

OpenFOAM勉強会 for beginner 2

- motorBike -

2011年9月3日
TAKEDA

トライ *motorBike!*

- 動機: SimpleFoamを試す&データが大きそう
自分のノートPCで解析できるのか?

検証!

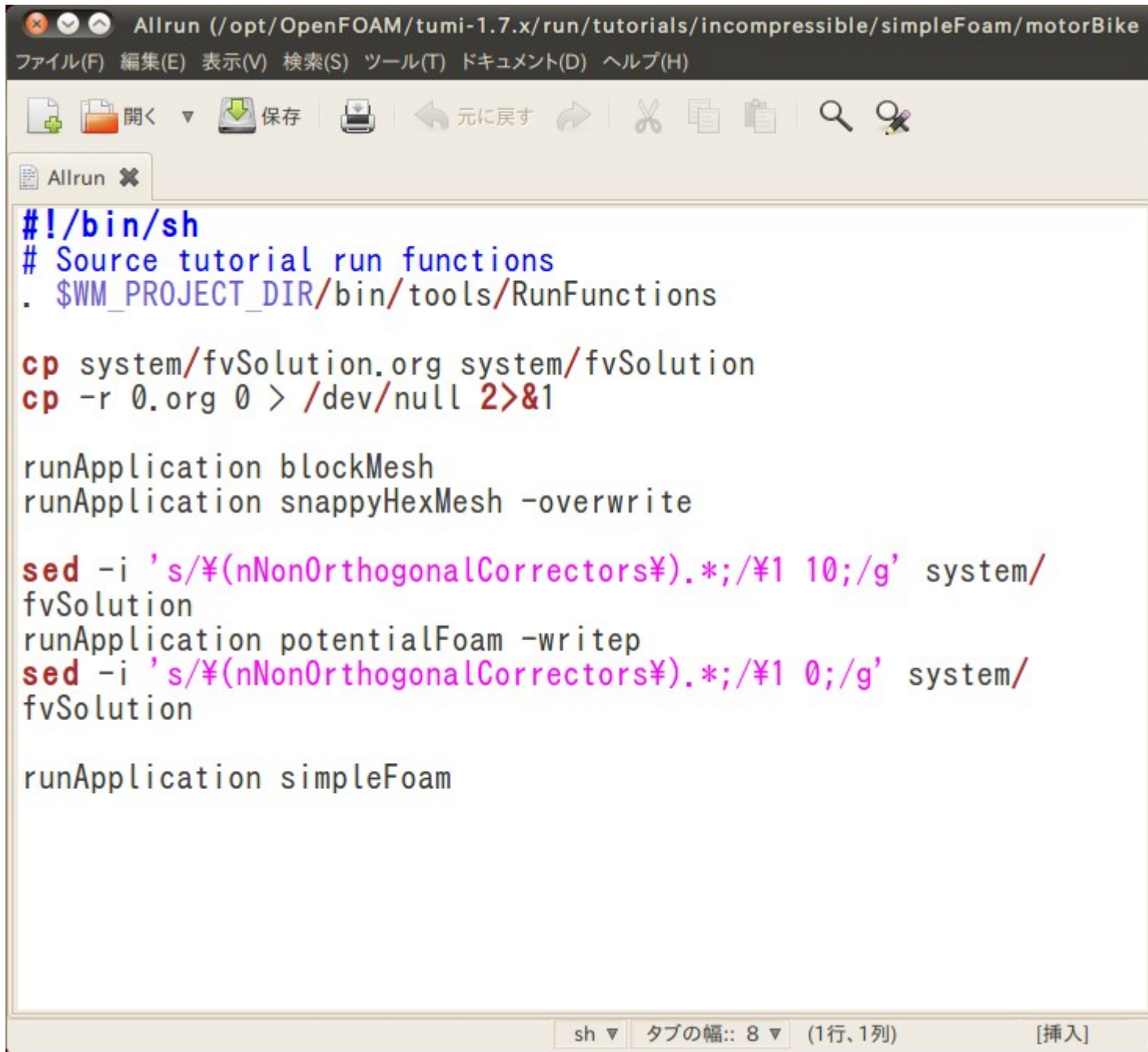
今後、どのくらい大きいモデルが解析可能?

- データの場所:
tutorials/incompressible/simpleFoam
/motorBike

My Laptop



MotorBike <Allrun>



```
Allrun (/opt/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBike)
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

[Icons: New, Open, Save, Print, Back, Forward, Copy, Paste, Find, Find & Replace]

Allrun x

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

cp system/fvSolution.org system/fvSolution
cp -r 0.org 0 > /dev/null 2>&1

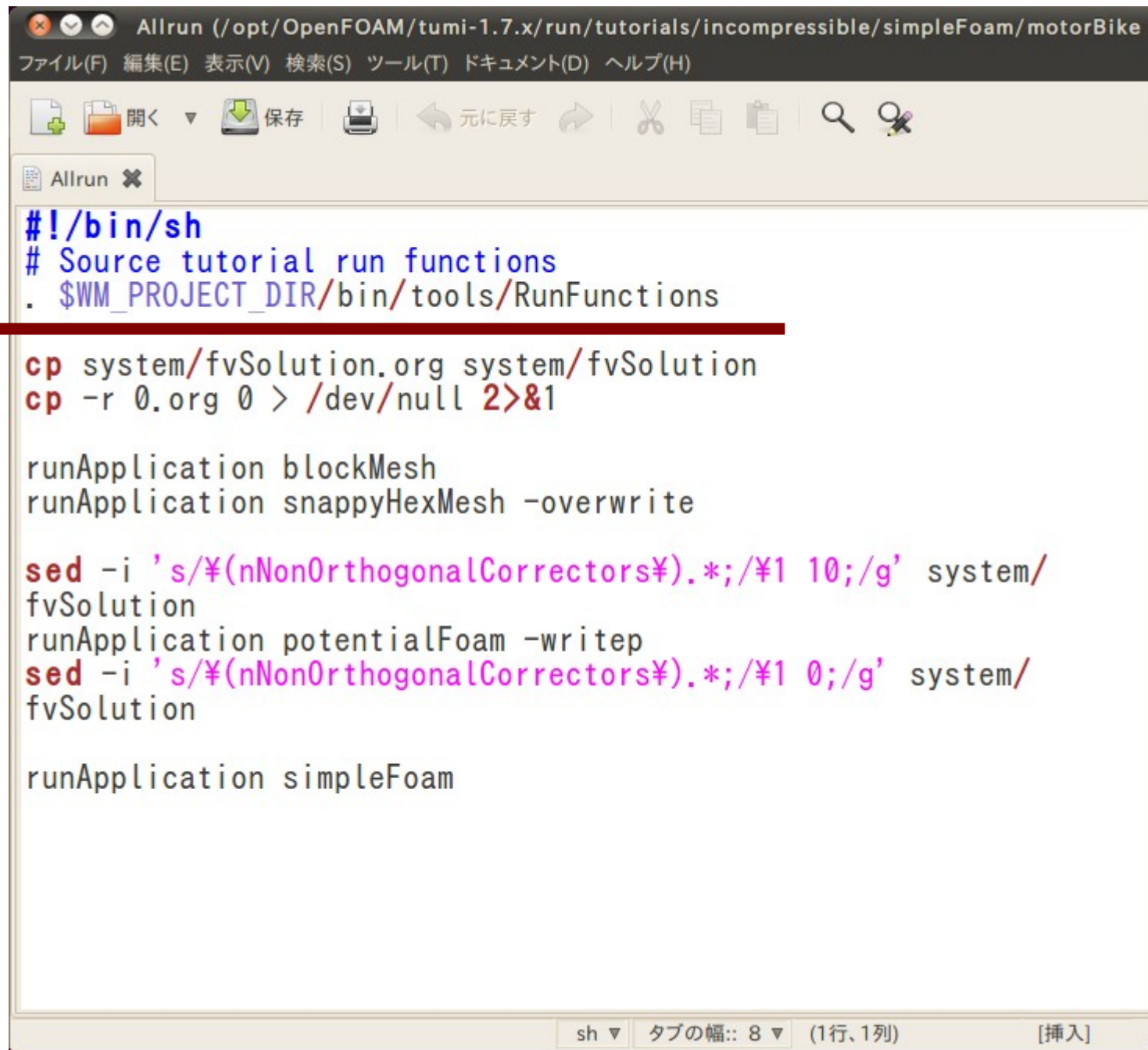
runApplication blockMesh
runApplication snappyHexMesh -overwrite

sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 10;/g' system/
fvSolution
runApplication potentialFoam -writep
sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 0;/g' system/
fvSolution

runApplication simpleFoam

sh ▼ タブの幅:: 8 ▼ (1行、1列) [挿入]
```

Allrun シェルスクリプト



The image shows a window titled "Allrun (/opt/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBike)". The window contains a shell script with the following content:

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

cp system/fvSolution.org system/fvSolution
cp -r 0.org 0 > /dev/null 2>&1

runApplication blockMesh
runApplication snappyHexMesh -overwrite

sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 10;/g' system/
fvSolution
runApplication potentialFoam -writep
sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 0;/g' system/
fvSolution

runApplication simpleFoam
```

The status bar at the bottom indicates "sh", "タブの幅: 8", "(1行, 1列)", and "[挿入]" (Insert).

. ドットコマンド

. \$WMM_PROJECT_DIR/bin/tools

/RunFunctions

- 別のスクリプトファイルを読み込む、
ここでは、RunFunctions

alias

```
tumi@tumi-desktop: ~/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBike
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

motorBike$ alias
alias app='cd $FOAM_APP'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias foam='cd $WM_PROJECT_DIR'
alias foam3rdParty='cd $WM_THIRD_PARTY_DIR'
alias foamfv='cd $FOAM_SRC/finiteVolume'
alias foamsrc='cd $FOAM_SRC/$WM_PROJECT'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias lib='cd $FOAM_LIB'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias run='cd $FOAM_RUN'
alias sol='cd $FOAM_SOLVERS'
alias src='cd $FOAM_SRC'
alias tut='cd $FOAM_TUTORIALS'
alias util='cd $FOAM_UTILITIES'
alias wm32='export WM_ARCH_OPTION=32; . $WM_PROJECT_DIR/etc/bashrc'
alias wm64='export WM_ARCH_OPTION=64; . $WM_PROJECT_DIR/etc/bashrc'
alias wmDP='export WM_PRECISION_OPTION=DP; . $WM_PROJECT_DIR/etc/bashrc'
alias wmSP='export WM_PRECISION_OPTION=SP; . $WM_PROJECT_DIR/etc/bashrc'
```


\$WM_PROJECT_DIR/bin/tools/RunFunctions

```
tumi@tumi-desktop: ~/OpenFOAM/OpenFOAM-1.7.x/bin/tools
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

tumi@tumi-desktop:~/OpenFOAM/OpenFOAM-1.7.x/bin/tools$ ls -al
合計 76
drwxr-xr-x 2 tumi tumi 4096 2010-11-11 23:22 .
drwxr-xr-x 4 tumi tumi 4096 2010-11-11 23:22 ..
-rw-r--r-- 1 tumi tumi 2726 2010-08-10 08:11 CleanFunctions
-rw-r--r-- 1 tumi tumi 114 2010-08-10 08:11 README
-rw-r--r-- 1 tumi tumi 2368 2010-08-10 08:11 RunFunctions
-rw-r--r-- 1 tumi tumi 1723 2010-08-10 08:11 doxyFilt-ignore.awk
-rw-r--r-- 1 tumi tumi 2282 2010-08-10 08:11 doxyFilt-top.awk
-rw-r--r-- 1 tumi tumi 2307 2010-08-10 08:11 doxyFilt.awk
-rw-r--r-- 1 tumi tumi 2000 2010-08-10 08:11 doxyFilt.sed
-rwxr-xr-x 1 tumi tumi 4822 2010-11-11 23:22 foamConfigurePaths
-rwxr-xr-x 1 tumi tumi 9380 2010-08-10 08:11 foamThirdParty
-rw-r--r-- 1 tumi tumi 2969 2010-08-10 08:11 gtagsrc
-rwxr-xr-x 1 tumi tumi 609 2010-08-10 08:11 inlineReplace
-rw-r--r-- 1 tumi tumi 547 2010-08-10 08:11 paraFoam.pvs
-rwxr-xr-x 1 tumi tumi 596 2010-08-10 08:11 replaceAllShellSun
-rw-r--r-- 1 tumi tumi 2248 2010-08-10 08:11 thirdPartyGetFunctions
tumi@tumi-desktop:~/OpenFOAM/OpenFOAM-1.7.x/bin/tools$
```


runApplication シェルスクリプト

```
tumi@tumi-desktop: ~/OpenFOAM/OpenFOAM-1.7.x/bin/tools
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

getApplication ()
{
    grep application system/controlDict | sed "s/application *¥([a-zA-Z]*¥);
/¥1/"
}
```

```
runApplication ()
{
    APP_RUN=$1; shift
    APP_NAME=${APP_RUN##*/}

    if [ -f log.$APP_NAME ] ; then
        echo "$APP_NAME already run on $PWD: remove log file to run"
    else
        echo "Running $APP_RUN on $PWD"
        $APP_RUN $* > log.$APP_NAME 2>&1
    fi
}
```

```
runParallel ()
{
    APP_RUN=$1; shift
```

runApplication シェルスクリプト

```
tumi@tumi-desktop: ~/OpenFOAM/OpenFOAM-1.7.x/bin/tools
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

getApplication ()
{
    grep application system/controlDict | sed "s/application *¥([a-zA-Z]*¥);
/¥1/"
}
```

```
runApplication ()
{
    APP_RUN=$1; shift
    APP_NAME=${APP_RUN##*/}

    if [ -f log.$APP_NAME ] ; then
        echo "$APP_NAME already run on $PWD: remove log file to run"
    else
        echo "Running $APP_RUN on $PWD"
        $APP_RUN $* > log.$APP_NAME 2>&1
    fi
}
```

```
runParallel ()
{
    APP_RUN=$1; shift
```

APP_RUN=\$1; shift

- \$1, \$2, \$3, : 引数を読み込む
- ; : 命令の連続実行
- shift : 引数を1つずつ前にシフトさせる。

例えば、\$2の値が\$1に代入される。

但し、シフト文を実行する前の\$1の値は\$0には代入されず破棄される。

- 例 : runApplication snappyHexMesh -overwrite
\$0 \$1 \$2 (シフト前)
\$1 (シフト後)

runApplication シェルスクリプト

```
tumi@tumi-desktop: ~/OpenFOAM/OpenFOAM-1.7.x/bin/tools
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

getApplication ()
{
    grep application system/controlDict | sed "s/application *¥([a-zA-Z]*¥);
/¥1/"
}
```

```
runApplication ()
{
    APP_RUN=$1; shift
    APP_NAME=${APP_RUN##*/}

    if [ -f log.$APP_NAME ] ; then
        echo "$APP_NAME already run on $PWD: remove log file to run"
    else
        echo "Running $APP_RUN on $PWD"
        $APP_RUN $* > log.$APP_NAME 2>&1
    fi
}
```

```
runParallel ()
{
    APP_RUN=$1; shift
```

APP_NAME=\${APP_RUN##*/}

- **##**: パターン照合演算子で、変数\${var**##**pattern}は、varの値にpatternがあれば、マッチした最初のもっとも長いpatternを削除して、残りを返す。

例: APP_RUNが、「../abc/def/blockMesh」
なら「***/**」が「**../abc/def/**」で、
\${APP_RUN**##*/**}は、「blockMesh」
だけを返す。

以上、と思うが・・・???

runApplication シェルスクリプト

```
tumi@tumi-desktop: ~/OpenFOAM/OpenFOAM-1.7.x/bin/tools
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

getApplication ()
{
    grep application system/controlDict | sed "s/application *¥([a-zA-Z]*¥);
/¥1/"
}
```

runApplication ()

```
{
    APP_RUN=$1; shift
    APP_NAME=${APP_RUN##*/}

    if [ -f log.$APP_NAME ] ; then
        echo "$APP_NAME already run on $PWD: remove log file to run"
    else
        echo "Running $APP_RUN on $PWD"
        $APP_RUN $* > log.$APP_NAME 2>&1
    fi
}
```

```
runParallel ()
```

```
{
```

```
    APP_RUN=$1; shift
```


if文の基本構文

- if 条件文 1 ; then

条件文 1 が真の時の処理

elif 条件文 2

条件文 2 が真の時の処理

else

いずれの条件も真でない時の処理

fi

runApplication シェルスクリプト

```
tumi@tumi-desktop: ~/OpenFOAM/OpenFOAM-1.7.x/bin/tools
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

getApplication ()
{
    grep application system/controlDict | sed "s/application *¥([a-zA-Z]*¥);
/¥1/"
}
```

runApplication ()

```
{
    APP_RUN=$1; shift
    APP_NAME=${APP_RUN##*/}

    if [ -f log.$APP_NAME ] ; then
        echo "$APP_NAME already run on $PWD: remove log file to run"
    else
        echo "Running $APP_RUN on $PWD"
        $APP_RUN $* > log.$APP_NAME 2>&1
    fi
}
```

```
runParallel ()
{
```

```
    APP_RUN=$1; shift
```

```
if [ -f log.$APP_NAME ] ; then
```

- [-f log.\$APP_NAME] により、log.\$APP_NAME
ファイルが存在するかどうかを判定する。
- 存在する時：
echo "\$APP_NAME already run on \$PWD:
remove log file to run" を実行する。
- 存在しない時：
echo "Running \$APP_RUN on \$PWD"
\$APP_RUN \$* > log.\$APP_NAME 2>&1
を実行する。

runApplication シェルスクリプト

```
tumi@tumi-desktop: ~/OpenFOAM/OpenFOAM-1.7.x/bin/tools
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

getApplication ()
{
    grep application system/controlDict | sed "s/application *¥([a-zA-Z]*¥);
/¥1/"
}
```

runApplication ()

```
{
    APP_RUN=$1; shift
    APP_NAME=${APP_RUN##*/}

    if [ -f log.$APP_NAME ] ; then
        echo "$APP_NAME already run on $PWD: remove log file to run"
    else
        echo "Running $APP_RUN on $PWD"
        $APP_RUN $* > log.$APP_NAME 2>&1
    fi
}
```

```
runParallel ()
```

```
{
```

```
    APP_RUN=$1; shift
```

\$APP_RUN \$* > log.\$APP_NAME 2>&1
runApplication blockMesh の場合

<shift前>

\$0 = runApplication

APP_RUN = \$1 = blockMesh

\$APP_RUN = blockMesh

<shift後>

\$0 = runApplication 、 \$1 = 「空白」

- \$* : \$0以外の引数を表示する。

ここでは、「空白」になる。

そうすると、

blockMesh > log.blockMesh 2>&1

\$APP_RUN \$* > log.\$APP_NAME 2>&1
runApplication snappyHexMesh
-overwrite の場合

<shift前>

\$0 = runApplication

APP_RUN = \$1 = snappyHexMesh

\$APP_RUN = snappyHexMesh

\$2 = -overwrite

<shift後>

\$0 = runApplication 、 \$ 1 = -overwrite

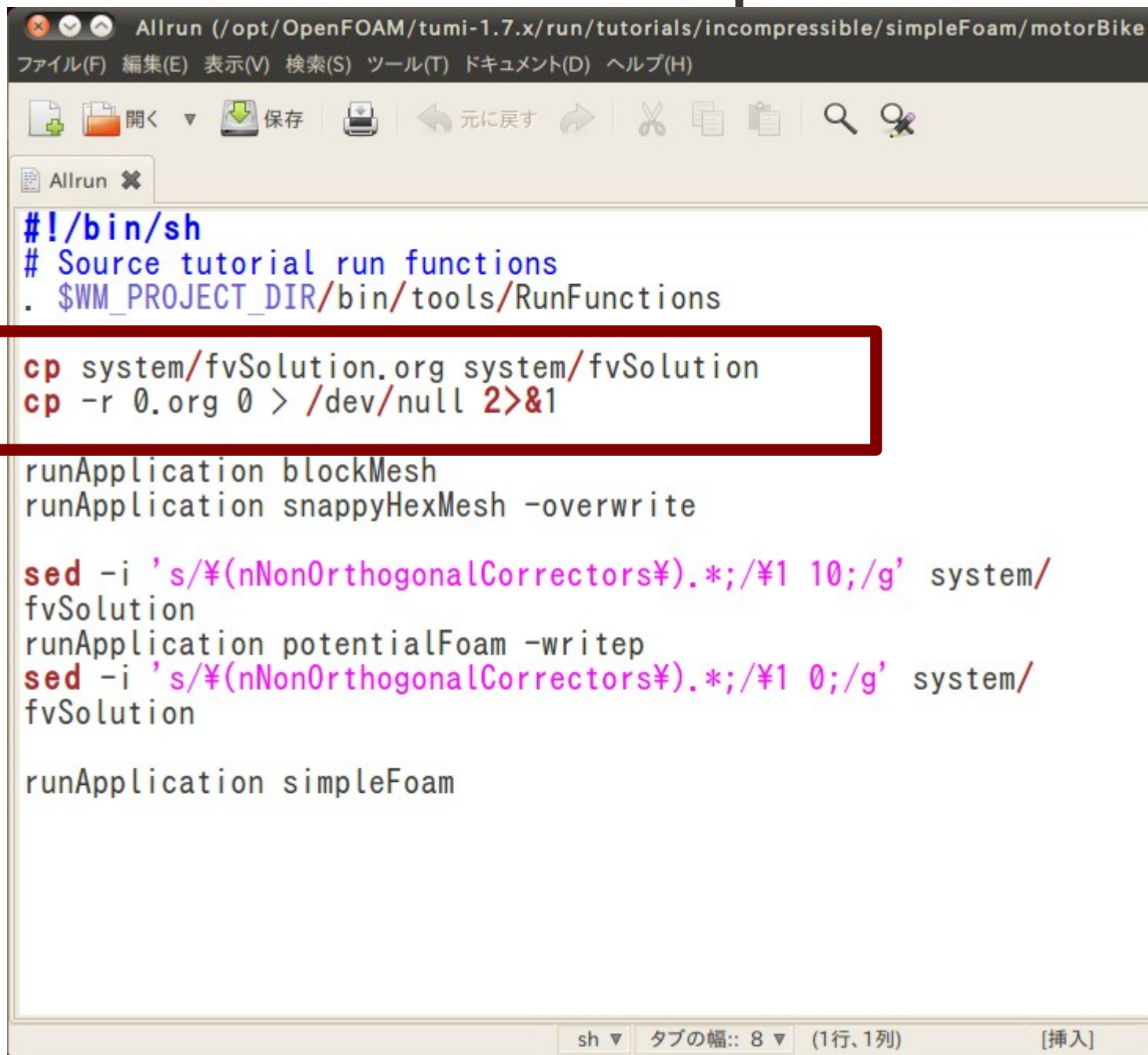
- \$* : \$0以外の引数を表示する。

ここでは、\$ 1 = -overwrite になる。

そうすると、

snappyHexMesh -overwrite > log.snappyHexMesh 2>&1

Allrun <cp>



```
Allrun (/opt/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBike)
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

[Icons: Open, Save, Print, Back, Forward, Copy, Paste, Find, Zoom]

Allrun x

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

cp system/fvSolution.org system/fvSolution
cp -r 0.org 0 > /dev/null 2>&1

runApplication blockMesh
runApplication snappyHexMesh -overwrite

sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 10;/g' system/
fvSolution
runApplication potentialFoam -writep
sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 0;/g' system/
fvSolution

runApplication simpleFoam

sh ▼ タブの幅:: 8 ▼ (1行, 1列) [挿入]
```

cp -r 0.org 0 > /dev/null 2>&1
コマンド > ファイル 2 > &1

> /dev/null :

標準出力をファイルnullに書き込む,

/devファイル → 特殊ファイルが存在する???

2>&1 :

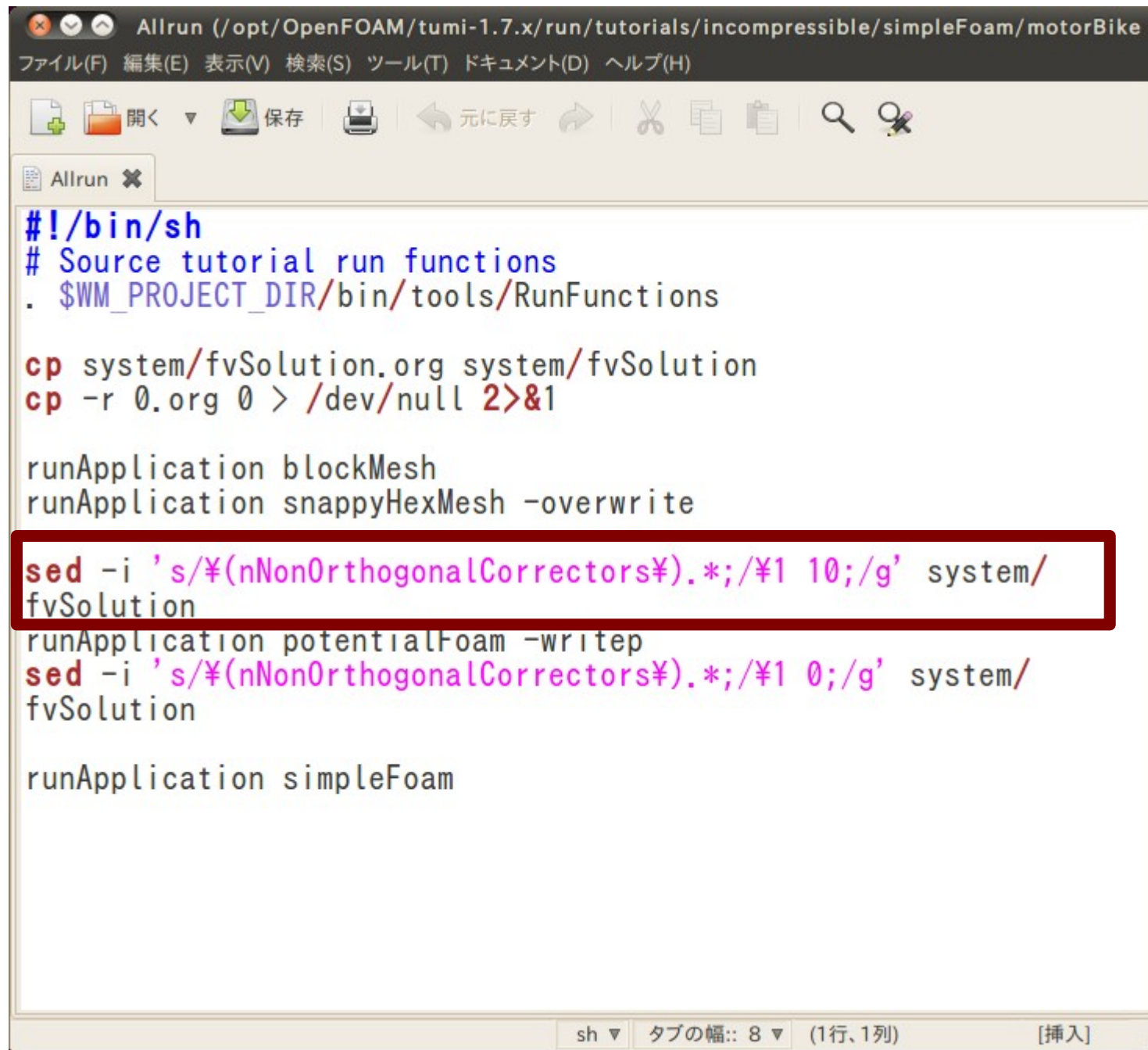
標準エラー出力(2>&1)の出力先を「&1」と記述することで、コマンドのエラー出力は標準出力と一緒にファイルに書き込まれる。

標準入力(キーボード入力)=0

標準出力(コンソール(ディスプレイ)画面出力)=1

標準エラー出力=2

Allrun <sed>



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

cp system/fvSolution.org system/fvSolution
cp -r 0.org 0 > /dev/null 2>&1

runApplication blockMesh
runApplication snappyHexMesh -overwrite

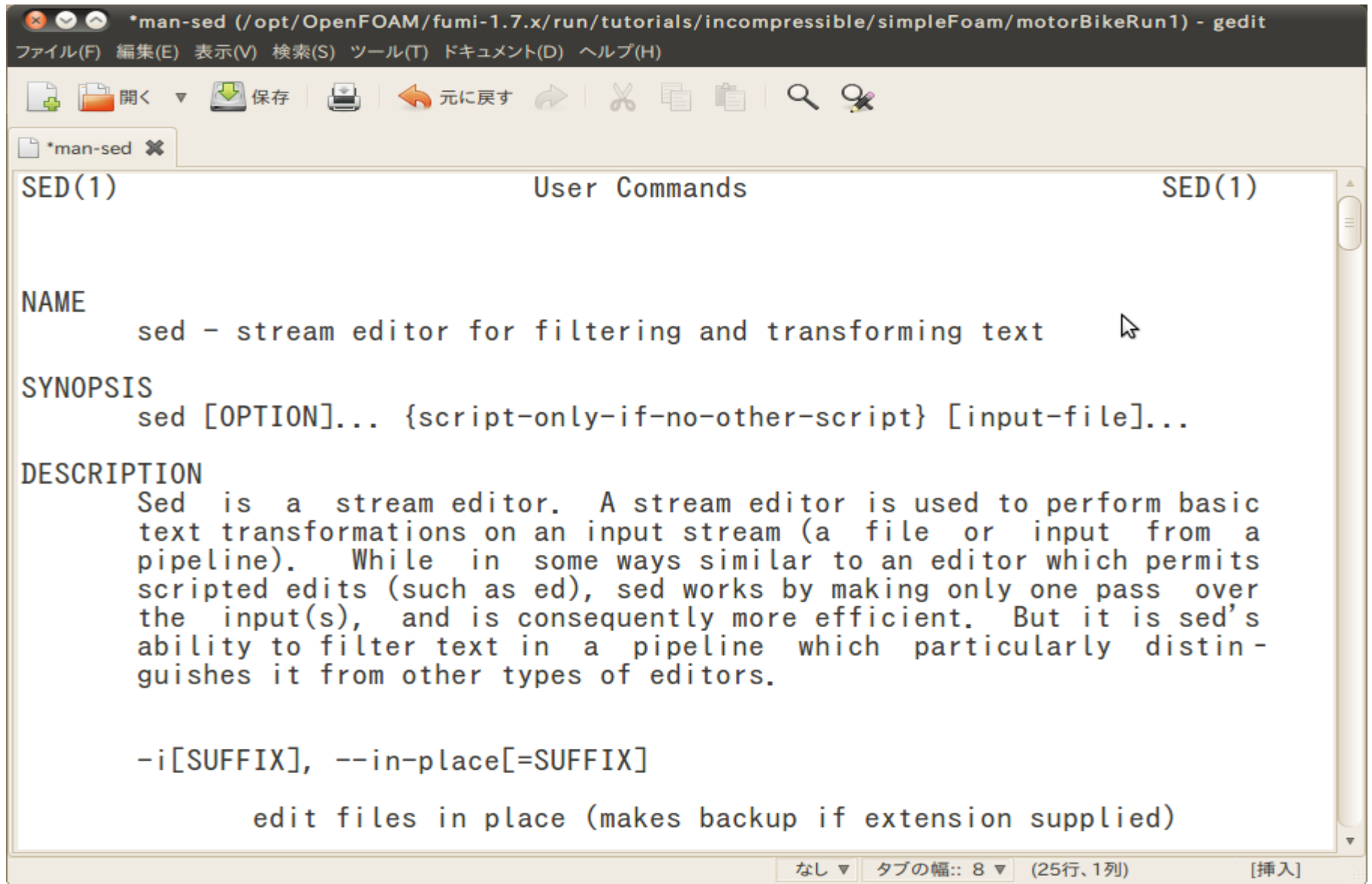
sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 10;/g' system/
fvSolution
runApplication potentialFoam -writep
sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 0;/g' system/
fvSolution

runApplication simpleFoam
```

sh ▾ タブの幅: 8 ▾ (1行, 1列) [挿入]

< man sed > ストリームエディター

```
sed -i 's/¥(nNonOrthogonalCorrectors¥).*/¥1 10;/g'  
system/fvSolution
```



```
*man-sed (/opt/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeRun1) - gedit  
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)  
開く 保存 元に戻す  
SED(1) User Commands SED(1)  
  
NAME  
    sed - stream editor for filtering and transforming text  
  
SYNOPSIS  
    sed [OPTION]... {script-only-if-no-other-script} [input-file]...  
  
DESCRIPTION  
    Sed is a stream editor. A stream editor is used to perform basic  
    text transformations on an input stream (a file or input from a  
    pipeline). While in some ways similar to an editor which permits  
    scripted edits (such as ed), sed works by making only one pass over  
    the input(s), and is consequently more efficient. But it is sed's  
    ability to filter text in a pipeline which particularly distin-  
    guishes it from other types of editors.  
  
    -i[SUFFIX], --in-place[=SUFFIX]  
        edit files in place (makes backup if extension supplied)
```

system/fvSolution ファイル

```
sed -i 's/¥(nNonOrthogonalCorrectors¥).*/¥1 10;/g'
system/fvSolution
```

- diff 実行前ファイル 実行後ファイル

63c63

< nNonOrthogonalCorrectors 0; 前

> nNonOrthogonalCorrectors 10; 後

非直交の補正数 (U-124) 意味不明????

system/FvSolution ファイル (potentialFoam実行前)

```
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeRun5sed/system
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

{
    solver            smoothSolver;
    smoother          GaussSeidel;
    tolerance         1e-8;
    relTol             0.1;
    nSweeps            1;
};

}

SIMPLE
{
    nNonOrthogonalCorrectors 10;
}

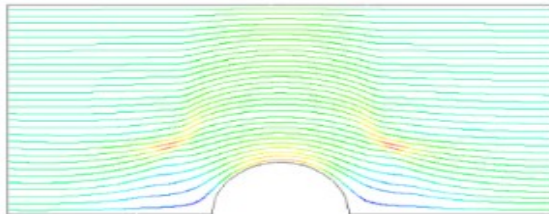
relaxationFactors
{
    p                0.3;
    U                0.7;
    k                0.7;
    omega            0.7;
}

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

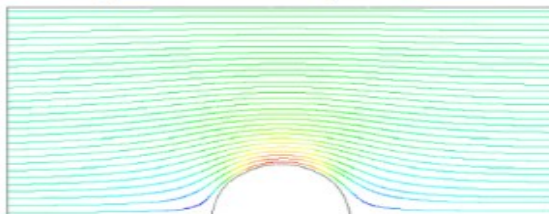

nNonOrthogonalCorrectors

```
01 runTimeModifiable yes;  
02  
03  
04 // *****
```

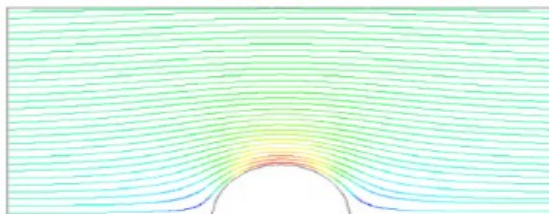
potentialFoam executes an iterative loop around the pressure equation which it solves in order that explicit terms relating to non-orthogonal correction in the Laplacian term may be updated in successive iterations. The number of iterations around the pressure equation is controlled by the `nNonOrthogonalCorrectors` keyword in *controlDict*. In the first instance we can set `nNonOrthogonalCorrectors` to 0 so that no loops are performed, i.e. the pressure equation is solved once, and there is no non-orthogonal correction. The solution is shown in Figure 3.3(a) (at $t = 1$, when the steady-state simulation is complete).



(a) With no non-orthogonal correction



(b) With non-orthogonal correction



(c) Analytical solution

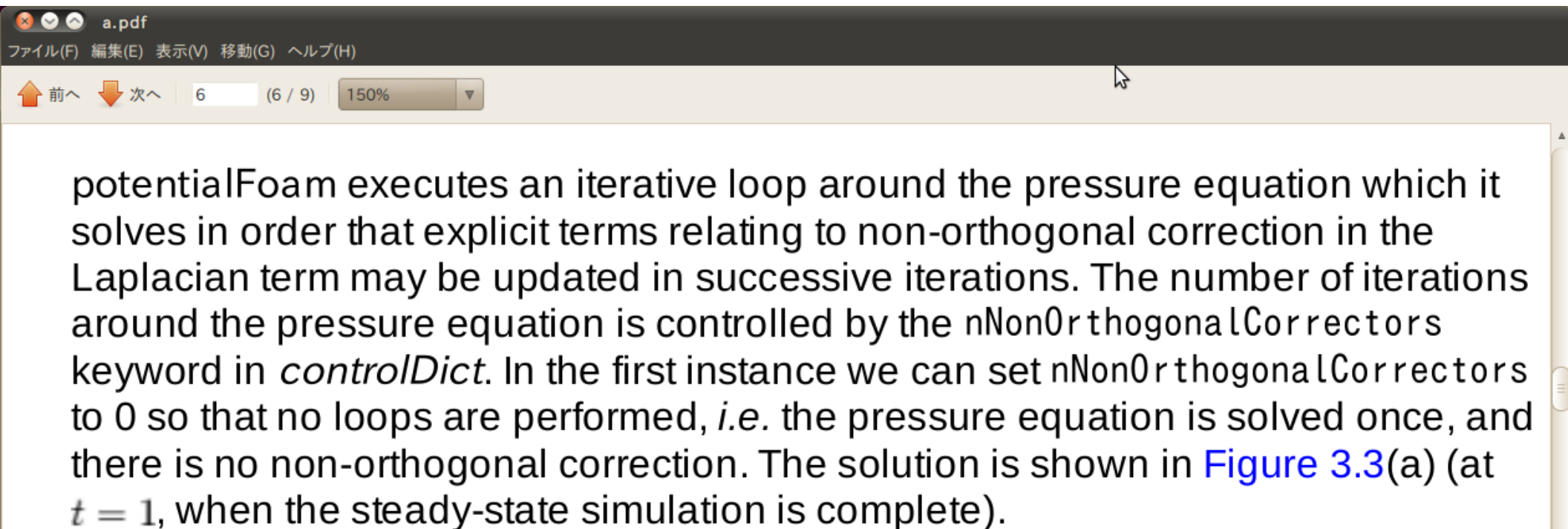
Figure 3.3: Streamlines of potential flow

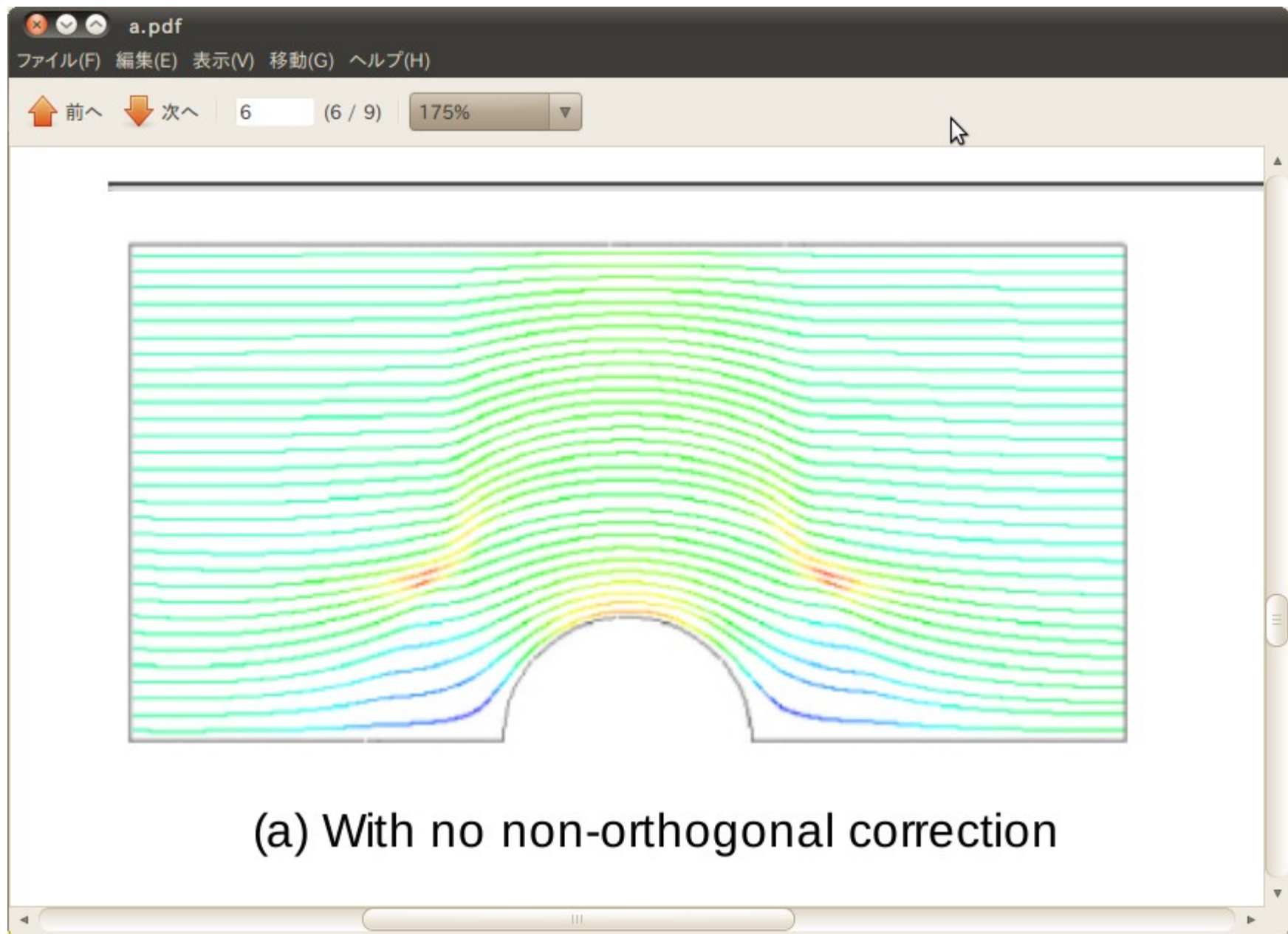
- <http://www.foamcfd.org/Nabla/guides/ProgrammersGuidese12.html>

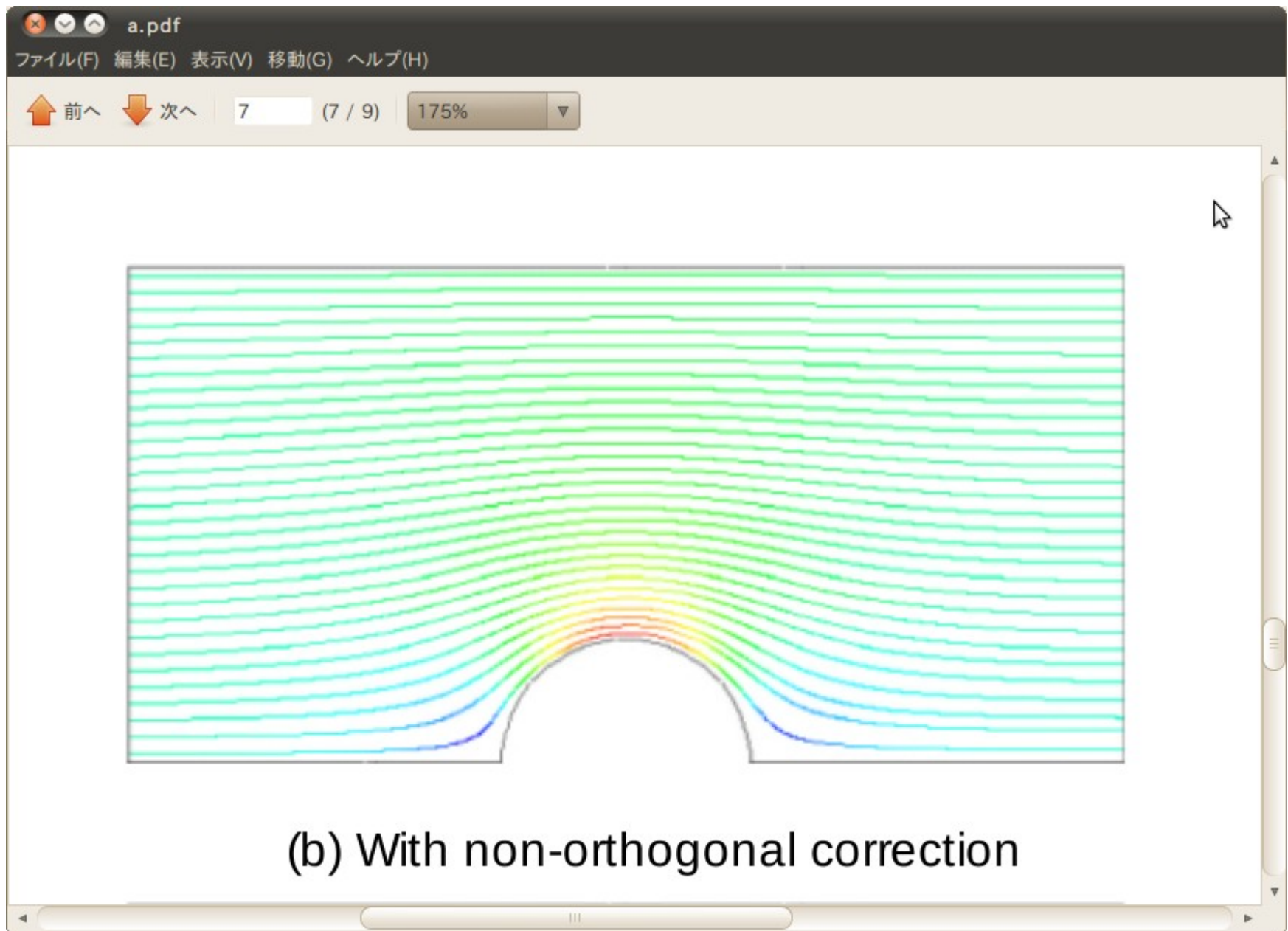


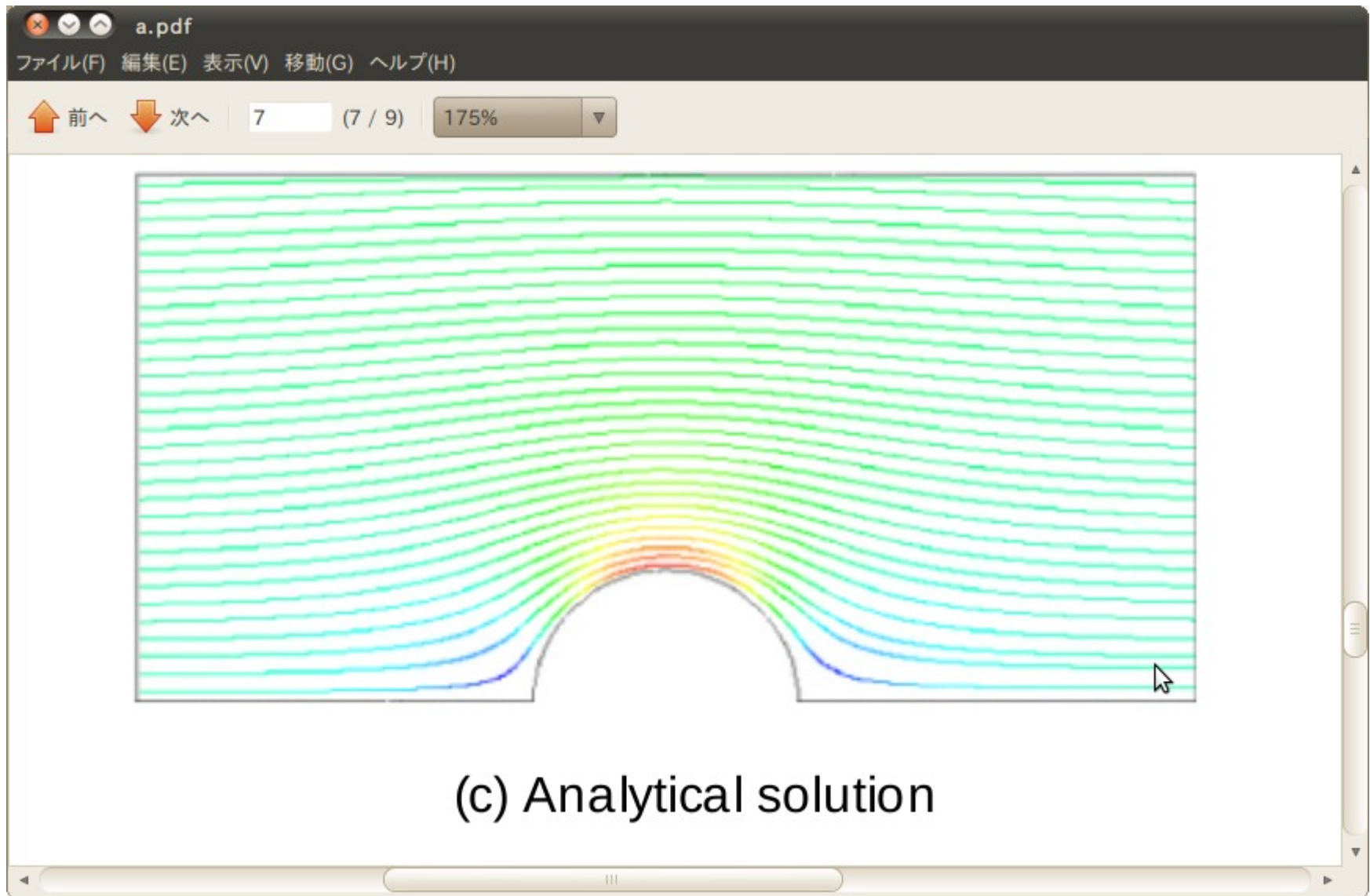
potentialFoam

- 圧力方程式において反復計算される。
- 反復数は”nNonOrthogonalCorrectors”により制御される。

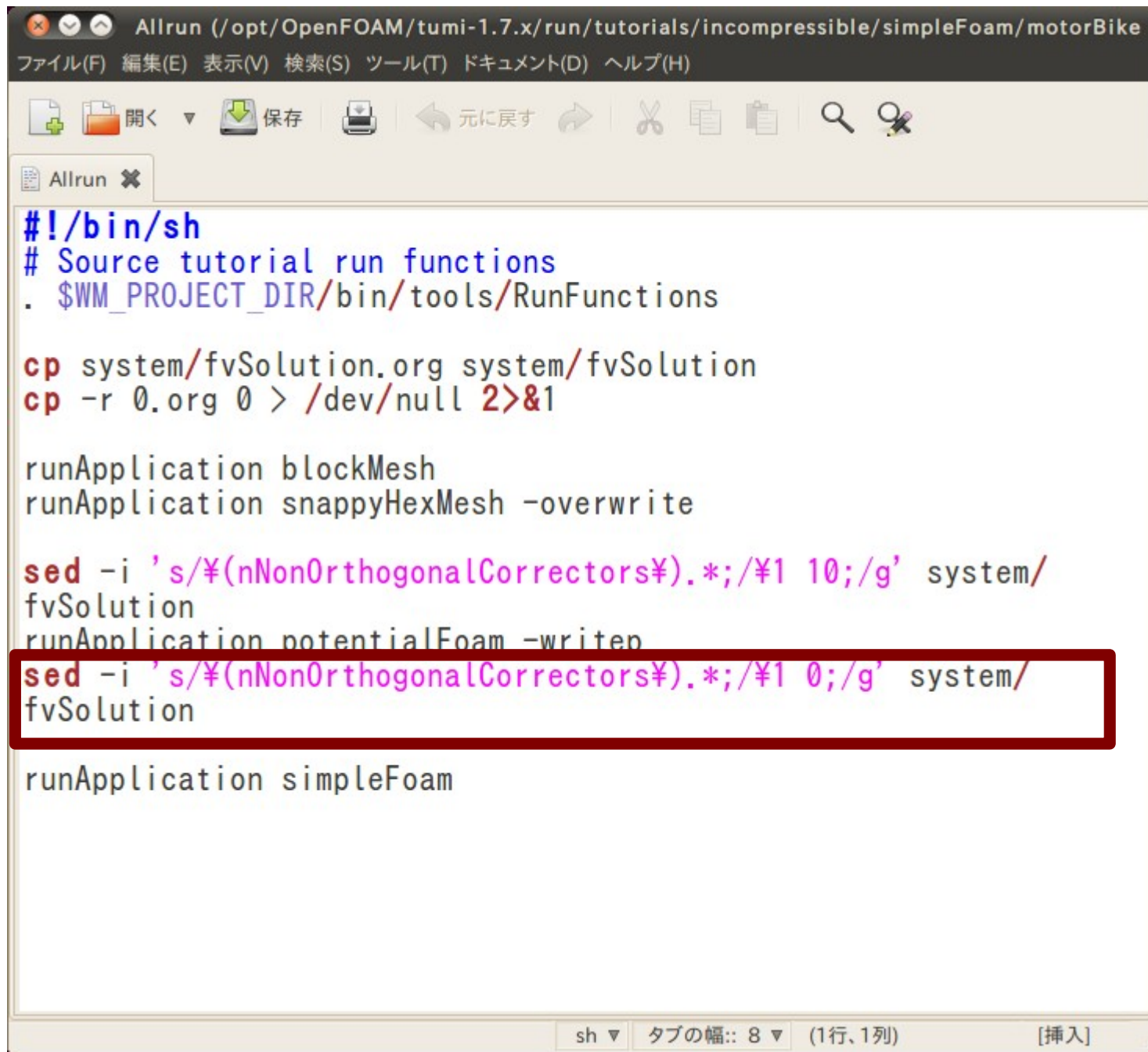








MotorBike <Allrun>



```
Allrun (/opt/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBike)
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

開く ▼ 保存 元に戻す

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

cp system/fvSolution.org system/fvSolution
cp -r 0.org 0 > /dev/null 2>&1

runApplication blockMesh
runApplication snappyHexMesh -overwrite

sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 10;/g' system/
fvSolution
runApplication potentialFoam -write
sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 0;/g' system/
fvSolution

runApplication simpleFoam
```

sh ▼ タブの幅:: 8 ▼ (1行, 1列) [挿入]

system/FvSolution ファイル (potentialFoam実行後 =simpleFoam実行前)

```
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeRun7sed/system
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)

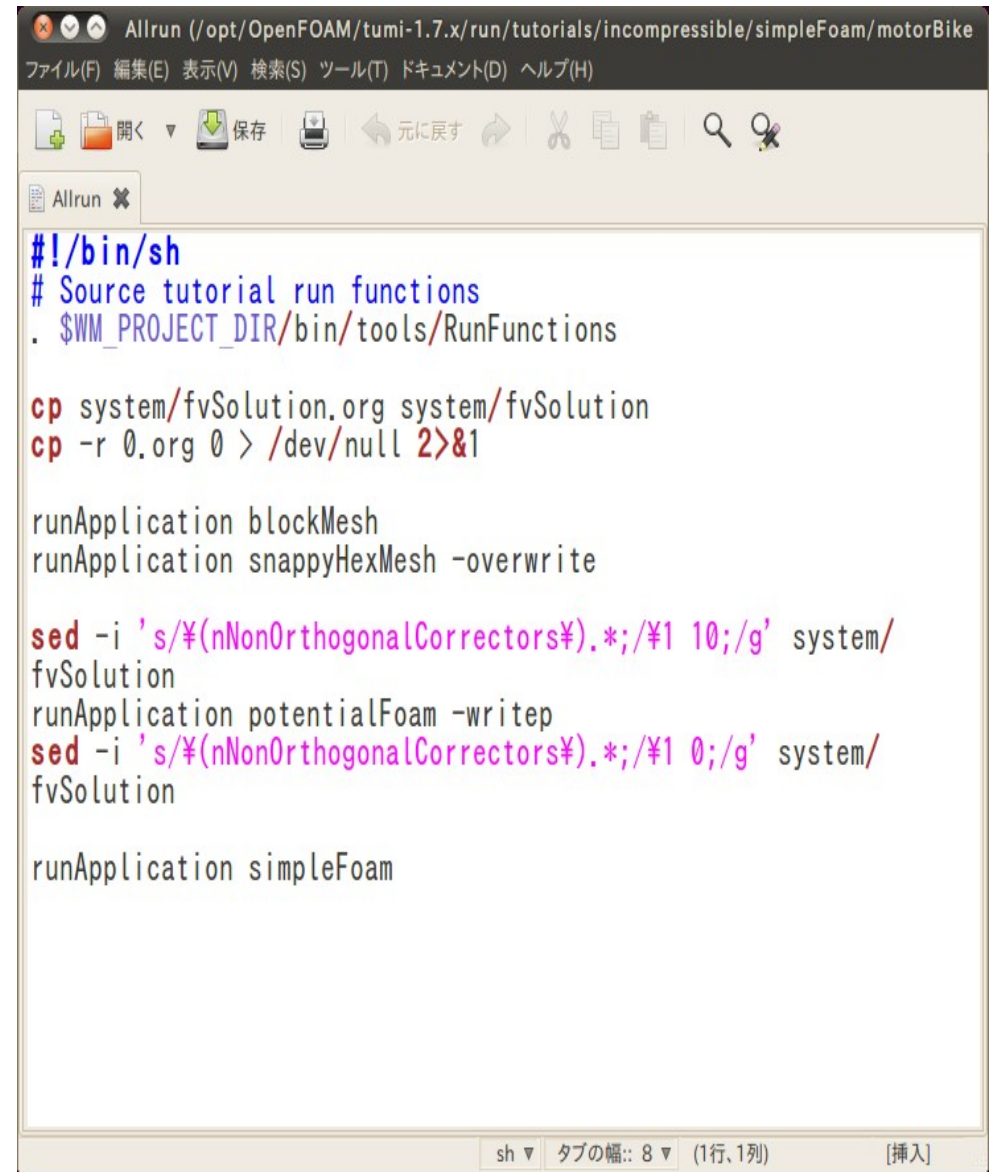
omega
{
    solver          smoothSolver;
    smoother        GaussSeidel;
    tolerance       1e-8;
    relTol          0.1;
    nSweeps         1;
};

SIMPLE
{
    nNonOrthogonalCorrectors 0;
}

relaxationFactors
{
    p              0.3;
    U              0.7;
    k              0.7;
    omega          0.7;
}
```

解析の流れ

- メッシュ作成
 - 1)blockMesh
 - 2)snappyHexMesh
- 初期値設定
 - 3)potentialFoam
- 解析
 - 4)simpleFoam



The screenshot shows a terminal window titled "Allrun (/opt/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBike)". The window contains the following commands and their outputs:

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

cp system/fvSolution.org system/fvSolution
cp -r 0.org 0 > /dev/null 2>&1

runApplication blockMesh
runApplication snappyHexMesh -overwrite

sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 10;/g' system/
fvSolution
runApplication potentialFoam -writep
sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 0;/g' system/
fvSolution

runApplication simpleFoam
```

The terminal window has a menu bar with options: ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H). The status bar at the bottom shows "sh", "タブの幅: 8", "(1行, 1列)", and "[挿入]".

入力データ (blockMesh前)

```
tumi@tumi-desktop: ~/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeRun1R$ ls ../constant/ -la
合計 24
drwxr-xr-x 4 tumi tumi 4096 2011-08-22 15:25 .
drwxr-xr-x 6 tumi tumi 4096 2011-08-22 15:26 ..
-rw-r--r-- 1 tumi tumi 920 2011-08-22 15:25 RASProperties
drwxr-xr-x 2 tumi tumi 4096 2011-08-22 15:25 polyMesh
-rw-r--r-- 1 tumi tumi 917 2011-08-22 15:25 transportProperties
drwxr-xr-x 2 tumi tumi 4096 2011-08-22 15:25 triSurface
tumi@tumi-desktop: ~/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeRun1R$ ls ../constant/polyMesh/ -al
合計 24
drwxr-xr-x 2 tumi tumi 4096 2011-08-22 15:25 .
drwxr-xr-x 4 tumi tumi 4096 2011-08-22 15:25 ..
-rw-r--r-- 1 tumi tumi 1460 2011-08-22 15:25 blockMeshDict
-rw-r--r-- 1 tumi tumi 10563 2011-08-22 15:25 boundary
tumi@tumi-desktop: ~/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeRun1R$ ls ../system/ -la
合計 44
drwxr-xr-x 2 tumi tumi 4096 2011-08-22 15:26 .
drwxr-xr-x 6 tumi tumi 4096 2011-08-22 15:26 ..
-rw-r--r-- 1 tumi tumi 1237 2011-08-22 15:25 controlDict
-rw-r--r-- 1 tumi tumi 1292 2011-08-22 15:25 decomposeParDict
-rw-r--r-- 1 tumi tumi 1600 2011-08-22 15:25 fvSchemes
-rw-r--r-- 1 tumi tumi 1913 2011-08-22 15:26 fvSolution
-rw-r--r-- 1 tumi tumi 1913 2011-08-22 15:25 fvSolution.org
-rw-r--r-- 1 tumi tumi 15197 2011-08-22 15:25 snappyHexMeshDict
tumi@tumi-desktop: ~/OpenFOAM/tumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeRun1R$
```

Sh Allrun or ./Allrun (1 core)

```
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeAllrun2$ sh Allrun
Running blockMesh on /home/fumi/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeAllrun2
Running snappyHexMesh on /home/fumi/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeAllrun2
Running potentialFoam on /home/fumi/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeAllrun2
Running simpleFoam on /home/fumi/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeAllrun2
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeAllrun2$ ls
0          200    500    constant    log.simpleFoam
0.org     300    Allclean  log.blockMesh  log.snappyHexMesh
100      400    Allrun    log.potentialFoam  system
```


log.simpleFoam (1337秒= 約22分)

```
log.simpleFoam (/opt/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeAllrun2) - gedi
ファイル(F) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

開く ▼ 保存 元に戻す

log.simpleFoam ✕

Time = 500

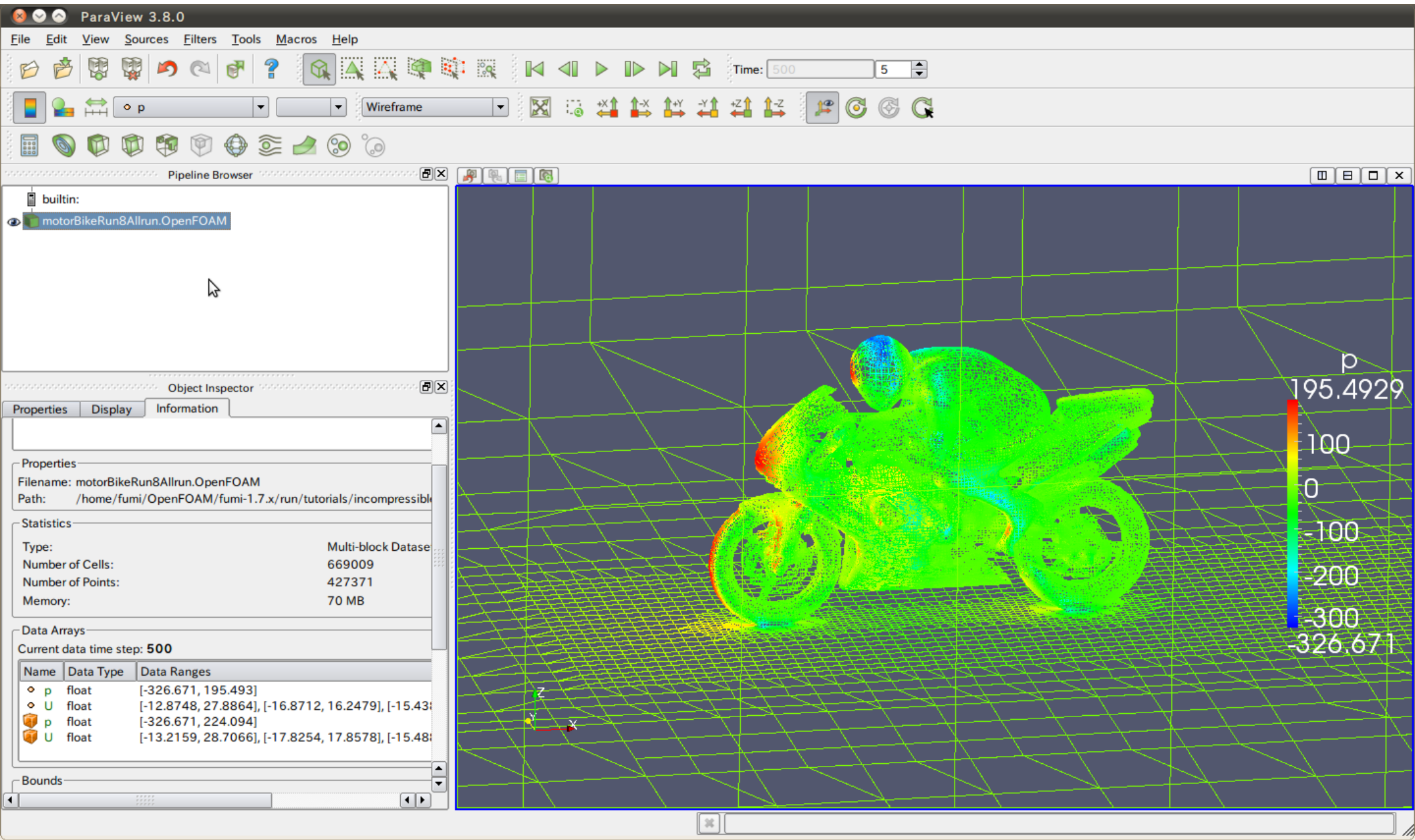
smoothSolver: Solving for Ux, Initial residual = 0.000239719, Final residual
= 1.10869e-05, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 0.00455322, Final residual
= 0.000215511, No Iterations 4
smoothSolver: Solving for Uz, Initial residual = 0.00515144, Final residual
= 0.000236923, No Iterations 4
GAMG: Solving for p, Initial residual = 0.0208451, Final residual =
0.000704974, No Iterations 2
time step continuity errors : sum local = 0.00011244, global = -1.78843e-06,
cumulative = -0.0126413
smoothSolver: Solving for omega, Initial residual = 6.43981e-05, Final
residual = 4.01627e-06, No Iterations 3
smoothSolver: Solving for k, Initial residual = 0.000876851, Final residual
= 6.50204e-05, No Iterations 3
ExecutionTime = 1336.09 s ClockTime = 1337 s

End
```

C ▼ タブの幅: 8 ▼ (1行、1列) [挿入]

ParaFoam Cells:669,009(約67万)

Points:427,371(約43万)




```
mpirun -np 4 simpleFoam -parallel
```

4coreでの解法

Allrun 一部変更

```
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBike
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
#!/bin/sh
# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

cp system/fvSolution.org system/fvSolution
cp -r 0.org 0 > /dev/null 2>&1

runApplication blockMesh
runApplication snappyHexMesh -overwrite

sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 10;/g' system/fvSolution
runApplication potentialFoam -writep
sed -i 's/$(nNonOrthogonalCorrectors$).*/$1 0;/g' system/fvSolution

decomposePar
mpirun -np 4 simpleFoam -parallel | tee log.simpleFoam
reconstructPar
```

Allrun後

```
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeDecompose$ ls
0          400          constant          processor0  run2
0.org      500          log.blockMesh     processor1  run3
100        Allclean    log.potentialFoam processor2   run4
200        Allrun      log.simpleFoam     processor3  run5
300        AllrunOrig  log.snappyHexMesh  run1        system
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeDecompose$ _
```

log.simpleFoam

(723秒= 約12分, 1core約22分)

```
fumi@fumi-desktop: ~/OpenFOAM/fumi-1.7.x/run/tutorials/incompressible/simpleFoam/motorBikeDecompose
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
ExecutionTime = 703.09 s  ClockTime = 719 s

Time = 500

smoothSolver: Solving for Ux, Initial residual = 0.000217543, Final residual = 1.06315e-05, No Iterations 4
smoothSolver: Solving for Uy, Initial residual = 0.00402833, Final residual = 0.000204182, No Iterations 4
smoothSolver: Solving for Uz, Initial residual = 0.00471395, Final residual = 0.00023341, No Iterations 4
GAMG: Solving for p, Initial residual = 0.0162106, Final residual = 0.000528995, No Iterations 2
time step continuity errors : sum local = 8.45725e-05, global = -5.8603e-07, cumulative = -0.00193854
smoothSolver: Solving for omega, Initial residual = 5.05247e-05, Final residual = 3.42191e-06, No Iterations 3
smoothSolver: Solving for k, Initial residual = 0.000755423, Final residual = 6.26567e-05, No Iterations 3
ExecutionTime = 707.11 s  ClockTime = 723 s

End

Finalising parallel run
```