OpenFOAM+のCo-simulation機能とFMU(Functional Mock-up Unit)の試作

第58回オープンCAE勉強会@関東（流体など）

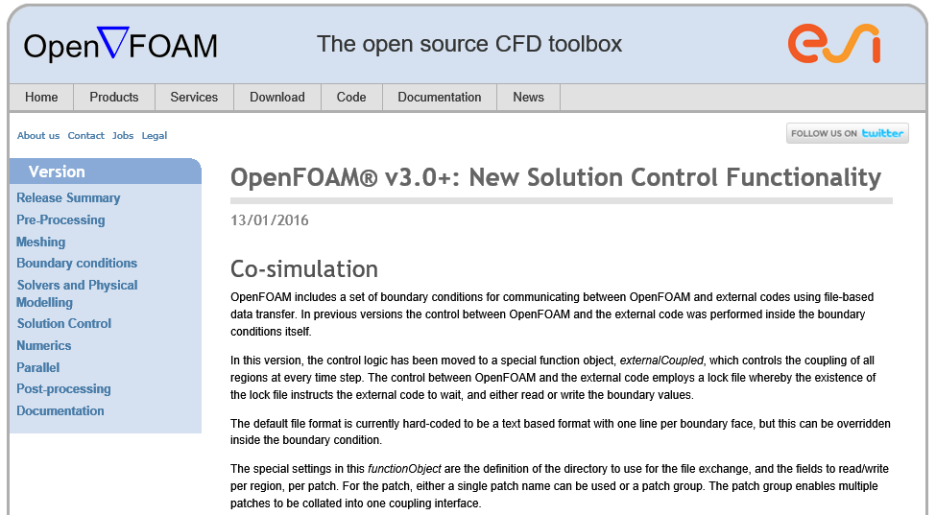
finback

OpenFOAM+のCo-simulation機能

- 概要
- Co-simulationのための設定方法
- 外部プログラムの動作
- データ交換のフォーマット

OpenFOAM+ の Co-simulation 機能の概要

http://www.openfoam.com/version-v3.0+/solution-control.php 13/01/2016



- 非定常シミュレーション中にファイルベースのデータ転送を使用してOpenFOAMと外部コードの間で通信ができる。
- function object externalCoupled に以下を定義する。
  - データ交換に使用するディレクトリ
  - リージョン毎またはパッチ、パッチグループ毎に読み書きするフィールド
- ロックファイルが存在する間、外部プログラムが停止（一時中断）して、OpenFOAMが境界値の読み書きを行なう。
- デフォルトのデータフォーマットは、境界面１つに対して１行の固定形式（ハードコーデッド）である。

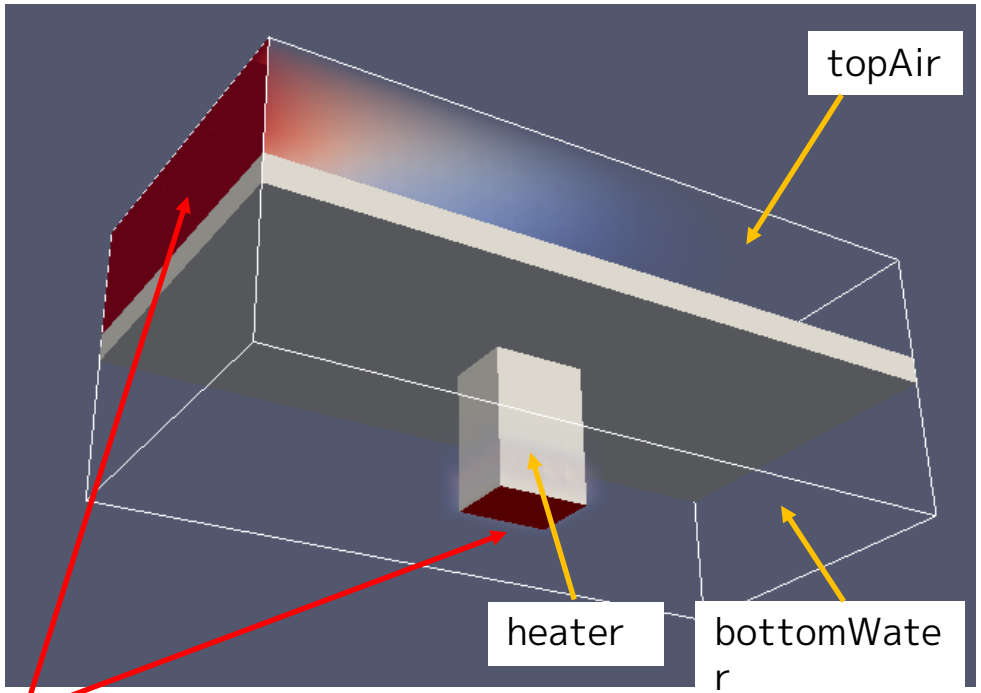
チュートリアル

$FOAM_TUTORIALS/heatTransfer/chtMultiRegionFoam/externalCoupledMultiRegionHeater

OpenFOAM v1606＋にも同じチュートリアルがあるので、今回はv1606+を調べる。

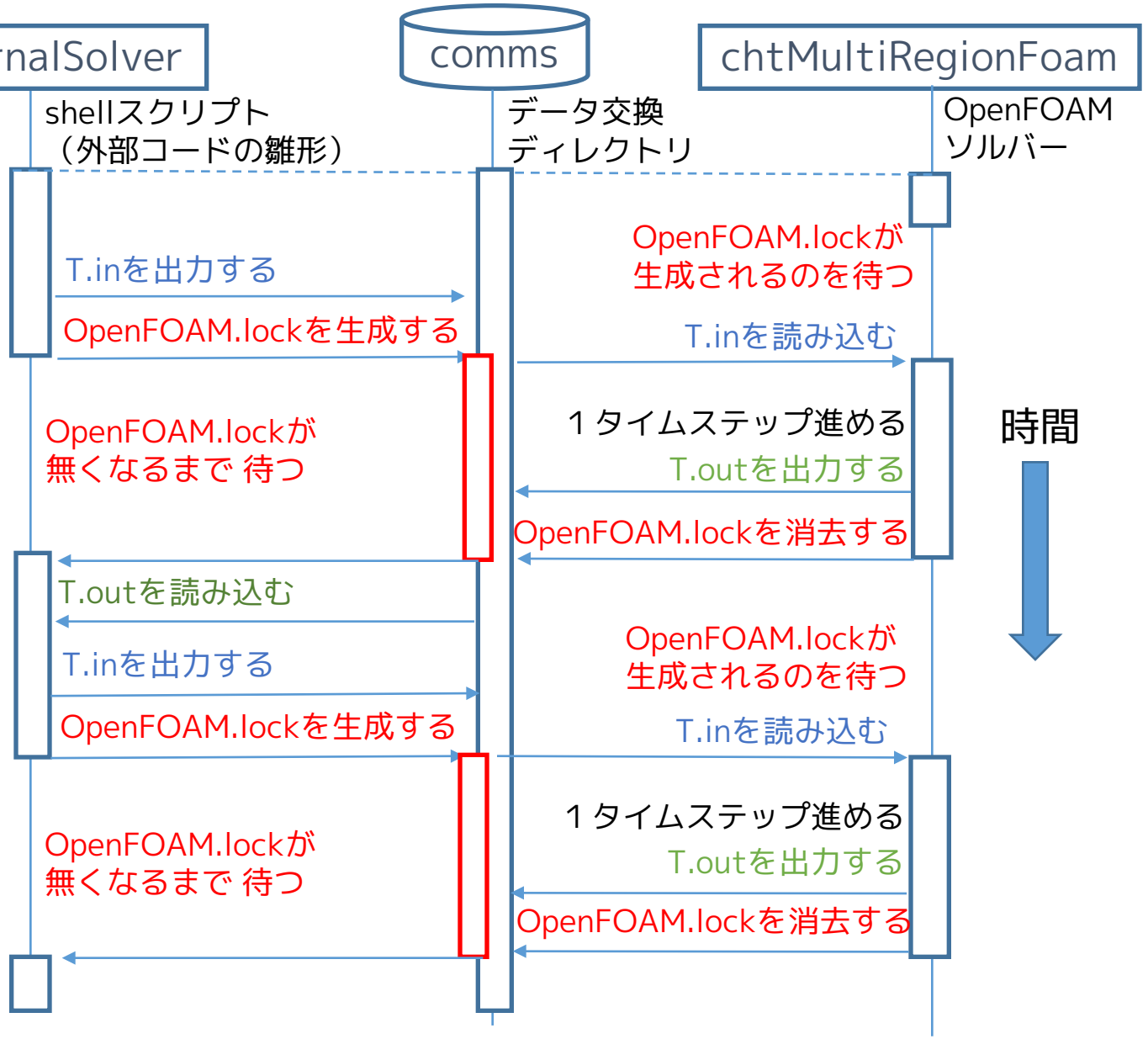
チュートリアル externalCoupledMultiRegionHeater の概要

非定常シミュレーション中に
externalSolver（外部ソルバー）と
chtMultiRegionFoamの間で境界条件を交換する。



coupleGroup（topAirの入口面とheaterの底面）の温度 T にexternalSolver (shellスクリプト)で計算した値を設定する。

T.in　：externalSolverが出力する温度データファイル
T.out：chtMultiRegionFoamが出力する温度データファイル
OpenFOAM.lock：ロックファイル



Co-simulationの流れ

Co-simulation のための設定方法

externalCoupledMultiRegionHeater のモデルは、

$FOAM_TUTORIALS/heatTransfer/chtMultiRegionFoam/multiRegionHeater

とほぼ同じである。違いを調べることによって、Co-simulation のための設定方法がわかる。

- system/controlDict
- system/topAir/changeDictionaryDict
- system/bottomWater/changeDictionaryDict
- Allrun.pre
- Allrunのソルバー実行部分
- externalSolver（外部コードの雛形）

設定が必要なファイル

Co-Simulationの実行手順

外部プログラムの動き

system/controlDict – function object externalCoupled の設定

```
*---------------------------------*- C++ -*----------------------------------*¥
| =========                 |                                                 |
| ¥¥      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  ¥¥    /   O peration     | Version:  v1606+                                |
|   ¥¥  /    A nd           | Web:      www.OpenFOAM.com                      |
|    ¥¥/     M anipulation  |                                                 |
¥*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// Library defines new boundary conditions
libs            ("libOpenFOAM.so" "libjobControl.so");
application     chtMultiRegionFoam;
startFrom       startTime;
startTime       0;
stopAt          endTime;
endTime         1;
deltaT          0.001;
writeControl    adjustableRunTime;
writeInterval   0.1;
purgeWrite      0;
writeFormat     ascii;
writePrecision  8;
writeCompression off;
timeFormat      general;
timePrecision   6;
runTimeModifiable yes;
maxCo           0.6;
// Maximum diffusion number
maxDi           10.0;
adjustTimeStep  yes;
```

```
functions
{
    externalCoupled
    {
        // Where to load it from (if not already in solver)
        functionObjectLibs ("libjobControl.so");

        type            externalCoupled;

        // Directory to use for communication
        commsDir        "${FOAM_CASE}/comms";

        // Does external process start first
        initByExternal  true;

        // Additional output
        log             true;

        regions
        {
            // Region name (wildcards allowed)
            "(topAir|heater)"
            {
                // In topAir adjust the minX patch (fixedValue)

                // Patch or patchGroup
                coupleGroup
                {
                    // Fields to output in commsDir
                    writeFields (T);
                    // Fields to read from commsDir
                    readFields  (T);
                }
            }
        }
    }
}
```

function object の externalCoupledの設定

データ交換用ディレクトリ

初期値も外部から設定する

ログ出力する

coupleGroup という patchGroup で で温度のフィールドデータを交換する。

system/topAir/changeDictionaryDict (一部分)

```
/*--------------------------------*- C++ -*----------------------------------*¥
| =========                 |                                                 |
| ¥¥    /  F ield           | OpenFOAM: The Open Source CFD Toolbox           |
|  ¥¥   /   O peration      | Version:  v1606+                                |
|   ¥¥ /    A nd            | Web:      www.OpenFOAM.com                      |
|    ¥¥/    M anipulation   |                                                 |
¥*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      changeDictionaryDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dictionaryReplacement
{
    boundary
    {
        minX
        {
            inGroups        (coupleGroup);
        }
    }

    U
    {
        internalField   uniform (0.1 0 0);
```

topAirの入り口を
coupleGroup に入れる

system/bottomWater/changeDictionaryDict(一部分)

```
*--------------------------------*- C++ -*----------------------------------*¥
| =========                 |                                                 |
| ¥¥    /  F ield           | OpenFOAM: The Open Source CFD Toolbox           |
|  ¥¥   /   O peration      | Version:  v1606+                                |
|   ¥¥ /    A nd            | Web:      www.OpenFOAM.com                      |
|    ¥¥/    M anipulation   |                                                 |
¥*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      changeDictionaryDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dictionaryReplacement
{
    boundary
    {
        minY
        {
            type            patch;
            inGroups        (coupleGroup);
        }
        minZ
        {
            type            patch;
        }
        maxZ/
        {
            type            patch;
        }
    }
```

Heaterの底面を
coupleGroup に入れる

Allrun.pre

① runApplication blockMesh
② runApplication topoSet
③ runApplication splitMeshRegions –cellZones –overwrite
④ 固体領域（heater, leftSolid, rightSolid）で
　0/領域名の nut, alphat, epsilon, k, U, p_rgh を消去する。
⑤ 全ての領域で
　　changeDictionary -region 領域名
　を実行する。
⑥ run Application createExternalCoupledPatchGeometry
　　–regions '(topAir heater)' coupleGroup
⑦ paraFoam -touchAll

comms/heater_topAir/coupleGroup に patchFaces と patchPoints が生成される。

Allrun のソルバー実行部分（シングルプロセッサの場合）

① runApplication $(getApplication) &
② runApplication ./externalSolver

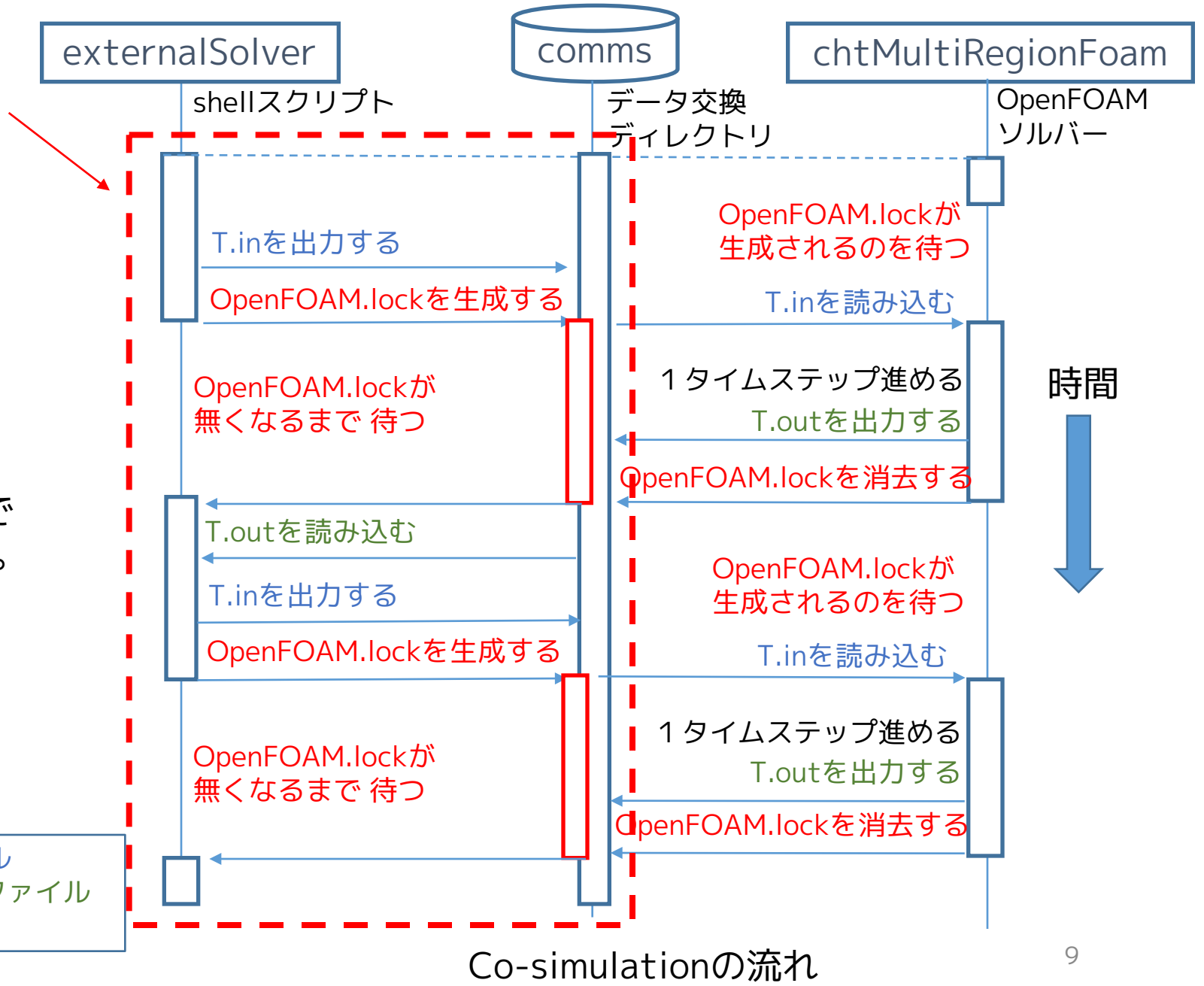
OpenFOAMのソルバーをバックグラウンド、外部ソルバーをフォアグラウンドで実行する

externalSolverの動作
（外部プログラムの雛形）

初期化
① 境界温度初期値 T.in を出力する。
② OpenFOAM.lock を生成する。

OpneFAM.lock が無くなると。。。
① 境界温度出力 T.out を読み込んで
温度を１度足してT.in に出力する。
② OpenFOAM.lock を生成する。

これを繰り返す。



T.in　：externalSolverが出力する温度データファイル
T.out：chtMultiRegionFoamが出力する温度データファイル
OpenFOAM.lock：ロックファイル

Co-simulationの流れ

データ交換のフォーマット

comms/heater_topAir/coupleGroup

出力データ　　入力データ

T.out　　T.in

```
531 47556.842
531 47556.841
531 47567.25
531 47567.25
531 47567.255
531 47567.255
531 47556.846
531 47556.846
331 2194.0633
331 1954.3748
331 1937.9028
331 1936.4325
331 2193.9723
331 1954.2926
331 1937.8204
331 1936.3528
331 2194.4991
...
```

```
528 0 1
528 0 1
528 0 1
528 0 1
528 0 1
528 0 1
528 0 1
528 0 1
328 0 1
328 0 1
328 0 1
328 0 1
328 0 1
328 0 1
328 0 1
328 0 1
328 0 1
328 0 1
...
```

T.out は１カラムが温度、2カラムは不明

T.in の各行の内容は、
　値、勾配、値と勾配の割合

heater底面の温度　8行　　行数は以下を参照。

constant/heater/polyMesh/boundary (一部分)

```
minY
{
    type            patch;
    inGroups        ( coupleGroup )
    nFaces          8;
    startFace       156;
}
```

topAir入り口面の温度 40行　　行数は以下を参照。

constant/topAir/polyMesh/boundary (一部分)

```
minX
{
    type            patch;
    nFaces                  40;
    startFace       3440;
    inGroup
            ( coupleGroup );
}
```

comms/heater_topAir/coupleGroup/patchFaces (一部分)

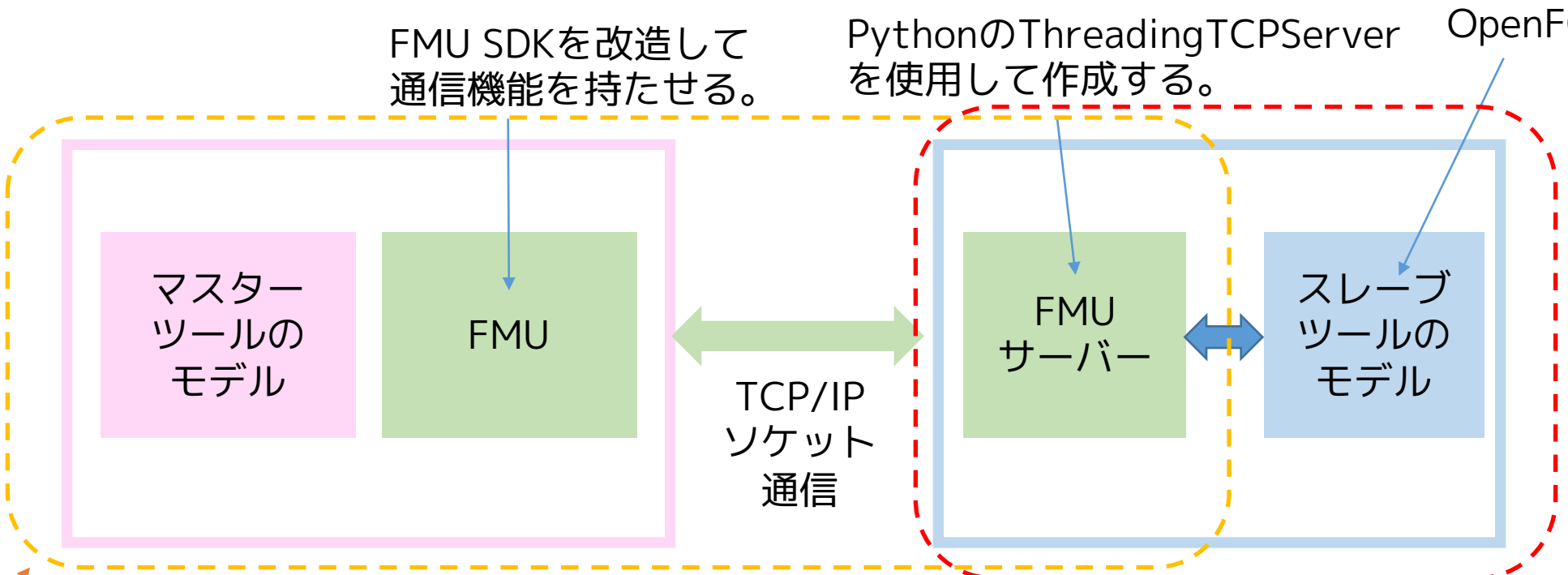
```
48
(
4(0 1 2 3)
4(3 2 4 5)
4(1 6 7 2)
...
```

全体の行数

FMU(Functional Mock-up Unit)の試作

- Co-Simulation システムの概要
- FMUの作成
- Xcos によるFMUの読み込みとシステムモデルの作成
- OpenFOAM 側の準備
- FMUサーバーの編集
- Co-Simulation の実行

FMUを使ったCo-Simulationシステムの概要



構想中のシステム

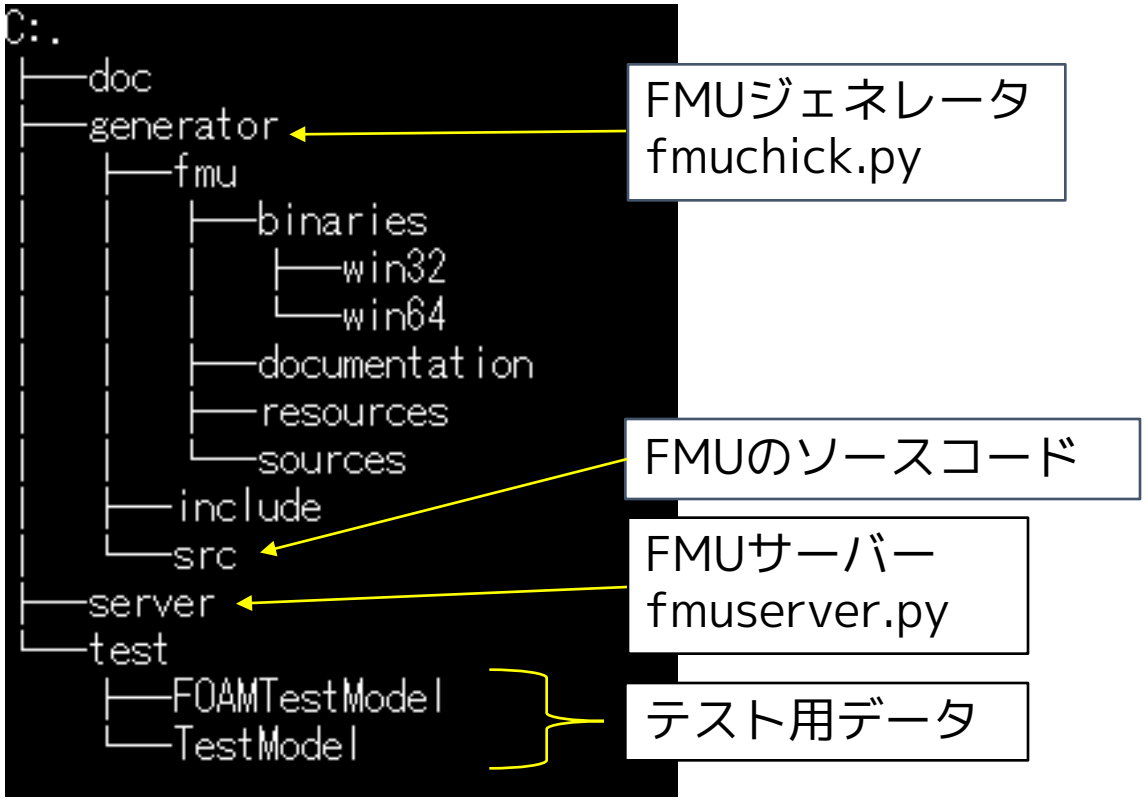
http://www.slideshare.net/AMANETANAKA/fmi10-fmi-for-cosimulation

作成したもの fmuchick (https://github.com/finback-at/fmuchick)

FMUジェネレータ（Python)
- FMUのソースコード（可変部、モデル固有の情報を含む）を出力する。
- XMLファイルを生成する。
- バッチファイルを使用してコンパイルし、ZIP化を行う。

FMUのソースコード（不変部）(C言語)
- FMUサーバーと通信を行うクライアント。
- FMU SDKのCo-Simulation用のソースコードを改造し通信機能を持たせる。

FMUサーバ (Python)
- FMU本体の要求によってスレーブツールをコントロールするサーバ。
- Pythonの標準ライブラリを使用し、マルチプラットフォームを目指す。



開発およびテスト環境
- Windows 10
- Visual Studio 2012
- Python 2.7.11 (32bit)
- wxPython 3.0

FMUの作成

チュートリアル
externalCoupledMultiRegionHeater
をベースとしてOpenFOAMと通信する
FOAMTestModel.fmu を作成する。

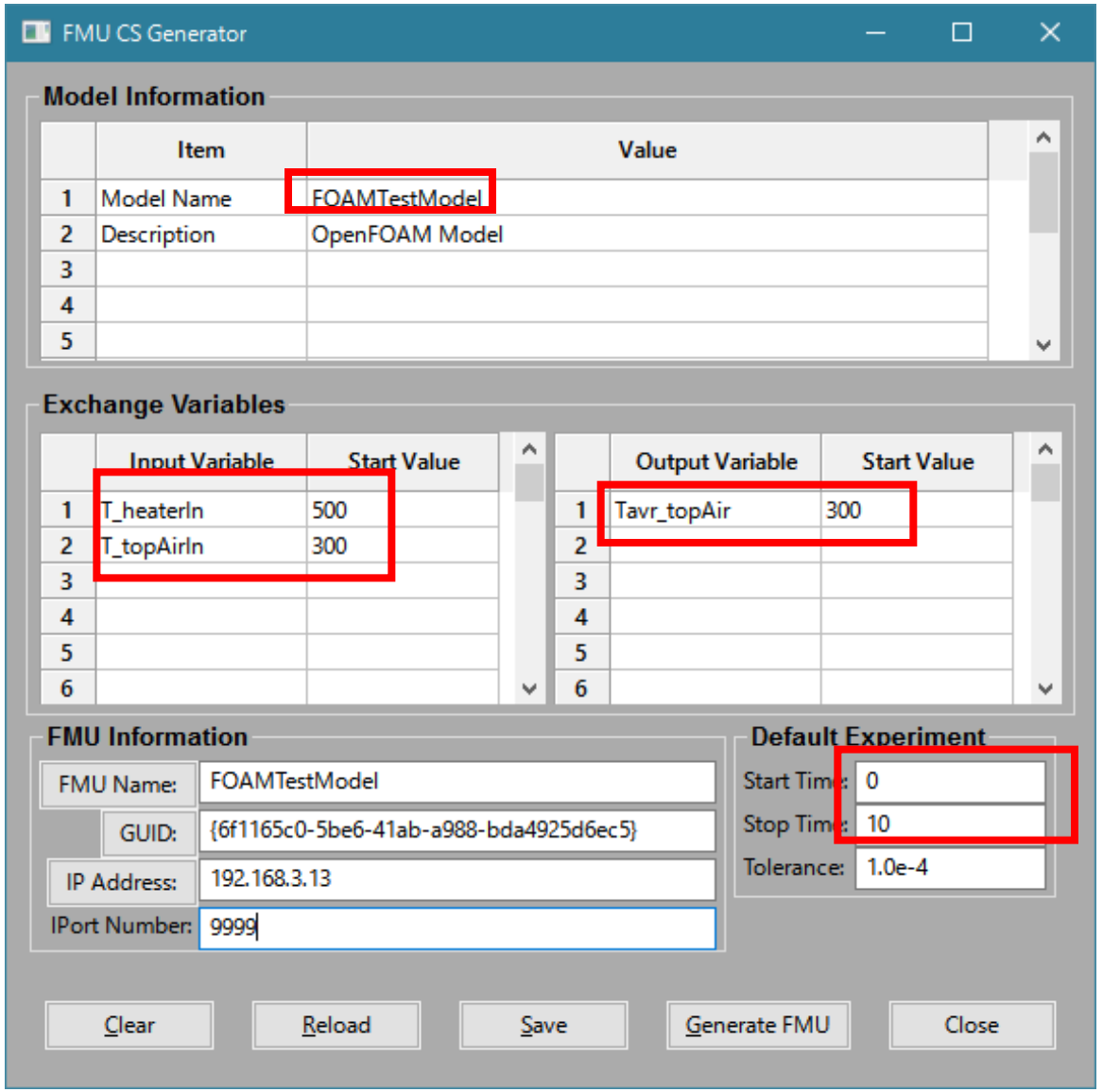
モデル名
- FOAMTestModel

入力変数
- T_heaterIn : heater 底面の温度
- T_topAirIn : topAir 入り口の温度

出力変数
- Tavr_topAir : topAirの平均温度

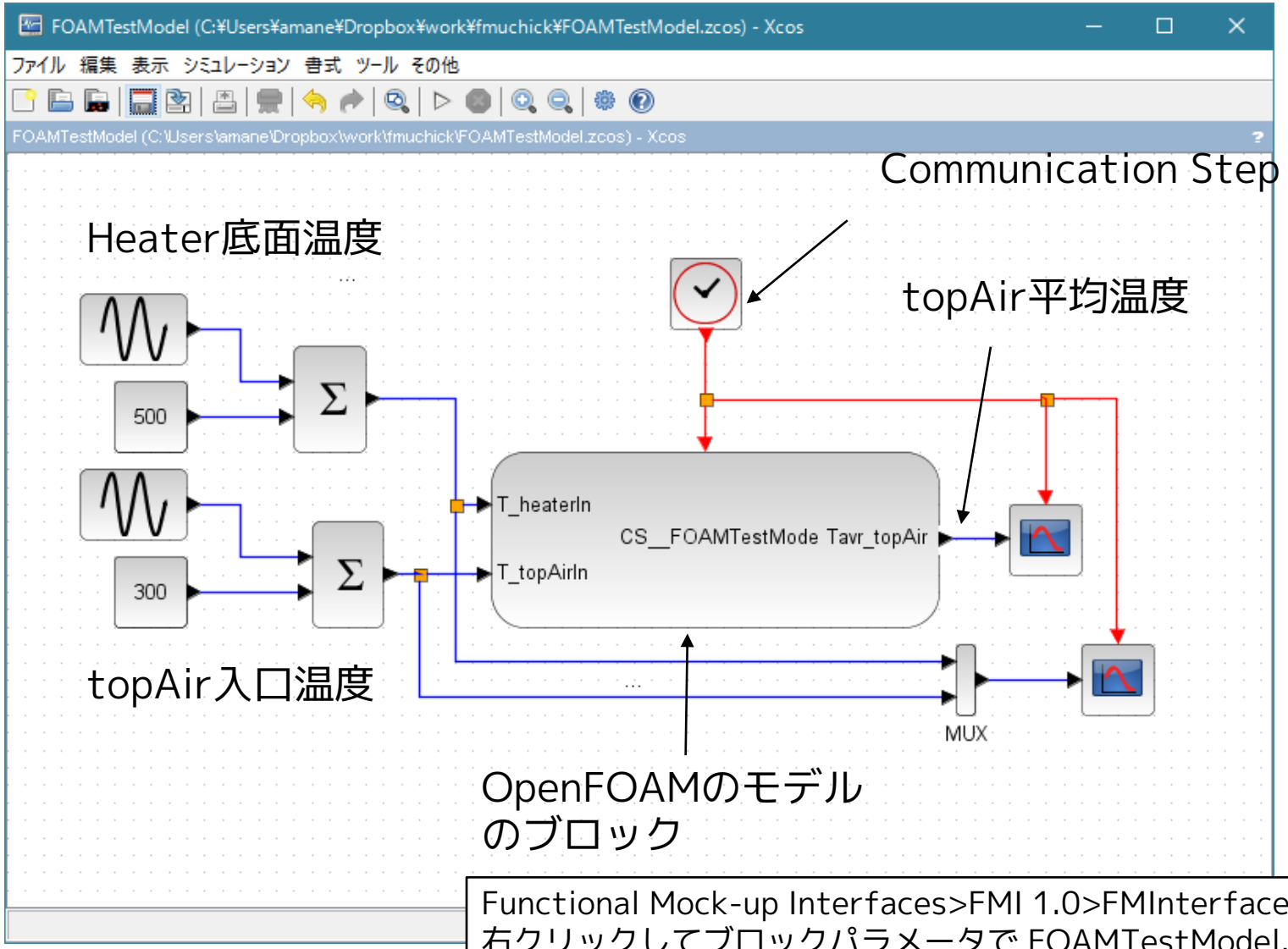
シミュレーション時間
Start Time　0 s
Stop Time　10 s
Communication Step 0.1 s

generator¥fmuchick.py によるFMUの作成



IP Address やポート番号はLinuxマシンに合わせて適宜変更する。

Xcos によるFMUの読み込みとシステムモデル作成



Functional Mock-up Interfaces>FMI 1.0>FMInterface
右クリックしてブロックパラメータで FOAMTestModel.fmu を読み込む。

シミュレーション＞設定

信号源>CLOCK_c

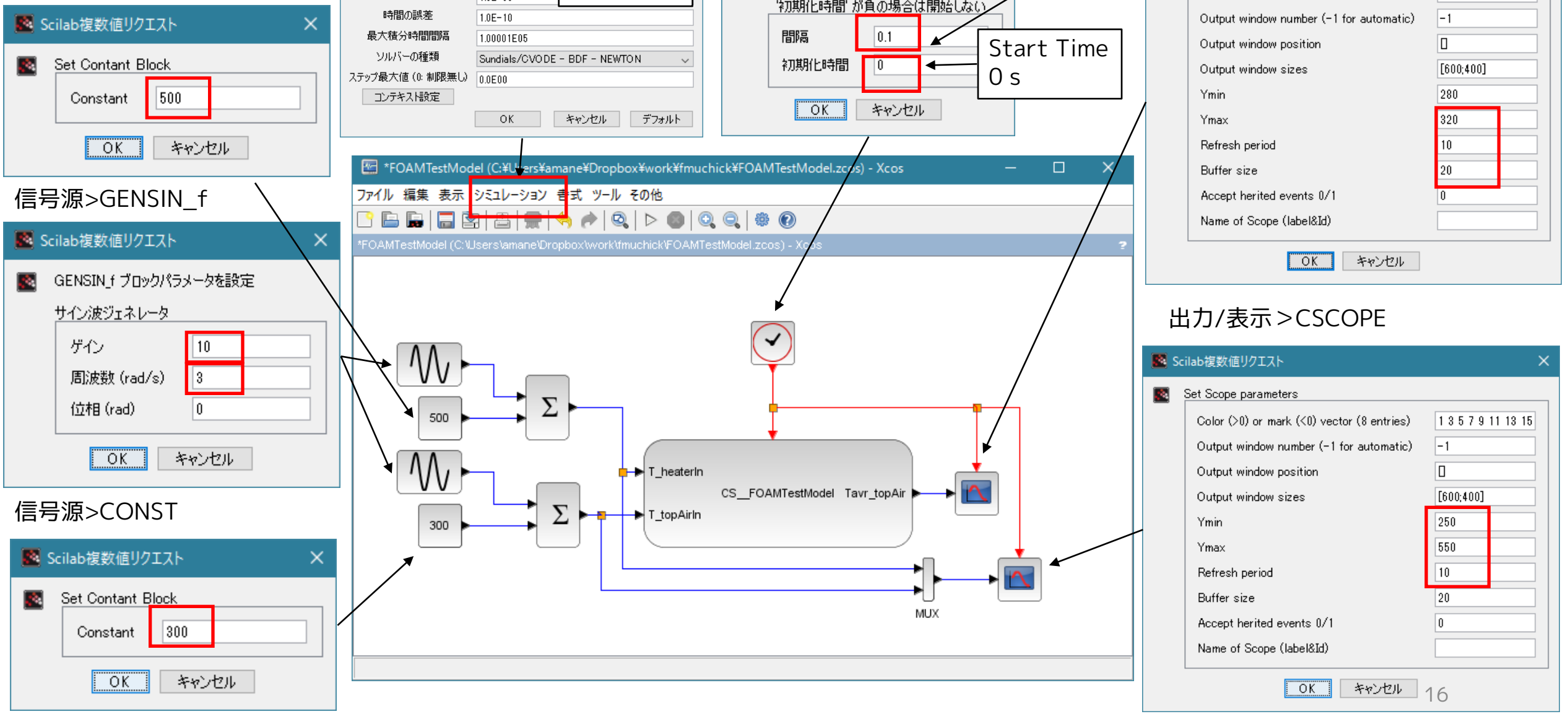
出力/表示＞CSCOPE

信号源>CONST

信号源>GENSIN_f

出力/表示＞CSCOPE

信号源>CONST



OpenFOAM 側の準備(1)

system/controlDict

```
/*--------------------------------*- C++ -*----------------------------------
*¥
| =========                 |                                                 |
| ¥¥      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  ¥¥    /   O peration     | Version:  v1606+                                |
|   ¥¥  /    A nd           | Web:      www.OpenFOAM.com                      |
|    ¥¥/     M anipulation  |                                                 |
¥*---------------------------------------------------------------------------
*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// Library defines new boundary conditions
libs            ("libOpenFOAM.so" "libjobControl.so"
                "libfieldFunctionObjects.so" "libFVFunctionObjects.so");

application     chtMultiRegionFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         10.0;

deltaT          0.01;
```

スタート時間

ストップ時間

```
writeControl    adjustableRunTime;

writeInterval   0.1;

purgeWrite      0;

writeFormat     ascii;

writePrecision  8;

writeCompression off;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;

maxCo           0.6;

// Maximum diffusion number
maxDi           10.0;

adjustTimeStep  on;

functions
{
    volAvgT
    {
        type cellSource;
        region "topAir";
        fields (T);
        valueOutput false;
        source all;
        operation volAverage;
        outputControl timeStep;
        outputInterval 1;
    }
```

コミュニケーションステップ

入口の流速が規定されているためクーラン条件からソルバーステップは約 0.03秒になっている。だいたい3ステップに1回データ交換する。

function object の追加

topAirの平均温度の出力の設定

system/controlDict (続き)

```
    externalCoupled
    {
        // Where to load it from (if not already in solver)
        functionObjectLibs ("libjobControl.so");

        type            externalCoupled;

        // Directory to use for communication
        commsDir        "${FOAM_CASE}/comms";

        // Does external process start first
        initByExternal  true;

        // Additional output
        log             true;

        regions
        {
            // Region name (wildcards allowed)
            "(topAir|heater)"
            {
                // In topAir adjust the minX patch (fixedValue)

                // Patch or patchGroup
                coupleGroup
                {
                    // Fields to output in commsDir
                    writeFields (T);
                    // Fields to read from commsDir
                    readFields  (T);
                }
            }
        }
    }
}

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

heater底面
topAir入口
の温度の書き出しと
読み込みの設定
（チュートリアルのまま）

出力変数の抽出のための function object の書き方は以下のリンクが参考になる。

PENGUINITIS – function objects

http://www.geocities.jp/penguinitis2002/study/OpenFOAM/function_objects.html

OpenFOAM の Function Object 機能について

http://www.slideshare.net/fumiyanozaki96/openfoam-function-object

OpenFOAM 側の準備(2)

Co-simulationでは、Allrun.pre を実行した後に、
FMUサーバー(fmuserver.py) を立ち上げる。
その前にFMUサーバーから起動されるスクリプトの作成や
FMUサーバーのカスタマイズを行う。

Run – OpenFOAM起動用スクリプト

```
!/bin/sh
. $HOME/OpenFOAM/OpenFOAM-
v1606+/etc/bashrc

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
rm -rf comms/OpenFOAM.lock
rm -rf postProcessing
rm -rf processor*
foamListTimes -rm -noZero
rm -rf log.decomposePar
rm -rf log.reconstructPar

#-- Run on single processor
#runApplication $(getApplication) &
# Simulated external solver
#runApplication ./externalSolver

# Decompose
runApplication decomposePar -allRegions

# Run OpenFOAM
runParallel $(getApplication)
```

以前の計算結果や不要なログファイル、ロックファイルなどを消去する

OpenFOAMの
並列計算の
開始

Post – 並列計算後処理用スクリプト

```
!/bin/sh
. $HOME/OpenFOAM/OpenFOAM-
v1606+/etc/bashrc

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

# Reconstruct
runApplication reconstructPar -allRegions
```

リコンストラクト

FMUサーバーの編集　OpenFOAMを操作するようにFMUサーバー（fmuserver.py）を編集する

```
def toolInitialize(self):
    self.proc = subprocess.Popen('./Run', shell=True)
    time.sleep(10)
    print "pid = "+str(self.proc.pid)
    self.stime = 0.0

def toolDoStep(self):
    t = self.ctime + self.cstep
    print str(self.stime)+" "+str(t)
    while self.stime < t:
        self.doSolverStep()
    print "t = "+str(self.ctime) + " dt = "+str(self.cstep)

def toolTerminate(self):
    subprocess.check_call('./Post', shell=True)
    if not (self.proc is None):
        os.killpg(os.getpgid(self.proc.pid), signal.SIGKILL)
    print "Terminate tool!"
```

初期化
(OpenFOAMの起動)
Runの実行

コミュニケーションステップ
のシミュレーション実行

シミュレーション
中止（終了処理）
Postの実行

```
def doSolverStep(self):
    grad0 = 0
    ratio0 = 1
    grad1 = 0
    ratio1 = 1
    f = open("comms/heater_topAir/coupleGroup/T.in","w")
    for i in range(0,8):
        f.write(str(self.inputValue[0])+" "+str(grad0)+" "+str(ratio0)+"¥n")
    for i in range(8,48):
        f.write(str(self.inputValue[1])+" "+str(grad1)+" "+str(ratio1)+"¥n")
    f.close()
    # generate lock file
    f = open("comms/OpenFOAM.lock","w")
    f.close()
    while (os.path.isfile("comms/OpenFOAM.lock")):
        time.sleep(0.02)
    cmd = "tail -n1 postProcessing/topAir/volAvgT/0/cellSource.dat"
    a = subprocess.check_output( cmd.split(" ") ).rstrip("¥n").split("¥t")
    self.stime = float(a[0].rstrip())
    self.outputValue[0] = float(a[1])
    print "t="+str(self.stime)+" Tavr="+str(self.outputValue[0])
```

heater底面温度、topAir入口温度
の入力データファイル作成

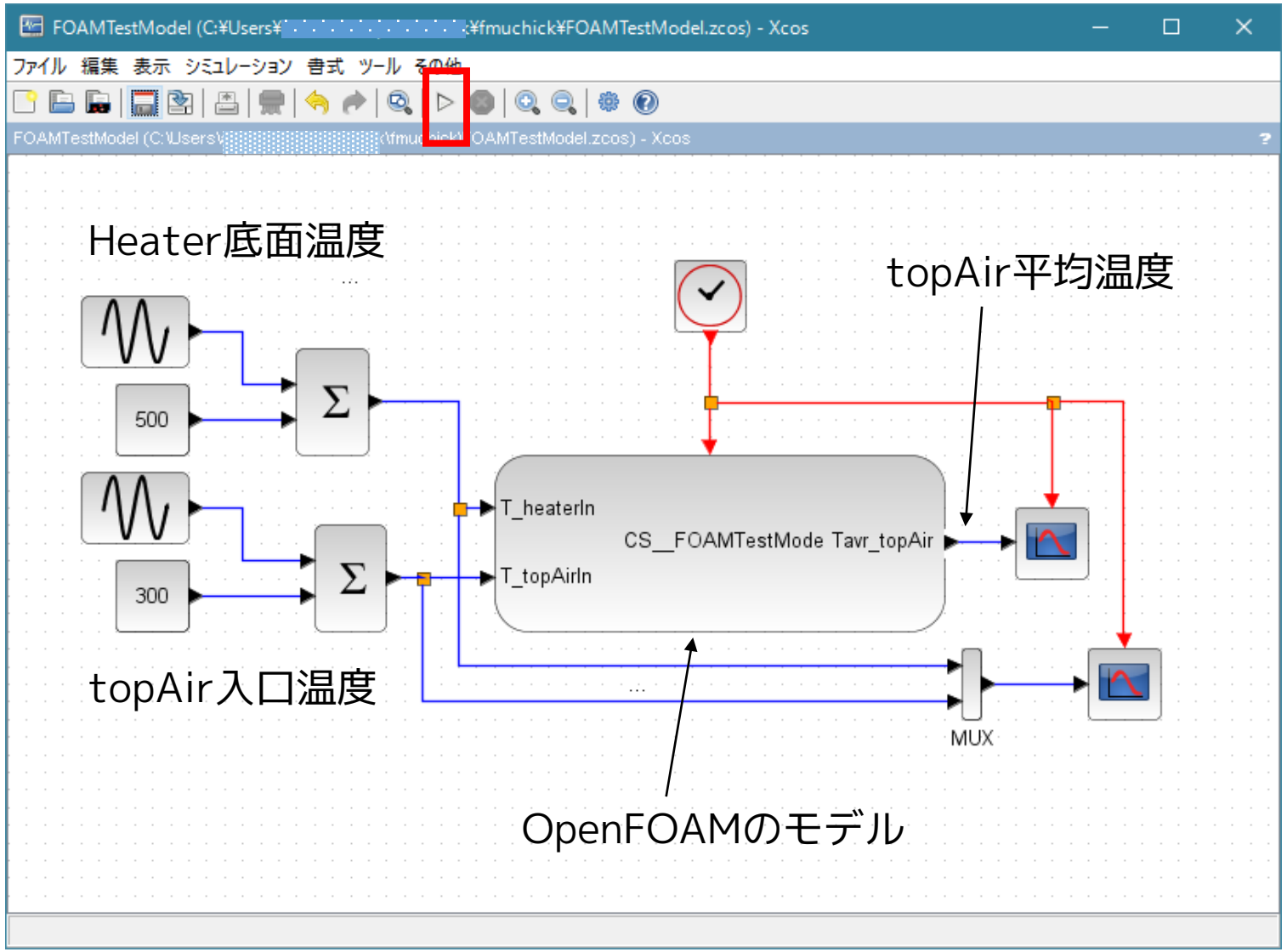
ロックファイルを生成
して無くなるまで待つ

topAir の平均温度と
ソルバー時間を取得する。

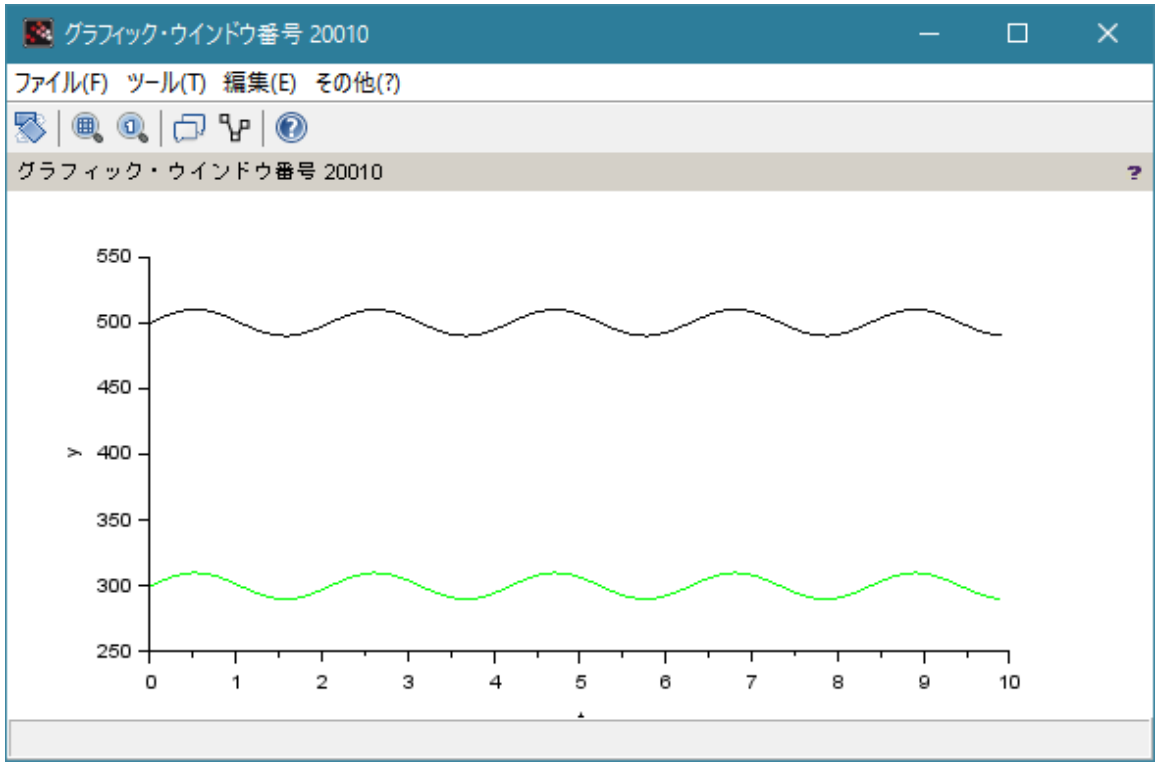
その他、fmuserver.py の最初の方にあるport number をFMUの設定値に合わせて変更する。

Co-Simulationの実行

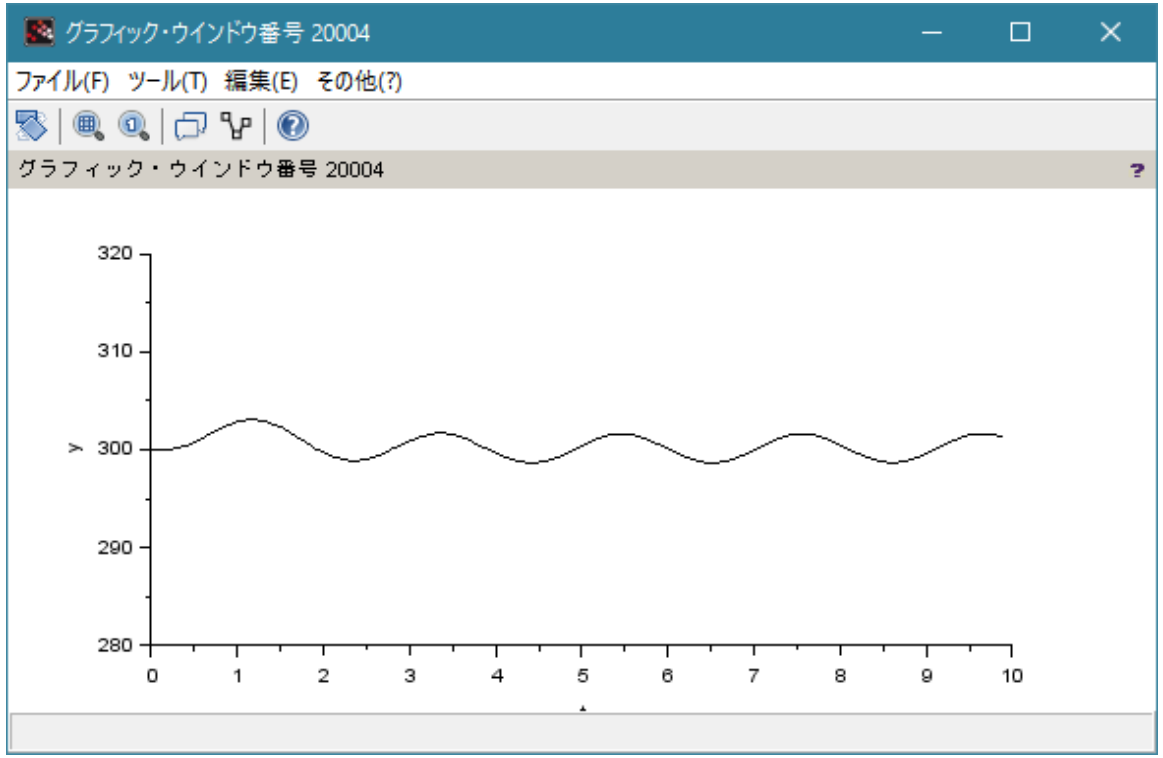
① Linux 上でFMUサーバを実行する。
② Xcosのシミュレーションを実行する。



Co-Simulationの実行結果

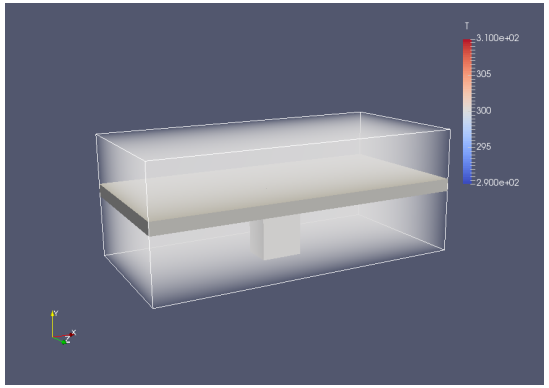


Heater底面温度（黒）とtopAir入口温度（緑）

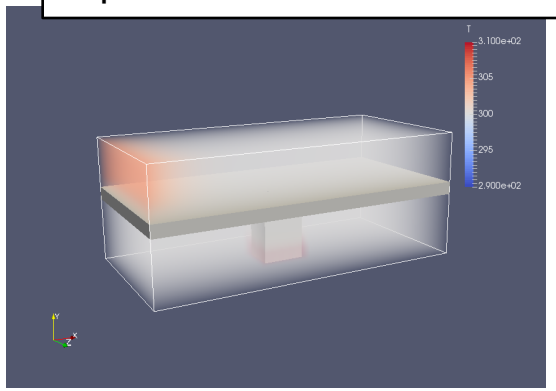


topAir平均温度

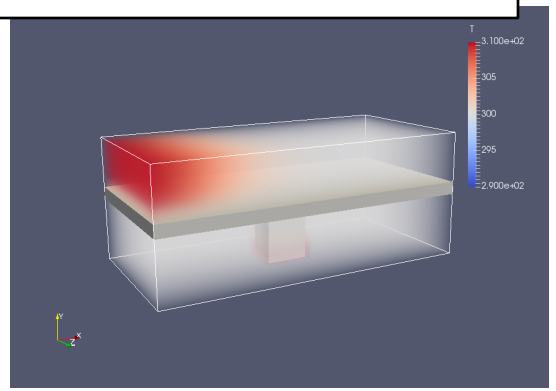
OpenFOAM+のシミュレーション結果（温度）



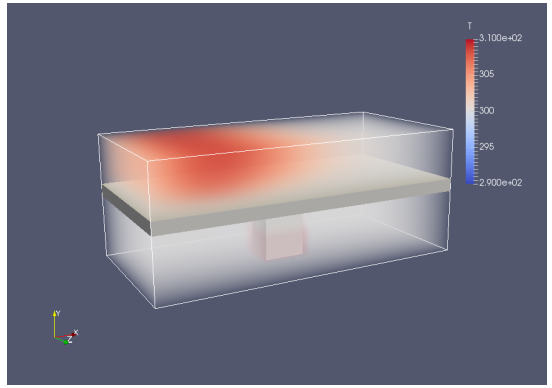
0.0 s



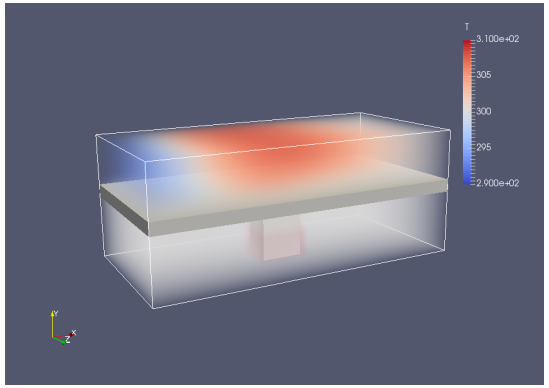
0.4 s



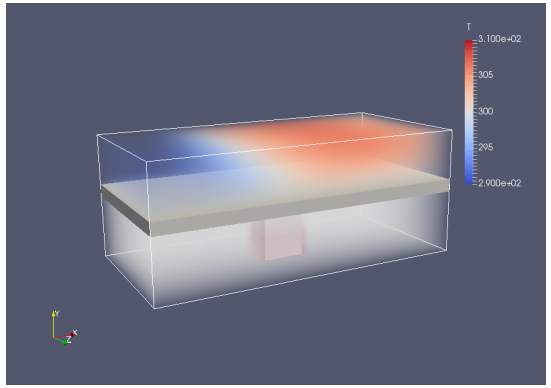
0.8 s



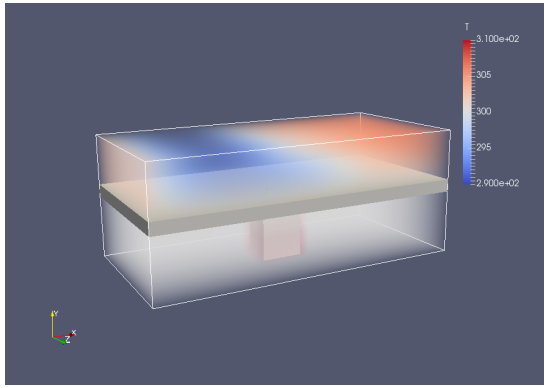
1.2 s



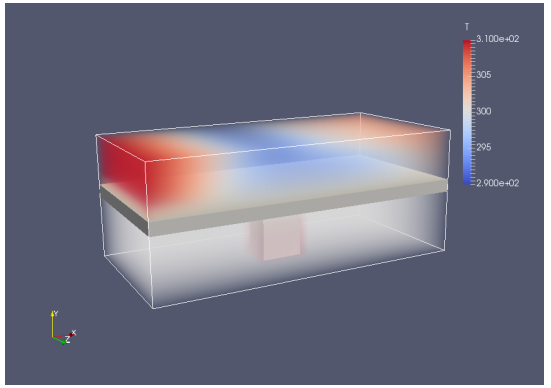
1.6 s



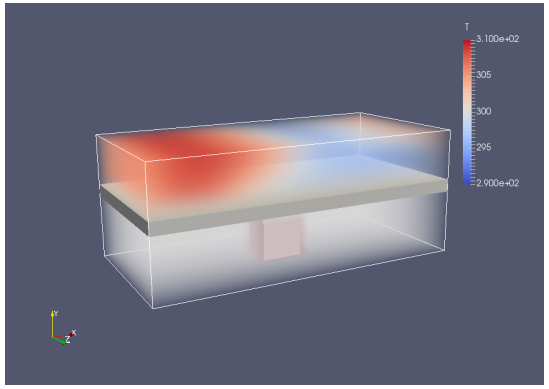
2.0 s



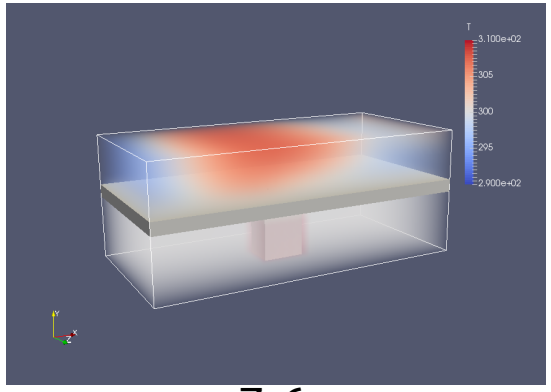
2.4 s



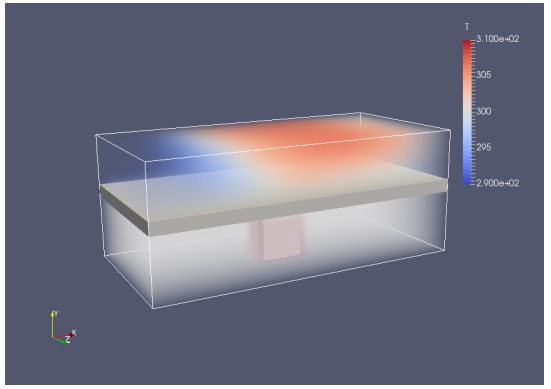
2.8 s



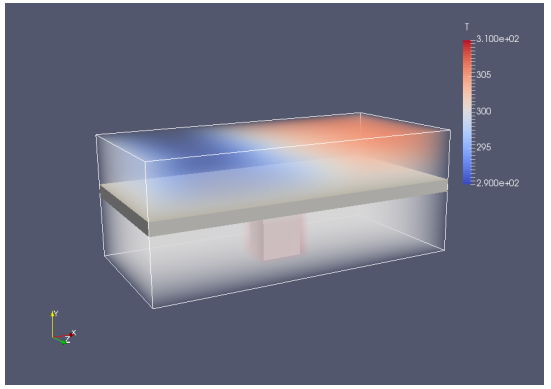
3.2 s



3.6 s



4.0 s



4.4 s

まとめ

- OpenFOAM+のCo-simulation機能について、チュートリアルモデルから使用方法を調べた。
- チュートリアルモデルをベースにしてFMUを作成し、Xcosをマスターソルバーとする簡単なCo-simulationを行った。