新TreeFoamの紹介

（TreeFoam ver 3.24.240723）

＜修正内容＞

Ubuntu24.04上で、TreeFoamを最新のOpenFOAMに対応

1. OF-12への対応
   snappyHexMesh、frontISTRとの連成解析
2. OF-v2406への対応
   field内変数定義
3. ubuntu24.04へのインストールについて
   環境変数QT_QPA_PLATFORMの定義、SalomeとpyFoamのインストール
4. まとめ

24/07/23　藤井

1.　OF-12への対応

TreeFoam操作マニュアルに基づきOpenFOAM-12の作動を確認

1-1.　snappyHexMeshDictに関する修正

geometryの定義方法が変更されている。

＜OF-11以前＞

```
        :
castellatedMesh false;
snap false;
addLayers true;

geometry
    {
        fineReg.stl
        {
            type triSurfaceMesh;
            name fineReg;
        }

        halfSp.stl
        {
            type triSurfaceMesh;
            name halfSp;
        }
      :
```

＜OF-12＞

```
        :
castellatedMesh false;
snap false;
addLayers true;

geometry
    {
        fineReg
        {
            type triSurfaceMesh;
            file "fineReg.stl";
        }

        halfSp
        {
            type triSurfaceMesh;
            file "halfSp.stl";
        }
      :
```

OF-12用の分岐を設け対応。

1-2.　FrontISTRとの流体構造連成解析

連成解析のsourceCode（C++）内で使用しているpatch名のindexIDを取得する関数名が変更された。
（findPatchID() ⇒　findIndex()　に変更）

＜OF-11以前＞

```
      :
//変数の定義
int pNo = Pstream::myProcNo();
int np = Pstream::nProcs();
scalar timeValue = runTime.value();
scalar dT = runTime.deltaTValue();          //deltaT取得
scalar beforeTime = timeValue - dT;
label timeIdx = runTime.timeIndex();
bool isWriteTime = runTime.writeTime();
bool isPressCoupling = false;
label patchId = mesh().boundaryMesh().findPatchID(patchName_);
word procNo = label2word(pNo);              //processNo
word nProcs = label2word(np);               //全process数
word tVal = scalar2word(timeValue);         //現在のtime
word btVal = scalar2word(beforeTime);       //前のtime
word timeIndex = label2word(timeIdx);       //timeIndex
      :
```

＜OF-12＞

```
      :
//変数の定義
int pNo = Pstream::myProcNo();
int np = Pstream::nProcs();
scalar timeValue = runTime.value();
scalar dT = runTime.deltaTValue();          //deltaT取得
scalar beforeTime = timeValue - dT;
label timeIdx = runTime.timeIndex();
bool isWriteTime = runTime.writeTime();
bool isPressCoupling = false;
label patchId = mesh().boundaryMesh().findIndex(patchName_);
word procNo = label2word(pNo);              //processNo
word nProcs = label2word(np);               //全process数
word tVal = scalar2word(timeValue);         //現在のtime
word btVal = scalar2word(beforeTime);       //前のtime
word timeIndex = label2word(timeIdx);       //timeIndex
      :
```

連成面を取得する箇所になるため、FSI, CHT等、全てのcodeをOF-12用に修正して対応した。

2. OF-v2406への対応

field内変数の定義方法が変更されており、gridEditorで読めないtutorialsのcaseが発生。

（変数定義方法　$:wall.T ⇒ $/wall/T に変更）

＜OF-11以前＞

```
        :
wall
{
        :
    T
    {
        type            zeroGradient;
    }
        :
}

dimensions      [0 0 0 1 0 0 0];

internalField   uniform $:outerInlet.T;

boundaryField
{
        :
    staticWalls
    {
        $:wall.T;
    }
        :
}
```

「type zeroGradient;」が入る

＜OF-12＞

```
        :
wall
{
        :
    T
    {
        type            zeroGradient;
    }
        :
}

dimensions      [0 0 0 1 0 0 0];

internalField   uniform $:outerInlet.T;

boundaryField
{
        :
    staticWalls
    {
        $/wall/T;
    }
        :
}
```

変数定義として「$:wall.T」「$/wall/T」とOF-12系の「$!wall/T」が解釈できるように修正。

3.　ubuntu24.04へのインストールについて

3-1.　環境変数QT_QPA_PLATFORMの定義追加

ubuntu24.04にTreeFoamをそのままインストールした場合、gridEditorが正常に作動しない。

gridEditorは、起動表示はするものの、マウスなどでカーソルを移動すると表示が乱れる。

gridEditor上でvtk表示させると、X errorで落ちる。

当初原因が全く不明だったが、以下の環境変数をセットする事で、問題が解決した。

```
export QT_QPA_PLATFORM=xcb
```

この環境変数は、Qtアプリ（gridEditor）について、DisplayServerのwaylandをXの様に動作させる為のもの。

Gtkアプリについても「export GDK_BACKEND=x11」をセットする事で、Xの様に動作させる事ができる。

ubuntu22.04は、「export GDK_BACKEND=x11」のみで正常に作動していたが、

ubuntu24.04は、「export QT_QPA_PLATFORM=xcb」を追加。

ubuntu22.04	ubuntu24.04	
export GDK_BACKEND=x11	export GDK_BACKEND=x11	#GTKアプリ用
	export QT_QPA_PLATFORM=xcb	#Qtアプリ用

この内容は、TreeFoamの起動スクリプト(treefoam)内で設定している。

3-2.　Salomeのインストール（mesh作成としての利用）

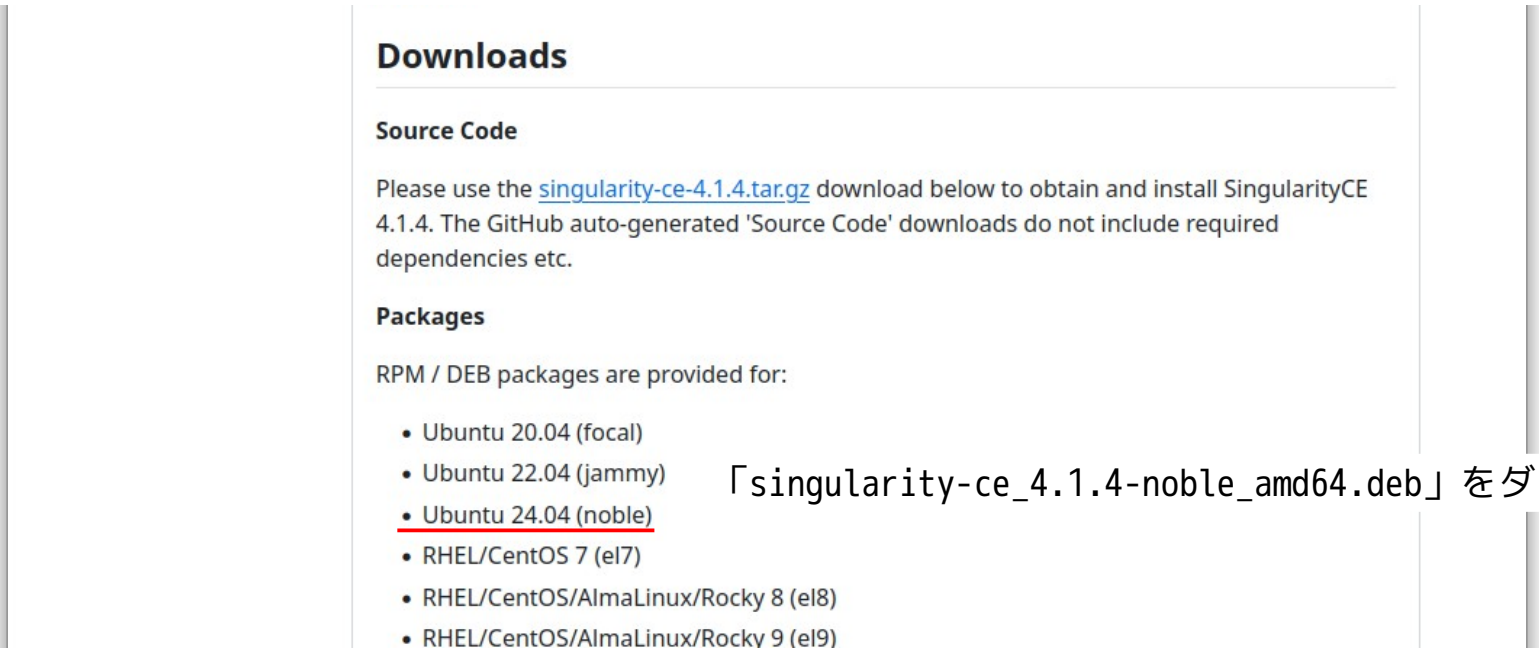
ubuntu22.04の場合、ubuntu22.04 nativeのSalomeを利用。

ubuntu24.04の場合、未だubuntu24.04 nativeのsalomeがリリースされていない。（24/07/23現在）

salome-Mecaは、singularityのコンテナ上で動かす為、singularityさえあれば、
ubuntuのバージョンを問わない。この為、Salome-Mecaのインストールを試みる。

ubuntu24.04用のsingularityのdebパッケージを探すと、以下にアップされている。

https://github.com/sylabs/singularity/releases



「singularity-ce_4.1.4-noble_amd64.deb」をダウンロード

salome-Mecaを「https://code-aster.org/spip.php?article303」からダウンロードする。
　　ダウンロードした「salome_meca-lgpl-2023.1.0-4-20240327-scibian-10.sif」を~/Salomeに保存

singularityのインストールとsalomeの実行ファイルを作成する。

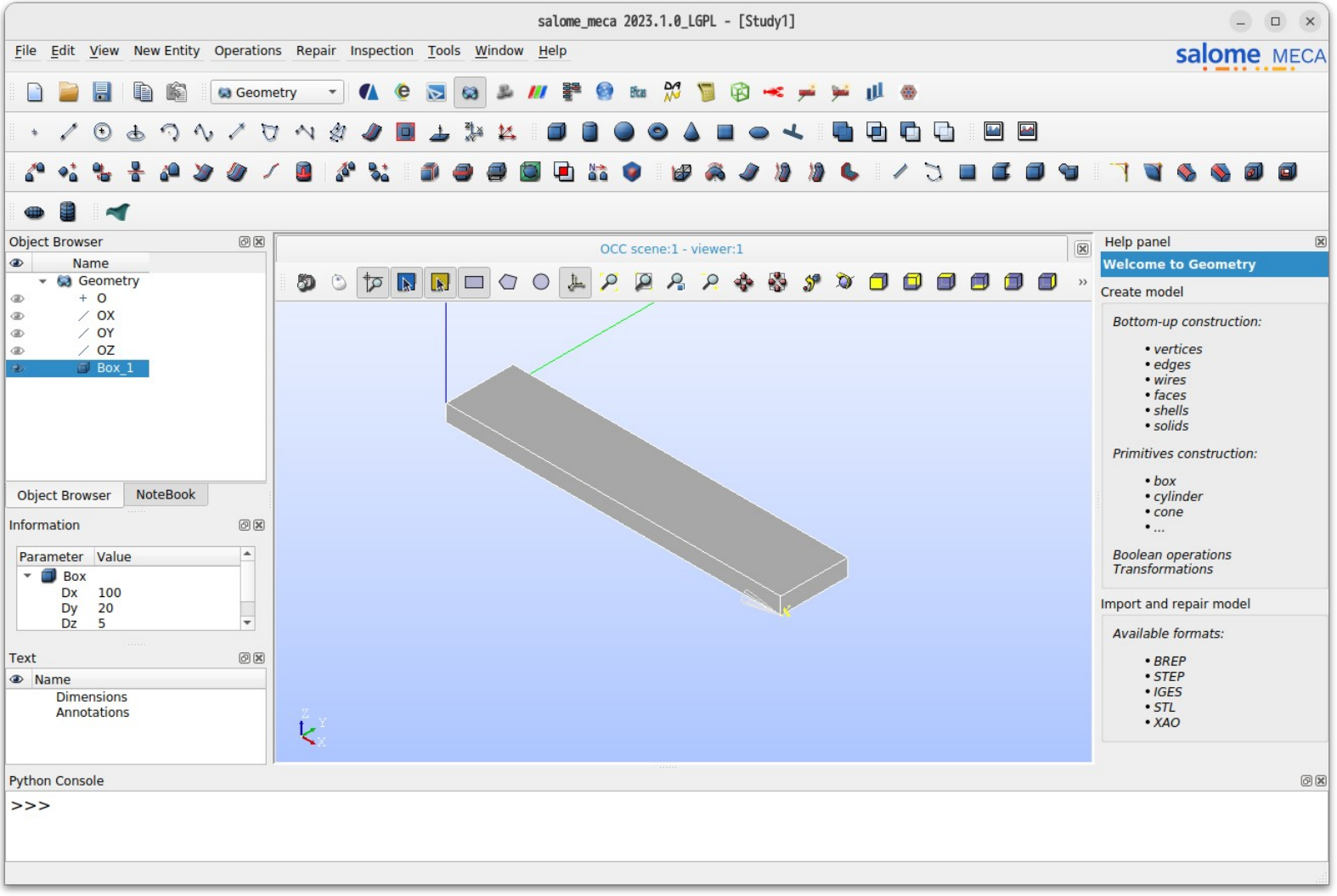
```
$ sudo apt --fix-broken install singularity-ce_4.1.4-noble_amd64.deb                  #singularityをインストール
$ cd ~/Salome                                                                         #保存場所に移動
$ singularity run --app install salome_meca-lgpl-2023.1.0-4-20240327-scibian-10.sif   #salomeMecaの実行ファイル作成
```

この操作により、実行ファイル「salome_meca-lgpl-2023.1.0-4-20240327-scibian-10」ができあがる。
実行は、以下を入力して実行する。

```
$ ./salome_meca-lgpl-2023.1.0-4-20240327-scibian-10 --soft
```

salome-Mecaが起動する。
singularity、Salome-Mecaも最新であり、安定して作動する。

Salome-Meca-2023.1.0（24/4/22 release）の画面



3-3.　PyFoamのインストール

現在のPyFoamは、「PyFoam-2023.7」であり、バージョンアップされていない。(24/07/24現在）
無理やりubuntu24.04にpipでインストールし、実行すると、importErrorが発生し、実行できず。

原因
　pyFoam側は、python3.11までしか対応していない。
　ubuntu24.04は、python3.12がデフォルトでインストールされている。
　（ubuntu22.04は、python3.10がインストール）
　pyFoamのsouceCodeは、入手できても、pythonのバージョンが異なるため、エラーが発生する。

エラーを回避するために、
ubuntu24.04上にpython3.10の仮想環境を準備し、ここにpyFoamをインストールする。
pythonの仮想環境は、pyenv、venvを使って作成する。
作成方法は、以下のwebを参考にしている。
　　「https://qiita.com/ysdyt/items/5008e607343b940b3480」
　　「https://qiita.com/yabish/items/93c4e043e4c8dbc60cad」

必要なライブラリとpyenvをインストール

```
$ sudo apt update
$ sudo apt install build-essential libbz2-dev libdb-dev libreadline-dev libffi-dev libgdbm-dev liblzma-dev
libncursesw5-dev libsqlite3-dev libssl-dev zlib1g-dev uuid-dev tk-dev
$ git clone https://github.com/pyenv/pyenv.git ~/.pyenv     #pyenvのインストール
```

pyenv関連の設定を.bashrcに追記し、その内容を反映する

```
.bashrcの最後に以下を追記
--------- 追記内容 ----------------------------------------------------
export PYENV_ROOT="$HOME/.pyenv"
command -v pyenv >/dev/null || export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init -)"
-----------------------------------------------------------------------
$ source ~/.bashrc                  #追記内容を反映させる
```

仮想環境folderを作成し、そこにpython3.10をインストールする

```
$ mkdir ~/local_python              #仮想環境のフォルダ作成
$ cd local_python                   #フォルダ移動
$ pyenv local 3.10.14               #python3.10.14を仮想環境にインストール
```

仮想環境上でPyFoamをインストール

```
$ python3 -m venv ~/local_python        #~/local_pythonを仮想環境に設定
$ source ~/local_python/bin/activate    #仮想環境を開始
$ python3 -m pip install PyFoam         #仮想環境上でpyFoamをpipでインストール
$ deactivate                            #仮想環境停止
```

仮想環境のpython3.10起動スクリプト作成する

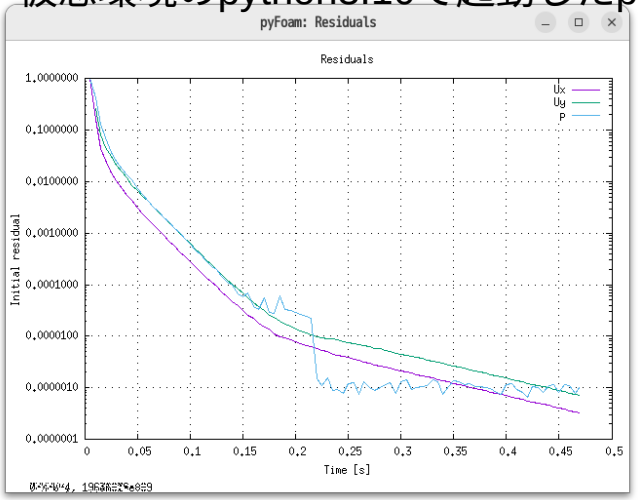
以下のスクリプトを作成し、~/binに「python3.10」として保存する

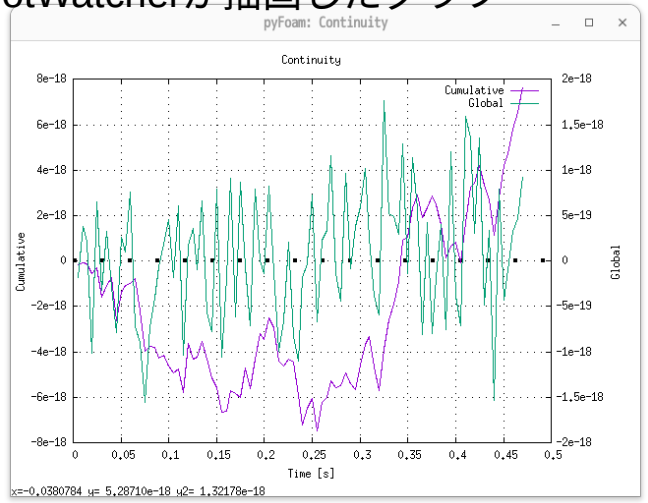
```
------------ python3.10 起動スクリプト ----------------------------
#!/bin/bash
cd ~/local_python
source ~/local_python/bin/activate      #仮想環境を開始
python3 $1 $2 $3 $4 $5 $6 $7 $8 $9      #python3.10を起動
deactivate                              #仮想環境停止
-------------------------------------------------------------------
```

python3.10起動スクリプトの作成により、TreeFoam内のplotWatcherの起動は、
以下の様に、通常のpython3と仮想のpython3.10使い分ける事ができる

```
python3 $pyFoamDir/bin/pyFoamPlotWatcher solve.log      #通常起動
python3.10 $pyFoamDir/bin/pyFoamPlotWatcher solve.log   #python3.10で起動
```

仮想環境のpython3.10で起動したplotWatcherが描画したグラフ





4.　まとめ

ubuntu24.04上に最新のTreeFoam（OF-12、OF-v2406対応）をインストールした。
SalomeとPyFoamが未だubuntu24.04に対応しておらず、インストールできなかったが、
　　salomeをsalome-Meca（24/3/27リリース）に変更してインストール
　　PyFoamは、旧版をpythonの仮想環境上（python3.10）にインストール
で対応している。

ubuntu24.04リリース以降、約3ヶ月経過しているが、SalomeとpyFoamが未だ対応していない為、
止むなく、salome-Meca、仮想環境内でのpyFoamをインストールしている。