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

The screenshot shows the OpenFOAM website interface. At the top, it says 'OpenFOAM The open source CFD toolbox' with the 'esi' logo. A navigation menu includes 'Home', 'Products', 'Services', 'Download', 'Code', 'Documentation', and 'News'. Below the menu, there are links for 'About us', 'Contact', 'Jobs', and 'Legal', along with a 'FOLLOW US ON twitter' button. The main content area is titled 'Version' and features a sidebar with a list of categories: 'Release Summary', 'Pre-Processing', 'Meshing', 'Boundary conditions', 'Solvers and Physical Modelling', 'Solution Control', 'Numerics', 'Parallel', 'Post-processing', and 'Documentation'. The main text is titled 'OpenFOAM@ v3.0+: New Solution Control Functionality' and dated '13/01/2016'. It has a sub-heading 'Co-simulation' and explains that OpenFOAM includes a set of boundary conditions for communicating between OpenFOAM and external codes using file-based data transfer. It notes that in previous versions, the control was performed inside the boundary conditions itself. In this version, the control logic has been moved to a special function object, `externalCoupled`, which controls the coupling of all regions at every time step. The control between OpenFOAM and the external code employs a lock file whereby the existence of the lock file instructs the external code to wait, and either read or write the boundary values. It also mentions that the default file format is currently hard-coded to be a text based format with one line per boundary face, but this can be overridden inside the boundary condition. Finally, it states that the special settings in this `functionObject` are the definition of the directory to use for the file exchange, and the fields to read/write per region, per patch. For the patch, either a single patch name can be used or a patch group. The patch group enables multiple patches to be collated into one coupling interface.

- 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 – 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 i e l d | OpenFOAM: The Open Source CFD Toolbox |  
|¥¥ / O p e r a t i o n | Version: v1606+ |  
| ¥¥ / 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;  
    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 i e l d | OpenFOAM: The Open Source CFD Toolbox |  
|¥¥ / O p e r a t i o n | Version: v1606+ |  
| ¥¥ / 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;  
    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, alphas, 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

データ交換のフォーマット

comms/heater_topAir/coupleGroup

出力データ

T.out

```
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
...
```

入力データ

T.in

```
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
...
```

comms/heater_topAir/coupleGroup/patchFaces (一部分)

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

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

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 );
}
```

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

FMU SDKを改造して
通信機能を持たせる。

PythonのThreadingTCPServer
を使用して作成する。

OpenFOAM

前回の作成範囲 Windows, Linux, Mac, ...

今回もWindows版のみ

Xcos を使用する。

Windows, Linux, Mac, ...

今回の作成範囲

Linux, OpenFOAM

構想中のシステム

作成したものの 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

generator¥fmuchick.py による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

The screenshot shows the FMU CS Generator application window with the following configuration details:

Model Information	
Item	Value
1 Model Name	FOAMTestModel
2 Description	OpenFOAM Model
3	
4	
5	

Exchange Variables					
	Input Variable	Start Value		Output Variable	Start Value
1	T_heaterIn	500	1	Tavr_topAir	300
2	T_topAirIn	300	2		
3			3		
4			4		
5			5		
6			6		

FMU Information		Default Experiment	
FMU Name:	FOAMTestModel	Start Time:	0
GUID:	{6f1165c0-5be6-41ab-a988-bda4925d6ec5}	Stop Time:	10
IP Address:	192.168.3.13	Tolerance:	1.0e-4
IPort Number:	9999		

Buttons: Clear, Reload, Save, Generate FMU, Close

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

シミュレーション>設定

信号源>CLOCK_c

出力/表示>CSCOPE

信号源>CONST

パラメータ設定

積分終了時間: 11.0E01 (Stop Time 10s)

実時間スケール: 0.0E00

積算絶対許容誤差: 1.0E-06

積算相対誤差: 1.0E-06

時間の誤差: 1.0E-10

最大積分時間間隔: 1.00001E05

ソルバーの種類: Sundials/CVODE - BDF - NEWTON

ステップ最大値 (0: 制限無し): 0.0E00

コンテキスト設定

OK キャンセル デフォルト

Scilab複数値リクエスト

CLOCK_c ブロックパラメータ

イベントクロック生成器

初期化時間 が負の場合は開始しない

間隔: 0.1 (Communication Step 0.1 s)

初期化時間: 0 (Start Time 0 s)

OK キャンセル

Scilab複数値リクエスト

Set Scope parameters

Color (>0) or mark (<0) vector (8 entries): 1 3 5 7 9 11 13 15

Output window number (-1 for automatic): -1

Output window position: 0

Output window sizes: [600;400]

Ymin: 280

Ymax: 320

Refresh period: 10

Buffer size: 20

Accept herited events 0/1: 0

Name of Scope (label&Id):

OK キャンセル

信号源>GENSIN_f

出力/表示>CSCOPE

Scilab複数値リクエスト

Set Scope parameters

Color (>0) or mark (<0) vector (8 entries): 1 3 5 7 9 11 13 15

Output window number (-1 for automatic): -1

Output window position: 0

Output window sizes: [600;400]

Ymin: 250

Ymax: 550

Refresh period: 10

Buffer size: 20

Accept herited events 0/1: 0

Name of Scope (label&Id):

OK キャンセル

信号源>CONST

Scilab複数値リクエスト

Set Contant Block

Constant: 300

OK キャンセル

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; コミュニケーションステップ
```

```
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;
```

```
}
```

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);
      }
    }
  }
}

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

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>

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)

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。