2012/06/19 中川作成

propeller (pimpleDyFoam) の実行手順解説：Allrun.pre版

propellerでAllrunを実行すると，その内部で，Allrun.preが実行されます。
Allrun.preでは，メッシュを生成するために必要な作業が実施されています。

この文書では，Allrun.pre内で行われている処理について説明します。

【Allrun.pre の解読：　背景灰色部はオリジナルファイルの記述】

```
#!/bin/sh
cd ${0%/*} || exit 1      # run from this directory
```
上記のコマンドは,実行場所をチェックするものです。実質的な意味はありません。

```
# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
```
上記コマンドは，tutorialで使う便利機能を読み込むためのものです。”runApplication”とかのコマンドを使うためです。

```
# copy propeller surface from resources folder
cp $FOAM_TUTORIALS/resources/geometry/propellerTip.obj.gz constant/triSurface/
```
プロペラ等の3Dモデルが入ったファイルを，ケースディレクトリ/constant/triSurface/にコピーしています。

```
# - meshing
runApplication blockMesh
```
blockMeshを実行しています。
snappyHexMeshに必要な大枠のメッシュを作っているはず。

```
surfaces="
        innerCylinder
        innerCylinderSmall
        outerCylinder
        propellerTip
        propellerStem1
        propellerStem2
        propellerStem3
"
```
surfacesという変数に，７つの名前を入れています。これらの名前は，/constant/triSurfaceディレクトリに置かれている3Dモデルのファイル名です。

```
for s in $surfaces; do
        runApplication surfaceFeatureExtract -includedAngle 150 -minElem 10    \
        constant/triSurface/$s.obj $s
        mv log.surfaceFeatureExtract log.surfaceFeatureExtract.$s
done
```
forループです。sという変数に，上記surfacesでセットした7つのファイル名が，順番に入ります。上記のコマンドが，各ファイルに対して実行されることになります。
surfaceFeatureExtractの詳細については，文章末に記載する。
注意：2行目行末の記号”\”は，その行と次の行がつながっていることを示します。

結局，一つ目のファイルinnnerCylinder.objに対して，下記の２つのコマンドを実行しています。下記の2行目は，surfaceFeatureExtract実行時の記録を書き込んだファイル（log.surfaceFeatureExtrace ← runApplicationによって作成された）を，拡張子にもとのモデルファイル名を追加した別名（log.surfaceFeatureExtract.innerCylinder）で保存しているものです。
surfaceFeatureExtract -includedAngle 150 -minElem 10 constant/triSurface/innerCylinder.obj innerCylinder > log.surfaceFeatureExtract
mv log.surfaceFeatureExtract log.surfaceFeatureExtract.innerCylinder
このようなコマンドを，前述の7つのファイルに対して実行しています。

runApplication snappyHexMesh -overwrite
snappyHexMeshを実行しています。

- generate face/cell sets and zones

#runApplication setSet -batch removeRedundantZones.setSet
#mv log.setSet log.removeRedundantZones.setSet
runApplication topoSet -dict system/removeRedundantZones.topoSetDict
mv log.topoSet log.removeRedundantZones.topoSet
systemの中にある"removeRedundantZones.topoSetDict"の情報を元にして，topoSetを実行する。重複するcellZoneSet（innerCylinderおよびinnerCylinderSmall）の削除？

#runApplication setSet -batch createInletOutletSets.setSet
#mv log.setSet log.createInletOutletSets.setSet
runApplication topoSet -dict system/createInletOutletSets.topoSetDict
mv log.topoSet log.createInletOutletSets.topoSet
systemの中にある"createInletOutletSets.topoSetDict"の情報を元にして，topoSetを実行する。流入口，流出口のためのsetを生成する。

#runApplication setSet -batch createAMIFaces.setSet
#mv log.setSet log.createAMIFaces.setSet
runApplication topoSet -dict system/createAMIFaces.topoSetDict
mv log.topoSet log.createAMIFaces.topoSet
systemの中にある"createAMIFaces.topoSetDict"の情報を元にして，topoSetを実行する。回転部のAMI面の生成を準備する。（faceZoneのinnerCylinderSmallから，後ほど，createBafflesコマンドを使って，AMI1とAMI2を生成する。）

- create the inlet/outlet patches
runApplication createPatch -overwrite
systemの中にある"createPatchDict"の情報を元にして，createPatchを実行する。
set inletFacesからpatch inletを，set outletFacesからpatch outletを作成する。

- create the AMI faces by creating baffles, and then splitting the mesh

runApplication changeDictionary
systemの中にある"changeDictionaryDict"の情報を元にして，dictionaryのentriesを変更する。
具体的には，boundaryファイルに，AMI1とAMI２の情報を追加する。

force removal of fields generated by snappy
\rm -rf 0

```
createBaffles -internalFacesOnly -overwrite innerCylinderSmall '(AMI1 AMI2)'   \
        > log.createBaffles 2>&1
# <faceZone>innerCylinderSmallをboundaryに変換する。AMI1をmasterPatch，AMI2をslavePatchにする。
#    -internalFacesOnlyオプション：do not convert boundary faces

runApplication mergeOrSplitBaffles -split -overwrite
#   -splitオプション：topologically split duplicate surfaces

# - apply the initial fields
cp -rf 0.org 0
# 0.orgに入っているファイルを0ディレクトリにコピーする。初期条件
```

【surfaceFeatureExtract -help コマンドで表示される説明】

```
Usage: surfaceFeatureExtract [OPTIONS] <surface> <output set>
options:
        -case <dir>
                specify alternate case directory, default is the cwd
-closeness <scalar>
                span to look for surface closeness
-deleteBox <((x0 y0 z0)(x1 y1 z1))>
                remove edges within specified bounding box
-featureProximity <scalar>
                distance to look for close features
         -includedAngle <degrees>
                construct feature set from included angle [0..180]
-manifoldEdgesOnly
                remove any non-manifold (open or more than two connected faces) edges
-minElem <int>
                remove features with fewer than the specified number of edges
-minLen <scalar>
                remove features shorter than the specified cumulative length
-noFunctionObjects
                do not execute functionObjects
-set <name>
                use existing feature set from file
-subsetBox <((x0 y0 z0)(x1 y1 z1))>
                remove edges outside specified bounding box
-writeObj
                write extendedFeatureEdgeMesh obj files
-writeVTK
                write surface property VTK files
-srcDoc
                display source code in browser
-doc
                display application documentation in browser
-help
                print the usage
```

【topoSet -help コマンドで表示される説明】

```
Usage: topoSet [OPTIONS]
options:
        -case <dir>     specify alternate case directory, default is the cwd
        -constant       include the 'constant/' dir in the times list
        -dict <file>    specify an alternative dictionary for the topoSet dictionary
        -latestTime     select the latest time
-noFunctionObjects
                        do not execute functionObjects
        -noSync         do not synchronise selection across coupled patches
        -noZero         exclude the '0/' dir from the times list, has precedence
                        over the -zeroTime option
        -parallel       run in parallel
        -region <name>          specify alternative mesh region
-roots <(dir1 .. dirN)>
                        slave root directories for distributed running
-time <ranges>
                        comma-separated time ranges - eg, ':10,20,40-70,1000:'
        -srcDoc         display source code in browser
        -doc            display application documentation in browser
        -help           print the usage
```

【createPatch -help コマンドで表示される説明】

```
Usage: createPatch [OPTIONS]
options:
  -case <dir>       specify alternate case directory, default is the cwd
  -noFunctionObjects
                do not execute functionObjects
  -overwrite        overwrite existing mesh/results files
  -parallel         run in parallel
  -region <name>    specify alternative mesh region
  -roots <(dir1 .. dirN)>
                slave root directories for distributed running
  -srcDoc           display source code in browser
  -doc              display application documentation in browser
  -help             print the usage
```

【createBaffles -help コマンドで表示される説明】

```
Usage: createBaffles [OPTIONS] <faceZone> <(masterPatch slavePatch)>
options:
  -additionalPatches <(patch2 .. patchN)>
                specify additional patches for creating baffles
  -case <dir>       specify alternate case directory, default is the cwd
  -internalFacesOnly
                do not convert boundary faces
  -noFunctionObjects
                do not execute functionObjects
  -overwrite        overwrite existing mesh/results files
  -parallel         run in parallel
  -region <name>    specify alternative mesh region
  -roots <(dir1 .. dirN)>
                slave root directories for distributed running
  -updateFields     update fields to include new patches: NOTE: updated field
                values may need to be edited
  -srcDoc           display source code in browser
```

```
  -doc            display application documentation in browser
  -help           print the usage
```

Makes internal faces into boundary faces.
Does not duplicate points, unlike mergeOrSplitBaffles.

【mergeOrSplitBaffles　-help コマンドで表示される説明】

```
Usage: mergeOrSplitBaffles [OPTIONS]
options:
  -case <dir>       specify alternate case directory, default is the cwd
  -detectOnly       find baffles only, but do not merge or split them
  -noFunctionObjects
                    do not execute functionObjects
  -overwrite        overwrite existing mesh/results files
  -parallel         run in parallel
  -region <name>    specify alternative mesh region
  -roots <(dir1 .. dirN)>
                    slave root directories for distributed running
  -split            topologically split duplicate surfaces
  -srcDoc           display source code in browser
  -doc              display application documentation in browser
  -help             print the usage
```

Detect faces that share points (baffles).
Merge them or duplicate the points.

【changeDictionary.C File Reference】
http://foam.sourceforge.net/docs/cpp/a03496.html

Utility to change dictionary entries, e.g. can be used to change the patch type in the field and polyMesh/boundary files. More...

Detailed Description
Utility to change dictionary entries, e.g. can be used to change the patch type in the field and polyMesh/boundary files.

Reads dictionaries (fields) and entries to change from a dictionary. E.g. to make the movingWall a fixedValue for p but all other Walls a zeroGradient boundary condition, the system/changeDictionaryDict would contain the following:

```
dictionaryReplacement
{
    p                       // field to change
    {
        boundaryField
        {
            ".*Wall"          // entry to change
            {
                type          zeroGradient;
            }
            movingWall        // entry to change
            {
                type          fixedValue;
```

```
            value           uniform 123.45;
        }
    }
  }
}
```

```
Usage
    changeDictionary [OPTION]
Parameters:
        -literalRE
    Do not interpret regular expressions; treat them as any other keyword.
        -enableFunctionEntries
    By default all dictionary preprocessing of fields is disabled
```

User Guide　3.6 Standard utilities
http://www.openfoam.org/docs/user/standard-utilities.php
createPatch
Utility to create patches out of selected boundary faces. Faces come either from existing patches or from a faceSet

topoSet
Operates on cellSets/faceSets/pointSets through a dictionary

createBaffles　Makes internal faces into boundary faces. Does not duplicate points, unlike mergeOrSplitBaffles

mergeOrSplitBaffles　Detects faces that share points (baffles). Either merge them or duplicate the points

【使われているLinuxコマンド】

mv：ファイル移動
使い方：　mv 元のファイル名　移動後のファイル名