e \rho dV \right) + \int_{CS} e \rho (\mathbf{U} \cdot \mathbf{n}) dA = \frac{dQ}{dt} - \frac{dW}{dt} + \text{Source}$$

$$\begin{aligned} \frac{d}{dt} \left( \int_{CV} e \rho dV \right) + \int_{CS} e \rho (\mathbf{U} \cdot \mathbf{n}) dA &= \dot{Q} - (\dot{W}_s + \dot{W}_p + \dot{W}_v) + \text{Source} \\ &= \dot{Q} - \left( \dot{W}_s + \int_{CS} p (\mathbf{U} \cdot \mathbf{n}) dA - \int_{CS} \boldsymbol{\tau} \cdot \mathbf{U} dA \right) + \text{Source} \end{aligned}$$

where,

$$e = e_{internal} + e_{kinetic} + e_{potential} + e_{other} = \hat{u} + \frac{1}{2} U^2 + gz$$

$$\frac{dW}{dt} = \dot{W} = \dot{W}_{shaft} + \dot{W}_{pressure} + \dot{W}_{viscous stresses}$$

# 3. Wall Temperature Issue

## Energy Equation (2), Differential equation

- Integral equation

$$\frac{d}{dt} \left( \int_{CV} e \rho dV \right) + \int_{CS} e \rho (\mathbf{U} \cdot \mathbf{n}) dA = \dot{Q} - \left( \dot{W}_s + \int_{CS} p (\mathbf{U} \cdot \mathbf{n}) dA + \dot{W}_v \right) + \text{Source}$$

- Differential equation

L. H. S.

$$= \left[ \frac{\partial}{\partial t} (\rho e) + \frac{\partial}{\partial x} (\rho u e) + \frac{\partial}{\partial y} (\rho v e) + \frac{\partial}{\partial z} (\rho w e) \right] dx dy dz$$

$$= \left[ \frac{\partial}{\partial t} (\rho e) + \nabla \cdot (\rho \mathbf{U} e) \right] dx dy dz$$

R. H. S.

$$= \left[ \nabla \cdot (k \nabla T) - \nabla \cdot (\mathbf{U} p) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + \text{source} \right] dx dy dz$$

$$\frac{\partial}{\partial t} (\rho e) + \nabla \cdot (\rho \mathbf{U} e) = \nabla \cdot (k \nabla T) - \nabla \cdot (\mathbf{U} p) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

where,

$$\dot{Q} = - \left[ \frac{\partial}{\partial x} (q_x) + \frac{\partial}{\partial y} (q_y) + \frac{\partial}{\partial z} (q_z) \right] dx dy dz$$

$$= -(\nabla \cdot \mathbf{q}) dx dy dz = -(\nabla \cdot (-k \nabla T)) dx dy dz$$

$$\dot{W}_v = - \left[ \frac{\partial}{\partial x} (u \tau_{xx} + v \tau_{xy} + w \tau_{xz}) + \frac{\partial}{\partial y} (u \tau_{yx} + v \tau_{yy} + w \tau_{yz}) + \frac{\partial}{\partial z} (u \tau_{zx} + v \tau_{zy} + w \tau_{zz}) \right] dx dy dz$$

$$= -\nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) dx dy dz$$

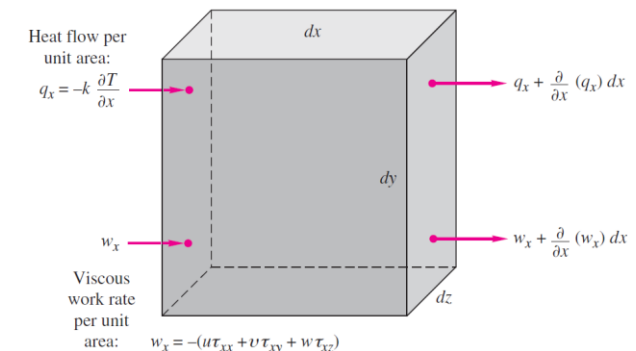


Fig. 4.6 Elemental cartesian control volume showing heat flow and viscous work rate terms in the x direction.

# 3. Wall Temperature Issue

## Energy Equation (3), Differential equations

- Expansion of equation

$$\frac{\partial}{\partial t}(\rho e) + \nabla \cdot (\rho \mathbf{U} e) = \nabla \cdot (k \nabla T) - \nabla \cdot (\mathbf{U} p) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

$$\frac{\partial}{\partial t}(\rho(\hat{u} + K)) + \nabla \cdot \left( \rho \mathbf{U} \left( \hat{u} + K + \frac{p}{\rho} \right) \right) = \nabla \cdot (k \nabla T) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

- Internal energy form

$$\frac{\partial}{\partial t}(\rho \hat{u}) + \frac{\partial}{\partial t}(\rho K) + \nabla \cdot (\rho \mathbf{U} \hat{u}) + \nabla \cdot (\rho \mathbf{U} K) + \nabla \cdot (\mathbf{U} p) = \nabla \cdot \left( \rho \alpha \frac{C_p}{C_v} \nabla \hat{u} \right) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

- Enthalpy form

$$\frac{\partial}{\partial t} \left( \rho \left( h - \frac{p}{\rho} \right) \right) + \frac{\partial}{\partial t}(\rho K) + \nabla \cdot \left( \rho \mathbf{U} \left( h - \frac{p}{\rho} \right) \right) + \nabla \cdot (\rho \mathbf{U} K) + \nabla \cdot (\mathbf{U} p) = \nabla \cdot (\rho \alpha \nabla h) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

- Total Enthalpy form

$$\frac{\partial}{\partial t}(\rho h_0) - \frac{\partial}{\partial t}(p) + \nabla \cdot (\rho \mathbf{U} h_0) = \nabla \cdot (\rho \alpha \nabla (h_0 - K)) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

- Then, Rothalpy form??

$$\frac{\partial}{\partial t}(\rho i) + \nabla \cdot (\rho \mathbf{U} (i + U_{rot} U_{\theta})) = \nabla \cdot (\rho \alpha \nabla ((i + U_{rot} U_{\theta}) - K)) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

where

$$e = \hat{u} + K + gz$$

$$K = \frac{1}{2} U^2$$

$$h = \hat{u} + \frac{p}{\rho}$$

$$h_0 = h + K$$

$$\alpha = \frac{k}{\rho C_p}$$

$$C_p = \frac{dh}{dT}$$

$$C_v = \frac{d\hat{u}}{dT}$$

$$\begin{aligned} i &= h_0 - U_{rot} U_{\theta} \\ &= h + \frac{1}{2} (U_{rel}^2 - U_{rot}^2) \end{aligned}$$

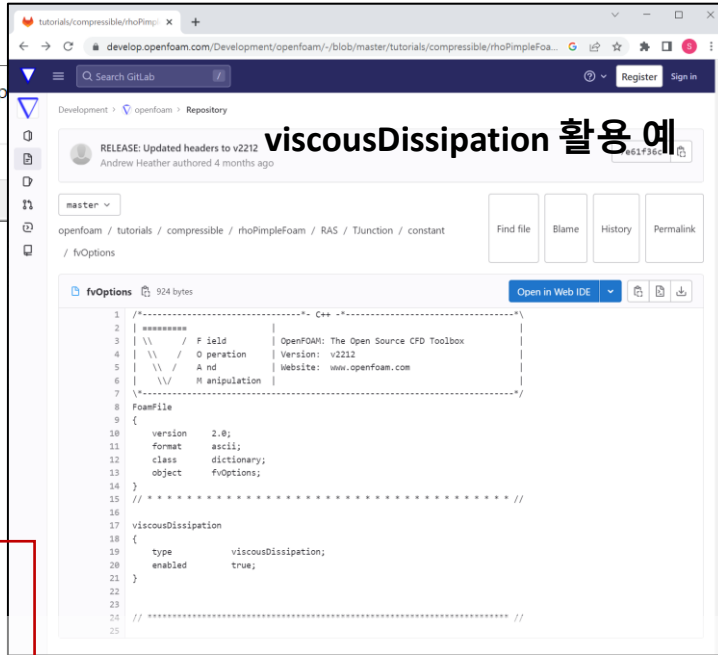
# 3. Wall Temperature Issue

## OpenFOAM energy equation, rhoPimpleFoam (v1906)

```

1 {
2   volScalarField& he = thermo.he ();
3
4   fvScalarMatrix EEqn
5   (
6     fvm::ddt(rho, he) + fvm::div(phi, he)
7   + fvc::ddt(rho, K) + fvc::div(phi, K)
8   + (
9     he.name() == "e"
10    ? fvc::div
11    (
12      fvc::absolute(phi/fvc::interpolate(rho), U),
13      p,
14      "div(phiiv,p)"
15    )
16    : -dpdt
17  )
18  - fvm::laplacian(turbulence->alphaEff(), he)
19  ==
20  fvOptions(rho, he)
21  );
22  EEqn.relax();
23
24  fvOptions.constrain(EEqn);
25
26  EEqn.solve();
27
28  fvOptions.correct(he);
29
30  thermo.correct();
31
32 }
33

```



• Energy equation of rhoPimpleFoam (v1906)

• Internal energy form

$$\frac{\partial}{\partial t}(\rho \hat{u}) + \nabla \cdot (\rho \mathbf{U} \hat{u}) + \frac{\partial}{\partial t}(\rho K) + \nabla \cdot (\rho \mathbf{U} K) + \nabla \cdot \left( \frac{\rho \mathbf{U}}{\rho} p \right) = \nabla \cdot \left( \rho \alpha \frac{C_p}{C_v} \nabla \hat{u} \right) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

• Enthalpy form

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\rho \mathbf{U} h) + \frac{\partial}{\partial t}(\rho K) + \nabla \cdot (\rho \mathbf{U} K) - \frac{\partial p}{\partial t} = \nabla \cdot (\rho \alpha \nabla h) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

(.com) 버전에서는 fvOptions를 통해 viscousDissipation을 추가할 수 있음.  
<https://www.openfoam.com/documentation/guides/latest/doc/guide-fvoptions-sources-viscous-dissipation.html>

# 3. Wall Temperature Issue

## OpenFOAM energy equation, rhoSimpleFoam (OF7)

```

1 {
2   volScalarField& he = thermo.he();
3
4   fvScalarMatrix EEqn
5   (
6     fvm::div(phi, he)
7     + (
8       he.name() == "e"
9       ? fvc::div(phi, volScalarField("Ekp", 0.5*magSqr(U) + p/rho))
10      : fvc::div(phi, volScalarField("K", 0.5*magSqr(U)))
11    )
12    - fvm::laplacian(turbulence->alphaEff(), he);
13  ==
14    fvOptions(rho, he)
15  );
16
17  EEqn.relax();
18
19  fvOptions.constrain(EEqn);
20
21  EEqn.solve();
22
23  fvOptions.correct(he);
24
25  thermo.correct();
26
27 }

```

- Energy equation of rhoSimpleFoam (OpenFOAM v.7)

- Internal energy form

$$\frac{\partial}{\partial t}(\rho \hat{u}) + \nabla \cdot (\rho \mathbf{U} \hat{u}) + \frac{\partial}{\partial t}(\rho K) + \nabla \cdot (\rho \mathbf{U} K) + \nabla \cdot \left( \rho \mathbf{U} \frac{p}{\rho} \right) = \nabla \cdot \left( \rho \alpha \frac{C_p}{C_v} \nabla \hat{u} \right) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

- Enthalpy form

$$\frac{\partial}{\partial t}(\rho h) + \nabla \cdot (\rho \mathbf{U} h) + \frac{\partial}{\partial t}(\rho K) + \nabla \cdot (\rho \mathbf{U} K) - \frac{\partial p}{\partial t} = \nabla \cdot (\rho \alpha \nabla h) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$



# 3. Wall Temperature Issue

## OpenFOAM energy equation, steadyUniversalMRFFoam

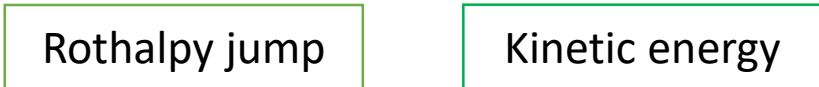
```

24 fvScalarMatrix iEqn
25 (
26     fvm::div(phi, i)
27     + fvm::SuSp(-fvc::div(phi), i)
28     - fvm::laplacian(turbulence->alphaEff(), i)
29     ==
30     // Viscous heating: note sign (devRhoReff has a minus in it)
31     - (turbulence->devRhoReff() && fvc::grad(U))
32 );
33
34 iEqn.relax();
35
36 iEqn.solve();
37
38 // From rothalpy, calculate enthalpy after solution of rothalpy
39 h = i + 0.5*(magSqr(Urot) - magSqr(Urel));
40 h.correctBoundaryConditions();
41
42 // Update thermo for new h
43 thermo.correct();
44 psis = thermo.psi()/thermo.Cp()*thermo.Cv();
45
46 }
    
```

- steadyUniversalMRFFoam (foam-extend-4.1)
- Rothalpy equation

$$\nabla \cdot (\rho \mathbf{U}(i + U_{rot} U_{\theta})) = \nabla \cdot (\rho \alpha \nabla ((i + U_{rot} U_{\theta}) - K)) + \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

$$\nabla \cdot (\rho \mathbf{U} i) + \nabla \cdot (\rho \mathbf{U} U_{rot} U_{\theta}) - \nabla \cdot (\rho \alpha \nabla i) - \nabla \cdot (\rho \alpha \nabla (U_{rot} U_{\theta} - K)) = \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

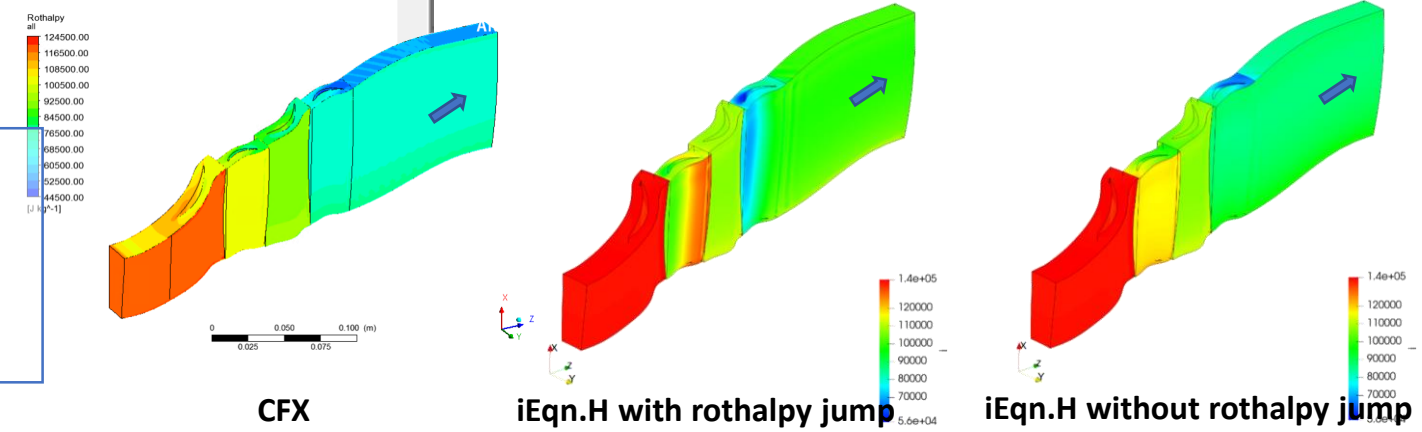


```

30 fvScalarMatrix iEqn
31 (
32     fvm::div(phi, i)
33     + fvc::div(phi, volScalarField("K1", mag(Urot)*mag(UthetaV))) //corrected 230302
34     + fvm::SuSp(-fvc::div(phi), i)
35     - fvm::laplacian(turbulence->alphaEff(), i)
36     - fvc::laplacian(turbulence->alphaEff(), volScalarField("K2", (mag(Urot)*mag(UthetaV)-0.5*magSqr(U))))
37     ==
38     // Viscous heating: note sign (devRhoReff has a minus in it)
39     - (turbulence->devRhoReff() && fvc::grad(U))
40 );
41
    
```

Corrected iEqn.H

- 단순 Rothalpy 수식 치환으로 해결 불가.
    - Rothalpy jump 항이 중복되어 오류 발생.
    - Rothalpy jump 항 제거 시에도 벽 온도 상승.
- 다른 접근 필요.

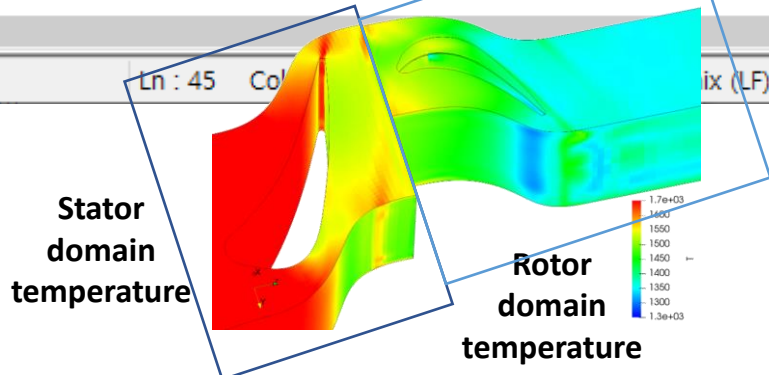


# 3. Wall Temperature Issue

## steadyUniversalMRFFoam – correction

```

30 fvScalarMatrix iEqn
31 (
32     fvm::div(phi, i)
33     + fvc::div(phi, volScalarField("K", 0.5*magSqr(Urel))) //corrected 230302
34     - fvm::laplacian(turbulence->alphaEff(), i)
35 //
36 // Viscous heating: note sign (devRhoReff has a minus in it)
37 //
38     - (turbulence->devRhoReff() && fvc::grad(U))
39 );
40 iEqn.relax();
41
42 iEqn.solve();
43
44 // From rothalpy, calculate enthalpy after solution of rothalpy equation
45 h = i + 0.5*(magSqr(Urot));
46 // h = i + 0.5*(magSqr(Urot) - magSqr(Urel));
47 // h = i + (URot&UThetaV);
    
```



- Enthalpy form energy equation
 
$$\nabla \cdot (\rho \mathbf{U} h) + \nabla \cdot (\rho \mathbf{U} K) - \nabla \cdot (\rho \alpha \nabla h) = \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

- Rothalpy 새 정의: Rothalpy에서 운동에너지 항을 분리.
 
$$i_0 = i + \frac{1}{2} U_{rel}^2 = h_0 - U_{rot} U_{\theta} = h + \frac{1}{2} U_{rel}^2 - \frac{1}{2} U_{rot}^2$$

$$\therefore i = h - \frac{1}{2} U_{rot}^2$$

- Corrected rothalpy form energy equation
 
$$\nabla \cdot (\rho \mathbf{U} i) + \nabla \cdot (\rho \mathbf{U} (0.5 U_{rel}^2)) - \nabla \cdot (\rho \alpha \nabla i) = \nabla \cdot (\mathbf{U} \cdot \boldsymbol{\tau}_{ij}) + s$$

- 정지 도메인에선 위 식은 Enthalpy 식과 동일.
- 회전 도메인에선 Rothalpy jump가 적용된 후 Enthalpy 식과 같은 형태로 계산되며, 이후 Rothalpy는 Enthalpy로 변환되고, 이후 온도가 산출됨.

# 3. Wall Temperature Issue

## Correct fixedEnthalpyFvPatchScalarField.C

```

1  //
2  //  F i e l d       foam-extend: Open Source CFD
3  //  O p e r a t i o n   Version: 4.1
4  //  A n d           Web: http://www.foam-extend.org
5  //  M a n i p u l a t i o n   For copyright notice see file Copyright
6
7  License
8
9  This file is part of foam-extend.
10
11  foam-extend is free software: you can redistribute it and/or modify it
12  under the terms of the GNU General Public License as published by the
13  Free Software Foundation, either version 3 of the License, or (at your
14  option) any later version.
15
16  foam-extend is distributed in the hope that it will be useful, but
17  WITHOUT ANY WARRANTY; without even the implied warranty of
18  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
19  General Public License for more details.
20
21  You should have received a copy of the GNU General Public License
22  along with foam-extend. If not, see <http://www.gnu.org/licenses/>.
23
24
25
26
27 #include "basicThermo.H"
28 #include "fvMesh.H"
29 #include "zeroGradientFvPatchFields.H"
30 #include "fixedEnthalpyFvPatchScalarField.H"
31 #include "gradientEnthalpyFvPatchScalarField.H"
32 #include "mixedEnthalpyFvPatchScalarField.H"
33 #include "fixedInternalEnergyFvPatchScalarField.H"
34 #include "gradientInternalEnergyFvPatchScalarField.H"
35 #include "mixedInternalEnergyFvPatchScalarField.H"
36
37 /* ***** Private Static Data ***** */
38
39 namespace Foam
40 {
41     defineTypeNameAndDebug(basicThermo, 0);
42
43     // ***** Protected Member Functions ***** //
44
45     Foam::wordList Foam::basicThermo::hBoundaryTypes()
46     {
47         const volScalarField::GeometricBoundaryField& tbf = T_.boundaryField();
48         wordList hbt = tbf.types();
49         forAll(tbf, patchi)
50         {
51             if (isA<fixedValueFvPatchScalarField>(tbf[patchi]))
52             {
53                 hbt[patchi] = fixedEnthalpyFvPatchScalarField::typeName;
54             }
55             else if
56             {
57                 isA<zeroGradientFvPatchScalarField>(tbf[patchi])
58                 || isA<fixedGradientFvPatchScalarField>(tbf[patchi])
59             }
60             {
61                 hbt[patchi] = gradientEnthalpyFvPatchScalarField::typeName;
62             }
63             else if (isA<mixedFvPatchScalarField>(tbf[patchi]))
64             {
65                 hbt[patchi] = mixedEnthalpyFvPatchScalarField::typeName;
66             }
67         }
68     }
69 }

```

basicThermo.C

```

102 fvPatchScalarField& Tw =
103     const_cast<fvPatchScalarField&>(thermo.T().boundaryField() [patchi]);
104 Tw.evaluate();
105
106 if
107 {
108     dimensionedInternalField().name() == db().mangleFileName("h")
109 }
110 {
111     operator==(thermo.h(Tw, patchi));
112 }
113 else if
114 {
115     dimensionedInternalField().name() == db().mangleFileName("i")
116 }
117 {
118     // Get access to relative and rotational velocities
119     const word UName("U");
120     const word URotName("URot");
121     const word UThetaName("UTheta");
122
123     if
124     {
125         !this->db().objectRegistry::found(URotName)
126         || !this->db().objectRegistry::found(UName)
127         || !this->db().objectRegistry::found(UThetaName)
128     }
129     {
130         // Velocities not available
131         InfoIn
132         (
133             "void gradientEnthalpyFvPatchScalarField::"
134             "updateCoeffs(const fvPatchVectorField& Up, const fvPatchVectorField& Urotp)"
135         ) << "Velocity fields not available" << endl;
136         const fvPatchVectorField& Up =
137             lookupPatchField<fvPatchVectorField, vector>(UName);
138         const fvPatchVectorField& Urotp =
139             lookupPatchField<fvPatchVectorField, vector>(URotName);
140         const fvPatchScalarField& UThetap =
141             lookupPatchField<fvPatchScalarField, scalar>(UThetaName);
142         operator==(
143             thermo.h(Tw, patchi)
144             + 0.5*magSqr(Up)
145             - mag(UThetap)*mag(Urotp)
146         );
147     }
148     else
149     {
150         const fvPatchVectorField& Up =
151             lookupPatchField<fvPatchVectorField, vector>(UName);
152         const fvPatchVectorField& Urotp =
153             lookupPatchField<fvPatchVectorField, vector>(URotName);
154         const fvPatchScalarField& UThetap =
155             lookupPatchField<fvPatchScalarField, scalar>(UThetaName);
156         operator==(
157             thermo.h(Tw, patchi)
158             + 0.5*magSqr(Up)
159             - mag(UThetap)*mag(Urotp)
160         );
161     }
162 }
163 else
164 {
165     operator==(thermo.hs(Tw, patchi));
166 }
167
168 fixedValueFvPatchScalarField::updateCoeffs();

```

fixedEnthalpyFvPatchScalarField.C

- basicThermo에서 참고하는 라이브러리 중 fixedEnthalpy 항목 존재.
- 원 코드에서 기존의 로탈피 정의가 사용됨:  $i_0 = h + \frac{1}{2}U^2 - U_{rot}U_{\theta}$
- 따라서 새로 정의한 로탈피 식 적용:  $i = h - \frac{1}{2}U_{rot}^2$
- gradientEnthalpyFvPatchScalarField.C도 동일하게 수정

```

145 }
146 else
147 {
148     const fvPatchVectorField& Urelp = //correct 230830
149         lookupPatchField<fvPatchVectorField, vector>(UName); //correct 230830
150     const fvPatchVectorField& Up =
151         lookupPatchField<fvPatchVectorField, vector>(UName);
152     const fvPatchVectorField& Urotp =
153         lookupPatchField<fvPatchVectorField, vector>(URotName);
154     const fvPatchScalarField& UThetap =
155         lookupPatchField<fvPatchScalarField, scalar>(UThetaName);
156     operator==(
157         thermo.h(Tw, patchi)
158         + 0.5*magSqr(Up) //original
159         - mag(UThetap)*mag(Urotp) //original
160         - 0.5*magSqr(Urotp) //correct 230830
161     );
162 }
163 else
164 {
165     operator==(thermo.hs(Tw, patchi));
166 }
167
168 fixedValueFvPatchScalarField::updateCoeffs();
169 }
170 }
171
172 }
173
174 fixedValueFvPatchScalarField::updateCoeffs();
175 }
176
177
178 // ***** Private Static Data ***** //
179 #include "fixedEnthalpyFvPatchScalarField.C" 190L, 5511C

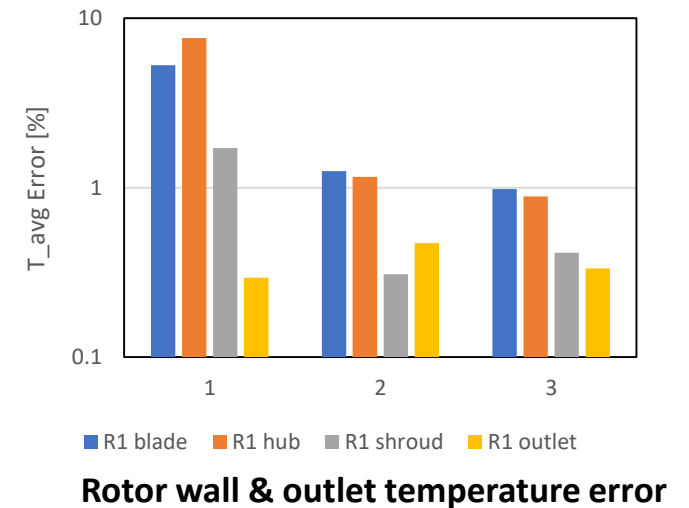
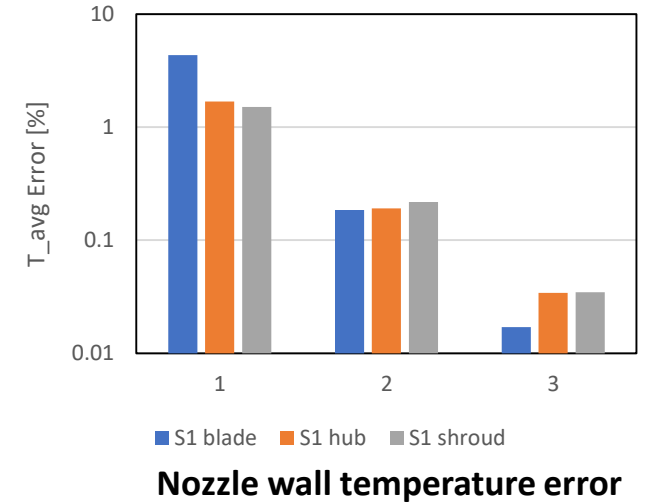
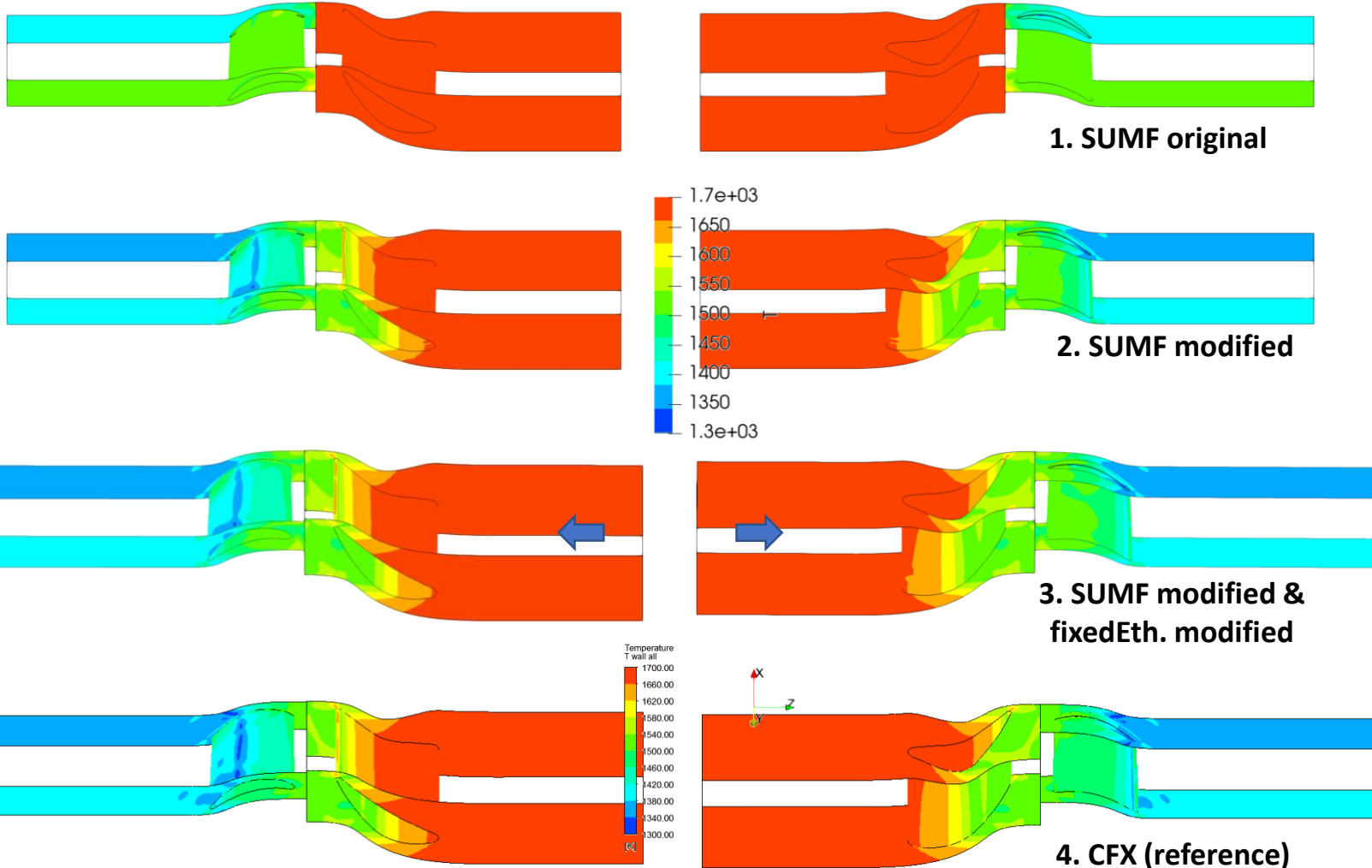
```

fixedEnthalpyFvPatchScalarField.C modified

# 3. Wall Temperature Issue

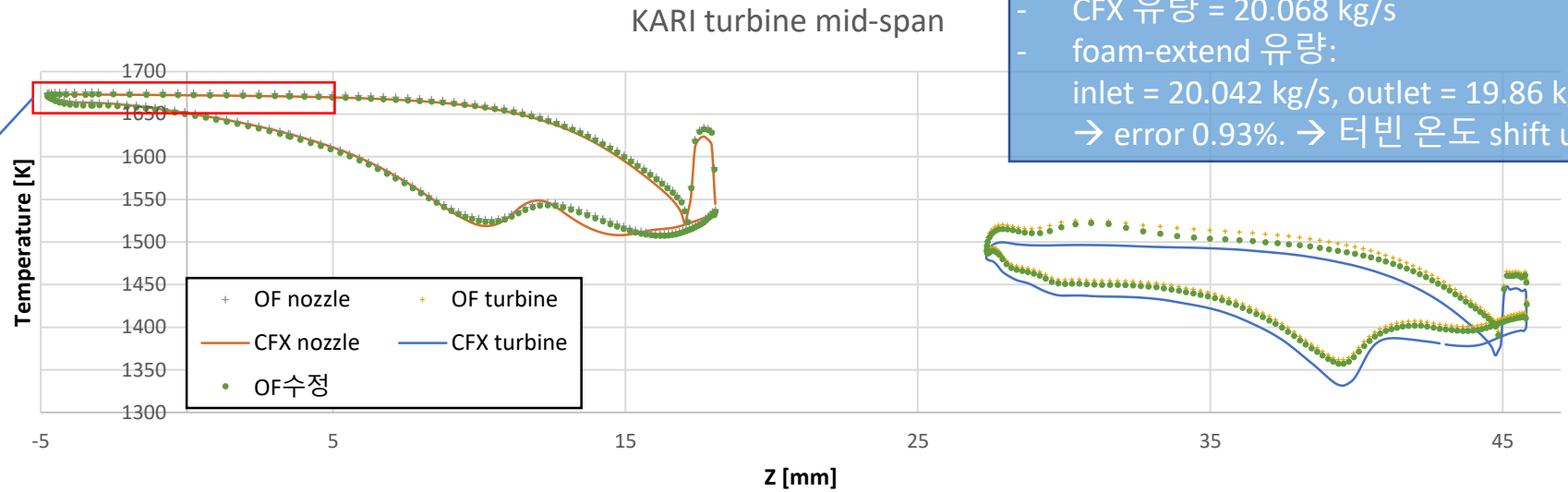
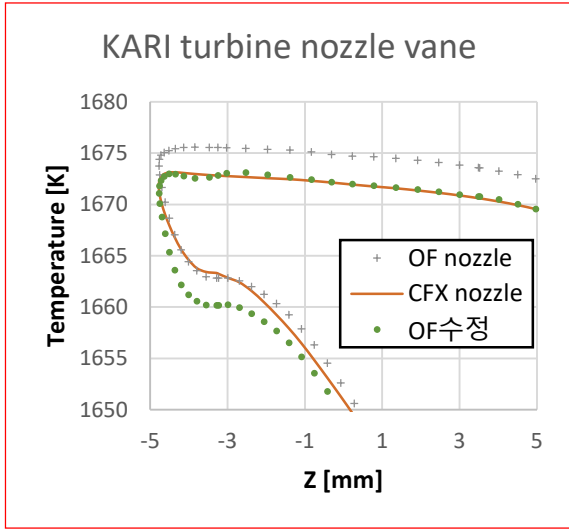
## KARI turbine wall temperature comparison

- *fixedEnthalpy* 수정 전/후, 그림에선 비슷해 보이지만 (2, 3번), CFX 결과와 비교 시 노즐 벽면의 오차가 상당히 줄어들음.



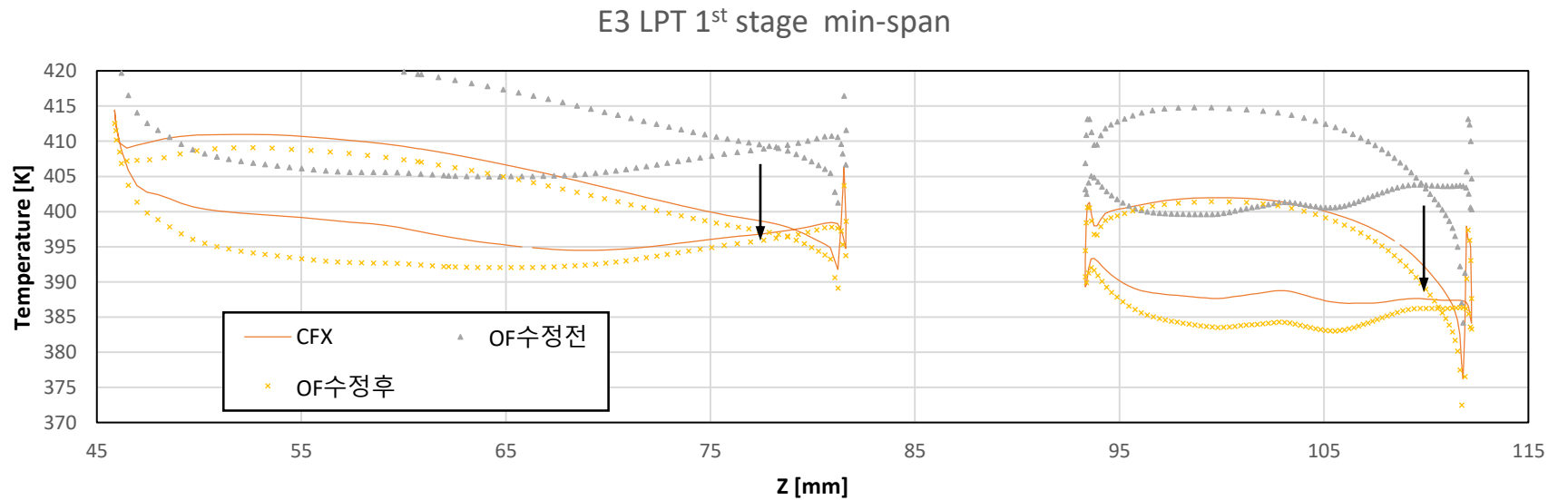
# 3. Wall Temperature Issue

## KARI & E3 LPT 1<sup>st</sup> stage mid-span vane temperature



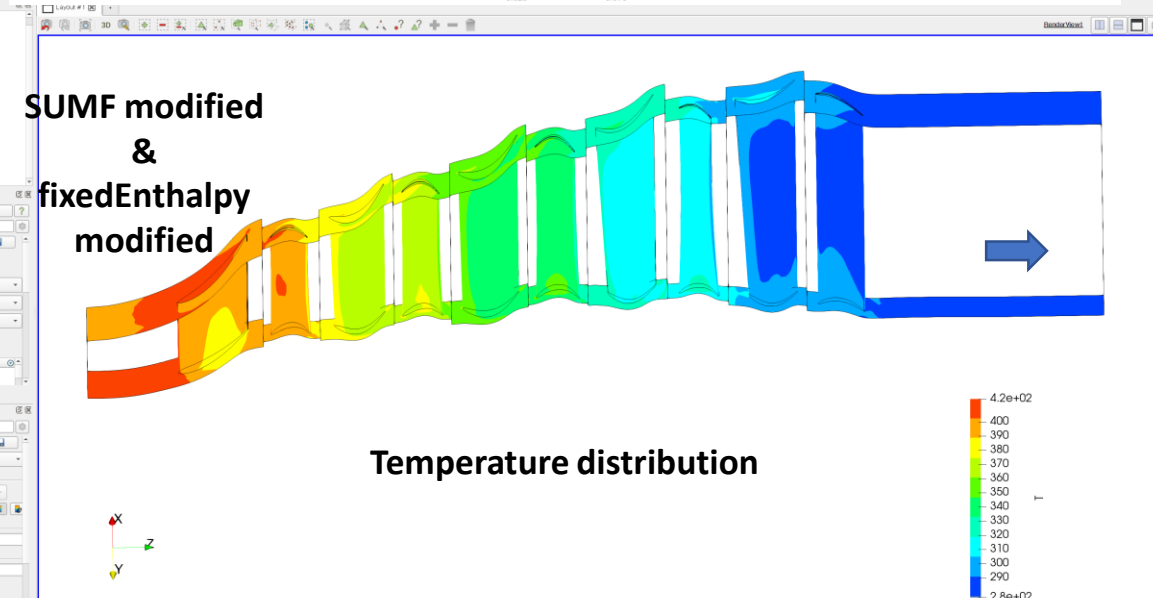
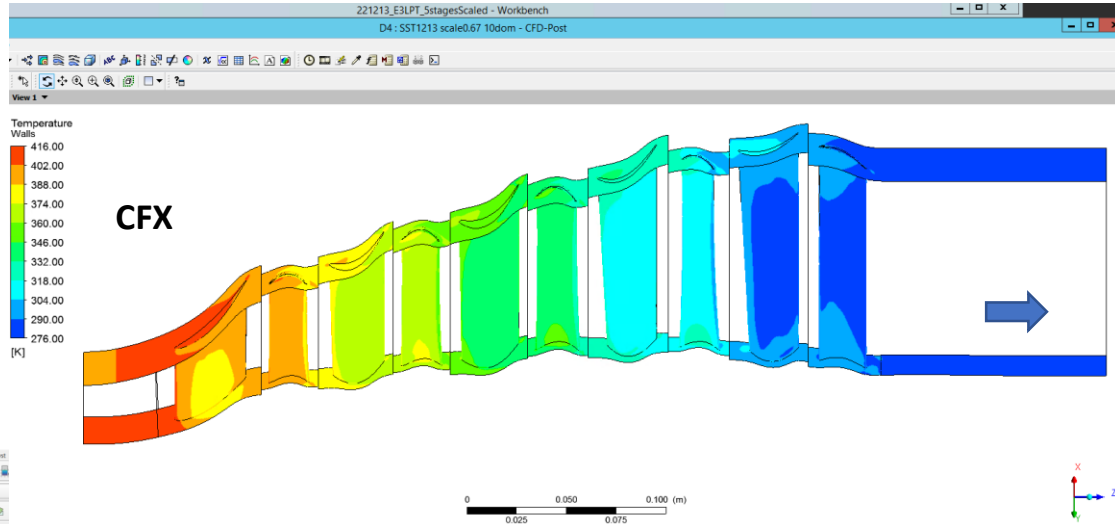
- CFX 유량 = 20.068 kg/s  
 - foam-extend 유량:  
 inlet = 20.042 kg/s, outlet = 19.86 kg/s  
 → error 0.93%. → 터빈 온도 shift up

- *fixedEnthalpy* 수정 후, KARI 터빈 노즐 vane의 온도가 입구온도(1673 K)를 넘지 않고 CFX와 잘 일치.
- 단, 터빈에서 온도 Shift 발생 (유량 오차에서 기인한 것으로 추정)
- E3 LPT의 경우, 온도가 위로 Shift 됐던 것이 *fixedEnthalpy* 수정 이후 회복되어 CFX 결과와 일치.

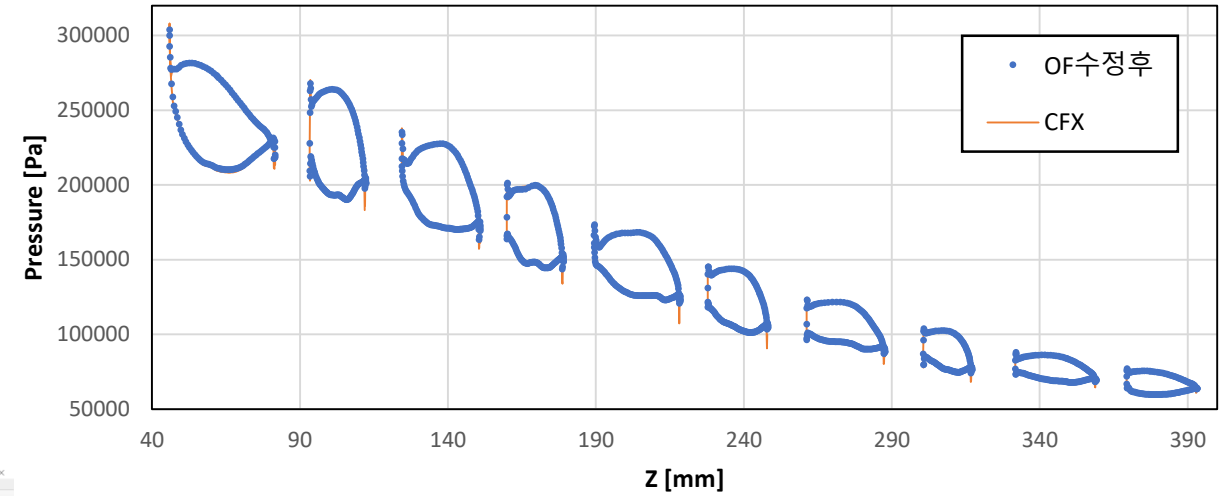


# 3. Wall Temperature Issue

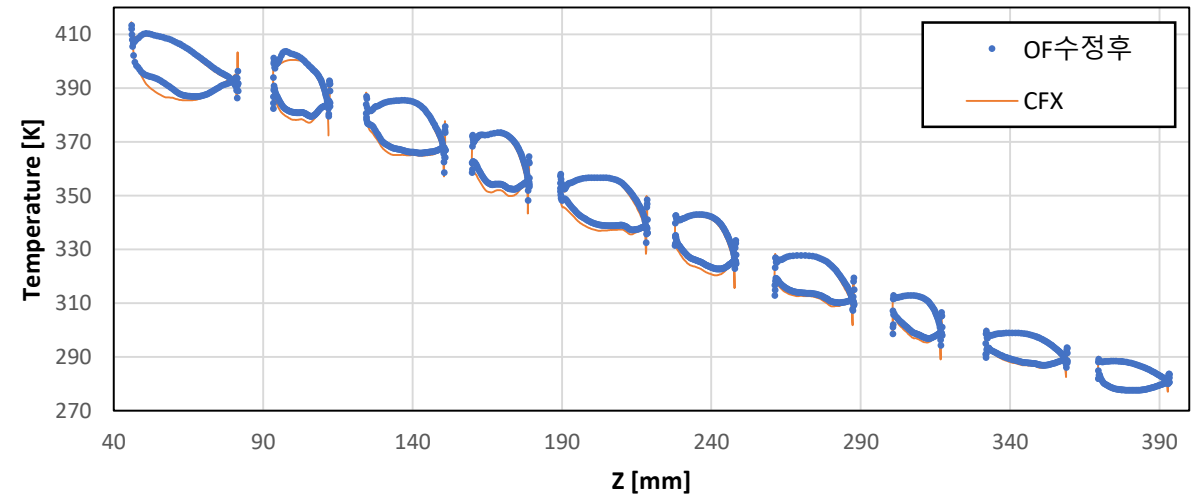
[cf.] E3 LPT 5 stages calculation results



Midspan Pressure

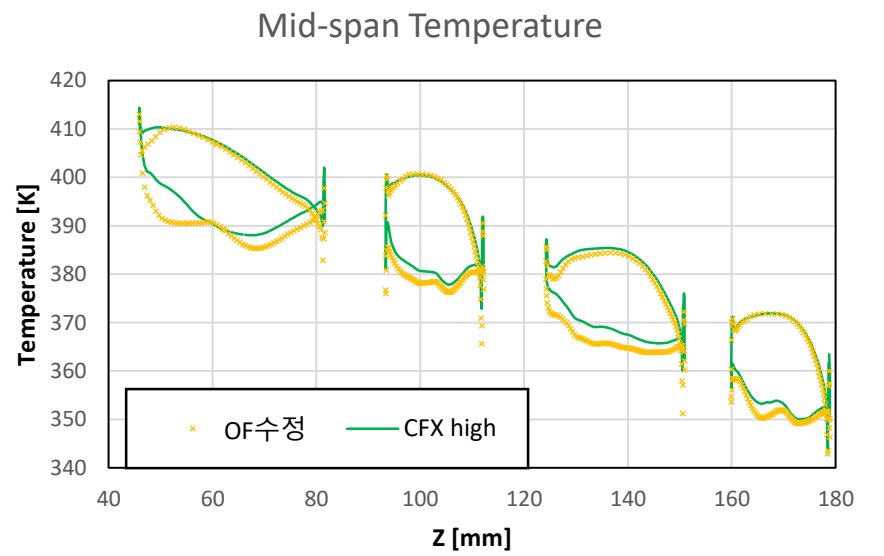
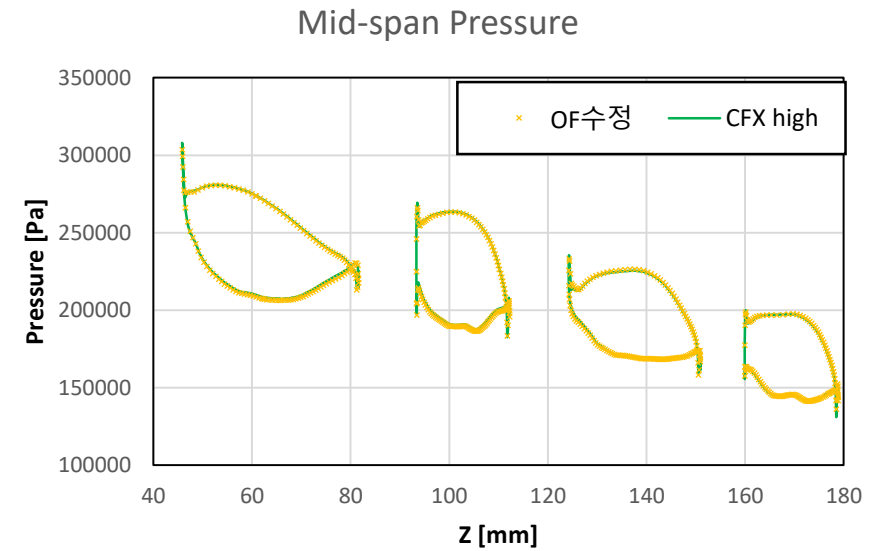
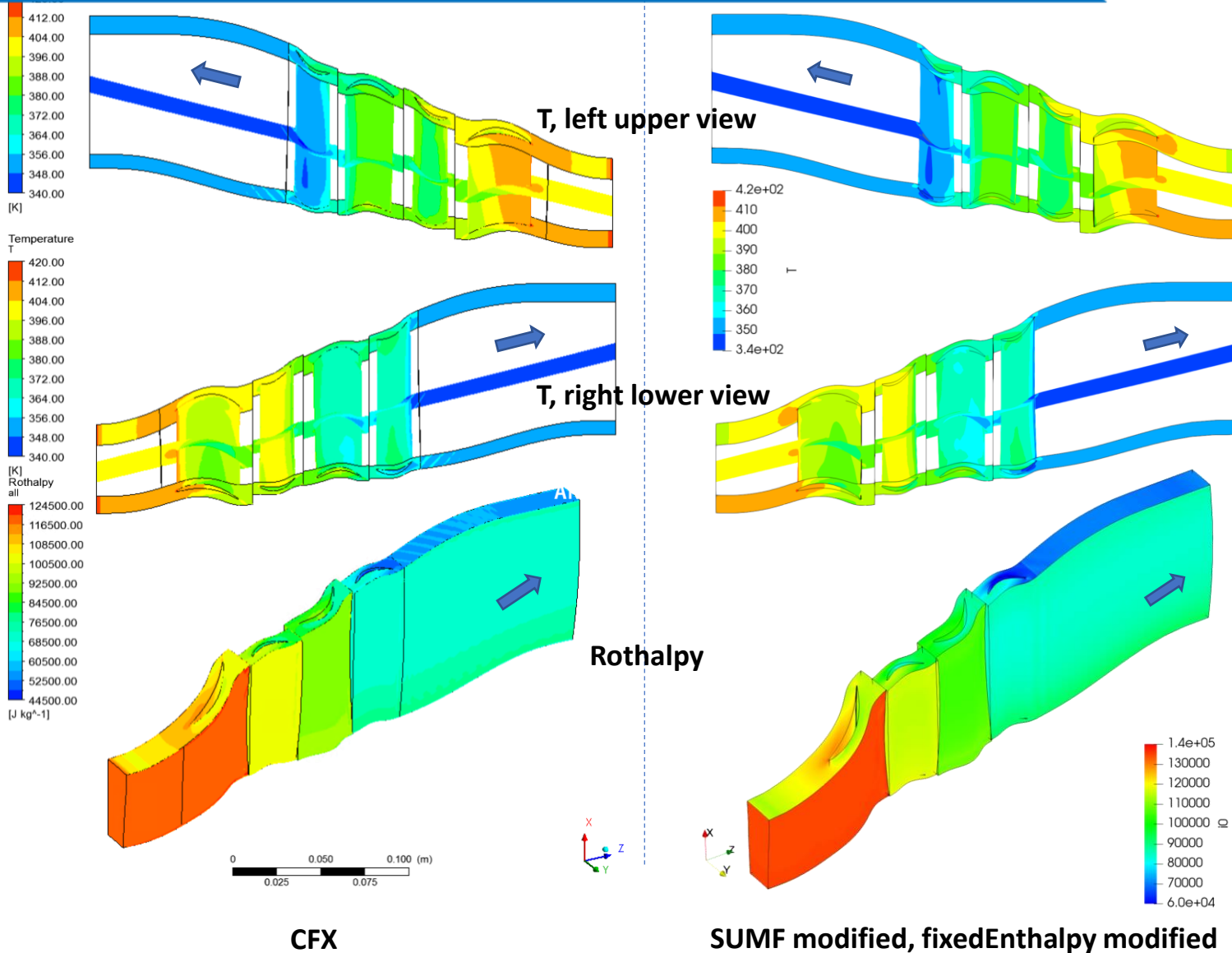


Midspan Temperature



# 3. Wall Temperature Issue

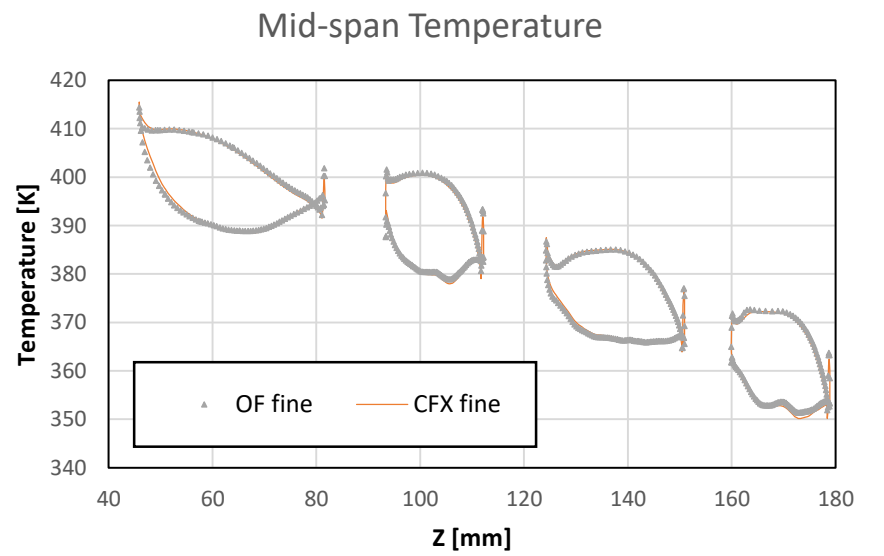
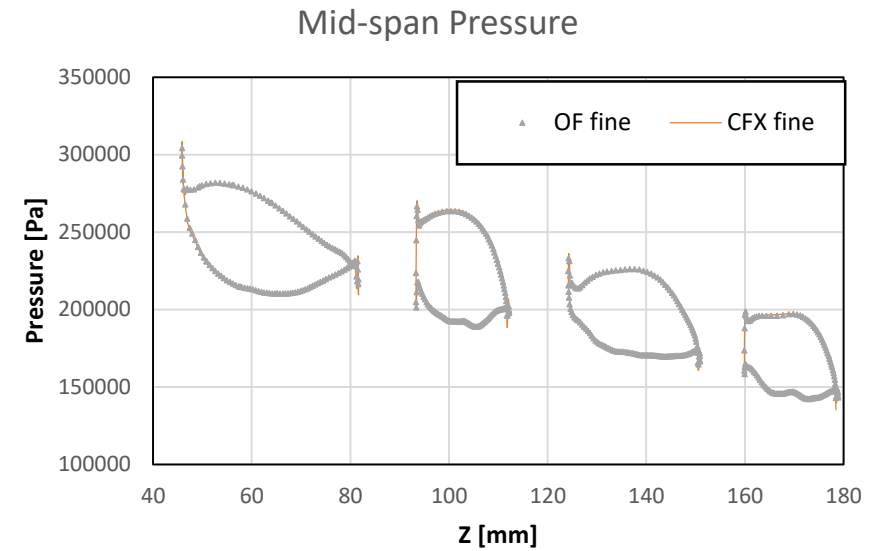
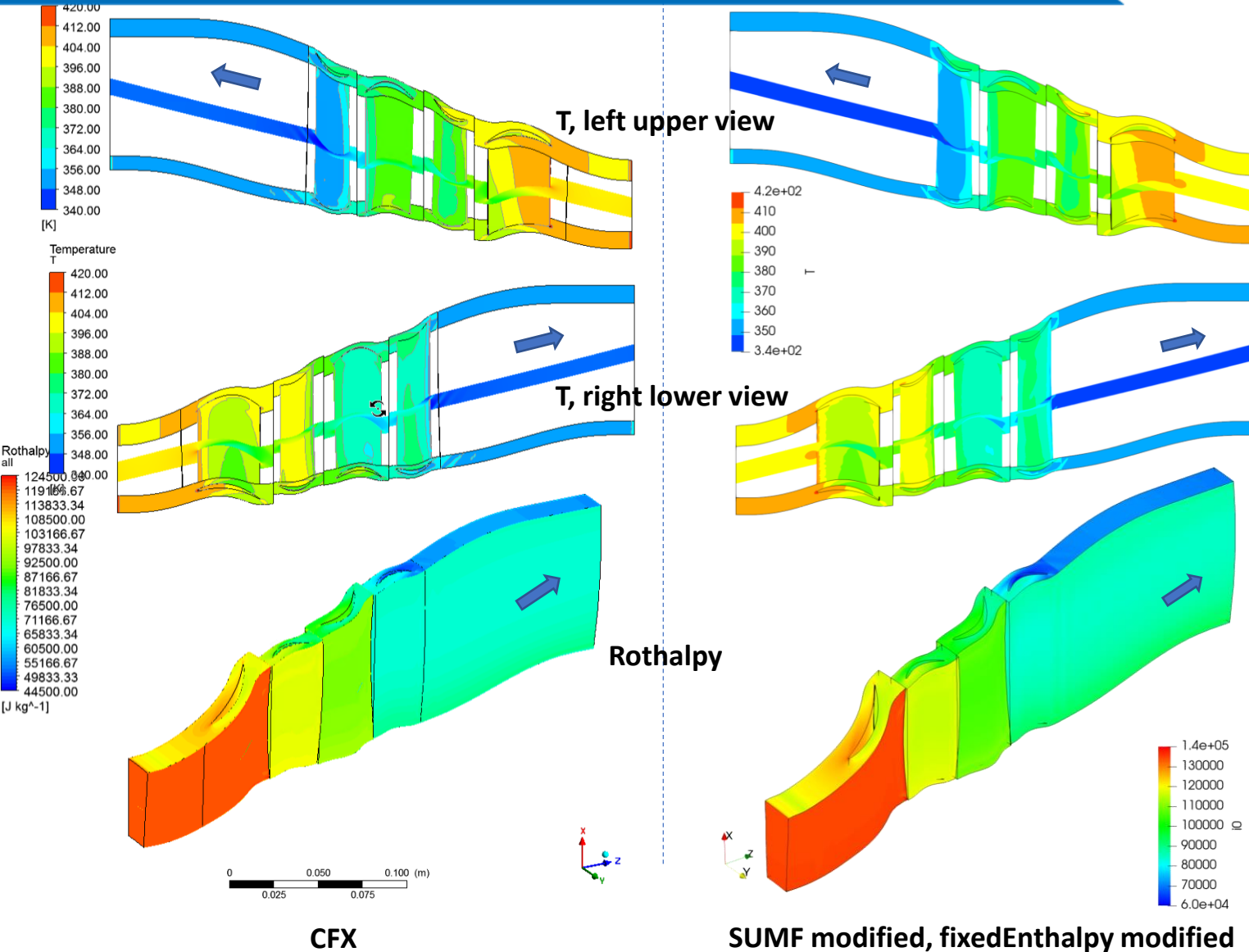
[cf.] E3 LPT 2 stages, calculation results (coarse grid)



→ coarse 격자 해석 시, 일부 오차 존재

# 3. Wall Temperature Issue

[cf.] E3 LPT 2 stages, calculation results (fine grid)



→ Fine 격자 해석 시, 서로 잘 일치.



# 4. The Other Issues

Parallel computation / Turbomachinery SIG

## foam-extend-5.0 on KISTI supercom Nurion

- <https://sourceforge.net/p/foam-extend/foam-extend-5.0/ci/master/tree/ReleaseNotes>
- HPC and parallelism, performance improvements
  - Major improvement in handling of processor boundaries and other cached coupled interface patch types. This resolves long-standing bugs in FOAM-OpenFOAM development lines and results in significant reduction in number of iterations of linear solvers. Further, numerous stability and consistency problems in parallel execution have been resolved
  - Improvements for large memory usage in large-scale HPC cases
  - Incremental consistency work on block-coupled solvers for incompressible flows
  - Incremental improvement in performance for parallel Oveset Mesh capability. Update in low-level communication and consistency
  - Clean-up of boundary condition updates in absence of database field access

→ 5.0 버전에서 병렬 연산 기능이 향상됐다는  
설명

```
- 설치 경로 :  
/apps/applications/foam/foam-extend-5.0  
  
- 설정 방법 예 :  
$ module purge  
$ export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-5.0/etc/bashrc  
$ module load gcc/10.2.0  
  
- 작업 스크립트 예 (2개 노드, 총 4 MPI 프로세스) :  
#!/bin/sh  
#PBS -V  
#PBS -N test  
#PBS -q debug  
#PBS -A openfoam  
#PBS -l select=2:ncpus=68:mpiprocs=2:ompthreads=1  
#PBS -l walltime=04:00:00  
  
cd $PBS_O_WORKDIR  
  
module purge  
export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-5.0/etc/bashrc  
module load gcc/10.2.0  
  
blockMesh  
decomposePar  
  
cat $PBS_NODEFILE > mf  
mpirun -np 4 --hostfile mf simpleFoam -parallel
```

# 4. The Other Issues 1

## Parallel computation test: extend-4.1

```
C:\Users\whahacar\Documents\Wrtg\연구정리2023\material230915\wfe50 core 수 test 0927\test\mpi0926t5mFE41.sh - Notepad++
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 도구(O) 매크로 실행 플러그인 창 관리 ?
E3_34x2c_t4m_Sp26, o13579644 E3_34x2c_t4m_Sp26, o13580259 mpi0926t5m, sh mpi0926t5mFE41, sh
1 #/bin/sh
2 #PBS -V
3 #PBS -N E3_34x2c_t4m_Sp26
4 #PBS -q normal
5 #PBS -A openfoam
6 #PBS -l select=2:ncpus=34:mpiprocs=34:ompthreads=1
7 #PBS -l walltime=00:04:59
8
9 cd $PBS_O_WORKDIR
10
11 module purge
12 module load craype-mic-knl intel/18.0.3 impi/18.0.3
13 #module load craype-x86-skylake intel/18.0.3 impi/18.0.3
14 export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-4.1/etc/bashrc
15
16 #export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-5.0/etc/bashrc
17 #module load gcc/10.2.0
18
19 #### Do not edit ####
20 TOTAL_CPUS=$(wc -l $PBS_NODEFILE | awk '{print $1}')
21 #####
22
23 mpirun -np 68 steadyUniversalMRFFoam230410_02 -parallel | tee log230926_solve2.logfile
24 #mpirun -np 68 steadyCompressibleMRFFoam -parallel | tee log230926_solve1.logfile
25 date
26
```

extend-4.1 Job script

```
C:\Users\whahacar\Documents\Wrtg\연구정리2023\material230915\wfe50 core 수 test 0927\test\E3_34x2c_t4m_Sp26.o13580259 - ...
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 도구(O) 매크로 실행 플러그인 창 관리 ?
E3_34x2c_t4m_Sp26, o13579644 E3_34x2c_t4m_Sp26, o13580259 mpi0926t5m, sh mpi0926t5mFE41, sh
1 /-----\
2 | \ \ \ / F i e l d | foam-extend: Open Source CFD
3 | \ \ \ / O p e r a t i o n | Version: 4.1
4 | \ \ \ / A n d | Web: http://www.foam-extend.org
5 | \ \ \ M a n i p u l a t i o n | For copyright notice see file Copyright
6 |-----\
7
8 Build : 4.1
9 Exec : steadyUniversalMRFFoam230410_02 -parallel
10 Date : Sep 26 2023
11 Time : 18:36:06
12 Host : "node4541"
13 PID : 20310
14 CtrlDict : "/scratch/r811a02/e3_0829_stgl_2_68c_fe50/system/controlDict"
15 Case : /scratch/r811a02/e3_0829_stgl_2_68c_fe50
16 nProcs : 68
17 Slaves :
18 67
19 {
20 "node4541.20311"
21 "node4541.20312"
22 "node4541.20313"
23 "node4541.20314"
24 "node4541.20315"
25 "node4541.20317"
26 "node4541.20318"
27 "node4541.20319"
28 "node4541.20321"
29 "node4541.20323"
30 "node4541.20324"
31 "node4541.20325"
32 "node4541.20326"
33 "node4541.20327"
34 "node4541.20328"
35 "node4541.20330"
36 "node4541.20331"
37 "node4541.20332"
38 "node4541.20333"
39 "node4541.20334"
40 "node4541.20335"
41 "node4541.20336"
42 "node4541.20337"
43 "node4541.20338"
44 "node4541.20339"
45 "node4541.20340"
46 "node4541.20341"
47 "node4541.20342"
48 "node4541.20343"
49 "node4541.20344"
50 "node4541.20345"
51 "node4541.20346"
52 "node4541.20347"
53 "node4541.20348"
54 "node4541.20349"
55 "node4541.20350"
56 "node4541.20351"
57 "node4541.20352"
58 "node4541.20353"
59 "node4541.20354"

```

Node 두 개로 나눠서 계산하도록 Queue를 걸었지만 실제 계산에선 Node 하나만 걸림.  
→ extend-4.1에선 노드 병렬 연산 불가능.

Log

# 4. The Other Issues 1

## Parallel computation test: extend-5.0 (1)

```
C:\Users\whahacar\Documents\Wrtg\연구정리2023\material230915\wfe50 core 수 test 0927\test\mpi0926t5m.sh - Notepad++
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(O) 도구(Q) 매크로 실행 플러그인 창 관리 ?
E3_34x2c_t4m_Sp26, o13579644 E3_34x2c_t4m_Sp26, o13580259 mpi0926t5m.sh mpi0926t5mFE41, sh
1 #/bin/sh
2 #PBS -V
3 #PBS -N E3_34x2c_t4m_Sp26
4 #PBS -q normal
5 #PBS -A openfoam
6 #PBS -l select=2:ncpus=34:mpiprocs=34:ompthreads=1
7 #PBS -l walltime=00:04:59
8
9 cd $PBS_O_WORKDIR
10
11 module purge
12 #module load craype-mic-knl intel/18.0.3 impi/18.0.3
13 #module load craype-x86-skylake intel/18.0.3 impi/18.0.3
14 #export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-4.1/etc/bashrc
15
16 export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-5.0/etc/bashrc
17 module load gcc/10.2.0
18
19 #### Do not edit ####
20 TOTAL_CPUS=$(wc -l $PBS_NODEFILE | awk '{print $1}')
21 #####
22
23 #mpirun -np 68 steadyUniversalMRFFoam230410_02 -parallel | tee log230901_solve.logfile
24 mpirun -np 68 steadyCompressibleMRFFoam -parallel | tee log230926_solve1.logfile
25 date
26
```

extend-5.0 Job script

```
C:\Users\whahacar\Documents\Wrtg\연구정리2023\material230915\wfe50 core 수 test 0927\test\E3_34x2c_t4m_Sp26.o13579644 - ...
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(O) 도구(Q) 매크로 실행 플러그인 창 관리 ?
E3_34x2c_t4m_Sp26, o13579644 E3_34x2c_t4m_Sp26, o13580259 mpi0926t5m.sh mpi0926t5mFE41, sh
1 /-----\
2 |         |
3 | \ \ \ /  | F i e l d           | foam-extend: Open Source CFD
4 | \ \ \ /  | O p e r a t i o n     | Version:      5.0
5 | \ \ \ /  | A n d                 | Web:          http://www.foam-extend.org
6 | \ \ \ /  | M a n i p u l a t i o n | For copyright notice see file Copyright
7 |         |
8 Build : 5.0-287705b4a589
9 Exec  : steadyCompressibleMRFFoam -parallel
10 Date  : Sep 26 2023
11 Time  : 17:44:35
12 Host  : "node5712"
13 PID   : 55604
14 CtrlDict : "/scratch/r811a02/e3_0829_stgl_2_68c_fe50/system/controlDict"
15 Case   : /scratch/r811a02/e3_0829_stgl_2_68c_fe50
16 nProcs : 68
17 Slaves :
18 67
19 {
20 "node5712.55603"
21 "node5712.55606"
22 "node5712.55605"
23 "node5712.55607"
24 "node5712.55608"
25 "node5712.55610"
26 "node5712.55609"
27 "node5712.55611"
28 "node5712.55612"
29 "node5712.55614"
30 "node5712.55613"
31 "node5712.55615"
32 "node5712.55616"
33 "node5712.55618"
34 "node5712.55617"
35 "node5712.55619"
36 "node5712.55620"
37 "node5712.55622"
38 "node5712.55621"
39 "node5712.55623"
40 "node5712.55624"
41 "node5712.55626"
42 "node5712.55625"
43 "node5712.55627"
44 "node5712.55628"
45 "node5712.55630"
46 "node5712.55629"
47 "node5712.55632"
48 "node5712.55631"
49 "node5712.55634"
50 "node5712.55633"
51 "node5712.55636"
52 "node5712.55635"
53 "node6287.19220"
54 "node6287.19221"
55 "node6287.19223"
56 "node6287.19222"
57 "node6287.19224"
58 "node6287.19225"
59 "node6287.19226"
}
Normal text file length : 62,407 lines : 883 Ln : 12 Col : 19 Sel : 8 | 1 Unix (LF) UTF-8 INS
```

Extend-5.0에선 Node 두 개로 나뉘서 계산이 걸림. → 노드 병렬 연산 가능.

Log

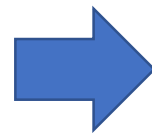
# 4. The Other Issues 1

## Parallel computation test: extend-5.0 (2)

```
C:\Users\whahacar\Documents\Wrtg\연구정리2023\material230915\fe50 core 수 test 0927\4x68c\mpi0926t2h.sh - Notepad++
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 도구(O) 매크로 실행 플러그인 창 관리 ?
mpi0926t2h.sh log230926_solve logfile
1 #/bin/sh
2 #PBS -V
3 #PBS -N E3_68x4c_t2h_Sp26
4 #PBS -q normal
5 #PBS -A openfoam
6 #PBS -l select=4:ncpus=68:mpiprocs=68:ompthreads=1
7 #PBS -l walltime=01:59:59
8
9 cd $PBS_O_WORKDIR
10
11 module purge
12 #module load craype-mic-knl intel/18.0.3 impi/18.0.3
13 #module load craype-x86-skylake intel/18.0.3 impi/18.0.3
14 #export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-4.1/etc/bashrc
15
16 export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-5.0/etc/bashrc
17 module load gcc/10.2.0
18
19 #### Do not edit ####
20 TOTAL_CPUS=$(wc -l $PBS_NODEFILE | awk '{print $1}')
21 #####
22
23 #mpirun -np 68 steadyUniversalMRFFoam230410_02 -parallel | tee log230901_solve.logfile
24 mpirun -np 272 steadyCompressibleMRFFoam -parallel | tee log230926_solve.logfile
25 date
26
```

extend-5.0 Job script

- 4 Nodes \* 68 cores = 272 cores로 계산이 잘 돌아가는 모습



```
C:\Users\whahacar\Documents\Wrtg\연구정리2023\material230915\fe50 core 수 test 0927\4x68c\log230926_solve.logfile - Notepad++
파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 도구(O) 매크로 실행 플러그인 창 관리 ?
mpi0926t2h.sh log230926_solve logfile
1 /-----\
2 | \ \ \ / Field | foam-extend: Open Source CFD
3 | \ \ \ / O peration | Version: 5.0
4 | \ \ \ / A nd | Web: http://www.foam-extend.org
5 | \ \ \ / M anipulation | For copyright notice see file Copyright
6 | \-----\
7
8 Build : 5.0-287705b4a589
9 Exec : steadyCompressibleMRFFoam -parallel
10 Date : Sep 26 2023
11 Time : 23:51:42
12 Host : "node4705"
13 PID : 17959
14 CtrlDict : "/scratch/r811a02/e3_0926_stgl_2_272c_fe50/system/controlDict"
15 Case : /scratch/r811a02/e3_0926_stgl_2_272c_fe50
16 nProcs : 272
17 Slaves :
18 271
19 {
20 "node4705.17960"
21 "node4705.17961"
22 "node4705.18023"
23 "node4705.18024"
24 "node4705.18025"
25 "node4705.18026"
26 "node4705.18027"
27 "node4705.18028"
28 "node4705.18029"
29 "node4705.18030"
30 "node4705.18031"
31 "node4705.18032"
32 "node4705.18033"
33 "node4705.18034"
34 "node4705.18035"
35 "node4705.18036"
36 "node4705.18037"
37 "node4705.18038"
38 "node4705.18039"
39 "node4705.18040"
40 "node4705.18041"
41 "node4705.18042"
42 "node4705.18043"
43 "node4705.18044"
44 "node4705.18045"
45 "node4705.18046"
46 "node4705.18047"
47 "node4705.18048"
48 "node4705.18049"
49 "node4705.18050"
50 "node4705.18051"
51 "node4705.18052"
52 "node4705.18053"
53 "node4705.18054"
54 "node4705.18055"
55 "node4705.18056"
56 "node4705.18057"
57 "node4705.18058"
58 "node4705.18059"
59 "node4705.18060"
60 "node4705.18061"
61 "node4705.18062"
62 "node4705.18063"
63 "node4705.18064"
64 "node4705.18065"
65 "node4705.18066"
66 "node4705.18067"
67 "node4705.18068"
68 "node4705.18069"
69 "node4705.18070"
70 "node4705.18071"
71 "node4705.18072"
72 "node4705.18073"
73 "node4705.18074"
74 "node4705.18075"
75 "node4705.18076"
76 "node4705.18077"
77 "node4705.18078"
78 "node4705.18079"
79 "node4705.18080"
80 "node4705.18081"
81 "node4705.18082"
82 "node4705.18083"
83 "node4705.18084"
84 "node4705.18085"
85 "node4705.18086"
86 "node4705.18087"
87 "node4705.18088"
88 "node4705.18089"
89 "node4705.18090"
90 "node4705.18091"
91 "node4705.18092"
92 "node4705.18093"
93 "node4705.18094"
94 "node4705.18095"
95 "node4705.18096"
96 "node4705.18097"
97 "node4705.18098"
98 "node4705.18099"
99 "node4705.18100"
100 "node4705.18101"
101 "node4705.18102"
102 "node4705.18103"
103 "node4705.18104"
104 "node4705.18105"
105 "node4705.18106"
106 "node4705.18107"
107 "node4705.18108"
108 "node4705.18109"
109 "node4705.18110"
110 "node4705.18111"
111 "node4705.18112"
112 "node4705.18113"
113 "node4705.18114"
114 "node4705.18115"
115 "node4705.18116"
116 "node4705.18117"
117 "node4705.18118"
118 "node4705.18119"
119 "node4705.18120"
120 "node4705.18121"
121 "node4705.18122"
122 "node4705.18123"
123 "node4705.18124"
124 "node4705.18125"
125 "node4705.18126"
126 "node4705.18127"
127 "node4705.18128"
128 "node4705.18129"
129 "node4705.18130"
130 "node4705.18131"
131 "node4705.18132"
132 "node4705.18133"
133 "node4705.18134"
134 "node4705.18135"
135 "node4705.18136"
136 "node4705.18137"
137 "node4705.18138"
138 "node4705.18139"
139 "node4705.18140"
140 "node4705.18141"
141 "node4705.18142"
142 "node4705.18143"
143 "node4705.18144"
144 "node4705.18145"
145 "node4705.18146"
146 "node4705.18147"
147 "node4705.18148"
148 "node4705.18149"
149 "node4705.18150"
150 "node4705.18151"
151 "node4705.18152"
152 "node4705.18153"
153 "node4705.18154"
154 "node4705.18155"
155 "node4705.18156"
156 "node4705.18157"
157 "node4705.18158"
158 "node4705.18159"
159 "node4705.18160"
160 "node4705.18161"
161 "node4705.18162"
162 "node4705.18163"
163 "node4705.18164"
164 "node4705.18165"
165 "node4705.18166"
166 "node4705.18167"
167 "node4705.18168"
168 "node4705.18169"
169 "node4705.18170"
170 "node4705.18171"
171 "node4705.18172"
172 "node4705.18173"
173 "node4705.18174"
174 "node4705.18175"
175 "node4705.18176"
176 "node4705.18177"
177 "node4705.18178"
178 "node4705.18179"
179 "node4705.18180"
180 "node4705.18181"
181 "node4705.18182"
182 "node4705.18183"
183 "node4705.18184"
184 "node4705.18185"
185 "node4705.18186"
186 "node4705.18187"
187 "node4705.18188"
188 "node4705.18189"
189 "node4705.18190"
190 "node4705.18191"
191 "node4705.18192"
192 "node4705.18193"
193 "node4705.18194"
194 "node4705.18195"
195 "node4705.18196"
196 "node4705.18197"
197 "node4705.18198"
198 "node4705.18199"
199 "node4705.18200"
200 "node4705.18201"
201 "node4705.18202"
202 "node4705.18203"
203 "node4705.18204"
204 "node4705.18205"
205 "node4705.18206"
206 "node4705.18207"
207 "node4705.18208"
208 "node4705.18209"
209 "node4705.18210"
210 "node4705.18211"
211 "node4705.18212"
212 "node4705.18213"
213 "node4705.18214"
214 "node4705.18215"
215 "node4705.18216"
216 "node4705.18217"
217 "node4705.18218"
218 "node4705.18219"
219 "node4705.18220"
220 "node4705.18221"
221 "node4705.18222"
222 "node4705.18223"
223 "node4705.18224"
224 "node4705.18225"
225 "node4705.18226"
226 "node4705.18227"
227 "node4705.18228"
228 "node4705.18229"
229 "node4705.18230"
230 "node4705.18231"
231 "node4705.18232"
232 "node4705.18233"
233 "node4705.18234"
234 "node4705.18235"
235 "node4705.18236"
236 "node4705.18237"
237 "node4705.18238"
238 "node4705.18239"
239 "node4705.18240"
240 "node4705.18241"
241 "node4705.18242"
242 "node4705.18243"
243 "node4705.18244"
244 "node4705.18245"
245 "node4705.18246"
246 "node4705.18247"
247 "node4705.18248"
248 "node4705.18249"
249 "node4705.18250"
250 "node4705.18251"
251 "node4705.18252"
252 "node4705.18253"
253 "node4705.18254"
254 "node4705.18255"
255 "node4705.18256"
256 "node4705.18257"
257 "node4705.18258"
258 "node4705.18259"
259 "node4705.18260"
260 "node4705.18261"
261 "node4705.18262"
262 "node4705.18263"
263 "node4705.18264"
264 "node4705.18265"
265 "node4705.18266"
266 "node4705.18267"
267 "node4705.18268"
268 "node4705.18269"
269 "node4705.18270"
270 "node4705.18271"
271 "node4705.18272"
272 "node4705.18273"
273 "node4705.18274"
274 "node4705.18275"
275 "node4705.18276"
276 "node4705.18277"
277 "node4705.18278"
278 "node4705.18279"
279 "node4705.18280"
280 "node4705.18281"
281 "node4705.18282"
282 "node4705.18283"
283 "node4705.18284"
284 "node4705.18285"
285 "node4705.18286"
286 "node4705.18287"
287 "node4705.18288"
288 "node4705.18289"
289 "node4705.18290"
290 "node4705.18291"
291 "node4705.18292"
292 "node4705.18293"
293 "node4705.18294"
294 "node4705.18295"
295 "node4705.18296"
296 "node4705.18297"
297 "node4705.18298"
298 "node4705.18299"
299 "node4705.18300"
300 // *****
301 Create time
302
303 Create mesh for time = 0
304
305 Initializing the GGI interpolator between master/shadow patches: R1-TO-R1-PERIODIC-1-SIDE-1/R1-TO-R1-PERIODIC-1-SIDE-2
306 Initializing the GGI interpolator between master/shadow patches: S1-TO-S1-PERIODIC-1-SIDE-1/S1-TO-S1-PERIODIC-1-SIDE-2
307 Initializing the GGI interpolator between master/shadow patches: S1-TO-S1-PERIODIC-2-SIDE-1/S1-TO-S1-PERIODIC-2-SIDE-2
308 Initializing the mixingPlane interpolator between master/shadow patches: R1-TO-S1-SIDE-1/R1-TO-S1-SIDE-2
309
310 FIMPLE: Operating solver in PISO mode
311
312 Reading thermophysical properties
313
314 Selecting thermodynamics package hPsiThermo<pureMixture<outerlandTransport<specieThermo<janafThermo<perfectGas>>>>
315
316 Reading field U
317
318 Reading/calculating face flux field phi
319
320 Creating turbulence model
321
322 Selecting RAS turbulence model kEpsilon
323 kEpsilonCoeffs
324 {
325 Cmu 0.09;
326 C1 1.44;
327 C2 1.92;
328 C3 -0.33;
329 sigmaK 1;
330 sigmaEps 1.3;
331 Prt 1;
332 }
333
334 Creating MRF for cell zone R1. rpm = -3208.7002
335
336 Starting time loop
337
338 Creating MachNumber for field U
339 Creating minMaxField for field U
340 Creating minMaxField for field p
341 Creating minMaxField for field rho
342 Creating minMaxField for field T
343 Time = 1
344
345 smoothSolver: Solving for Ux, Initial residual = 1, Final residual = 0.08509255, No Iterations 2
346 smoothSolver: Solving for Uy, Initial residual = 1, Final residual = 0.03209208, No Iterations 2
347 smoothSolver: Solving for Uz, Initial residual = 1, Final residual = 0.032486419, No Iterations 2
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
Normal text file length: 4,342,213 lines: 54,462 Ln: 155 Col: 10 Sel: 8 | 1 Unix (LF) UTF-8 INS
```

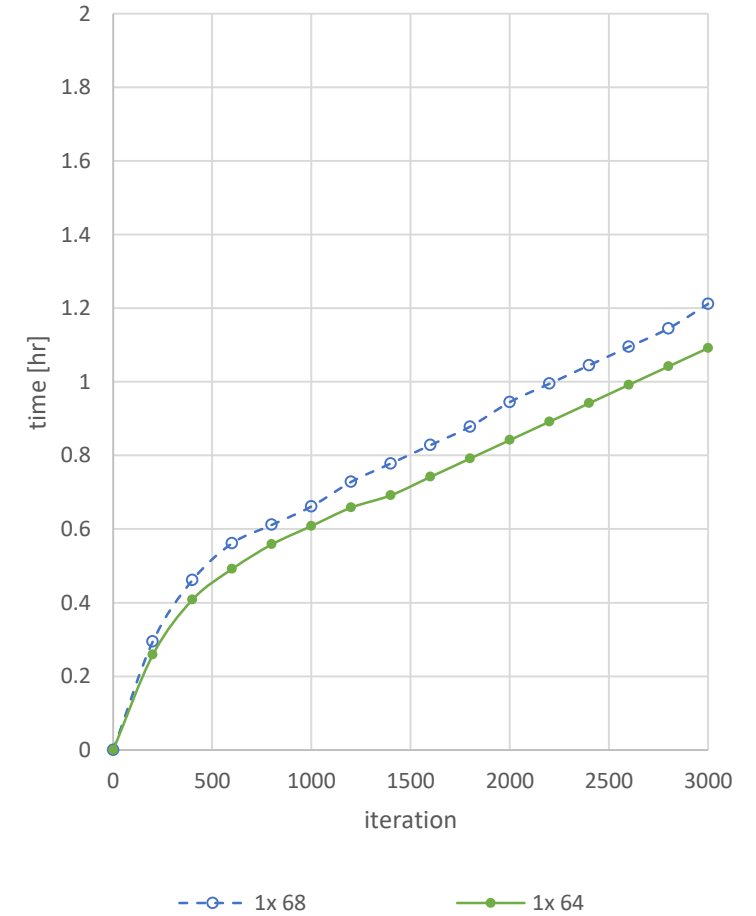
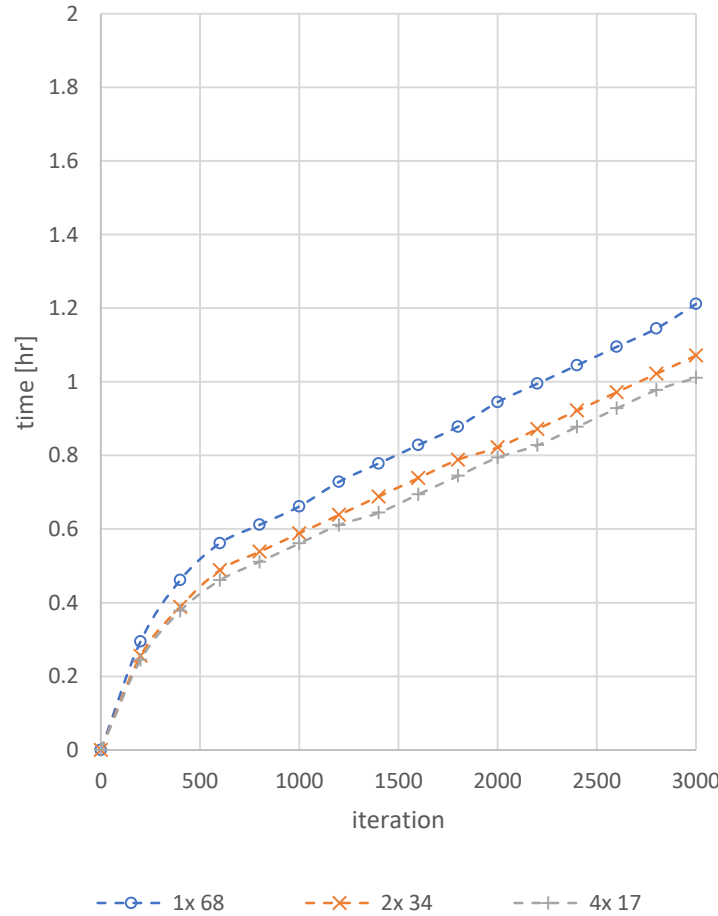
Log

# 4. The Other Issues 1

## Node parallel computation test (1)

- E3 LPT 1단. 333,393 개의 격자.
- 노드/코어 수 조합으로 계산, 계산 시간 정리.
- 총 core 수가 같을 때 node를 나눌 수록 계산 시간 빨라짐.
- 또한 68 core보다 64core를 썼을 때 계산 시간이 짧았음.
- 하지만 본 모델에선, 한 node에 같은 core 수를 설정하고 node 수를 늘릴 수록 계산 시간이 늘어남.

→ 병렬최적화 어려움.



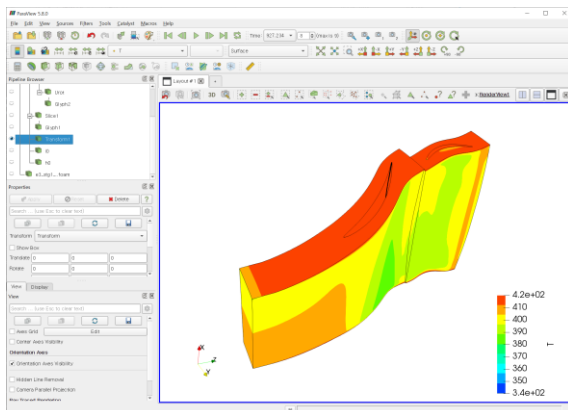
Computation time

# 4. The Other Issues 1

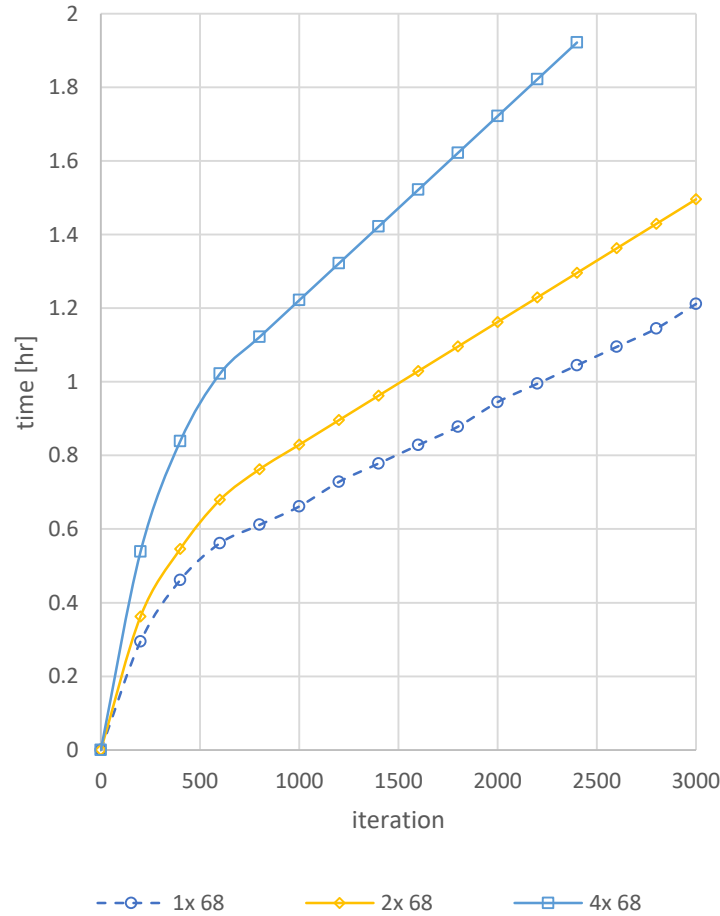
## Node parallel computation test (2)

- E3 LPT 1단. 333,393 개의 격자.
- 노드/코어 수 조합으로 계산, 계산 시간 정리.
- 총 core 수가 같을 때 node를 나눌 수록 계산 시간 빨라짐.
- 또한 68 core보다 64core를 썼을 때 계산 시간이 짧았음.
- 하지만 본 모델에선, 한 node에 같은 core 수를 설정하고 node 수를 늘릴 수록 계산 시간이 늘어남.

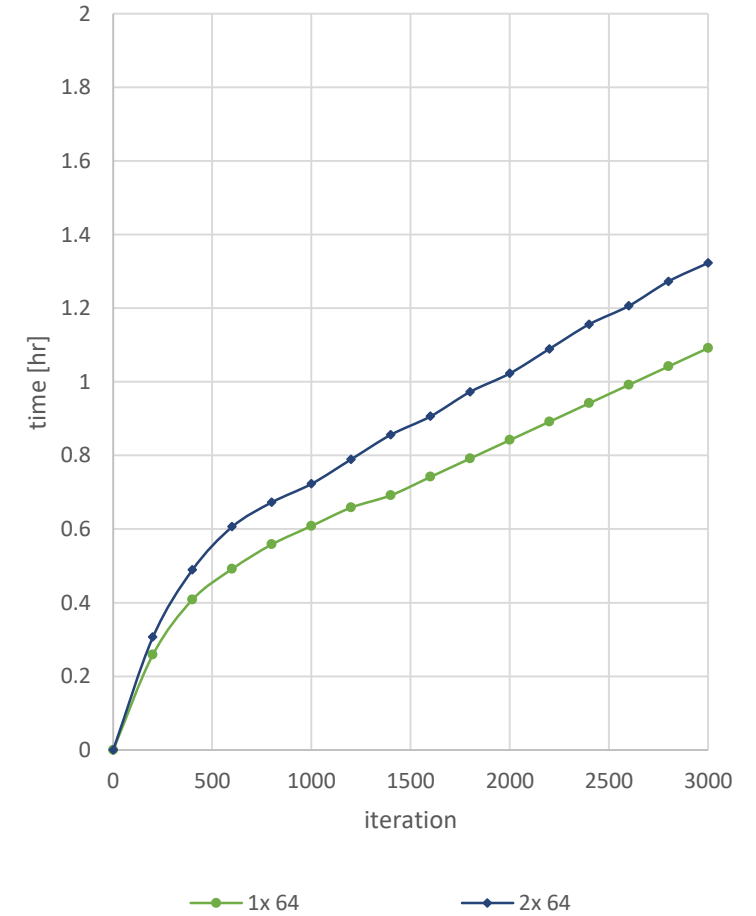
→ 병렬최적화 어려움.



extend-5.0, steadyCompressibleMRFFoam



Computation time



# 4. The Other Issues 2

## 18th OpenFOAM Workshop – Turbo SIG (1)

- **Special Interest Group:** 관심 있는 주제에 대해 자유롭게 그룹을 구성하고 모여서 토론하는 세션 (HVAC, 연소, 등)
- Turbomachinery SIG 참여인원: On/Off 라인 합쳐서 10명 이상 참여
- Turbomachinery 소개:  
[https://wiki.openfoam.com/Turbomachinery\\_Special\\_Interest\\_Group](https://wiki.openfoam.com/Turbomachinery_Special_Interest_Group)

**Turbomachinery Special Interest Group**

The Turbomachinery Special Interest Group advances the modelling capabilities of OpenFOAM for Turbomachinery applications.

**Contents (hide)**

- 1 Scope
- 2 Work process
- 3 Meetings
- 4 Deliverables
- 5 Members
- 6 List of meetings
- 7 List of prioritized developments

**Scope**

- Both incompressible and compressible turbomachines are of interest.
- Open developments in the OpenFOAM framework are promoted, not limited to a particular version of OpenFOAM.
- A first goal is to sum up the present capabilities and differences between different versions (and in applications and libraries that do not yet belong to any version), and to identify the need of further developments and alignments.
- A second step is to put in place a test harness with cases that specifically test turbomachinery functionality, aiming at minimizing problems at version upgrades.
- A third step is to identify the most important developments that need to be done, and to form groups that can realize them. Finally, a continuous task for this committee will be to document the turbomachinery-related functionalities that are available.

**Work process**

Not being limited to a single version of OpenFOAM, a project has been set up at SourceForge, at <https://sourceforge.net/projects/turbowg/>. It was first populated with updated versions of the work done by the Turbomachinery Working Group that started in 2007 and was very active for a number of years. Those test-cases were originally set up for foam-extend, but will now as far as possible also be set up for the most recent ESI OpenFOAM version. Some additional stand-alone applications and libraries are also gathered with the aim to eventually integrate them into the main releases. A summary of the status of the test cases, tutorials, applications, libraries and test harness is made available through a Wiki, at <https://sourceforge.net/p/turbowg/wiki/Home/>.

**Meetings**

To stimulate the group members to stay active, we have one hybrid (on-line/IRL) meeting at each of the annual OpenFOAM workshop and OpenFOAM conference, and two intermediate on-line meetings between those occasions. At the workshops there will as well be an open Turbomachinery SIG meeting for anyone that is interested. A list of meetings is supplied at the bottom of the page.

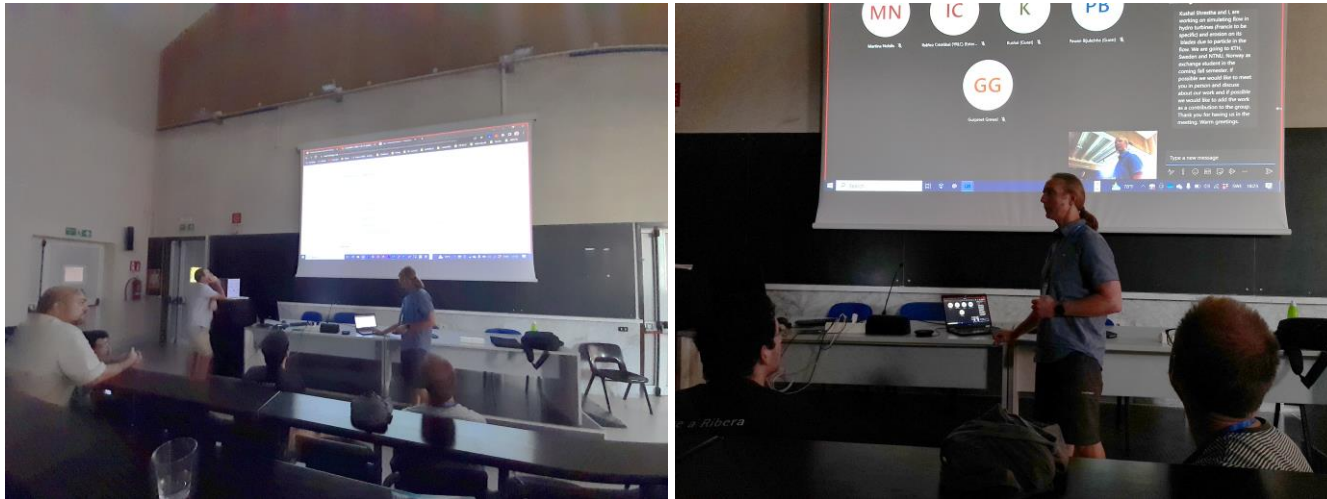
**Deliverables**

The committee progress will be reported once a year in conjunction with the OpenFOAM conference, occasionally as a presentation, but at minimum as a report to the OpenFOAM Governance Steering Committee. The developments will continuously get delivered through the SourceForge project. Once a development has reached a matured and "polished" status, it will be suggested to be merged into the main release(s) and discontinued in the SourceForge project.

**Members**

Here are the present members, led by Håkan Nilsson who acts as chairperson.

|               |                 |      |              |               |              |               |               |             |
|---------------|-----------------|------|--------------|---------------|--------------|---------------|---------------|-------------|
|               |                 |      |              |               |              |               |               |             |
| Håkan Nilsson | Martin Beaudoin | Fred | Saeed Salehi | Jiri Polansky | Eric Lilberg | Remo De Donno | Greg Burgreen | Tessa Uroic |



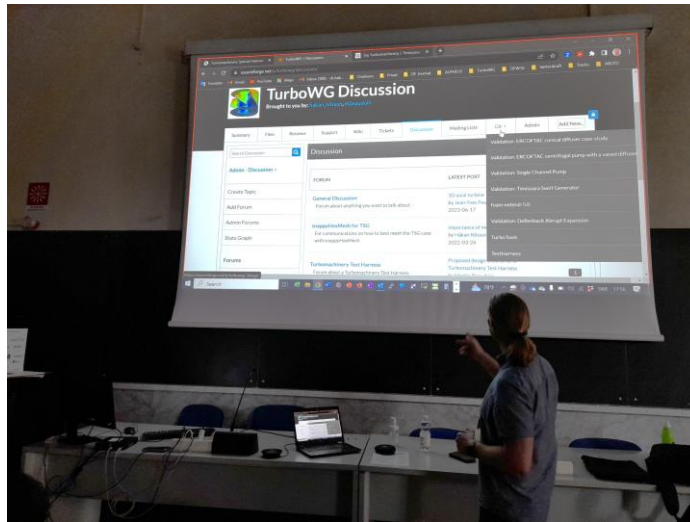
Turbomachinery SIG meeting scene



# 4. The Other Issues 2

## 18th OpenFOAM Workshop – Turbo SIG (2)

- Turbomachinery Working Group 자료실: <https://sourceforge.net/projects/turbowg/>



Turbomachinery working group archive



| List of updated TurboWG applications, libraries and test cases |                 |                |  |
|--|-----------------|----------------|--|
| Authors: 👤 👤   |                 |                |  |
| <b>Test cases</b>  | foam-extend-5.0 | OpenFOAM v2112 | Notes  |
| ERCOFTAC Conical Diffuser                                      | Experimental    | ToDo           |  |
| ERCOFTAC Centrifugal Pump                                      | Experimental    | ToDo           |  |
| Single Channel Pump  | Experimental    | ToDo           |  |
| Timisoara Swirl Generator                                      | Experimental    | ToDo           |  |
| Francis-99   | ToDo            | ToDo           |  |
| Dellenback Abrupt Expansion                                    | ToDo            | ToDo           |  |
| <b>Tutorials</b>   | foam-extend-5.0 | OpenFOAM v2112 | Notes  |
| axialTurbine (SRF, MRF, DyM)                                   | Experimental    | ToDo           |  |
| <b>Applications</b>  | foam-extend-5.0 | OpenFOAM v2206 | Notes  |
| addSwirlAndRotation  | Experimental    | Experimental   |  |
| simpleTurboMFRFoam   | Experimental    | ToDo           |  |
| CGNS Converters  | Experimental    | Experimental   | Updated to cgns-4.3.0  |
| turboPassageCopy   | ToDo            | ToDo           | Read a single passage mesh and output a mesh of two to all passages. |
| createMixingPlaneRibbons                                       | ToDo            | ToDo           | Based on interface mesh distributions and shape.                     |
| <b>Libraries</b>   | foam-extend-5.0 | OpenFOAM v2206 | Notes  |
| profile1DFixedValue  | Experimental    | Experimental   |  |
| turboPerformance   | Experimental    | ToDo           | See propellerInfo in OFv2112   |

- 터보머신 기능은 주로 OpenFOAM extend-5.0에 구현됨.
- 따라서 현재 터보머신 기능 및 예제를 ESI 버전에 심으려는 작업 진행 중.
- 이 과정에 관심있는 User는 테스트에 참여 가능.

# **5. Summary & Conclusion**

## Summary

- 가스터빈 터보유체기계의 고정밀도 해석을 위해 오픈 소스 CFD 코드 중 하나인 OpenFOAM을 사용한 대규모 병렬 계산이 요구됨.
- 필요한 Interface 유틸리티가 사용 가능한 압축성 MRF 솔버로는 → foam-extend 버전의 steadyUniversalMRFFoam
- *mixingPlane* 온도 불연속 이슈는 정의 안 된 변수와 Rothalpy jump의 스칼라 곱으로 인해 발생.
- foam-extend 버전의 압축성 솔버의 에너지 방정식에선 운동에너지 항이 결여됨. → 잘못된 온도장 산출.
- 병렬 연산 이슈 소개 / Turbomachinery SIG 소개
- (Thermo-physical properties 공유)

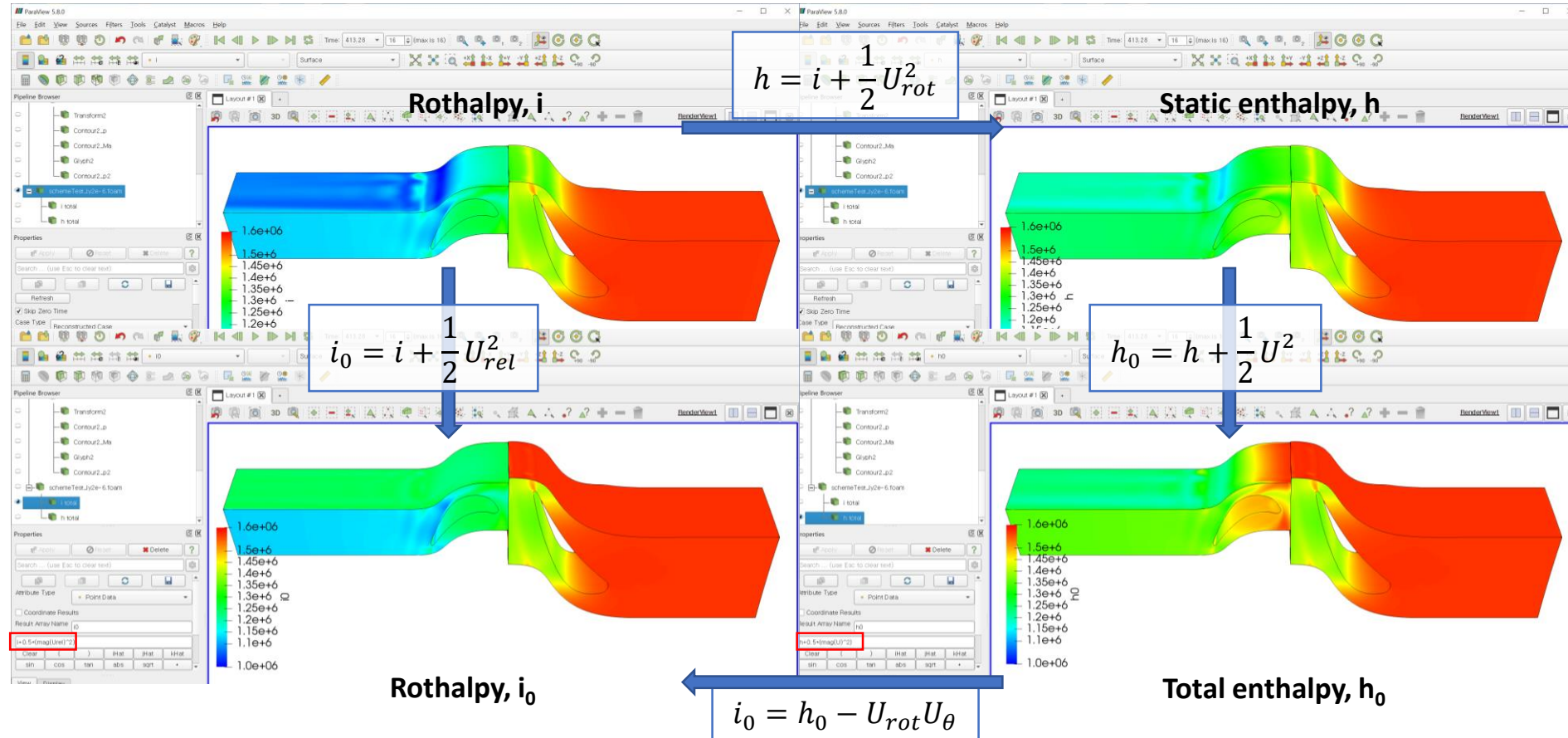
## Conclusion (1)

- *mixingPlane*과 같은 Interface 유틸리티에서 **내적**을 이용하여 **Rothalpy jump** 계산 시 연속적인 온도 결과 산출.
- steadyUniversalMRFFoam 솔버에서 벽 온도 상승 문제 교정:
  - 운동에너지 항을 구분하여 **Rothalpy**를 재정의하고, 운동에너지에 대한 divergence 항을 포함하여 Enthalpy 형태의 에너지 방정식을 모사하여 Rothalpy 식을 수정할 경우 올바른 벽 온도 산출.

# 4. Summary & Conclusion

## Conclusion (2)

- 새로 정의된 Rothalpy에 따른  $i, i_0, h, h_0$ 의 관계



- 향후 연구: *mixingPlane* 유량 오차, Viscous work term

**감사합니다.**

강승환KANG, Sueng-Hwan ([hahacar@kari.re.kr](mailto:hahacar@kari.re.kr))

# Appendix

## Thermophysical properties (1)

- 아래의 조건 사용 시 *thermophysicalProperties*의 thermoType에서 가능한 옵션들
 

|                 |              |
|-----------------|--------------|
| transport       | Sutherland;  |
| mixture         | pureMixture; |
| equationOfState | perfectGas;  |



| type        | mixture     | transport  | thermo | equationOfState | specie | energy           |
|-------------|-------------|------------|--------|-----------------|--------|------------------|
| hePsiThermo | pureMixture | sutherland | hConst | perfectGas      | specie | sensibleEnthalpy |
| hePsiThermo | pureMixture | sutherland | janaf  | perfectGas      | specie | sensibleEnthalpy |
| heRhoThermo | pureMixture | sutherland | hConst | perfectGas      | specie | sensibleEnthalpy |
| heRhoThermo | pureMixture | sutherland | janaf  | perfectGas      | specie | sensibleEnthalpy |

### 7.1.2 Transport model

The transport modelling concerns evaluating dynamic viscosity  $\mu$ , thermal conductivity  $\kappa$  and thermal diffusivity  $\alpha$  (for internal energy and enthalpy equations). The current transport models are as follows:

**const** assumes a constant  $\mu$  and Prandtl number  $Pr = c_p \mu / \kappa$  which is simply specified by a two keywords, mu and Pr, respectively.

**sutherland** calculates  $\mu$  as a function of temperature  $T$  from a Sutherland coefficient  $A_s$  and Sutherland temperature  $T_s$ , specified by keywords As and Ts;  $\mu$  is calculated according to:

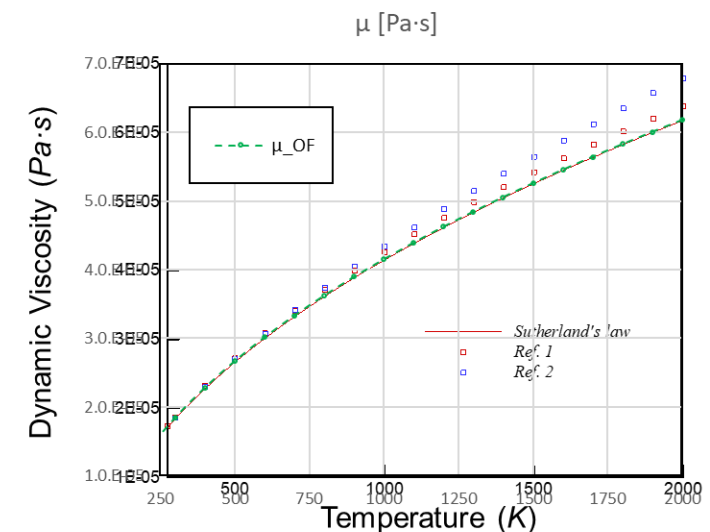
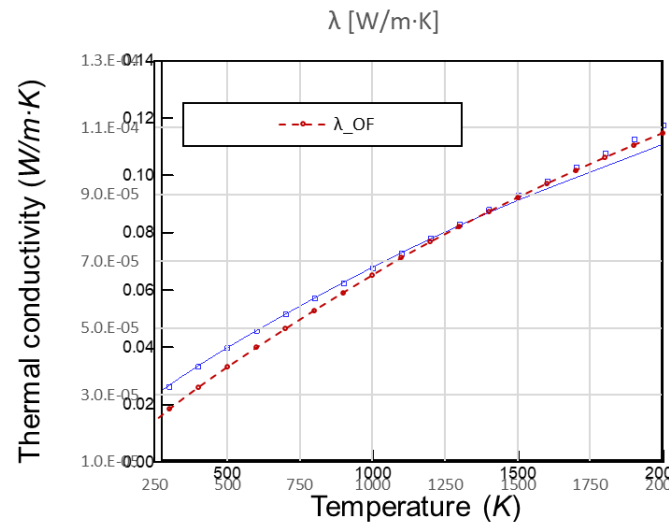
$$\mu = \frac{A_s \sqrt{T}}{1 + T_s/T} \quad (7.1)$$

(OpenFOAM User Guide ver.7)

\* New coefficients

$$\mu = \frac{A_s \sqrt{T}}{1 + T_s/T}$$

- $A_s = 1.4579e-06$ ;
- $T_s = 110.4$ ;
- $Pr = 0.72$ ;



### Verification of new coefficients for thermal conductivity and dynamic viscosity of air

Gas viscosity data from

- <http://www.lmnoeng.com/Flow/GasViscosity.htm>
- "Viscosity and Thermal Conductivity of Dry Air in the Gaseous Phase", K.Kadoya, N. Matsunaga, and A. Nagashima, J. Phys. Chem. Ref. Data, Vol. 14, No. 4, 1985

## Thermophysical properties (2)

- 온도의 함수인 Cp 를 사용하기 위해 → JANAF option. 아래와 같이 설정:  
thermo janaf

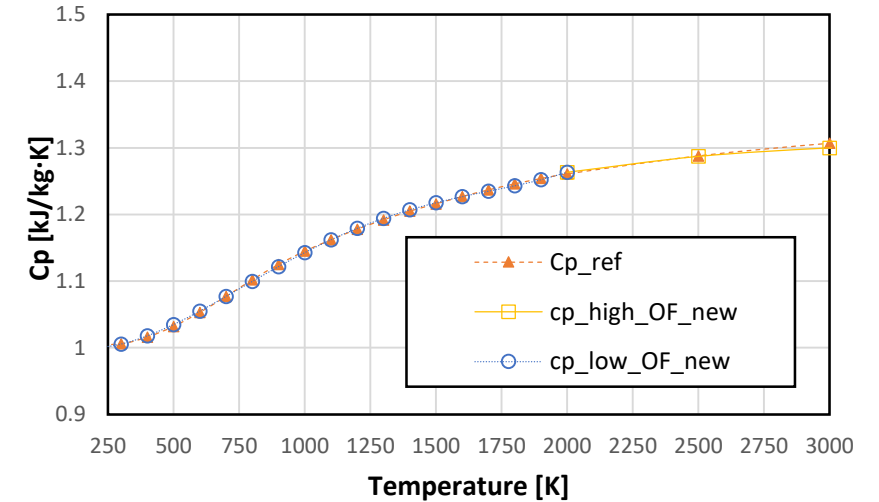
- JANAF function

$$c_p = R(\{(a_4 T + a_3)T + a_2\}T + a_1)T + a_0$$

$$H = \left\{ \left( \left[ \left( \left( \frac{a_4}{5} T + \frac{a_3}{4} \right) T + \frac{a_2}{3} \right) T + \frac{a_1}{2} \right] T + a_0 \right) T + a_5 \right\}$$

$$S = \left( \left[ \left( \left( \frac{a_4}{4} T + \frac{a_3}{3} \right) T + \frac{a_2}{2} \right) T + a_1 \right] T + a_0 \ln(T) + a_6 \right)$$

- molWeight\_air=28.9645 kg/kmol, R=287.058 J/kg·K
- Low: 100~2000, high: 2000~5000
- Tcommon(2000K)에서 두 곡선이 만나도록 Coefficients a0, a5 및 a6 수정  
⇒ 두 곡선이 만나지 않을 시 오류 발생.



Verification of new coefficients for specific heat capacity at constant pressure of air

### New coefficient of JANAF function

|    | highCpCoeffs   | lowCpCoeffs  |
|----|----------------|--------------|
| a0 | 3.0225407      | 3.53881      |
| a1 | 1.3968838E-03  | -6.77619E-04 |
| a2 | -4.9262577E-07 | 2.26946E-06  |
| a3 | 7.8600091E-11  | -1.44141E-09 |
| a4 | -4.6074978E-15 | 2.91951E-13  |
| a5 | -923.93753     | -990.9637    |
| a6 | 5.8718221      | 3.4374391    |

janaf calculates  $c_p$  as a function of temperature  $T$  from a set of coefficients taken from JANAF tables of thermodynamics. The ordered list of coefficients is given in Table 7.1. The function is valid between a lower and upper limit in temperature  $T_l$  and  $T_h$  respectively. Two sets of coefficients are specified, the first set for temperatures above a common temperature  $T_c$  (and below  $T_h$ ), the second for temperatures below  $T_c$  (and above  $T_l$ ). The function relating  $c_p$  to temperature is:

$$c_p = R(\{(a_4 T + a_3)T + a_2\}T + a_1)T + a_0 \tag{7.4}$$

In addition, there are constants of integration,  $a_5$  and  $a_6$ , both at high and low temperature, used to evaluating  $h$  and  $s$  respectively.

| Description                      | Entry           | Keyword                         |
|----------------------------------|-----------------|---------------------------------|
| Lower temperature limit          | $T_l$ (K)       | Tlow                            |
| Upper temperature limit          | $T_h$ (K)       | Thigh                           |
| Common temperature               | $T_c$ (K)       | Tcommon                         |
| High temperature coefficients    | $a_0 \dots a_4$ | highCpCoeffs (a0 a1 a2 a3 a4... |
| High temperature enthalpy offset | $a_5$           | a5...                           |
| High temperature entropy offset  | $a_6$           | a6)                             |
| Low temperature coefficients     | $a_0 \dots a_4$ | lowCpCoeffs (a0 a1 a2 a3 a4...  |
| Low temperature enthalpy offset  | $a_5$           | a5...                           |
| Low temperature entropy offset   | $a_6$           | a6)                             |

Table 7.1: JANAF thermodynamics coefficients.



## Thermophysical properties (3)

```

/*----- C++ -----*/
|=====|
| \ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox
| \ / | O p e r a t i o n | Version: v1906
| \ / | A n d | Web: www.OpenFOAM.com
| \ / | M a n i p u l a t i o n |
|-----|
FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       thermophysicalProperties;
}
// ***** //

thermoType
{
  type         hePsiThermo;
  mixture      pureMixture;
  transport    sutherland;
  thermo       janaf;
  equationOfState perfectGas;
  specie       specie;
  energy       sensibleEnthalpy;
}

mixture
{
  specie
  {
    molWeight      28.9645;
  }
  thermodynamics
  {
    Tlow           100;
    Thigh          5000;
    Tcommon        2000;
    highCpCoeffs  ( 3.0225407 1.3968838E-03 -4.9262577E-07 7.8600091E-11 -4.6074978E-15 -923.93753 5.8718221);
    lowCpCoeffs   ( 3.5388100 -6.7761900E-04 2.2694600E-06 -1.4414100E-09 2.9195100E-13 -990.9637 3.4374391);
  }
  transport
  {
    As             1.4579e-06;
    Ts             110.4;
    Pr             0.72;
  }
}
// ***** //

```

thermophysicalProperties for the general OpenFOAM version with new coefficients

```

/*----- C++ -----*/
|=====|
| \ / | F i e l d | foam-extend: Open Source CFD
| \ / | O p e r a t i o n | Version: 4.1
| \ / | A n d | Web: http://www.foam-extend.org
| \ / | M a n i p u l a t i o n |
|-----|
FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       thermophysicalProperties;
}
// ***** //

Pr           Pr [ 0 0 0 0 0 0 ] 0.72;

thermoType   hPsiThermo<pureMixture<sutherlandTransport<specieThermo<janafThermo<perfectGas>>>>;

mixture      air 1 28.9645 /* specie ? molWeight */
             100 5000 2000 /* Tlow Thigh Tcommon */
             3.0225407 1.3968838E-03 -4.9262577e-07 7.8600091e-11 -4.6074978e-15 -923.93753 5.8718221 /* highCpCoeffs a0 ~ a6 */
             3.5388100 -6.7761900E-04 2.2694600E-06 -1.4414100E-09 2.9195100E-13 -990.9637 3.4374391 /* lowCpCoeffs a0 ~ a6 */
             1.4579e-06 110.4; /* As Ts */

// ***** //

```

thermophysicalProperties for the foam-extend with new coefficients