하늘로 띄운 꿈, 우주에서 찾는 미래

Foam-extend를 이용한 가스터빈 유동해석

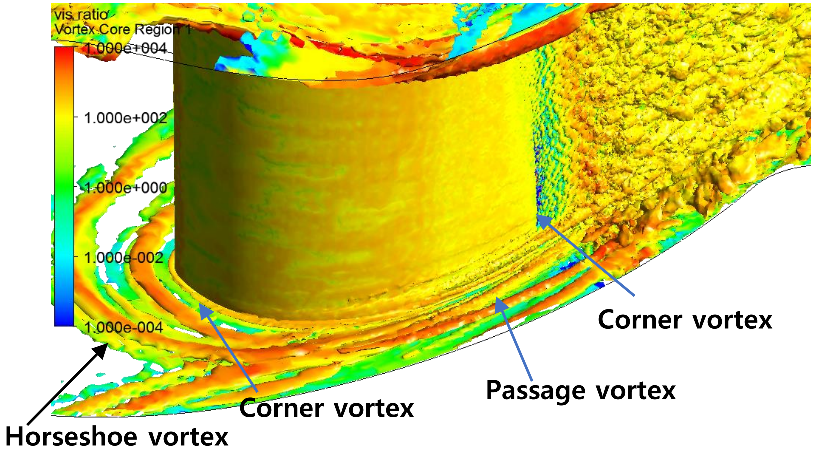
한국항공우주연구원
항공추진연구부
강승환

Contents

1. Introduction

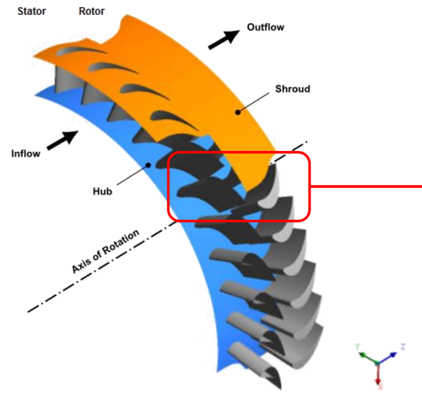
Background

- (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**

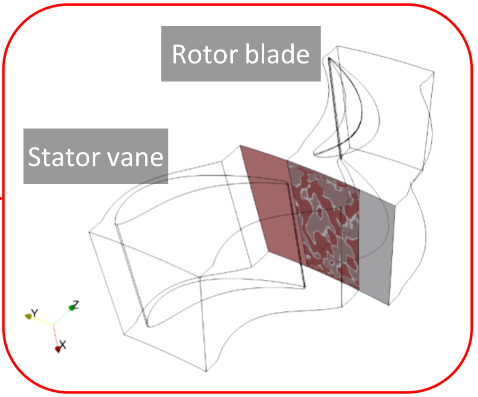


A Large Eddy Simulation of Axial Turbine using ANSYS CFX

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/

foam-extend
Open Source CFD Toolbox
Status: Beta Brought to you by: bgsch

- 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.

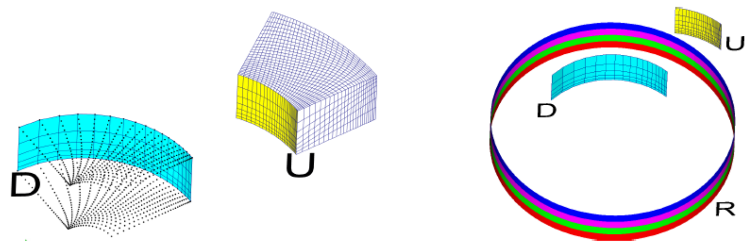


Figure 2. Illustration of a simple mixingPlane interface configuration.

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

mixingPlane

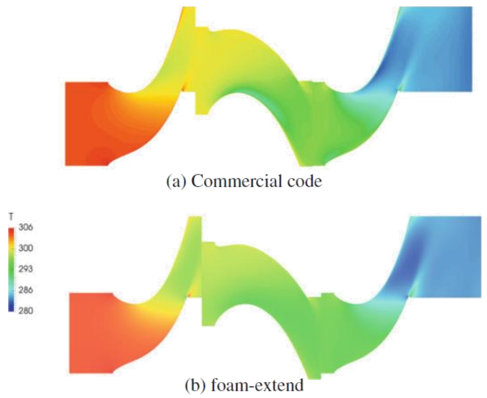
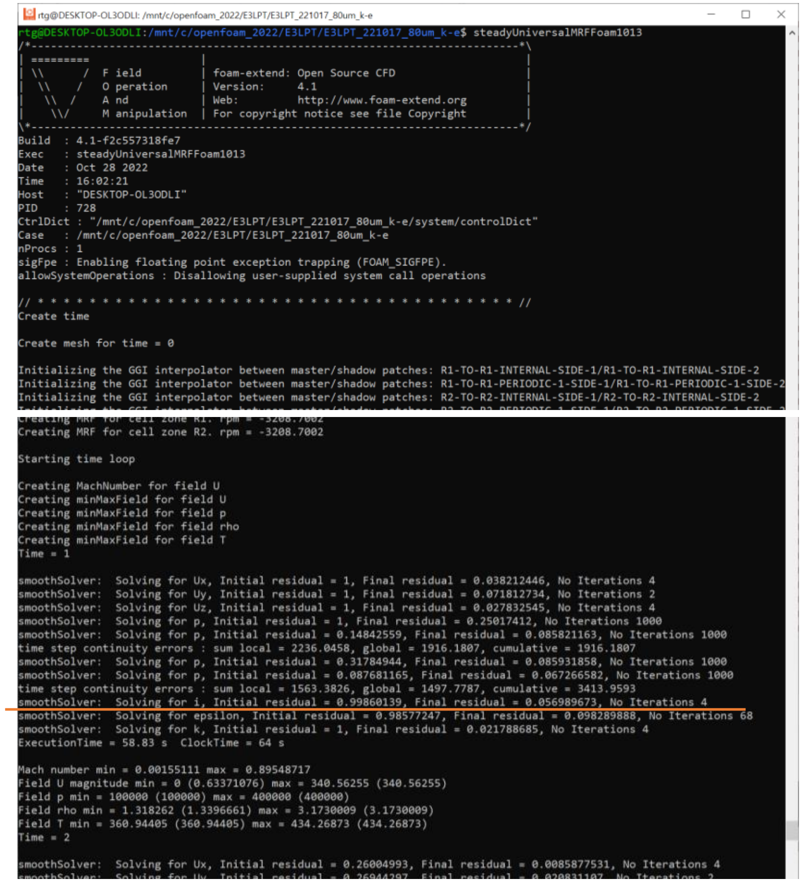


Fig. 6 Temperature field comparison

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

A calculation of 1.5 axial turbine with foam-extend



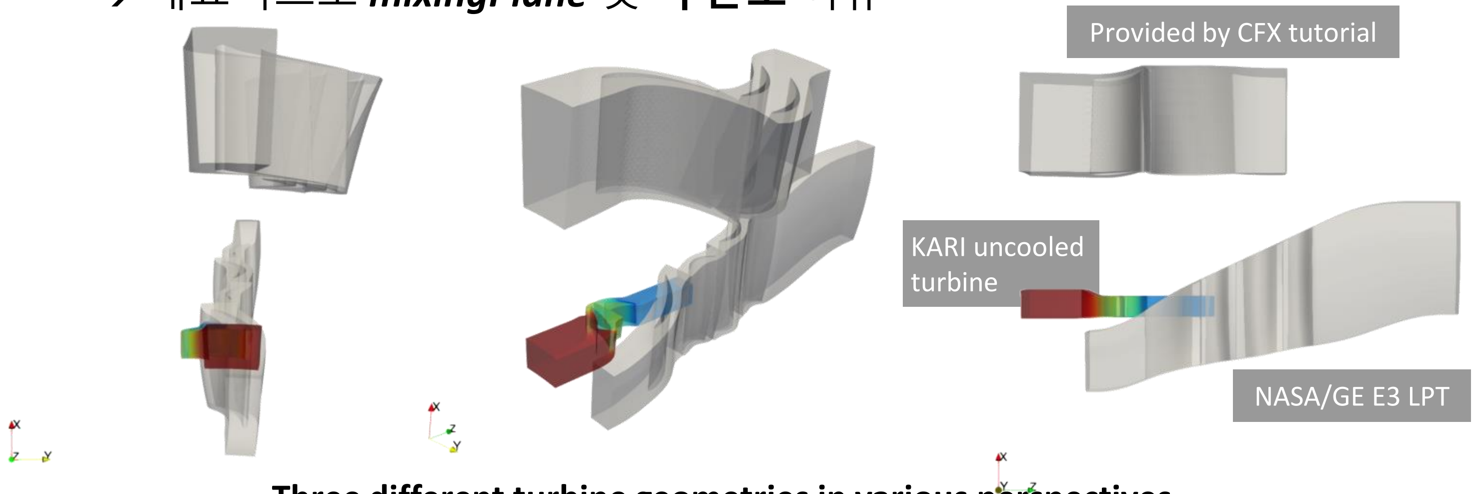
An Execution of steadyUniversalMRFFoam

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 5th national supercomputer, Cray CS500
https://www.ksc.re.kr/eng/resources/nurion



Three different turbine geometries in various perspectives

```
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

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, C는 유체의 절대 속도, C_θ는 유체의 접선 속도

- 일률(Power)

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

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

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

$$\therefore i = h_0 \boxed{- U_{rot} C_\theta}$$

Rothalpy jump

- 회전이 없다면 “Rothalpy = Total Enthalpy”

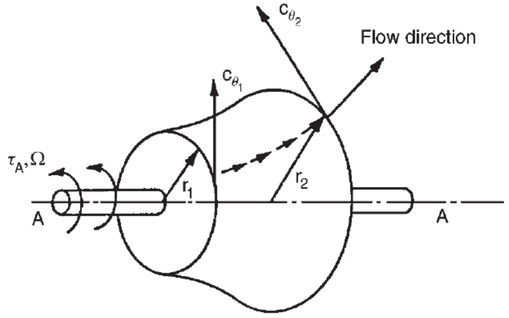


Fig. 2.4. Control volume for a generalised turbomachine.

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

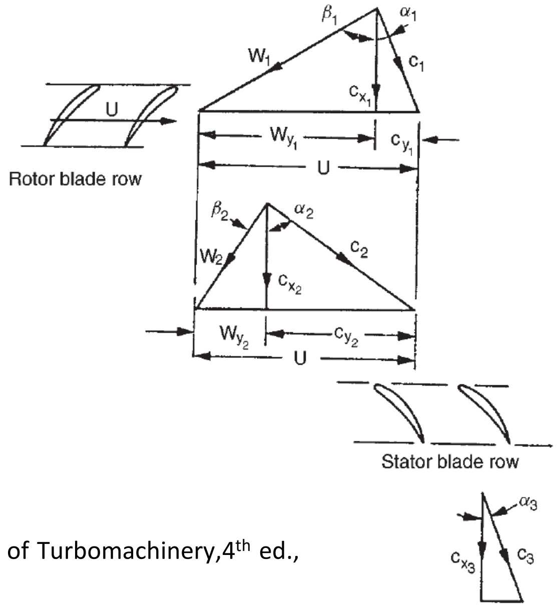


Fig. 5.2. Velocity diagrams for a compressor stage.

※ 속도 삼각형으로 Rothalpy (W는 유체의 상대속도)

$$h_0 = h + \frac{1}{2} C^2$$

$$i = h + \frac{1}{2} C^2 - U_{rot} C_\theta = h + \frac{1}{2}\left(C_x^2 + C_\theta^2 - 2 U_{rot} C_\theta\right)$$

$$= h + \frac{1}{2}\left(C_x^2 + (C_\theta - U_{rot})^2 - U_{rot}^2\right)$$

$$= h + \frac{1}{2}\left(C_x^2 + W_\theta^2 - U_{rot}^2\right) = h + \frac{1}{2}\left(W^2 - U_{rot}^2\right)$$

$$= h + \frac{1}{2}\left(U_{rel}^2 - U_{rot}^2\right)$$

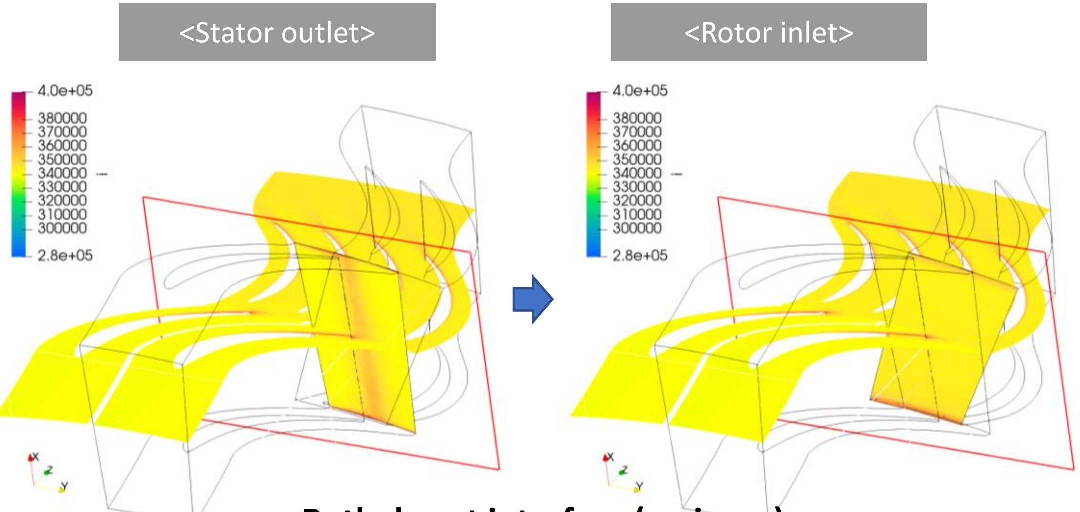
2. mixingPlane Issue

MixingPlane error

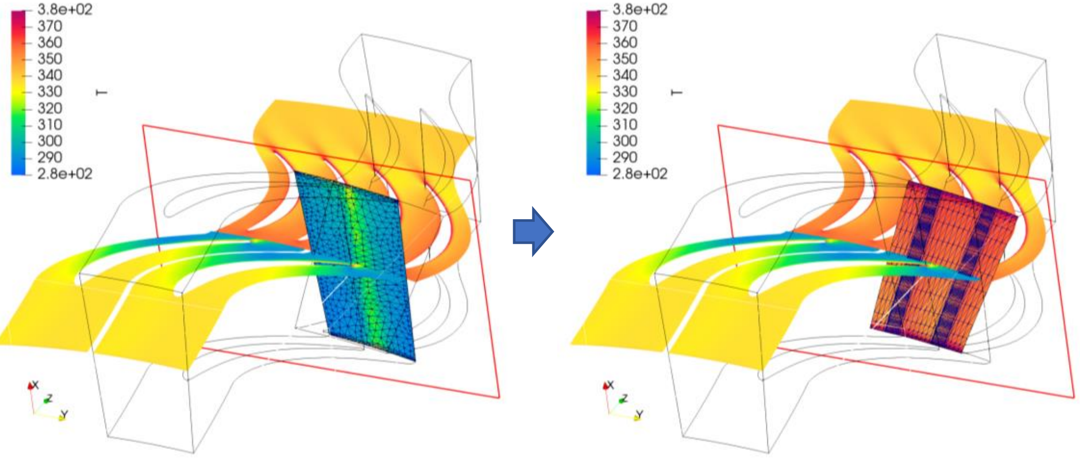
- 경고 메시지
  - 솔버와 라이브러리에서 정의된 변수의 대소문자가 달라서 발생하는 문제.
  - Rothalpy jump 미작동 → 온도 불연속

```
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



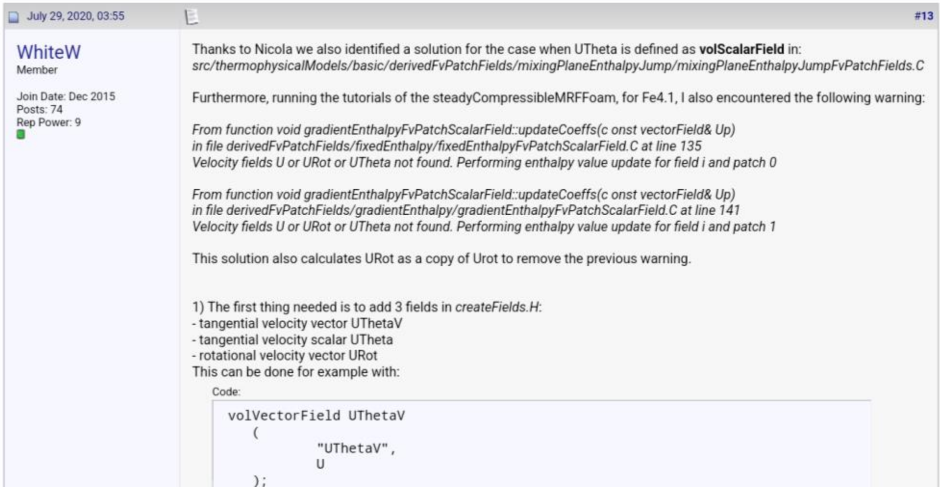
Rothalpy at interface (no jump)



(Discontinuous) Temperature at interface

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/fixedEnthalpy/fixedEnthalpyFvPatchScalarField.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/gradientEnthalpy/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:

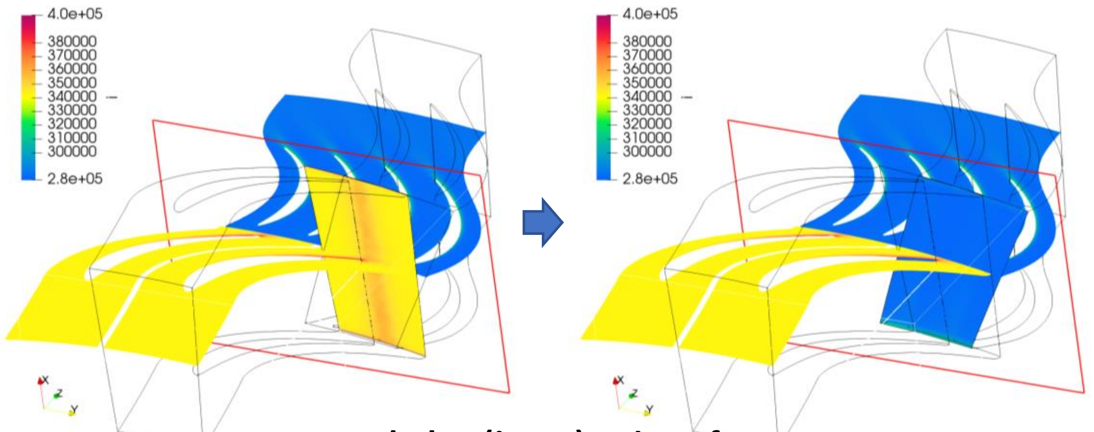
Code:

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

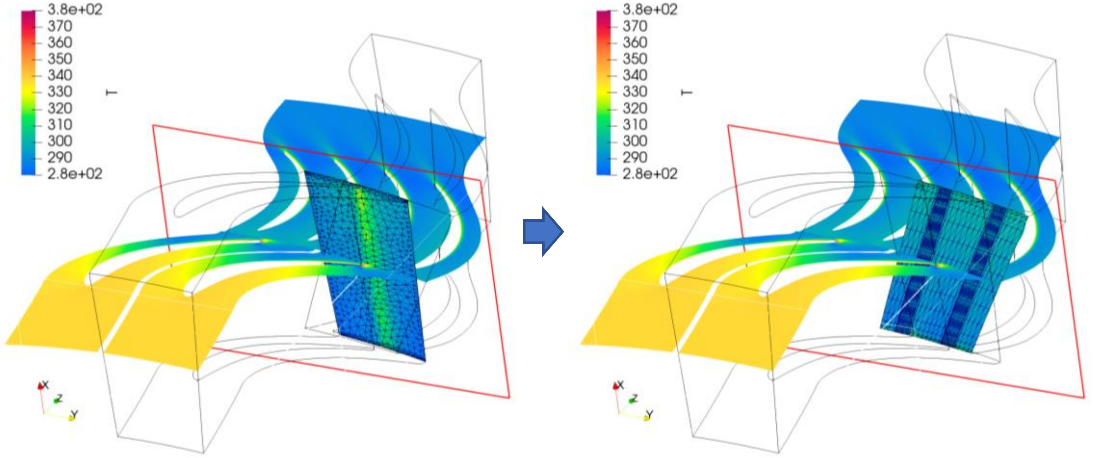
- 관련된 MRFZone files에 UTheta function code 추가.
- createFields.H 및 iEqn.H에 URot 및 UTheta 변수 추가.

→ Rothalpy jump 작동 → 온도 연속

<Stator outlet> <Rotor inlet>



Rothalpy (jump) at interface

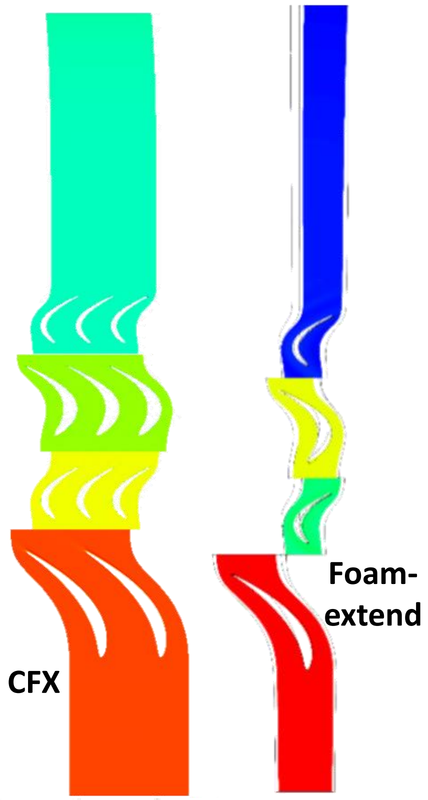


(Continuous) Temperature at interface

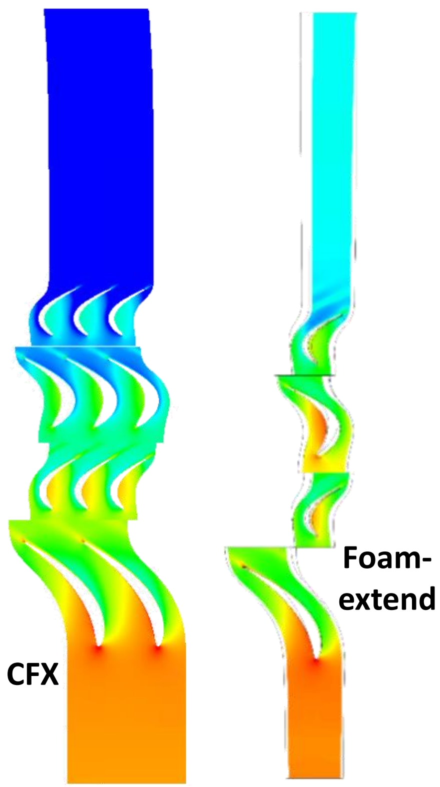
2. mixingPlane Issue

Multi-stage error

- 단과 단 사이의 온도 불연속 다시 발생.
- $U_{rot} = U - U_{rel}$
- Rothalpy: $i = h_0 - U_{rot}C_\theta$
- C_θ와 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.



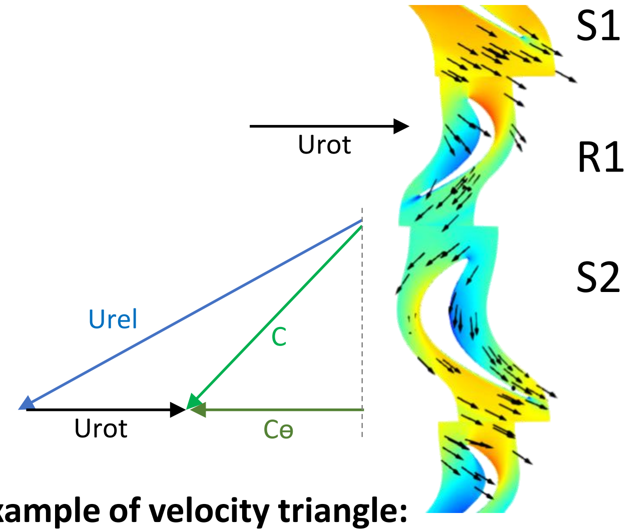
Comparison of rothalpy



Comparison of temperature

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | foam-extend: Open Source CFD                    |
|  \\    /   O peration     | Version:     4.1                                |
|   \\  /    A nd           | Web:         http://www.foam-extend.org         |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       volVectorField;
    location    "2000";
    object      URot;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

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

internalField   nonuniform List<vector>
704520
(
(-1.343462 -41.384598 0)
(-1.2631974 -41.387237 0)
(-1.1831929 -41.389721 0)
(-1.1029095 -41.392048 0)
(-1.0222642 -41.394203 0)
```

An example of U_{rot}



An example of velocity triangle: different direction between C_θ and U_{rot}

2. mixingPlane Issue

KARI 한국항공우주연구원 Korea Aerospace Research Institute

Original code

```
// corrections
volVectorField URot //corrected(1008)
(
    "URot",
    U - Urel
);
/*  volVectorField URot
(
    IOobject
    (
        "URot",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    U - Urel
);*/

// create U_thetaV
volVectorField UThetaV
(
    "UThetaV",
    U
);

// Create UTheta
volScalarField UTheta
(
    "UTheta",
    mag(U)
);
/*  volScalarField UTheta
(
    IOobject
    (
        "UTheta",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    mag(U)
);
mrfZones.UTheta(UThetaV);
UTheta = mag(UThetaV); */
//
```

length : 2,273 lines : 135 Ln : 117 Col : 1 Pos : 1,886 Unix (LF)

UTheta defined as scalar

UTheta transformed to magnitude

createFields.H

C:\Users\hahacar\AppData\Local\Temp\BNZ.637b992f69e1d310\mixingPlaneEnthalpyJumpFvPatc...

```
if
(
    !this->db().objectRegistry::found(URotName)
 || !this->db().objectRegistry::found(UThetaName)
)
{
    // Velocities not available, do not update
    InfoIn
    (
        "void gradientEnthalpyFvPatchScalarField::"
        "updateCoeffs(const vectorField& Up)"
    )   << "Velocity fields " << URotName << " or "
        << UThetaName << " not found.  "
        << "Performing enthalpy value update" << endl;

    jump_ = 0;
}
else
{
    const fvPatchVectorField& URotp =
        lookupPatchField<volVectorField, vector>(URotName);

    const fvPatchScalarField& UThetap =
        lookupPatchField<volScalarField, scalar>(UThetaName);

    if (rotating)
    {
        // We can either make jump_ on neighbour field and interpolate (in
        // jumpOverlapGgi) or interpolate first, then add jump_ for
        // internalField
        const scalarField UThetaIn = UThetap.patchInternalField();

        jump_ = - (
                      mag(UThetaIn)
                     *mag(URotp.patchInternalField())
                  );
    }
    else
    {
        const scalarField UThetaIn = UThetap.patchNeighbourField();

        jump_ =  (
                     mag(UThetaIn)
                    *mag(URotp.patchNeighbourField())
                 );
    }
}

jumpMixingPlaneFvPatchField<scalar>::updateCoeffs();
}
```

length : 3,735 lines : 120 Ln : 92 Col : 22 Sel : 9 | 1 Unix (LF) UTF-8 INS

Scalar product with magnitude value of UTheta and URot

mixingPlaneEnthalpyJumpFvPatchFields.C

2. mixingPlane Issue

KARI 한국항공우주연구원 Korea Aerospace Research Institute

Corrected code

rtg@DESKTOP-OL3ODLI: ~

```
#include "surfaceFields.H"

// * * * * * * * * * * * * * * Static Data Members * * * * * * * * * * * * * //

namespace Foam
{
    makeTemplatePatchTypeField
    (
        fvPatchScalarField,
        mixingPlaneEnthalpyJumpFvPatchScalarField
    );
}

// * * * * * * * * * * * * * * * Member Functions  * * * * * * * * * * * * * //

template<>
void Foam::mixingPlaneEnthalpyJumpFvPatchField<Foam::scalar>::updateCoeffs()
{
    if (updated())
    {
        return;
    }

    // Get access to relative and rotational velocity
    const word URotName("URot");
    /*const word UThetaName("UTheta");*/
    const word UThetaName("UThetaV");

    jump_ = 0;

    if
    (
        !this->db().objectRegistry::found(URotName)
     || !this->db().objectRegistry::found(UThetaName)
    )
    {
        // Velocities not available, do not update
        InfoIn
        (
            "void gradientEnthalpyFvPatchScalarField::"
            "updateCoeffs(const vectorField& Up)"
        )   << "Velocity fields " << URotName << " or "
            << UThetaName << " not found.  "
            << "Performing enthalpy value update" << endl;
```

Vector value, UThetaV

```
        jump_ = 0;
    }
    else
    {
        const fvPatchVectorField& URotp =
            lookupPatchField<volVectorField, vector>(URotName);

/*        const fvPatchScalarField& UThetap =
            lookupPatchField<volScalarField, scalar>(UThetaName);*/
        const fvPatchVectorField& UThetap =
            lookupPatchField<volVectorField, vector>(UThetaName);

        if (rotating_)
        {
            // We can either make jump_ on neighbour field and interpolate (in
            // jumpOverlapGgi) or interpolate first, then add jump_ for
            // internalField
            /*const scalarField UThetaIn = UThetap.patchInternalField();*/

            jump_ =  - (
                        // mag(UThetaIn)
                        //*mag(URotp.patchInternalField())
                         UThetap.patchInternalField()
                        &URotp.patchInternalField()
                       );
        }
        else
        {
            /*const scalarField UThetaIn = UThetap.patchNeighbourField();*/

            jump_ = (
                       // mag(UThetaIn)
                       //*mag(URotp.patchNeighbourField())
                        UThetap.patchNeighbourField()
                       &URotp.patchNeighbourField()
                    );
        }
    }

    jumpMixingPlaneFvPatchField<scalar>::updateCoeffs();
}


// ************************************************************************* //
```

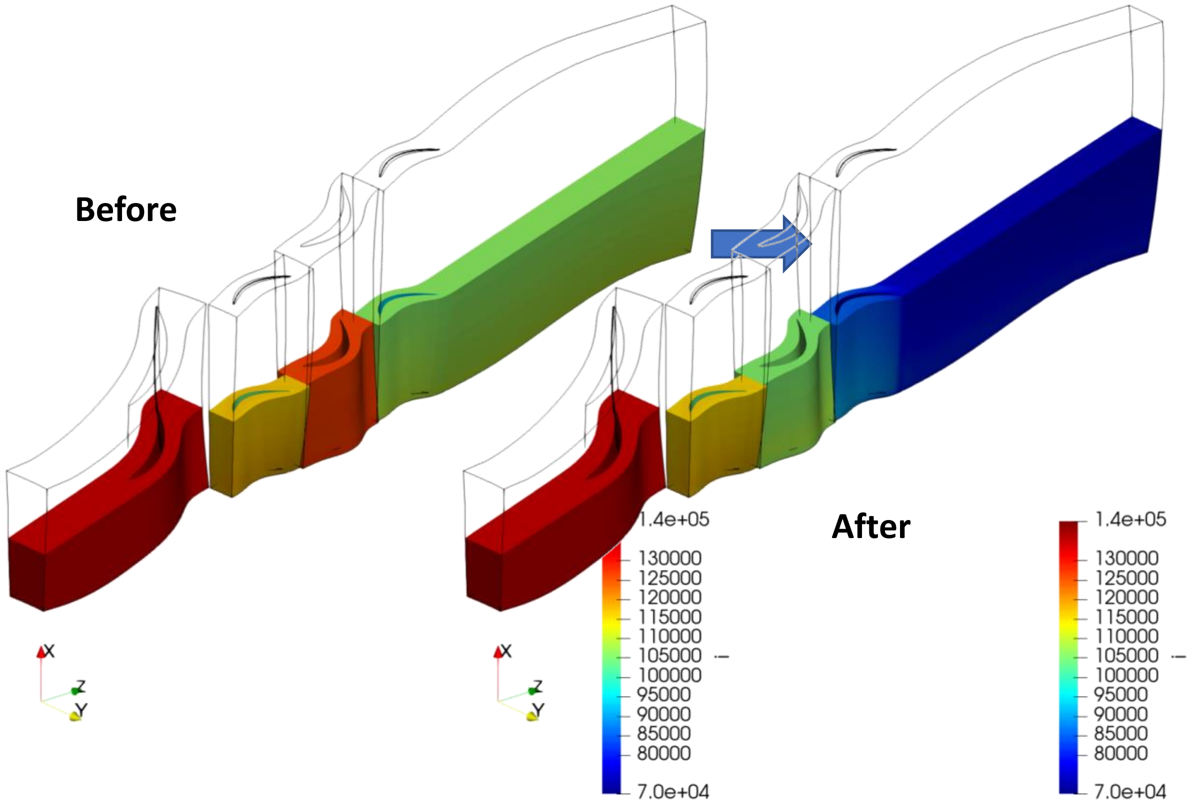
Inner product with vector value of UthetaV and URot

mixingPlaneEnthalpyJumpFvPatchFields.C

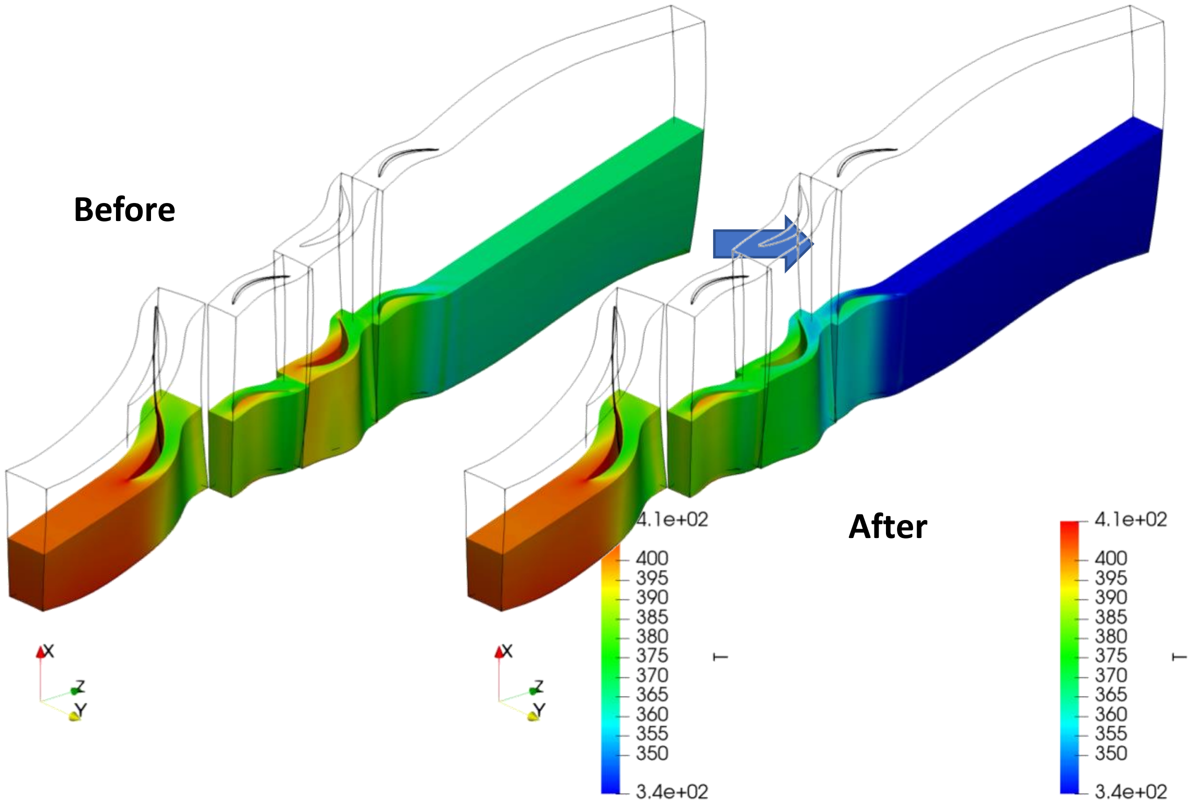
2. mixingPlane Issue

Corrected result of E3 LPT 2 stages

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



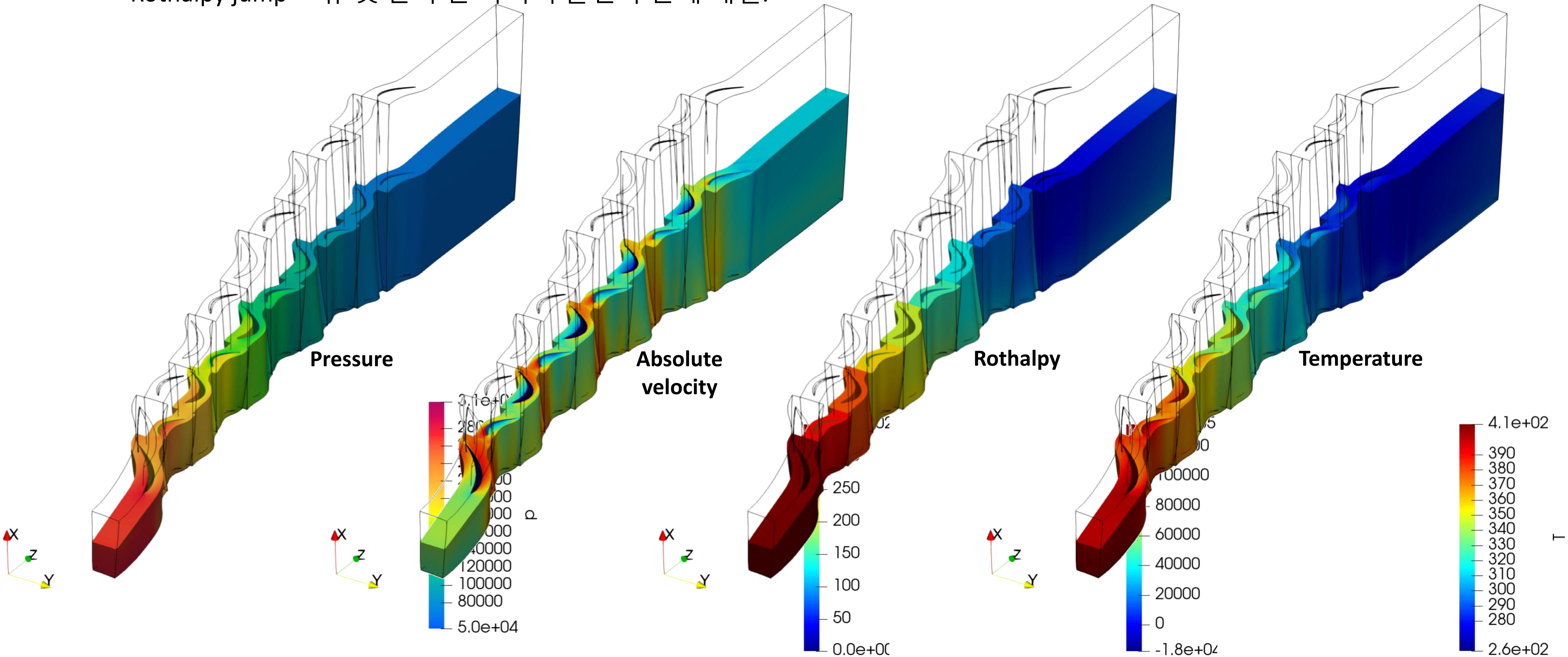
Rotalpy



Temperature

Corrected result of E3 LPT 5 stages

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

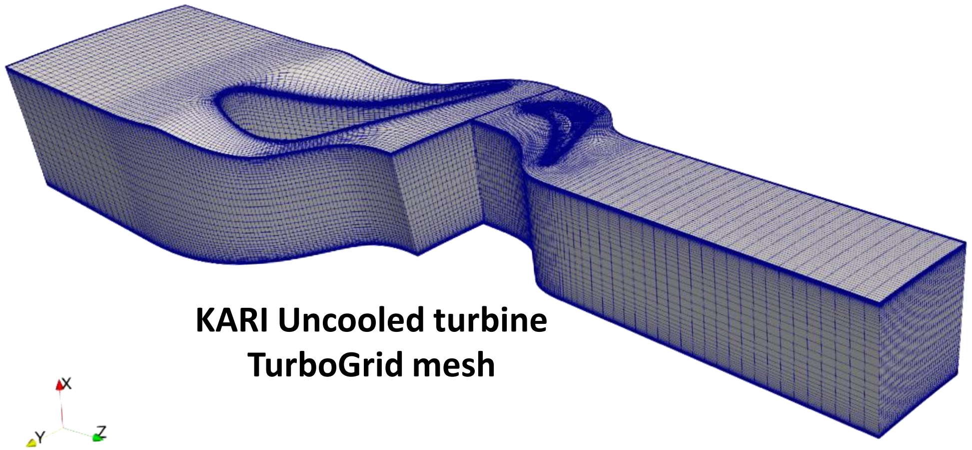


3. Wall Temperature Issue

3. Wall Temperature Issue

KARI turbine

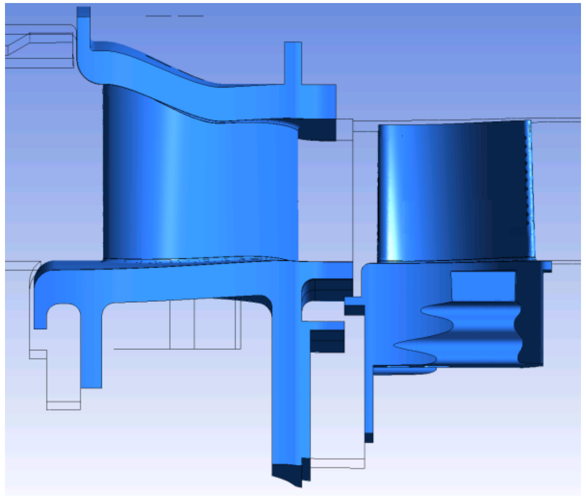
- 항우연에서 설계한 (un)cooled turbine
  - 56 Nozzle vanes
  - 104 Turbine blades
- ANSYS TurboGrid → OpenFOAM grid로 변환
  - (Fine grid)
  - Wall distance = 2e-6 m
  - Number of element of stator: 643,376
  - Number of element of rotor: 1,102,895



KARI Uncooled turbine TurboGrid mesh

Boundary conditions of design point

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



Side view of 1st turbine stage

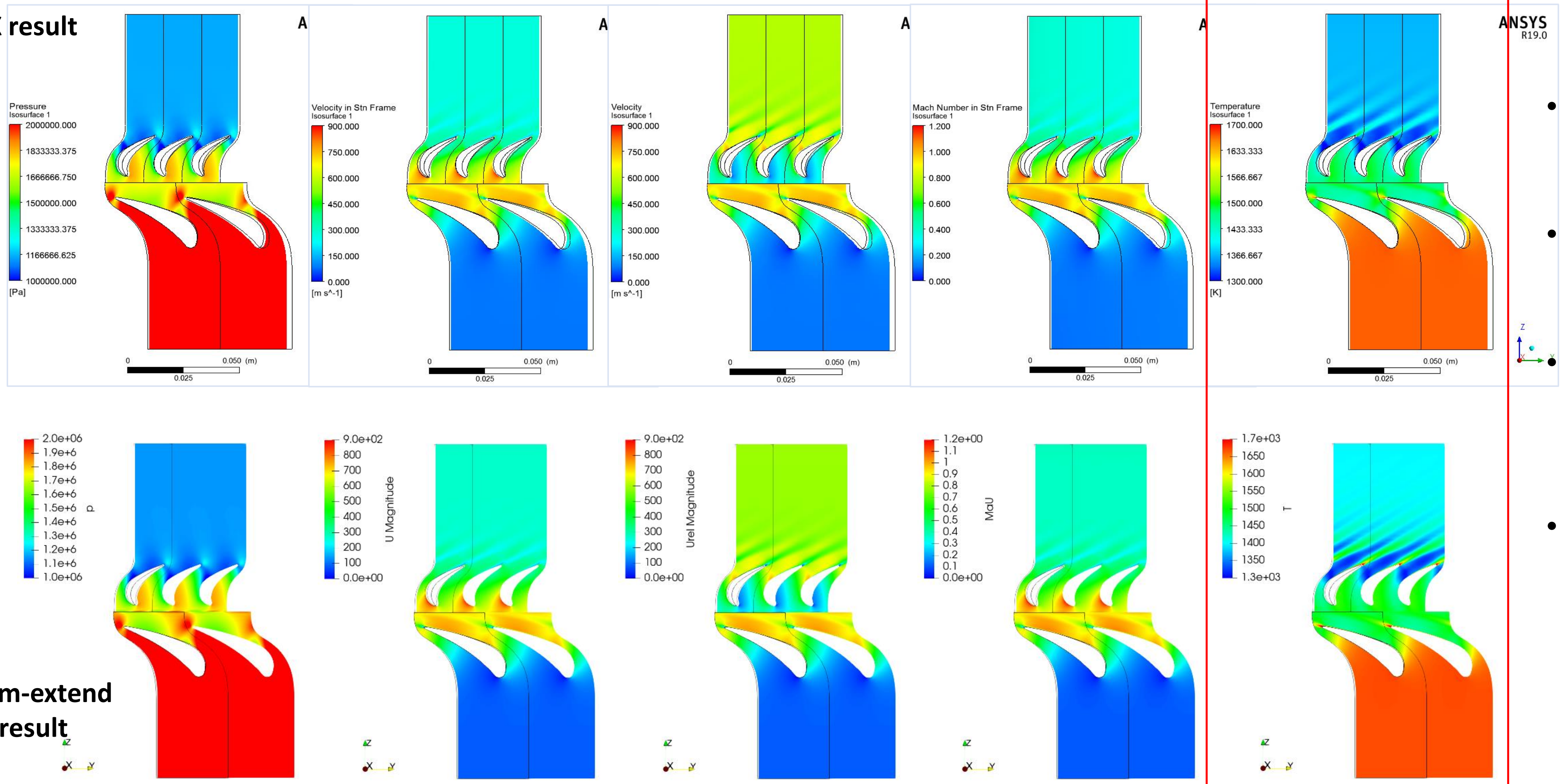
3. Wall Temperature Issue

Calculation results at span 0.5 of KARI turbine

Pressure | Velocity | Relative velocity | Ma | Temperature

CFX result

Foam-extend result

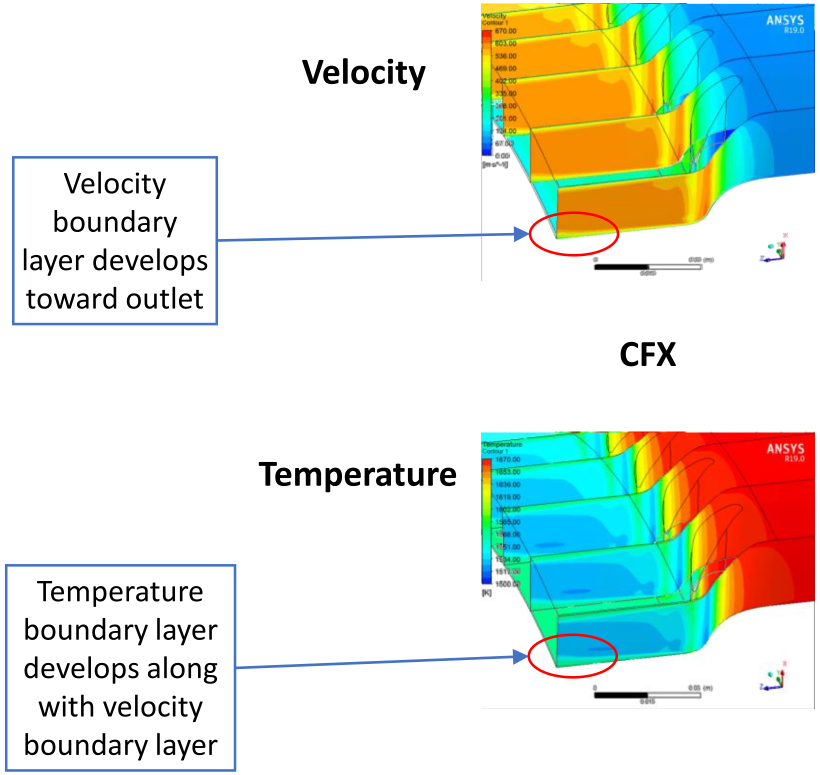


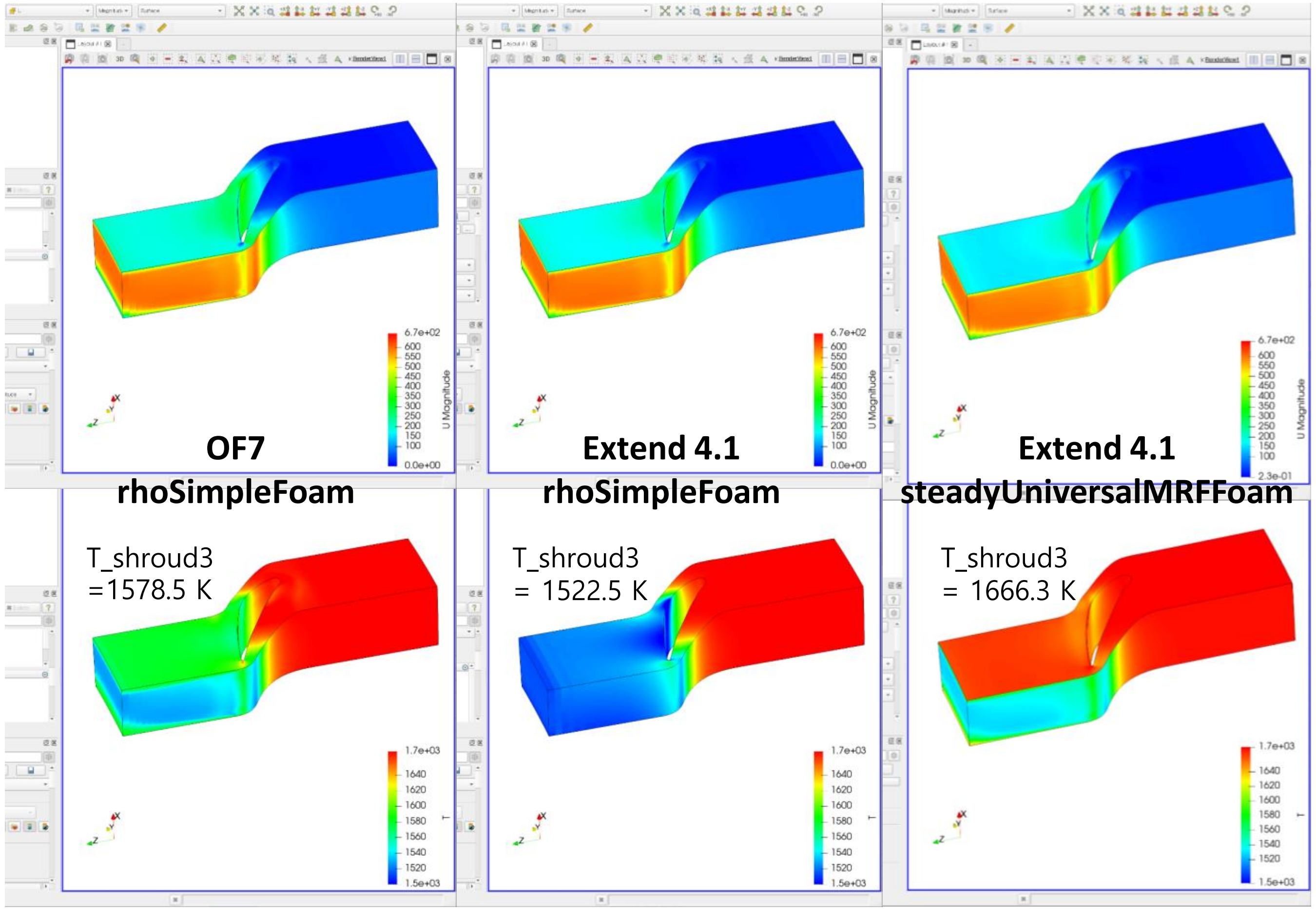
- 오픈폼 결과의 온도 상승 → 벽 온도 상승에서 기인
- Viscous heating term에 상관 없이 벽 온도 상승.
- *gradientEnthalpy* 또는 *zeroGradient*에 상관 없이 벽 온도 상승.
- Rotating domain 뿐 아니라 stationary domain에서도 벽 온도 상승.

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.

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





KARI 한국항공우주연구원 Korea Aerospace Research Institute

Energy Equation (1), Intergral equation

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

$$\frac{d}{dt}\left(\int_{\mathrm{CV}} e\rho d\mathrm{V}\right) + \int_{\mathrm{CS}} e\rho(\boldsymbol{U}\cdot\boldsymbol{n})dA = \dot{Q} - \left(\dot{W}_s + \dot{W}_p + \dot{W}_v\right) + \mathrm{Source}$$

$$= \dot{Q} - \left(\dot{W}_s + \int_{\mathrm{CS}} p(\boldsymbol{U}\cdot\boldsymbol{n})dA - \int_{\mathrm{CS}} \boldsymbol{\tau}\cdot\boldsymbol{U}dA\right) + \mathrm{Source}$$

where,

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

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

(White, Frank M., 2011, "Fluid Mechanics," 7th ed., McGraw-Hill, NY)

3. Wall Temperature Issue

Energy Equation (2), Differential equation

- Integral equation

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

- Differential equation

$L.H.S.$
$$= \left[\frac{\partial}{\partial t}(\rho e) + \frac{\partial}{\partial x}(\rho u e) + \frac{\partial}{\partial y}(\rho v e) + \frac{\partial}{\partial z}(\rho w e)\right] dxdydz$$
$$= \left[\frac{\partial}{\partial t}(\rho e) + \nabla\cdot(\rho \boldsymbol{U} e)\right] dxdydz$$

$R.H.S.$
$$= \left[\nabla\cdot(k\nabla T) - \nabla\cdot(\boldsymbol{U}p) + \nabla\cdot\left(\boldsymbol{U}\cdot\boldsymbol{\tau}_{ij}\right) + \text{source}\right] dxdydz$$

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

where,

$$\dot{Q} = -\left[\frac{\partial}{\partial x}(q_x) + \frac{\partial}{\partial y}(q_y) + \frac{\partial}{\partial z}(q_z)\right] dxdydz$$
$$= -(\nabla\cdot\mathbf{q})dxdydz = -\left(\nabla\cdot(-k\nabla T)\right)dxdydz$$

$$\dot{W}_v$$
$$= -\left[\frac{\partial}{\partial x}(u\tau_{xx} + v\tau_{xy} + w\tau_{xz}) + \frac{\partial}{\partial y}(u\tau_{yx} + v\tau_{yy} + w\tau_{yz}) + \frac{\partial}{\partial z}(u\tau_{zx} + v\tau_{zy} + w\tau_{zz})\right] dxdydz$$
$$= -\nabla\cdot\left(\boldsymbol{U}\cdot\boldsymbol{\tau}_{ij}\right)dxdydz$$

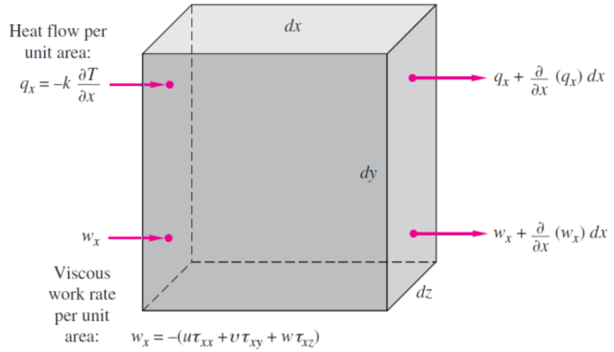


Fig. 4.6 Elemental cartesian control volume showing heat flow and viscous work rate terms in the x direction.

(White, Frank M., 2011, "Fluid Mechanics," 7th ed., McGraw-Hill, NY)

Energy Equation (3), Differential equations

- Expansion of equation

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

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

- Internal energy form

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

- Enthalpy form

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

- Total Enthalpy form

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

- Then, Rothalpy form??

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

where

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

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

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

$$h_0 = h + K$$

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

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

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

$$i = h_0 - U_{rot} U_\theta$$
$$= h + \frac{1}{2}\left(U_{rel}^2 - U_{rot}^2\right)$$

3. Wall Temperature Issue

OpenFOAM energy equation, rhoPimpleFoam (v1906)

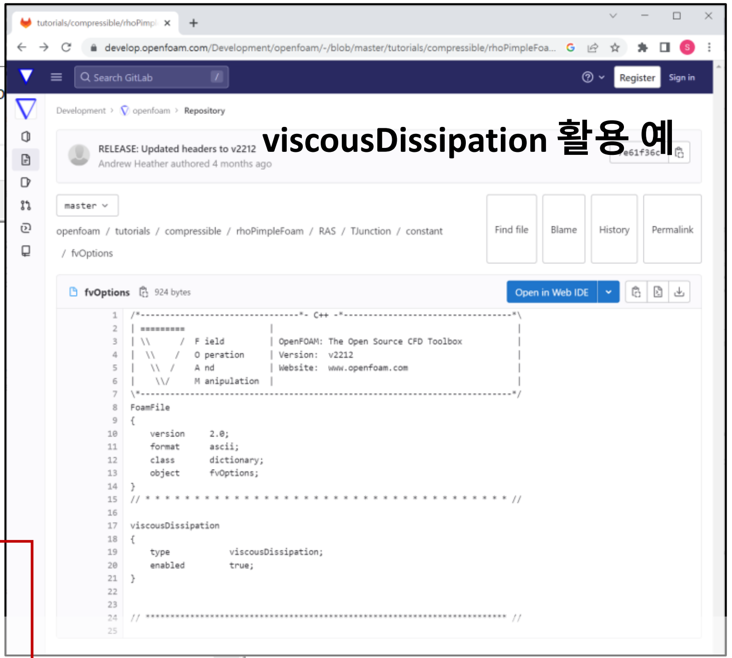
...ers₩compressible₩rhoPimpleFoam₩EEqn.H - Notep

파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 도구(O) 매크로 실행 플러그인 창 관리 ?

sonicFoam.C | EEqn.H | rhoPimpleFoam.C | EEqn.H

```
{
    volScalarField& he = thermo.he();

    fvScalarMatrix EEqn
    (
        fvm::ddt(rho, he) + fvm::div(phi, he)
      + fvc::ddt(rho, K) + fvc::div(phi, K)
      + (
            he.name() == "e"
          ? fvc::div
            (
                fvc::absolute(phi/fvc::interpolate(rho), U),
                p,
                "div(phiv,p)"
            )
          : -dpdt
        )
      - fvm::laplacian(turbulence->alphaEff(), he)
     ==
        fvOptions(rho, he)
    );

    EEqn.relax();

    fvOptions.constrain(EEqn);

    EEqn.solve();

    fvOptions.correct(he);

    thermo.correct();
}
```

C++ source file | length : 603 | lines : 33



- Energy equation of rhoPimpleFoam (v1906)
- Internal energy form

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

- Enthalpy form

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

(.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)

EEqn.H | hEqn.H | iEqn.H

```
{
    volScalarField& he = thermo.he();

    fvScalarMatrix EEqn
    (
        fvm::div(phi, he)
      + (
            he.name() == "e"
          ? fvc::div(phi, volScalarField("Ekp", 0.5*magSqr(U) + p/rho))
          : fvc::div(phi, volScalarField("K", 0.5*magSqr(U)))
        )
      - fvm::laplacian(turbulence->alphaEff(), he)
     ==
        fvOptions(rho, he)
    );

    EEqn.relax();

    fvOptions.constrain(EEqn);

    EEqn.solve();

    fvOptions.correct(he);

    thermo.correct();
}
```

- Energy equation of rhoSimpleFoam (OpenFOAM v.7)
- Internal energy form

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

- Enthalpy form

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

3. Wall Temperature Issue

KARI 한국항공우주연구원 Korea Aerospace Research Institute

OpenFOAM energy equation, steadyUniversalMRFFoam

파일(F) 편집(E) 찾기(S) 보기(V) 인코딩(N) 언어(L) 설정(T) 도구(O) 매크로 실행 플러그인 창 관리 ?

EEqn.H | hEqn.H | iEqn.H

```
{
    fvScalarMatrix iEqn
    (
        fvm::div(phi, i)
      + fvm::SuSp(-fvc::div(phi), i)
      - fvm::laplacian(turbulence->alphaEff(), i)
     ==
        // Viscous heating: note sign (devRhoReff has a minus in it)
      - (turbulence->devRhoReff() && fvc::grad(U))
    );

    iEqn.relax();

    iEqn.solve();

    // From rothalpy, calculate enthalpy after solution of roth
    h = i + 0.5*(magSqr(Urot) - magSqr(Urel));
    h.correctBoundaryConditions();

    // Update thermo for new h
    thermo.correct();
    psis = thermo.psi()/thermo.Cp()*thermo.Cv();
}
```

Rothalpy jump

Kinetic energy

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

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

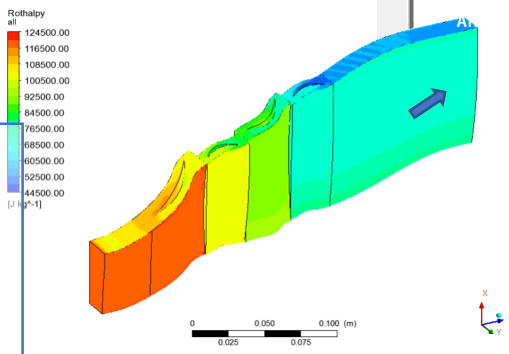
$$\nabla\cdot(\rho \boldsymbol{U} i) + \nabla\cdot(\rho \boldsymbol{U} U_{rot}U_\theta) - \nabla\cdot(\rho\alpha\nabla i) - \nabla\cdot\left(\rho\alpha\nabla(U_{rot}U_\theta - K)\right) = \nabla\cdot\left(\boldsymbol{U}\cdot\boldsymbol{\tau}_{ij}\right) + s$$

```
    fvScalarMatrix iEqn
    (
        fvm::div(phi, i)
      + fvc::div(phi, volScalarField("K1", mag(URot)*mag(UThetaV))) //corrected 230302
//      + fvm::SuSp(-fvc::div(phi), i)
      - fvm::laplacian(turbulence->alphaEff(), i)
      - fvc::laplacian(turbulence->alphaEff(), volScalarField("K2", (mag(URot)*mag(UThetaV)-0.5*magSqr(U))))
//     ==
      // Viscous heating: note sign (devRhoReff has a minus in it)
//        - (turbulence->devRhoReff() && fvc::grad(U))
    );
```

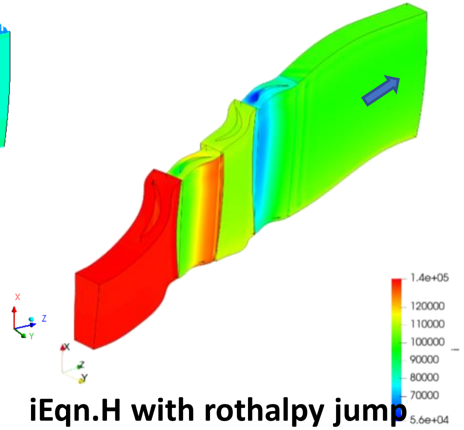
Corrected iEqn.H

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

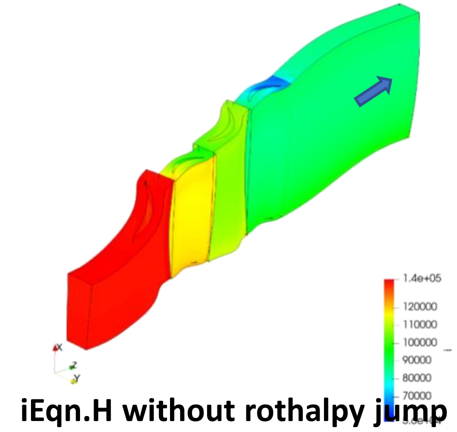
→ 다른 접근 필요.



CFX



iEqn.H with rothalpy jump



iEqn.H without rothalpy jump

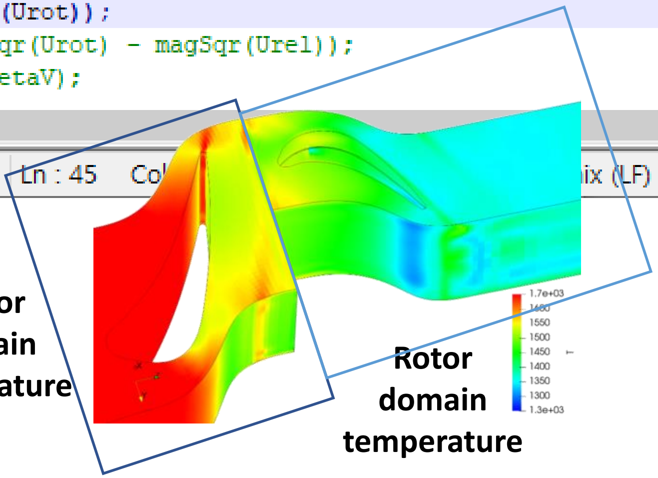
3. Wall Temperature Issue

KARI 한국항공우주연구원 Korea Aerospace Research Institute

steadyUniversalMRFFoam – correction

```
fvScalarMatrix iEqn
(
    fvm::div(phi, i)
  + fvc::div(phi, volScalarField("K", 0.5*magSqr(Urel))) //corrected 230302
  - fvm::laplacian(turbulence->alphaEff(), i)
//   ==
   // Viscous heating: note sign (devRhoReff has a minus in it)
//     - (turbulence->devRhoReff() && fvc::grad(U))
);

iEqn.relax();

iEqn.solve();

// From rothalpy, calculate enthalpy after solution of rothalpy equation
h = i + 0.5*(magSqr(Urot));
//   h = i + 0.5*(magSqr(Urot) - magSqr(Urel));
//   h = i + (URot&UThetaV);
```

length : 1,398   lines : 55   Ln : 45   UTF-8



- Enthalpy form energy equation

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

- 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

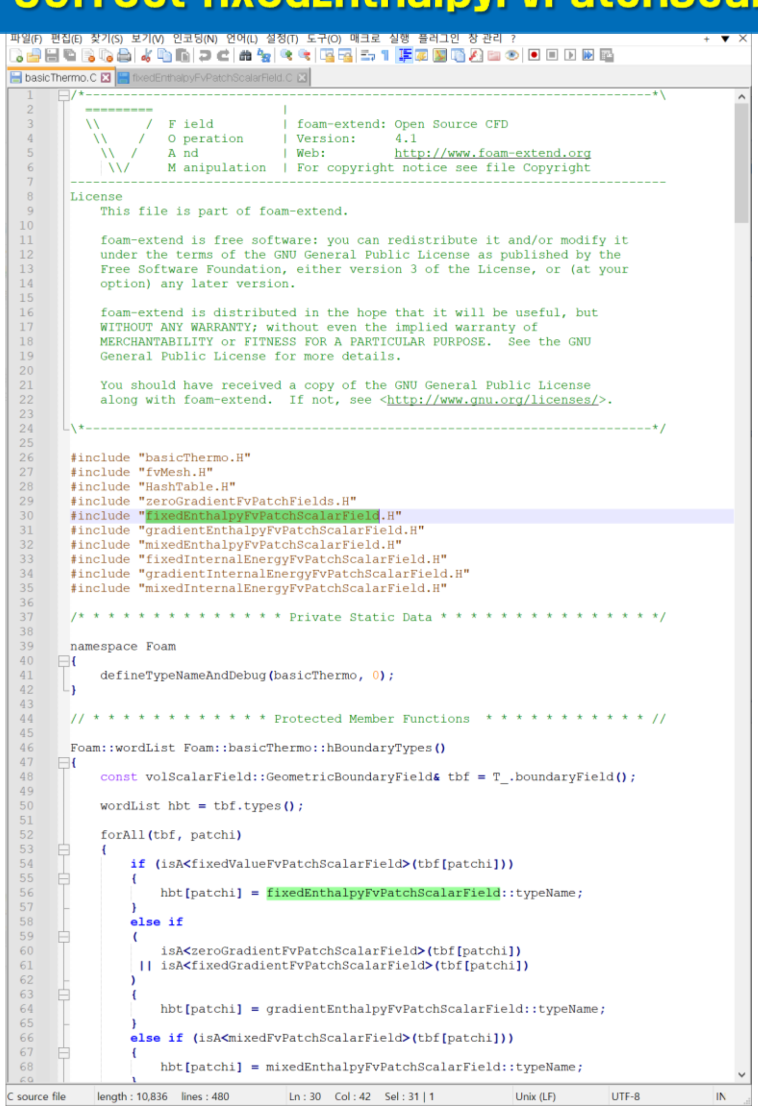
$$\nabla\cdot(\rho\boldsymbol{U}i)+\nabla\cdot\left(\rho\boldsymbol{U}(0.5U_{rel}^2)\right)-\nabla\cdot(\rho\alpha\nabla i) = \nabla\cdot\left(\boldsymbol{U}\cdot\boldsymbol{\tau}_{ij}\right)+s$$

- 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.

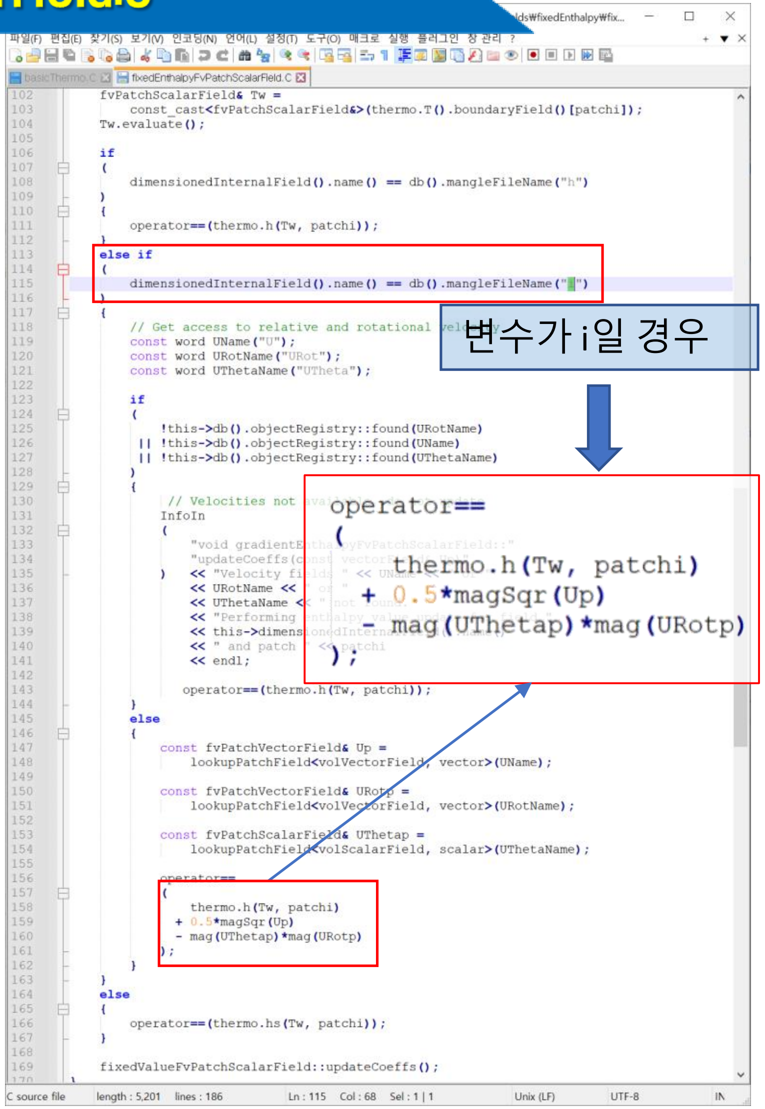
3. Wall Temperature Issue

KARI 한국항공우주연구원 Korea Aerospace Research Institute

Correct fixedEnthalpyFvPatchScalarField.C

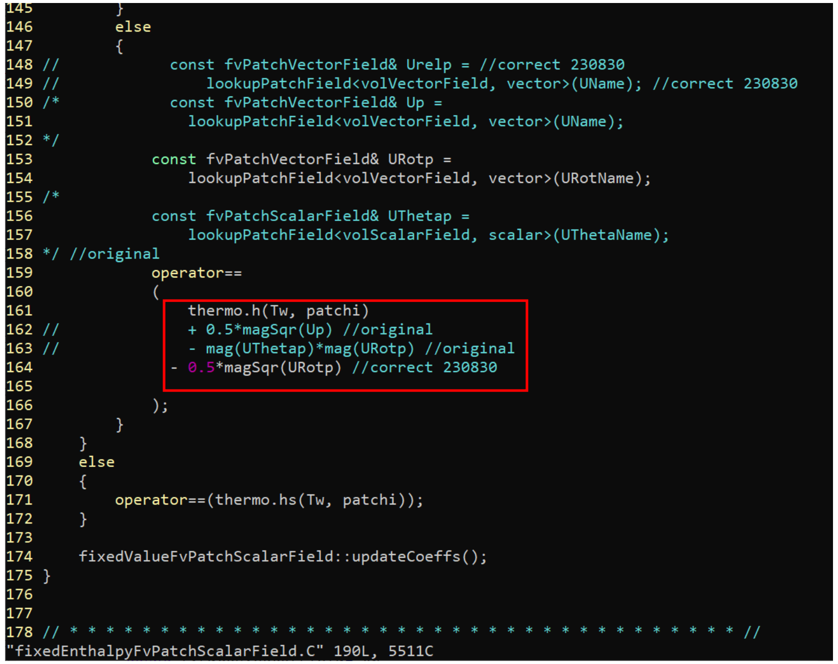


basicThermo.C



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

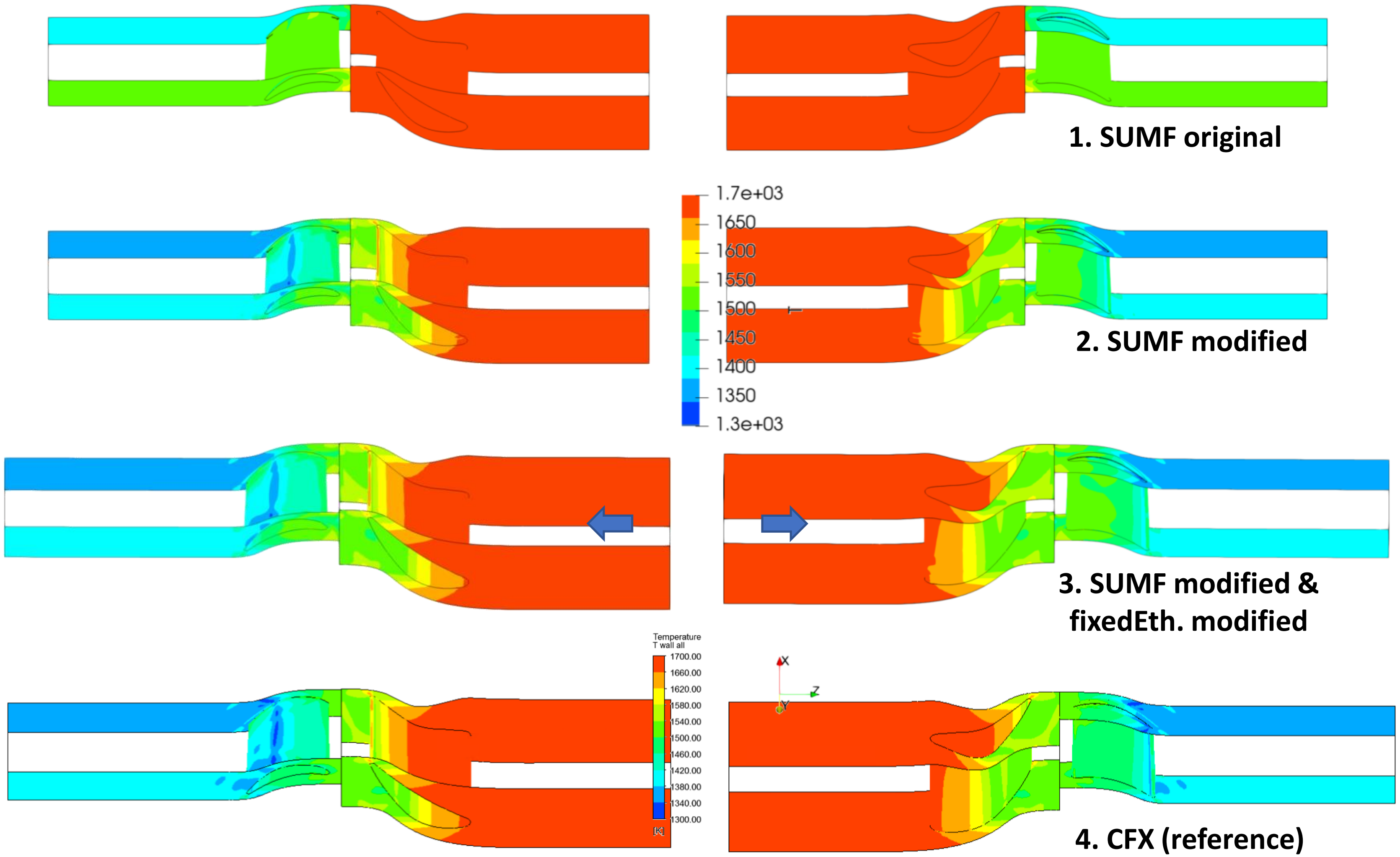


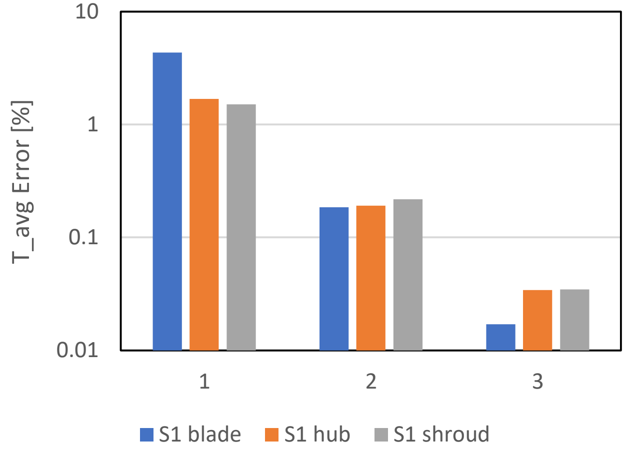
fixedEnthalpyFvPatchScalarField.C modified

KARI 한국항공우주연구원 Korea Aerospace Research Institute

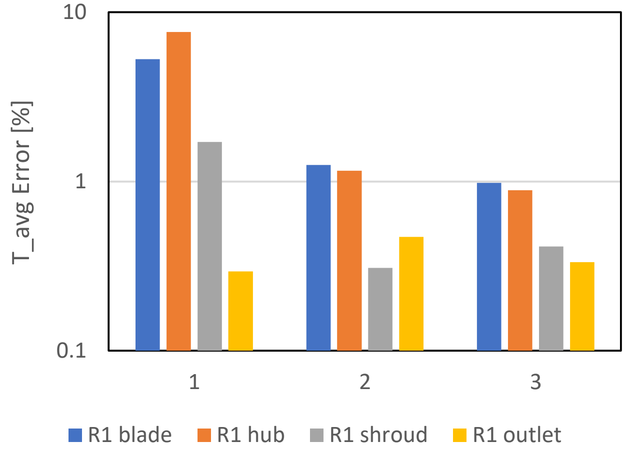
KARI turbine wall temperature comparison

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





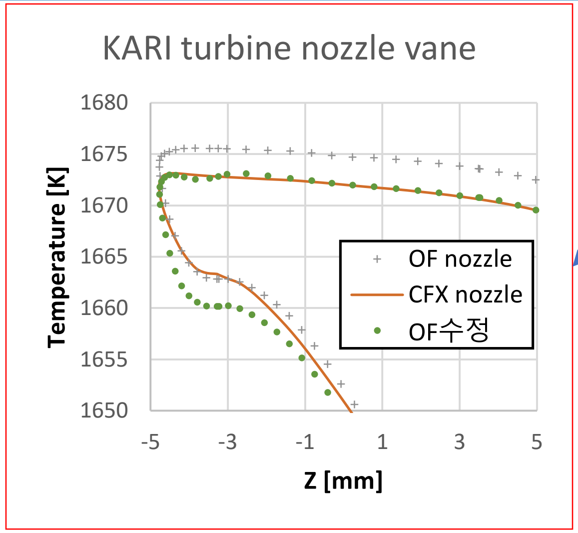
Nozzle wall temperature error

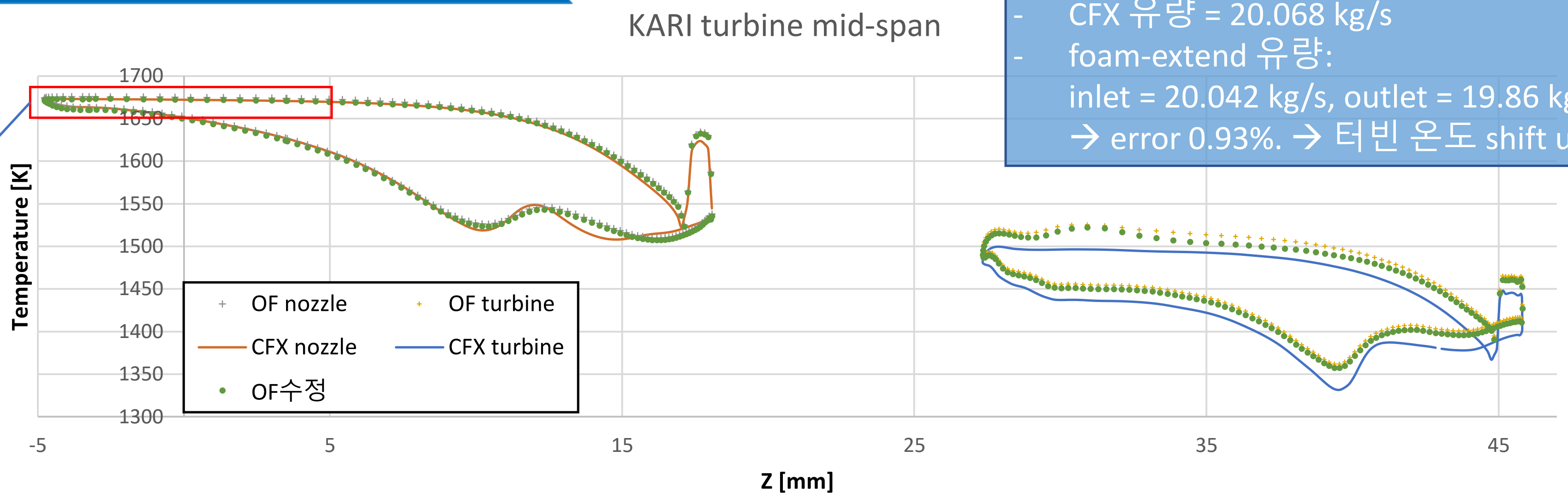


Rotor wall & outlet temperature error

3. Wall Temperature Issue

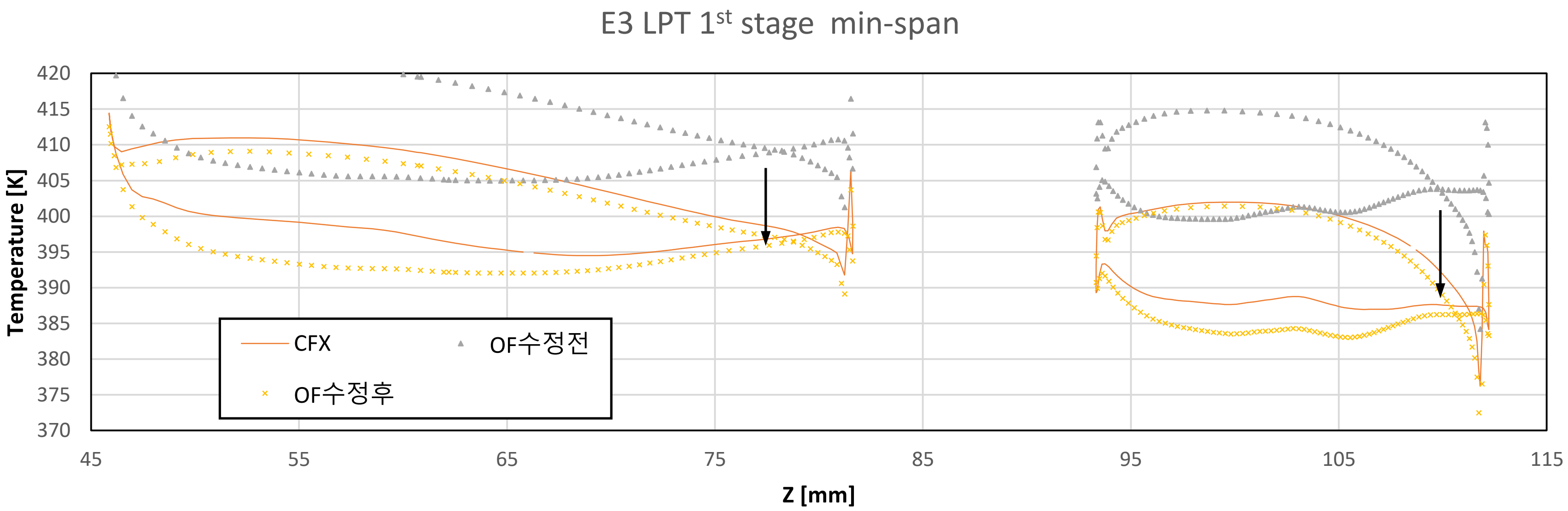
KARI & E3 LPT 1st stage mid-span vane temperature





- 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

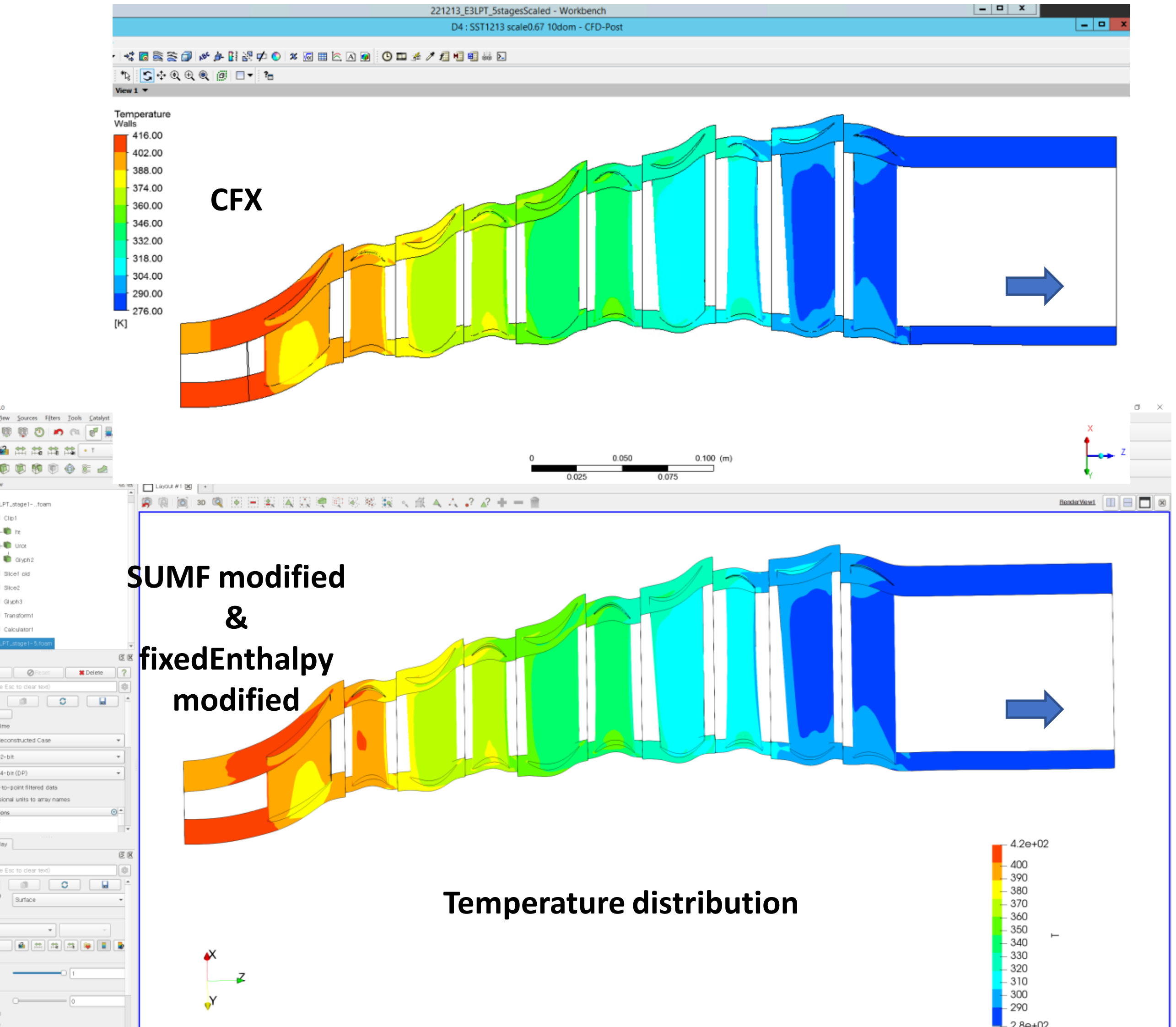
- *fixedEnthalpy* 수정 후, KARI 터빈 노즐 vane의 온도가 입구온도(1673 K)를 넘지 않고 CFX와 잘 일치.
- 단, 터빈에서 온도 Shift 발생 (유량 오차에서 기인한 것으로 추정)
- E3 LPT의 경우, 온도가 위로 Shift 됐던 것이 *fixedEnthalpy* 수정 이후 회복되어 CFX 결과와 일치.

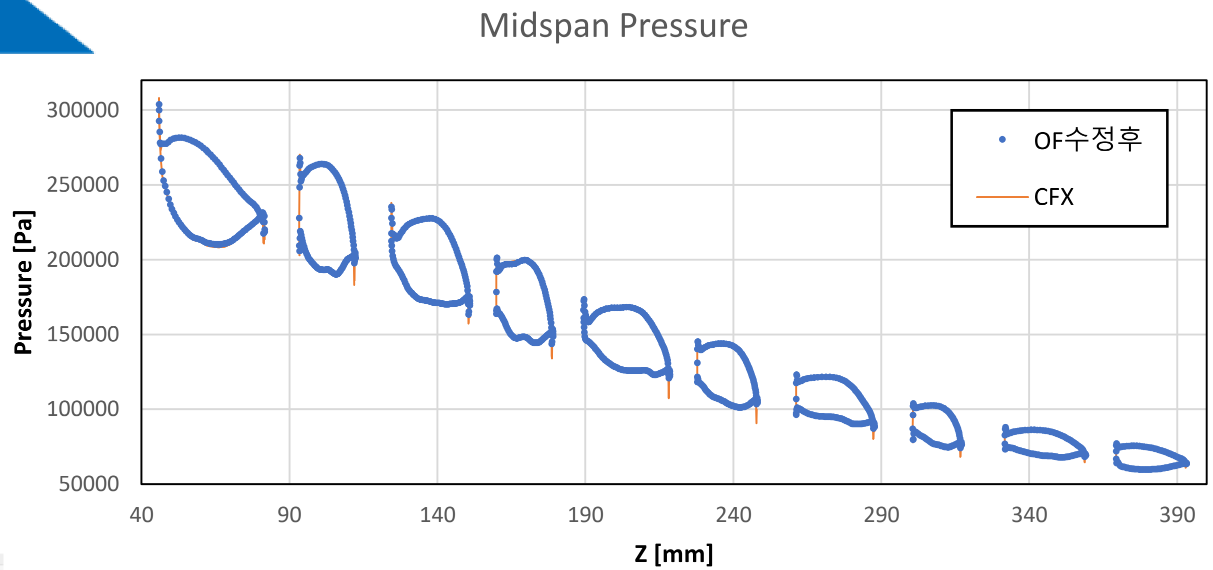


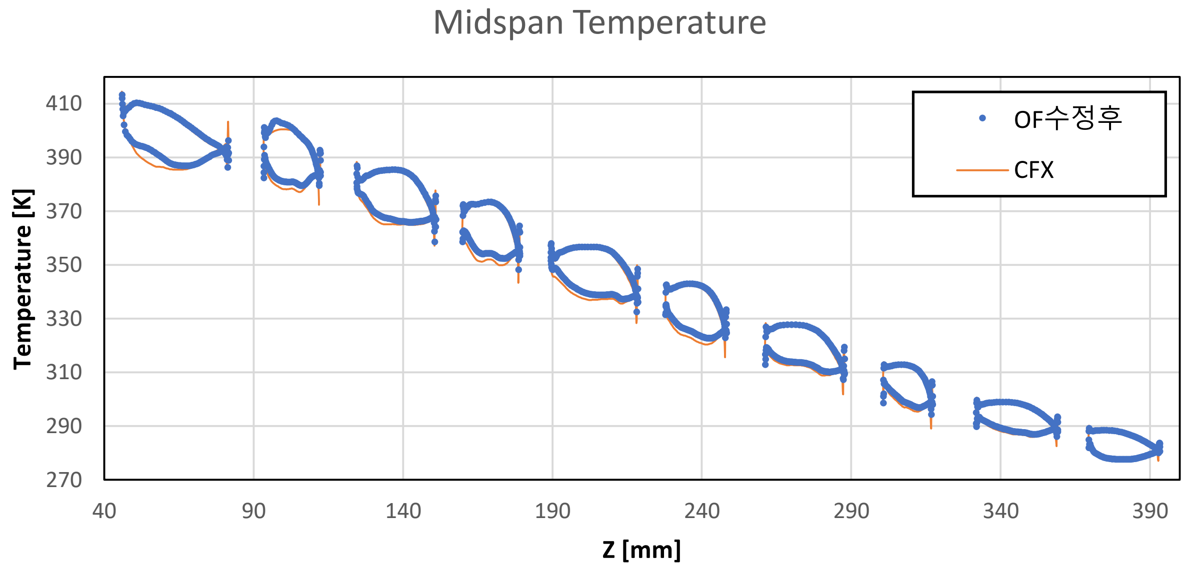
3. Wall Temperature Issue

KARI 한국항공우주연구원 Korea Aerospace Research Institute

(cf.) E3 LPT 5 stages calculation results



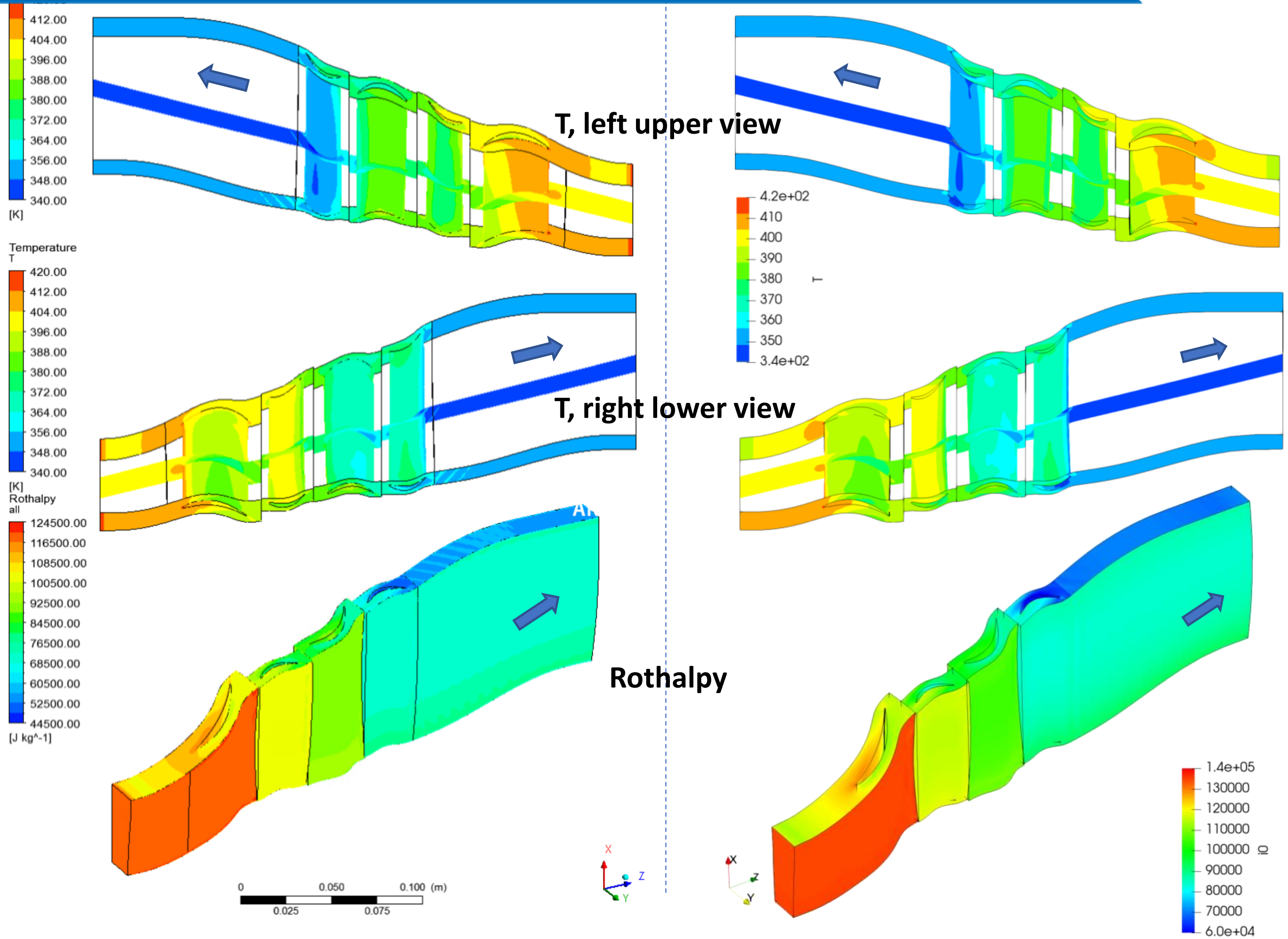




3. Wall Temperature Issue

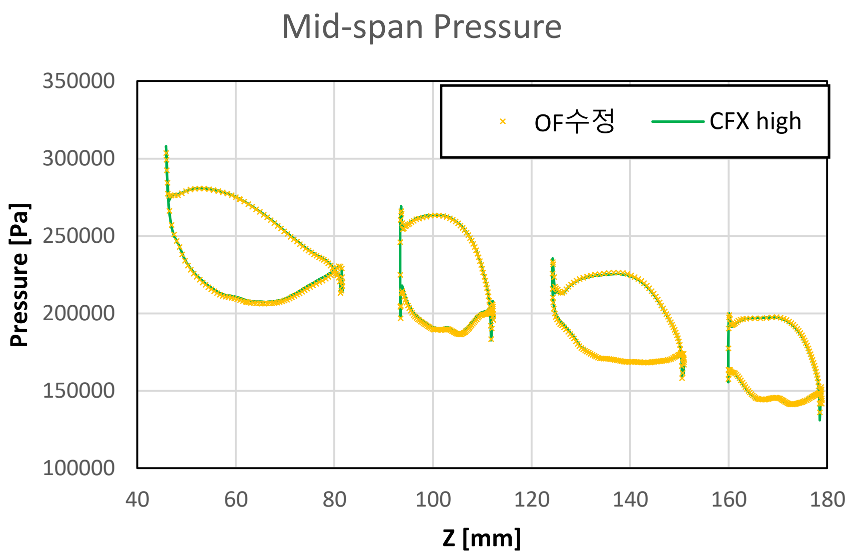
KARI 한국항공우주연구원 Korea Aerospace Research Institute

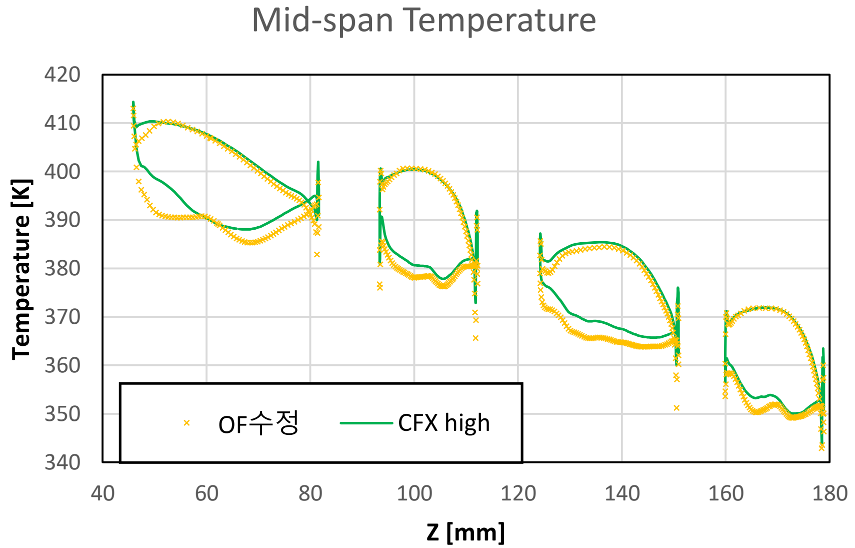
(cf.) E3 LPT 2 stages, calculation results (coarse gird)



CFX

SUMF modified, fixedEnthalpy modified



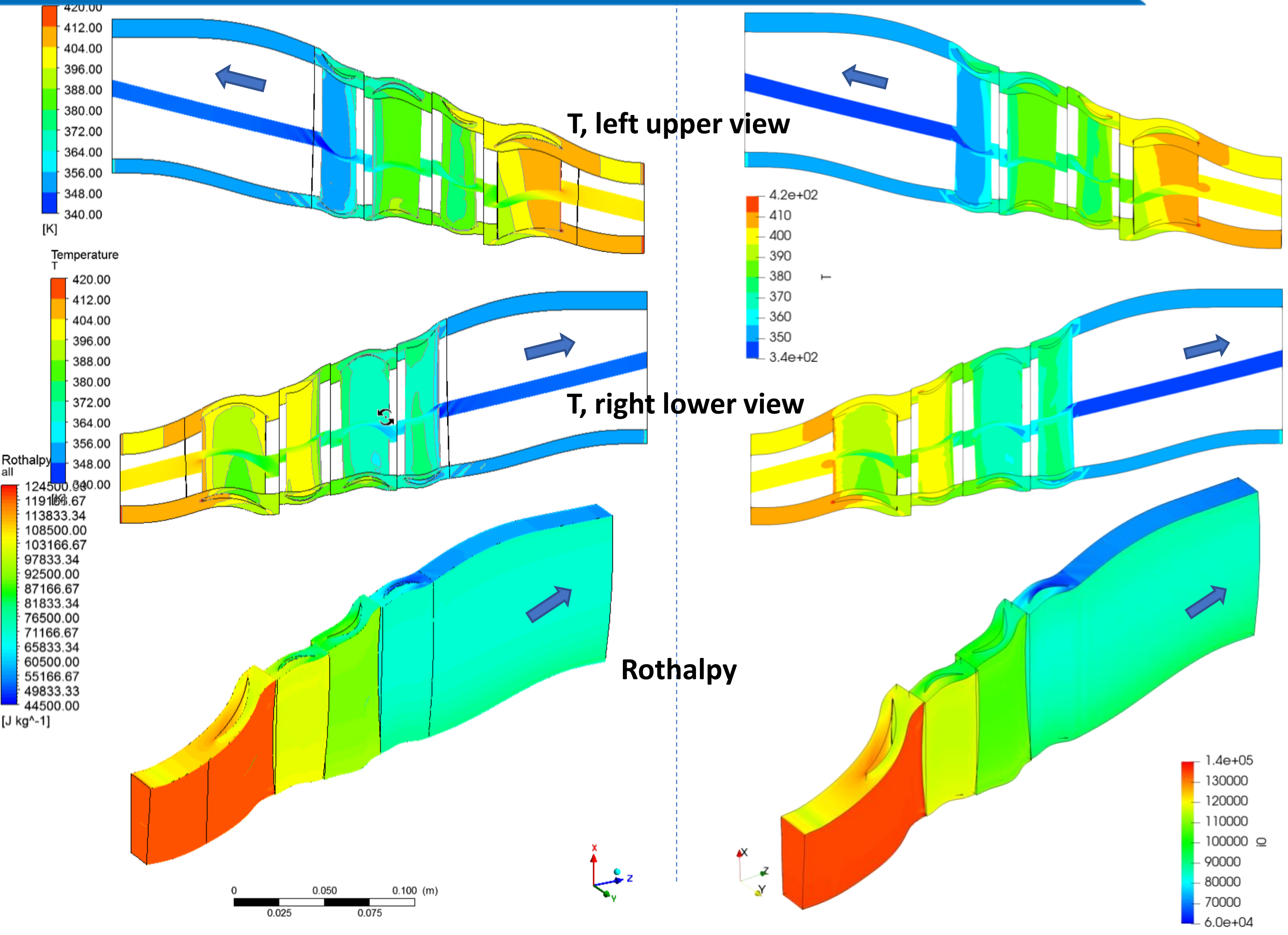


→ coarse 격자 해석 시, 일부 오차 존재

3. Wall Temperature Issue

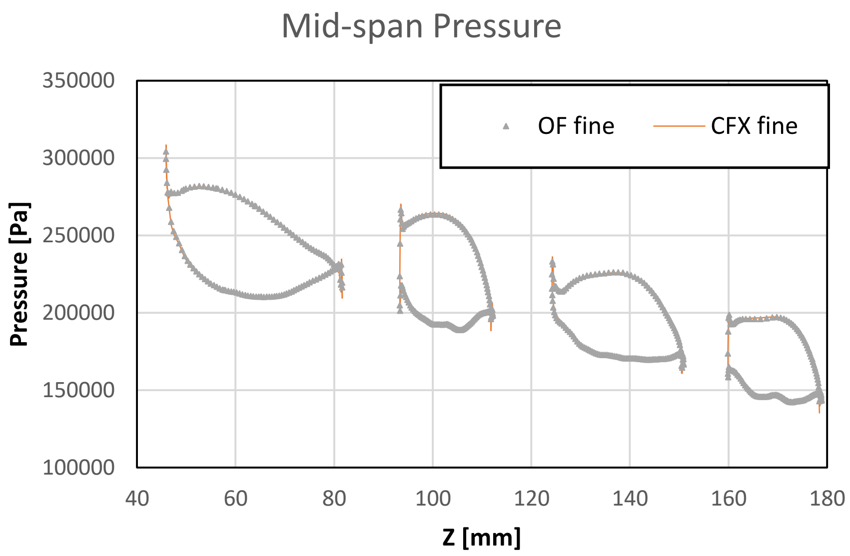
KARI 한국항공우주연구원 Korea Aerospace Research Institute

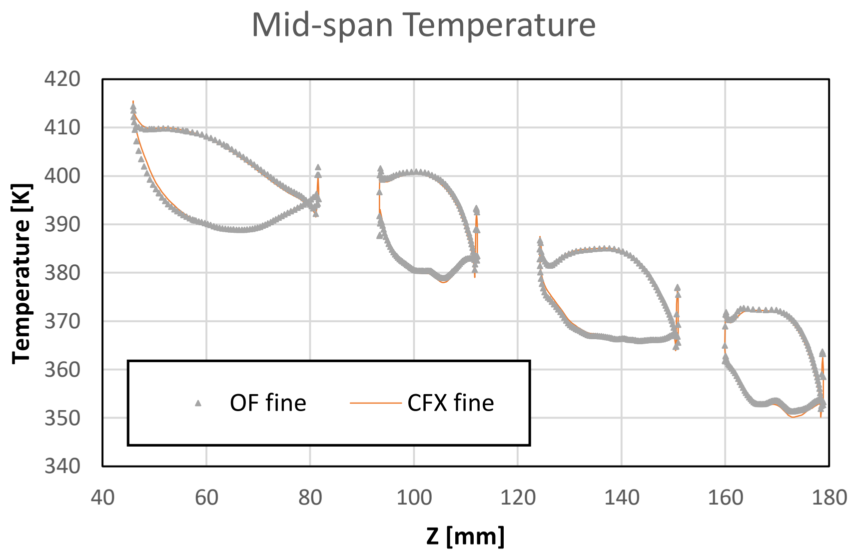
(cf.) E3 LPT 2 stages, calculation results (fine grid)



CFX

SUMF modified, fixedEnthalpy modified





→ Fine 격자 해석 시, 서로 잘 일치.

4. The Other Issues

Parallel computation / Turbomachinery SIG

4. The Other Issues 1

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
```

KARI 한국항공우주연구원 Korea Aerospace Research Institute

Parallel computation test: extend-4.1

C:₩Users₩hahacar₩Documents₩rtg₩연구정리2023₩material230915₩fe50 core 수 test 0927₩test₩mpi0926t5mFE41.sh - Notepad++

```
#/bin/sh
#PBS -V
#PBS -N E3_34x2c_t4m_Sp26
#PBS -q normal
#PBS -A openfoam
#PBS -l select=2:ncpus=34:mpiprocs=34:ompthreads=1
#PBS -l walltime=00:04:59

cd $PBS_O_WORKDIR

module purge
module load craype-mic-knl intel/18.0.3 impi/18.0.3
#module load craype-x86-skylake intel/18.0.3 impi/18.0.3
export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-4.1/etc/bashrc

#export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-5.0/etc/bashrc
#module load gcc/10.2.0

#### Do not edit #####
TOTAL_CPUS=$(wc -l $PBS_NODEFILE | awk '{print $1}')
#####################

mpirun -np 68 steadyUniversalMRFFoam230410_02 -parallel | tee log230926_solve2.logfile
#mpirun -np 68 steadyCompressibleMRFFoam -parallel | tee log230926_solve1.logfile
date
```

extend-4.1 Job script

C:₩Users₩hahacar₩Documents₩rtg₩연구정리2023₩material230915₩fe50 core 수 test 0927₩test₩E3_34x2c_t4m_Sp26.o13580259 - ...

```
/*---------------------------------------------------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | foam-extend: Open Source CFD                    |
|  \\    /   O peration     | Version:     4.1                                |
|   \\  /    A nd           | Web:         http://www.foam-extend.org         |
|    \\/     M anipulation  | For copyright notice see file Copyright         |
\*---------------------------------------------------------------------------*/
Build  : 4.1
Exec   : steadyUniversalMRFFoam230410_02 -parallel
Date   : Sep 26 2023
Time   : 18:36:06
Host   : "node4541"
PID    : 20310
CtrlDict : "/scratch/r811a02/e3_0829_stgl_2_68c_fe50/system/controlDict"
Case   : /scratch/r811a02/e3_0829_stgl_2_68c_fe50
nProcs : 68
Slaves :
67
(
"node4541.20311"
"node4541.20312"
"node4541.20313"
"node4541.20314"
"node4541.20315"
"node4541.20317"
"node4541.20318"
"node4541.20319"
"node4541.20321"
"node4541.20323"
"node4541.20324"
"node4541.20325"
"node4541.20326"
"node4541.20327"
"node4541.20328"
"node4541.20330"
"node4541.20331"
"node4541.20332"
"node4541.20333"
"node4541.20334"
"node4541.20335"
"node4541.20336"
"node4541.20337"
"node4541.20338"
"node4541.20339"
"node4541.20340"
"node4541.20341"
"node4541.20342"
"node4541.20343"
"node4541.20344"
"node4541.20345"
"node4541.20346"
"node4541.20347"
"node4541.20348"
"node4541.20349"
"node4541.20350"
"node4541.20351"
"node4541.20352"
"node4541.20353"
"node4541.20354"
```

Normal text file | length : 7,549 lines : 190 | Ln : 12 Col : 19 Sel : 8 | 1 | Unix (LF) | UTF-8 | INS

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₩hahacar₩Documents₩rtg₩연구정리2023₩material230915₩fe50 core 수 test 0927₩test₩mpi0926t5m.sh - Notepad++

```
#!/bin/sh
#PBS -V
#PBS -N E3_34x2c_t4m_Sp26
#PBS -q normal
#PBS -A openfoam
#PBS -l select=2:ncpus=34:mpiprocs=34:ompthreads=1
#PBS -l walltime=00:04:59

cd $PBS_O_WORKDIR

module purge
#module load craype-mic-knl intel/18.0.3 impi/18.0.3
#module load craype-x86-skylake intel/18.0.3 impi/18.0.3
#export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-4.1/etc/bashrc

export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-5.0/etc/bashrc
module load gcc/10.2.0

#### Do not edit #####
TOTAL_CPUS=$(wc -l $PBS_NODEFILE | awk '{print $1}')
#####################

#mpirun -np 68 steadyUniversalMRFFoam230410_02 -parallel | tee log230901_solve.logfile
mpirun -np 68 steadyCompressibleMRFFoam -parallel | tee log230926_solve1.logfile
date
```

extend-5.0 Job script

C:₩Users₩hahacar₩Documents₩rtg₩연구정리2023₩material230915₩fe50 core 수 test 0927₩test₩E3_34x2c_t4m_Sp26.o13579644 - ...

```
/*---------------------------------------------------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | foam-extend: Open Source CFD                    |
|  \\    /   O peration     | Version:     5.0                                |
|   \\  /    A nd           | Web:         http://www.foam-extend.org         |
|    \\/     M anipulation  | For copyright notice see file Copyright         |
\*---------------------------------------------------------------------------*/
Build  : 5.0-287705b4a589
Exec   : steadyCompressibleMRFFoam -parallel
Date   : Sep 26 2023
Time   : 17:44:35
Host   : "node5712"
PID    : 55604
CtrlDict : "/scratch/r811a02/e3_0829_stgl_2_68c_fe50/system/controlDict"
Case   : /scratch/r811a02/e3_0829_stgl_2_68c_fe50
nProcs : 68
Slaves :
67
(
"node5712.55603"
"node5712.55606"
"node5712.55605"
"node5712.55607"
"node5712.55608"
"node5712.55610"
"node5712.55609"
"node5712.55611"
"node5712.55612"
"node5712.55614"
"node5712.55613"
"node5712.55615"
"node5712.55616"
"node5712.55618"
"node5712.55617"
"node5712.55619"
"node5712.55620"
"node5712.55622"
"node5712.55621"
"node5712.55623"
"node5712.55624"
"node5712.55626"
"node5712.55625"
"node5712.55627"
"node5712.55628"
"node5712.55630"
"node5712.55629"
"node5712.55632"
"node5712.55631"
"node5712.55634"
"node5712.55633"
"node5712.55636"
"node5712.55635"
"node6287.19220"
"node6287.19221"
"node6287.19223"
"node6287.19222"
"node6287.19224"
"node6287.19225"
"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 나눠서 계산이 걸림. → 노드 병렬 연산 가능.

Log

4. The Other Issues 1

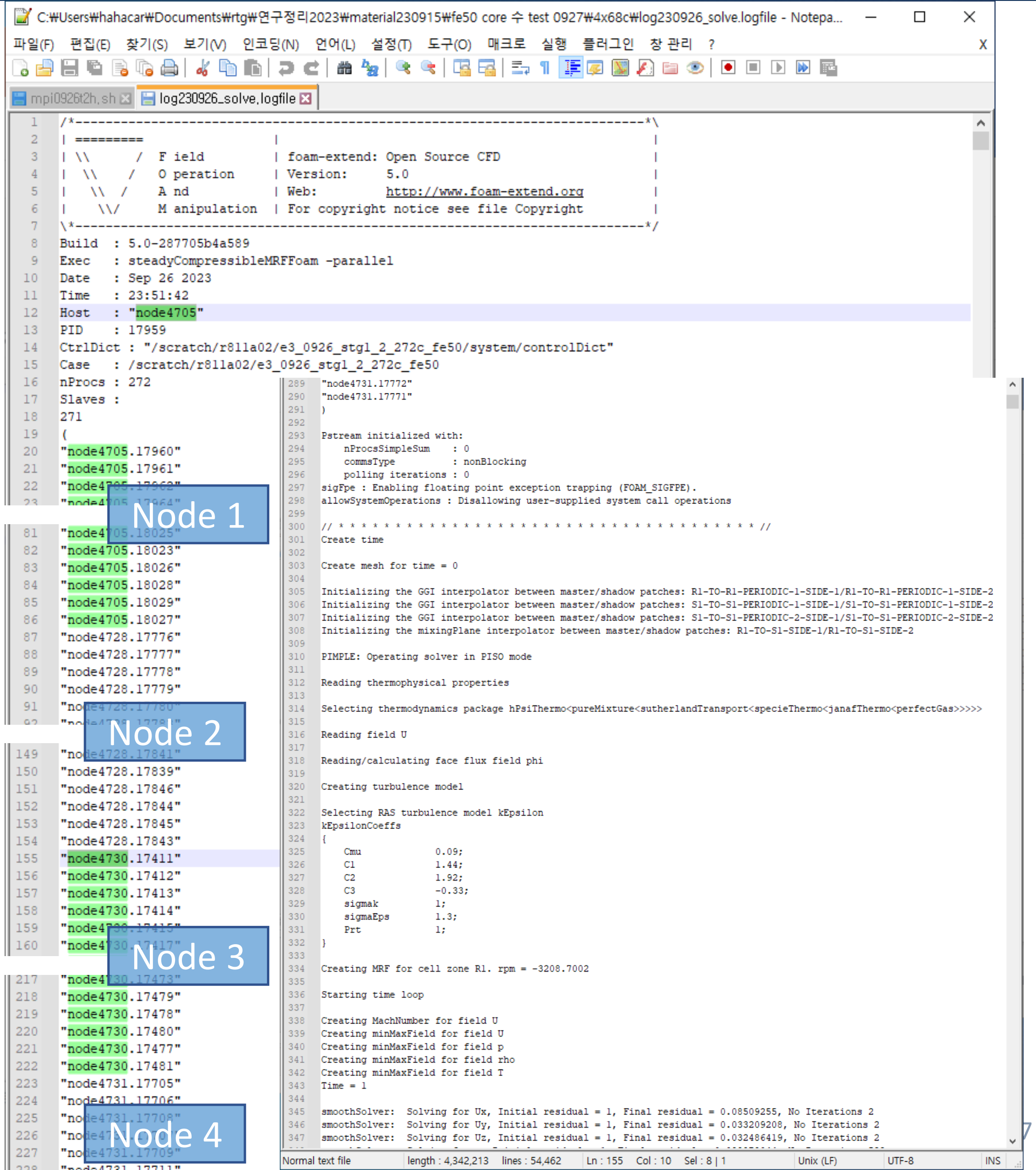
Parallel computation test: extend-5.0 (2)

C:\Users\hahacar\Documents\rtg\연구정리2023\material230915\fe50 core 수 test 0927\4x68c\mpi0926t2h.sh - Notepad++

```
#/bin/sh
#PBS -V
#PBS -N E3_68x4c_t2h_Sp26
#PBS -q normal
#PBS -A openfoam
#PBS -l select=4:ncpus=68:mpiprocs=68:ompthreads=1
#PBS -l walltime=01:59:59

cd $PBS_O_WORKDIR

module purge
#module load craype-mic-knl intel/18.0.3 impi/18.0.3
#module load craype-x86-skylake intel/18.0.3 impi/18.0.3
#export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-4.1/etc/bashrc

export FOAM_INST_DIR=/apps/applications/foam; . $FOAM_INST_DIR/foam-extend-5.0/etc/bashrc
module load gcc/10.2.0

#### Do not edit #####
TOTAL_CPUS=$(wc -l $PBS_NODEFILE | awk '{print $1}')
#####################

#mpirun -np 68 steadyUniversalMRFFoam230410_02 -parallel | tee log230901_solve.logfile
mpirun -np 272 steadyCompressibleMRFFoam -parallel | tee log230926_solve.logfile
date
```

extend-5.0 Job script

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



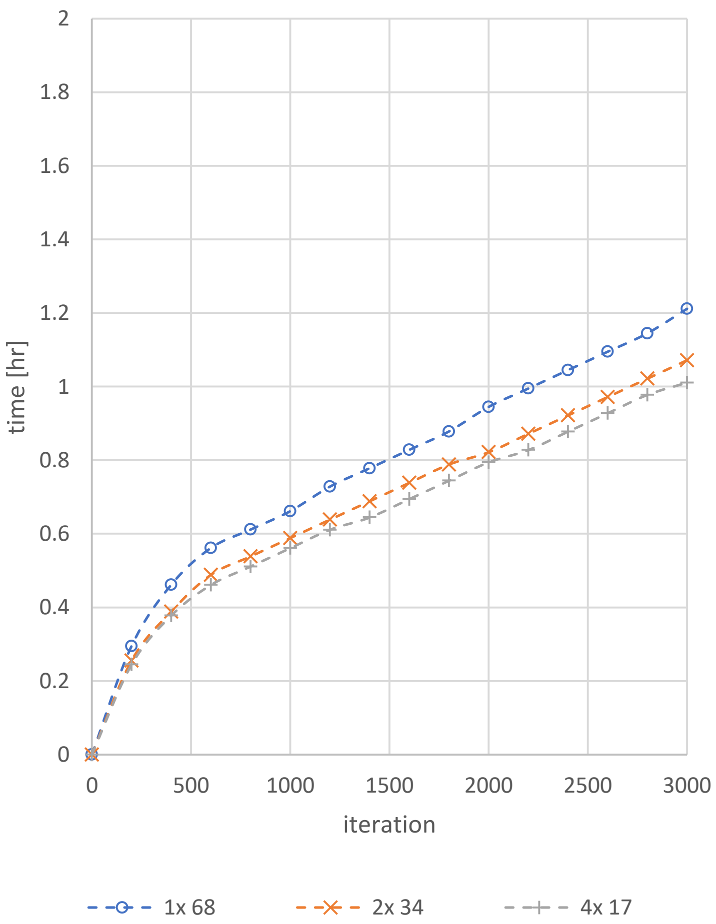
Log

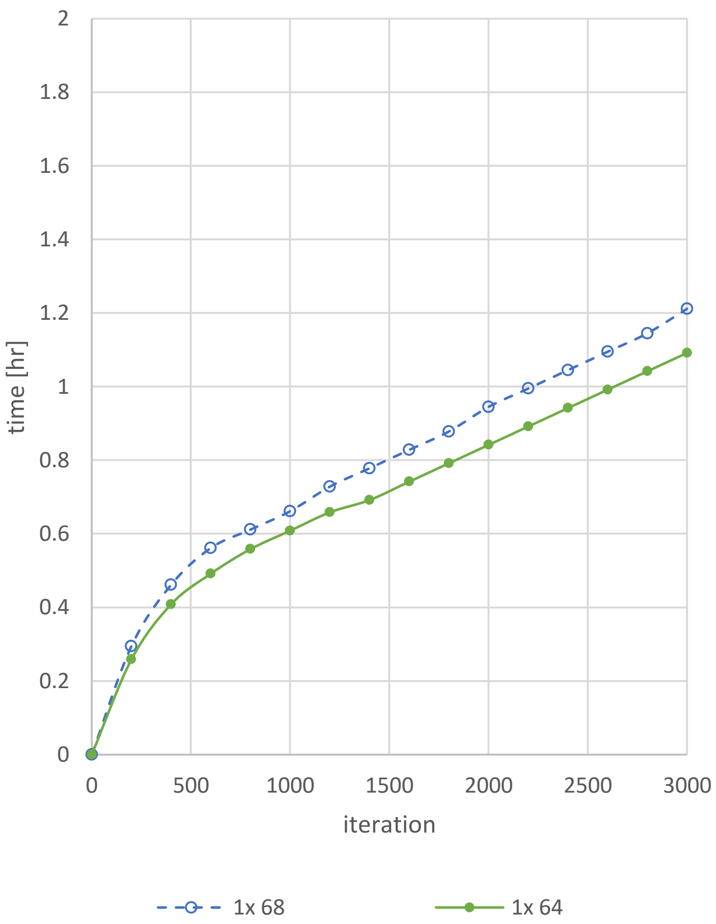
4. The Other Issues 1

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.



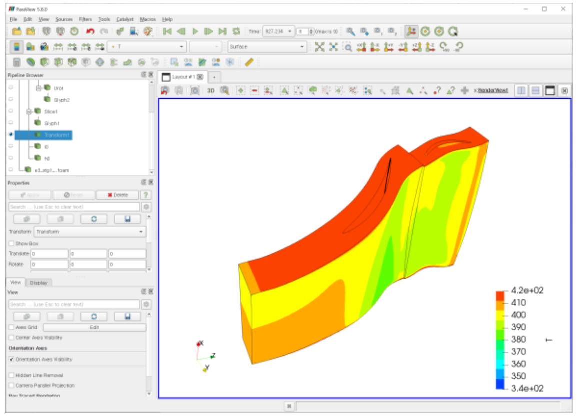


Computation time

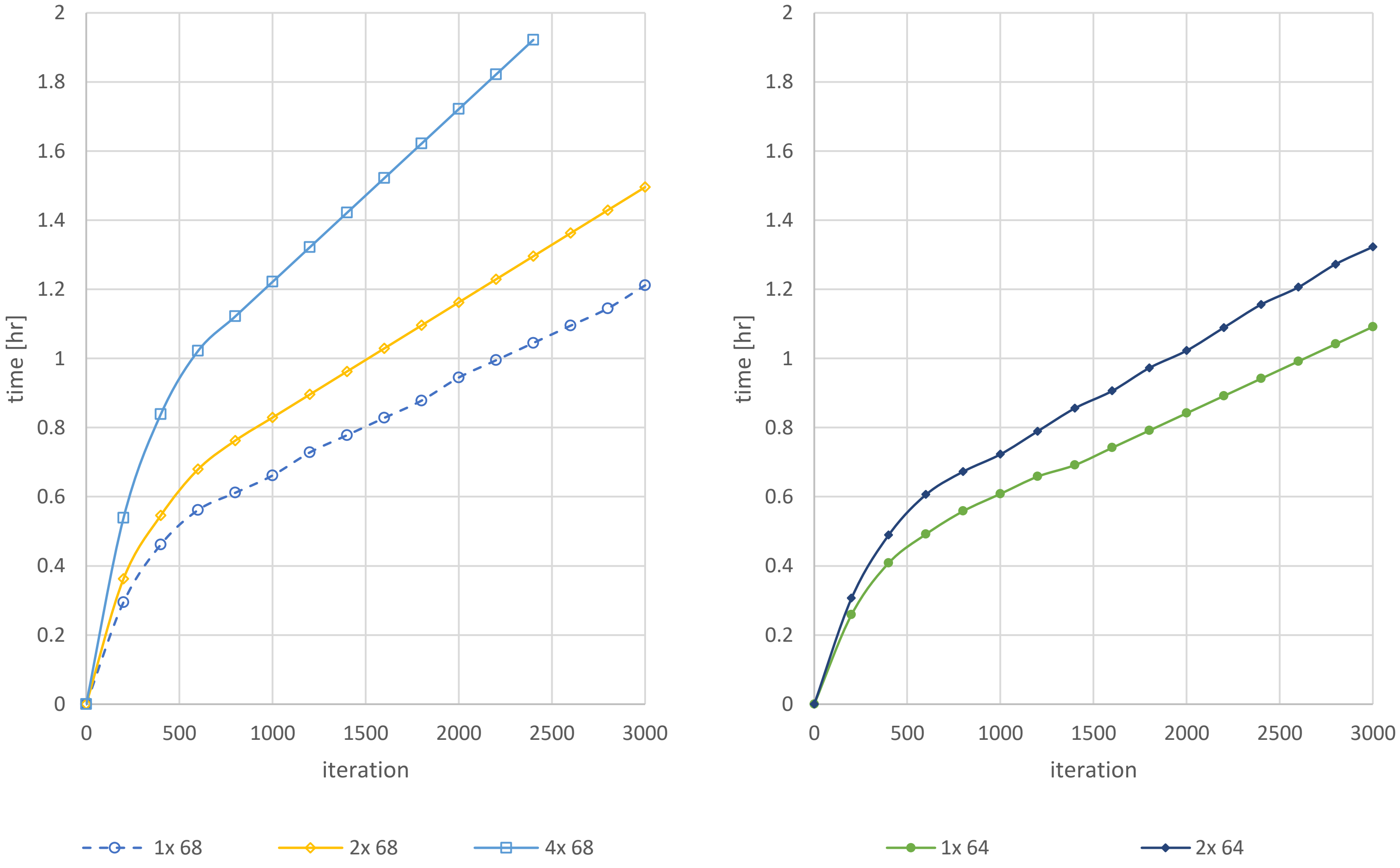
Node parallel computation test (2)

- 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.**



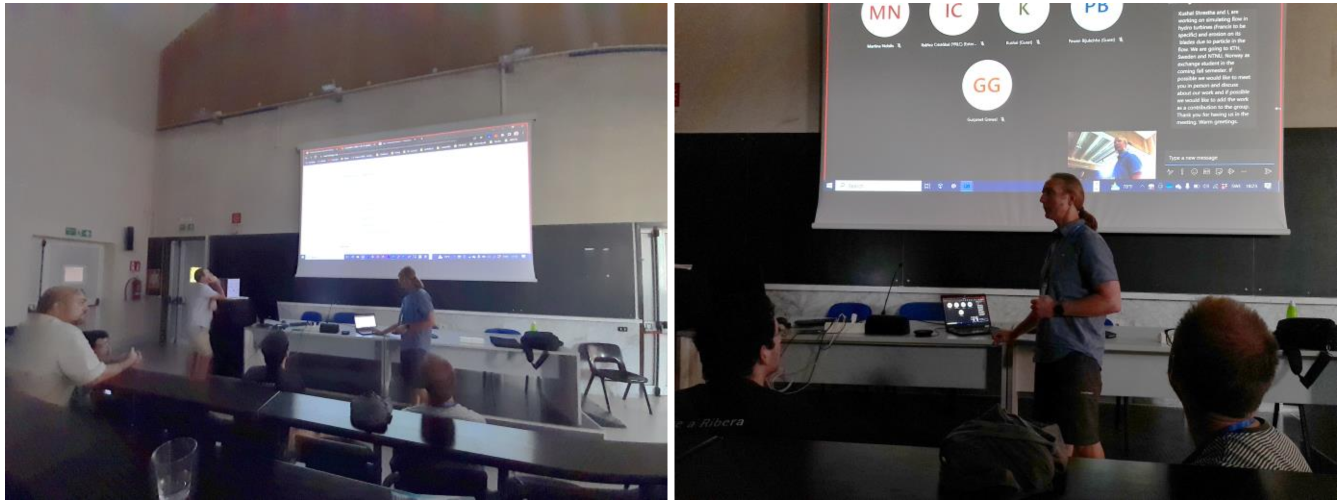
extend-5.0, steadyCompressibleMRFFoam



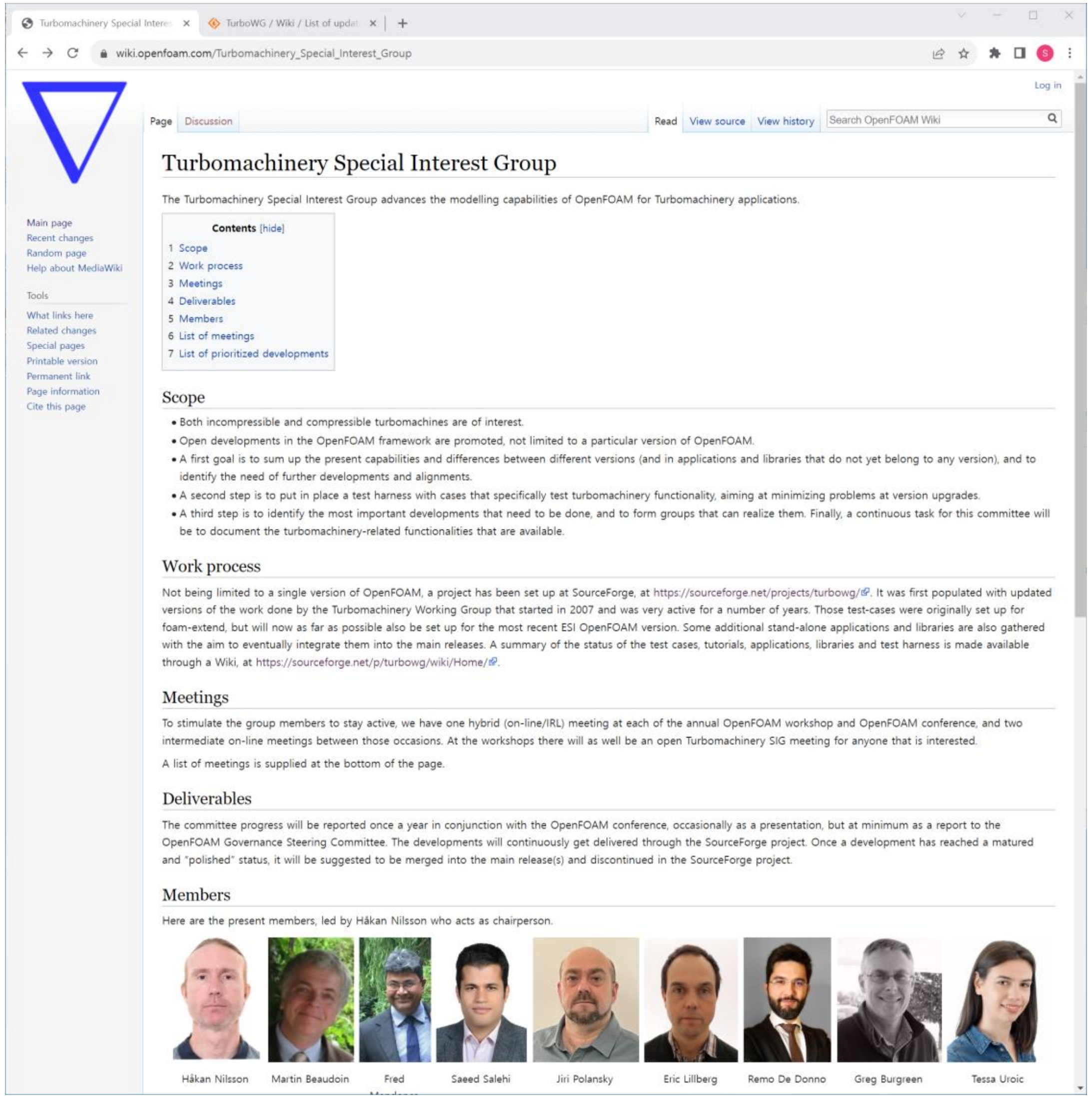
Computation time

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 10 명 이상 참여
- Turbomachinery 소개: https://wiki.openfoam.com/Turbomachinery_Special_Interest_Group



Turbomachinery SIG meeting scene



Turbomachinery Special Interest Group

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

Contents [hide]

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

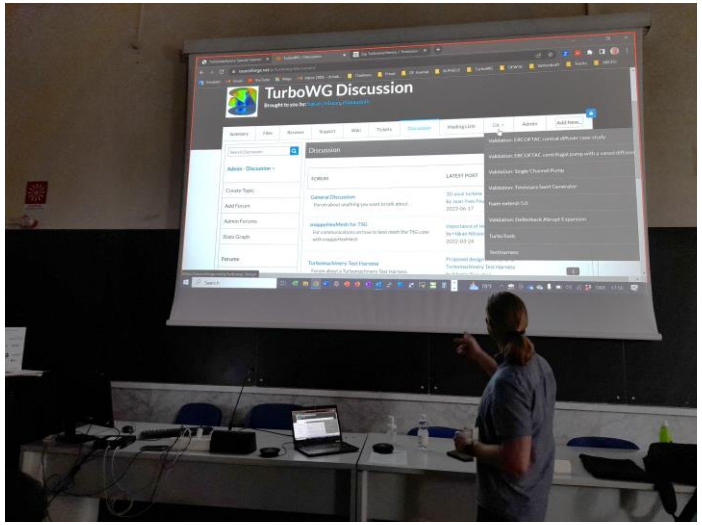
Here are the present members, led by Håkan Nilsson who acts as chairperson.

Håkan Nilsson, Martin Beaudoin, Fred Mendonça, Saeed Salehi, Jiri Polansky, Eric Lillberg, Remo De Donno, Greg Burgreen, Tessa Uroic

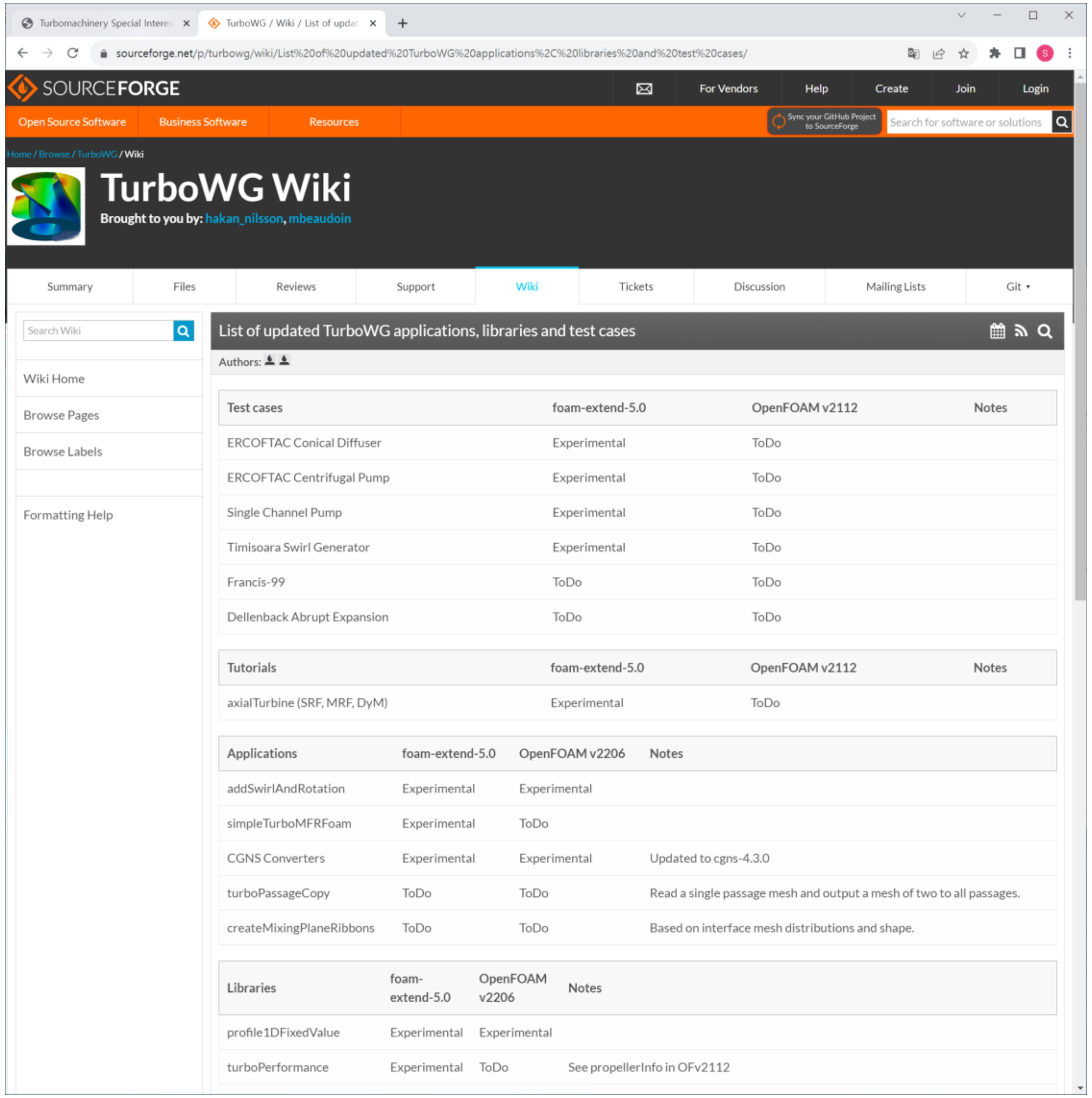
KARI 한국항공우주연구원 Korea Aerospace Research Institute

18th OpenFOAM Workshop – Turbo SIG (2)

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



Turbomachinery working group archive



- 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

4. Summary & Conclusion

Summary

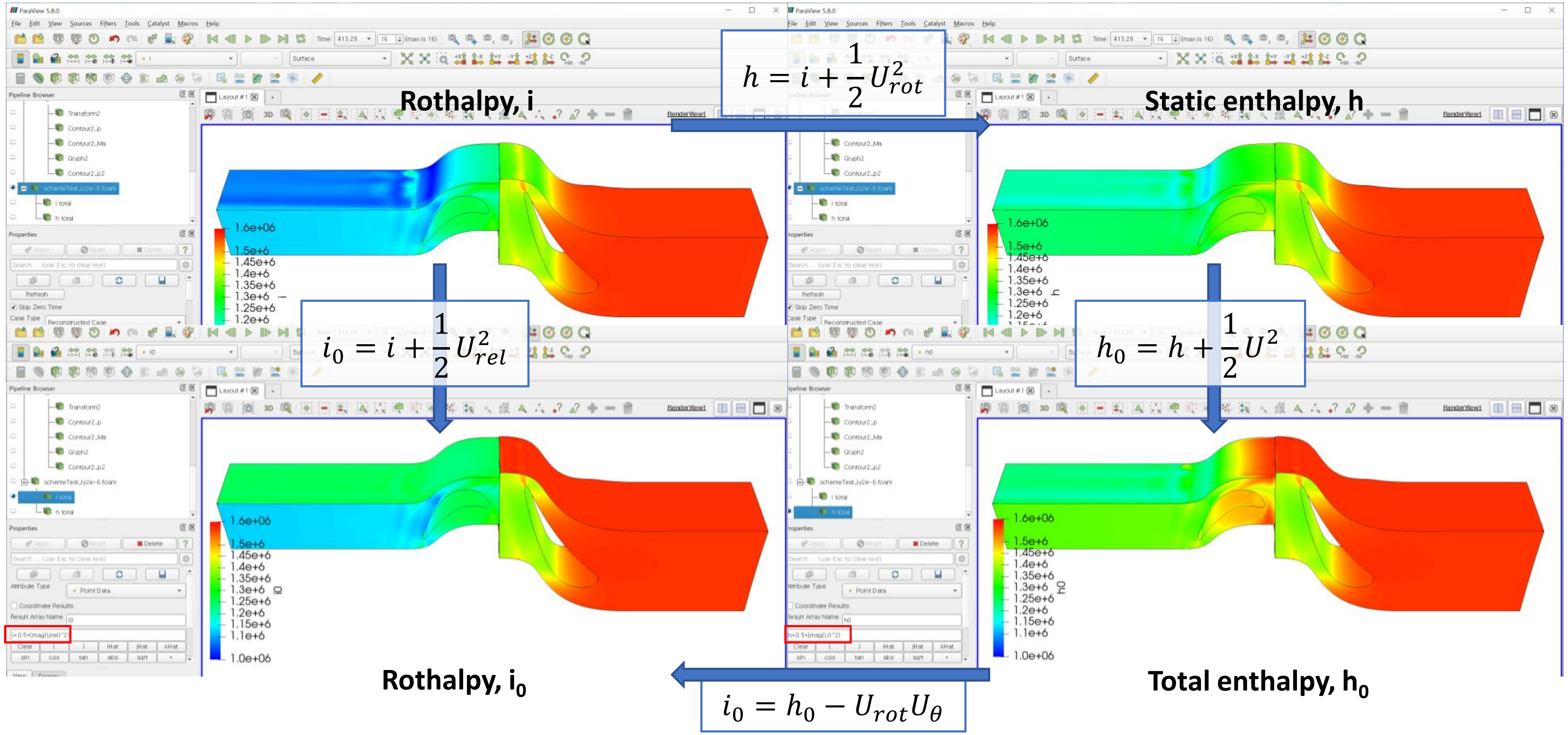
- 가스터빈 터보유체기계의 고정밀도 해석을 위해 오픈 소스 CFD 코드 중 하나인 OpenFOAM을 사용한 대규모 병렬 계산이 요구됨.
- 필요한 Interface 유틸리티가 사용 가능한 압축성 MRF 솔버로는 → foam-extend 버전의 steadyUniversalMRFFoam
- *mixingPlane* 온도 불연속 이슈는 정의 안 된 변수와 Rothalpy jump의 스칼라 곱으로 인해 발생.
- foam-extend 버전의 압축성 솔버의 에너지 방정식에선 운동에너지 항이 결여됨. → 잘못된 온도장 산출.
- 병렬 연산 이슈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

Conclusion (2)

- 새로 정의된 Rothalpy에 따른 i, i_0, h, h_0의 관계



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

감사합니다.

강승환KANG, Sueng-Hwan (hahacar@kari.re.kr)

Appendix

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 μ, thermal conductivity κ and thermal diffusivity α (for internal energy and enthalpy equations). The current transport models are as follows:

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

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

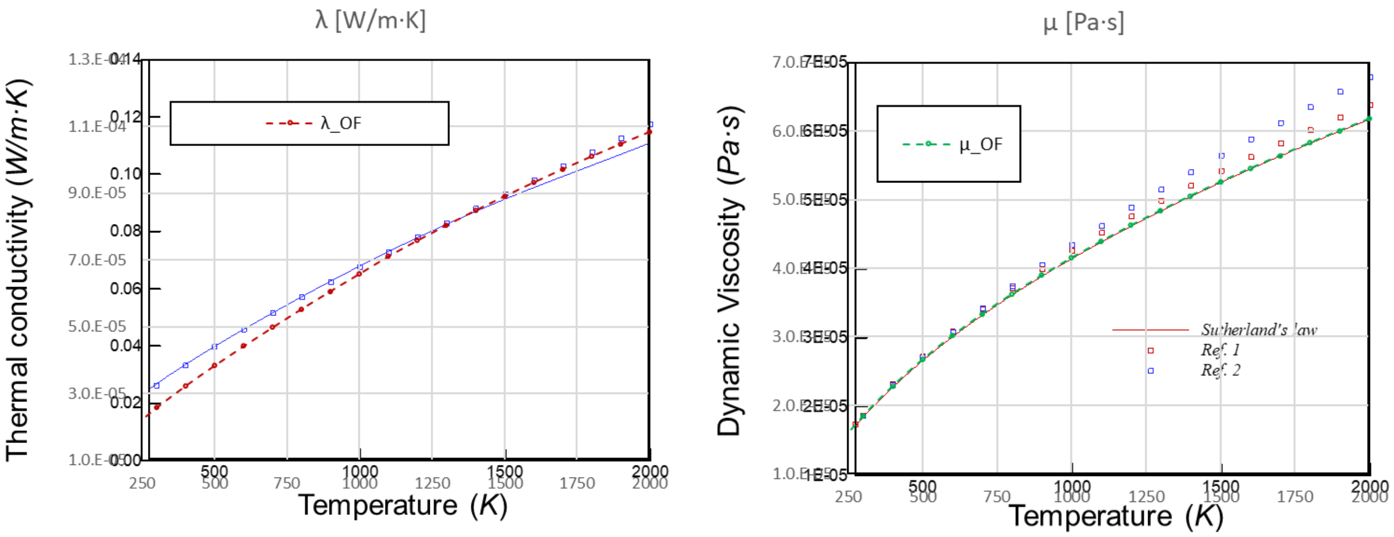
$$\mu = \frac{A_s\sqrt{T}}{1 + T_s/T}. \quad (7.1)$$

(OpenFOAM User Guide ver.7)

* New coefficients

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

- As = 1.4579e-06;
- Ts = 110.4;
- Pr = 0.72;



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

Gas viscosity data from

1. http://www.lmnoeng.com/Flow/GasViscosity.htm
2. "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

Appendix

KARI 한국항공우주연구원 Korea Aerospace Research Institute

Thermophysical properties (2)

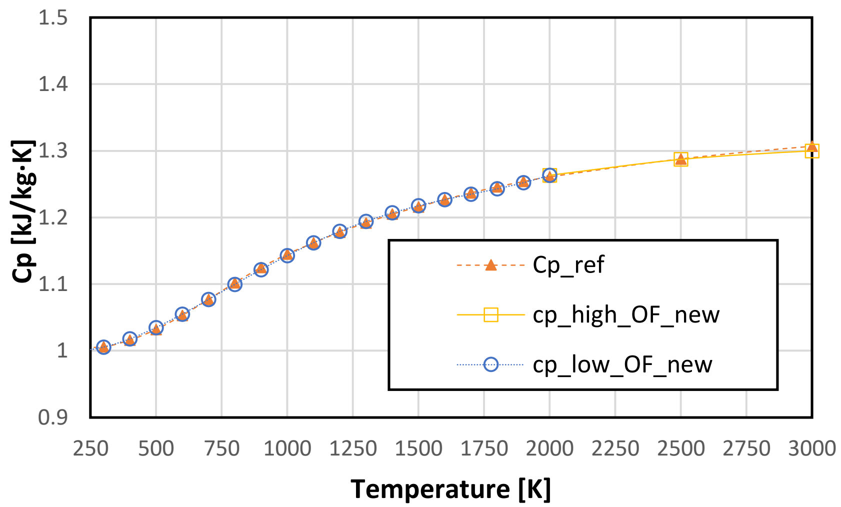
- 온도의 함수인 Cp 를 사용하기 위해 → JANAF option. 아래와 같이 설정:
  `thermo                    janaf`
- JANAF function

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

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

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

- 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:

$$c_p = R((((a_4 T + a_3)T + a_2)T + a_1)T + a_0). \quad (7.4)$$

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	$a_0 \ldots a_4$	highCpCoeffs (a0 a1 a2 a3 a4...
High temperature enthalpy offset	a_5	a5...
High temperature entropy offset	a_6	a6)
Low temperature coefficients	$a_0 \ldots a_4$	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.

(OpenFOAM User Guide ver.7)

Thermophysical properties (3)

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1906                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
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 ield         | foam-extend: Open Source CFD                    |
|  \\    /   O peration     | Version:     4.1                                |
|   \\  /    A nd           | Web:         http://www.foam-extend.org         |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      thermophysicalProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

Pr              Pr [ 0 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