

pyFoamの訳 (まだ途中)

Runner-Utilities

pyFoamRunner.py

OpenFOAMのソルバーを走らせる。通常 `pyFoamRunner.py <solver> <directory> <case>` のように使う。 `output` はターミナルとログファイル `PyFoamSolver.logfile` に記述される。`PyFoamSolver.analyzed` ディレクトリに以下の情報が保存される。

- 残差とソルバーに関するその他の情報
- 実行時間
- 連続性の情報
- 変数の境界情報 `bounding of variables`

pyFoamUtilityRunner.py

`output` を解析するOpenFOAMのUtilityを起動する。通常 `pyFoamUtilityRunner.py <solver> <directory> <case>` のように使う。`PyFoamUtility.analyzed` ディレクトリに各タイムステップ毎の情報が保存される。

pyFoamSteadyRunner.py

OpenFOAMの定常ソルバーを走らせる。通常 `pyFoamSteadyRunner.py <solver> <directory> <case>` のように使う。出力は `pyFoamRunner.py` と同じ。収束した場合 (しきい値を下回った場合)、停止し、解析の最後の状態が記録される。

pyFoamPlotRunner.py

OpenFOAMのソルバーを走らせる。通常 `pyFoamRunner.py <solver> <directory> <case>` のように使う。出力は、`pyFoamRunner.py`、`pyFoamSteadyRunner.py` と同じ。さらに以下のものがグラフ主として出力される。

- ソルバーの残差
- 連続性
- 変数の境界情報
- クーラン数 (オプション)
- 実行時間 (オプション)
- 反復回数 (オプション)
- arbitrary regular expressions

定常ソルバーでは、`pyFoamSteady.py` と同じように動作する。

pyFoamMeshUtilityRunner.py

いくつかのutilityでは、新規に作成したメッシュを最初のタイムステップに保存する。オリジナルのメッシュは上書きされないが、これを `constant` ディレクトリにコピーしなければならないので、不便である。

1. このutilityは
2. メッシュutilityを起動する。

作成されたメッシュを最初のタイムステップから `constant` ディレクトリにコピーする。

pyFoamPotentialRunner.py

potentialFoamを走らせ、結果をpとUに書きこむ。

potentialFoamを起動できるように、最初にsystemディレクトリのdictionariesを操作する。

potentialFoamのrun後、古いものをリセットする。結果はU, pに書き込まれる。初期のU, pはU.prepotential, p.prepotentialに書き込まれる。

pyFoamRunAtMultipleTimes.py

-timeオプションを用いることで、utilityを複数の時間で起動できる。single time-stepでしか使えないutilityに対する回避策となる。

Utilities for Logfiles

pyFoamPlotWatcher.py

OpenFOAMソルバーのlogfileを指定することで、収束情報を解析し、gnuplotのグラフに出力する。中断されるまでウォッチする。

pyFoamStandardLogAnalyzer.py

foamJobで記録されたログを解析する。_analyzedが付与されたログファイル名を指定する。