

DEXCS2009-OpenFOAMの基礎

DEXCS研究会

はじめに

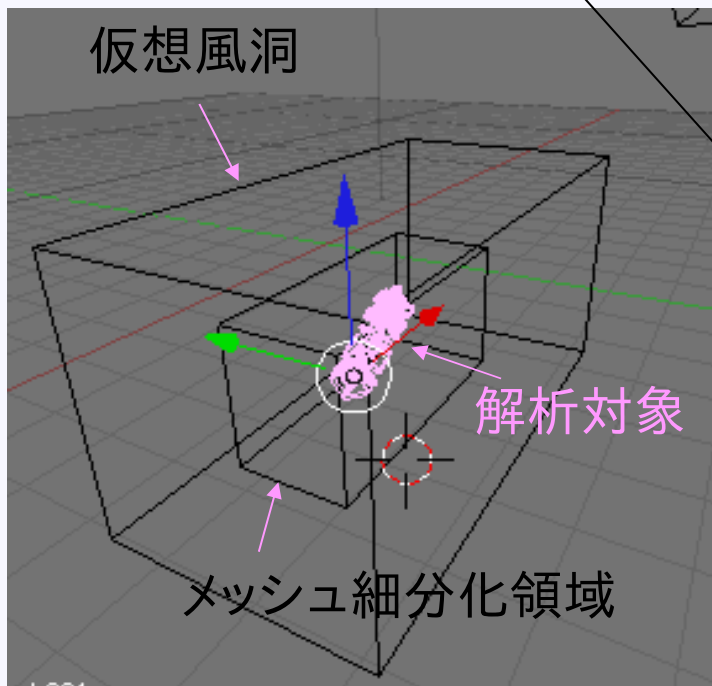
DEXCS2009-OpenFOAMの嬉しさと、
本講習会の狙いについて説明します

DEXCS2009-OpenFOAMの嬉しさ

OpenFOAMのインストール不要

simpleFoam: 定常非圧縮流れ解析
($k-\varepsilon$ 系乱流モデル使用可能)

仮想風洞試験が、ボタンを順番に押していくだけで実行可能



コマンドライン入力は一切不要
GUIエディタでパラメタ変更可能

(パラメタファイルの収納場所・名前を知らなくとも、
解析シーン毎にボタンを順番に押していけばよい)

推奨利用手順

1. セットアップ
2. ランチャー起動⇒ヘルプ⇒ランチャーの使い方
3. 標準モデルを使ったパラメタスタディ
4. 形状変更のケーススタディ
5. 応用・・・

しかし・・・

- 形状変更、試験品だけでなく、風洞形状も変更したい
- 内部/外部流れは？
- simpleFOAM以外は？
- ランチャーのカスタマイズって？
- 汎用CAD,PrePostソフトとのインタフェース

DEXCS-OpenFOAMの基礎知識が前提条件

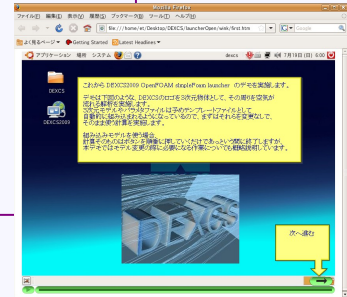
本講習の狙い

講習A-1 DEXCS-OpenFOAMの基礎知識

- 第二回オープンソースCAEワークショップ発表資料「DEXCS-OpenFOAMの紹介」(PDF) 

DEXCSとOpenFOAMの関係
解析手順、ファイル構成、組込ツール

DEXCSによる解析手順の説明
(各ステップ、ボタン操作で、
何をやっているのか)



1. セットアップ

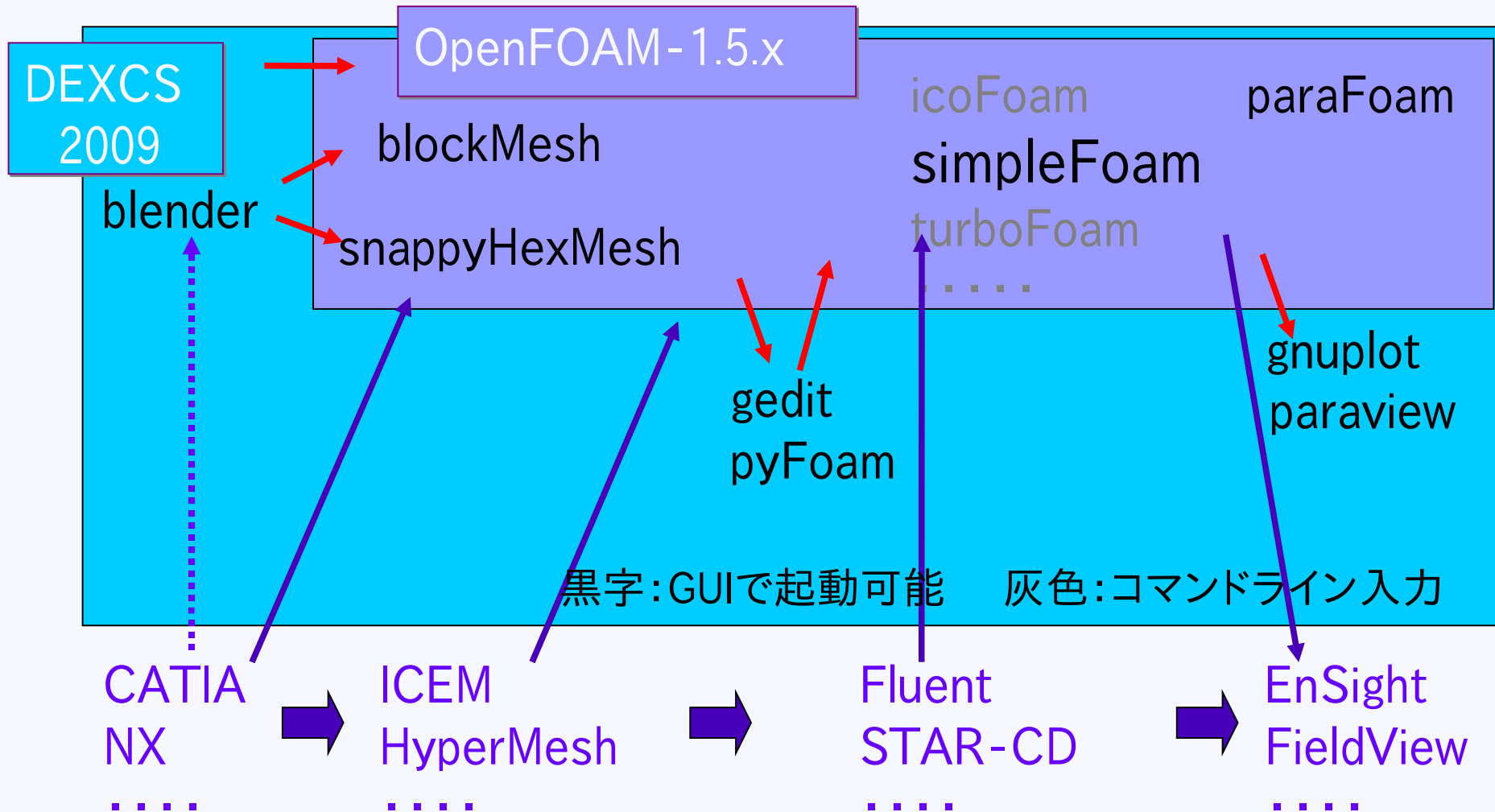
- ・ランチャー起動⇒ヘルプ⇒ランチャーの使い方
- ・標準モデルを使ったパラメタスタディ

講習A-2 4. 形状変更のケーススタディ

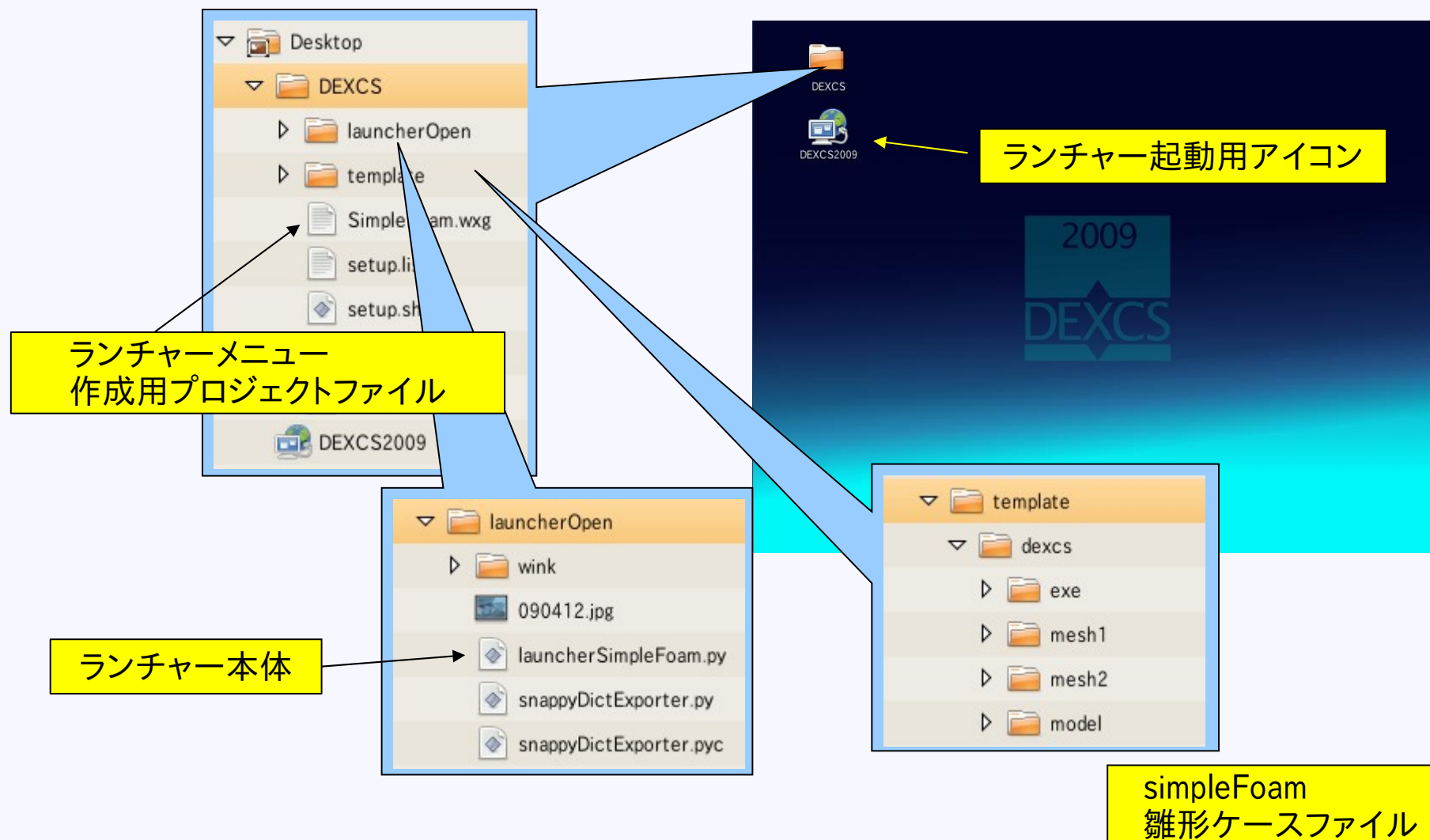
DEXCSとOpenFOAMの関係

解析の流れ

形状作成 メッシュ作成 境界条件 計算実行 後処理



DEXCSのファイル構成



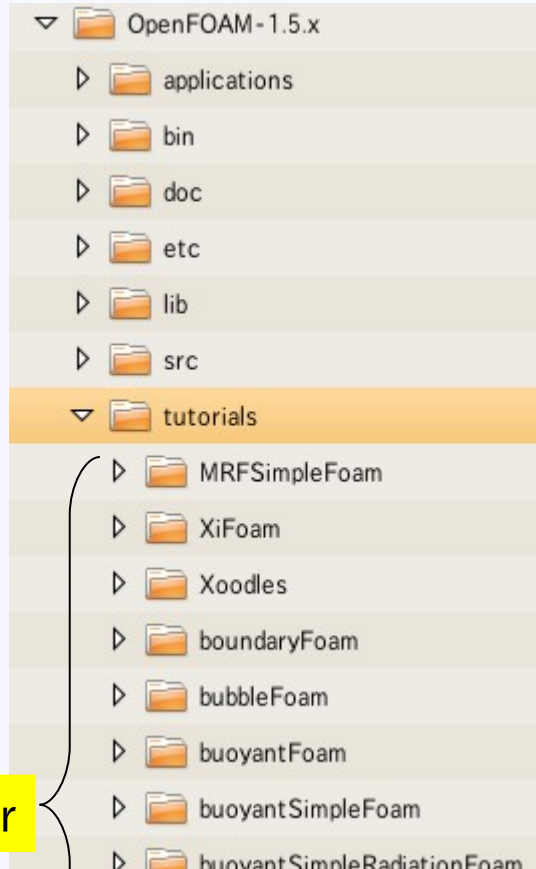
OpenFOAMのファイル構成

第4章 OpenFOAMのケース

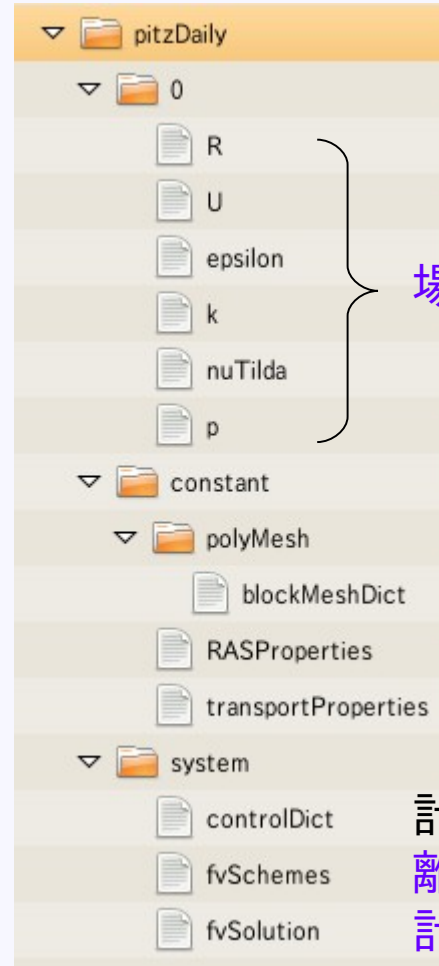
- 4.1 OpenFOAMのケースのファイル構造
- 4.2 基本的な入出力ファイルのフォーマット
- 4.3 時間とデータの入出力制御
- 4.4 数値スキーム
- 4.5 解法とアルゴリズム 制御

<case>

計算前



solvers



場の変数

メッシュデータ

モデル定数

計算制御パラメタ

離散化スキーム

計算解法、収束条件

<case>

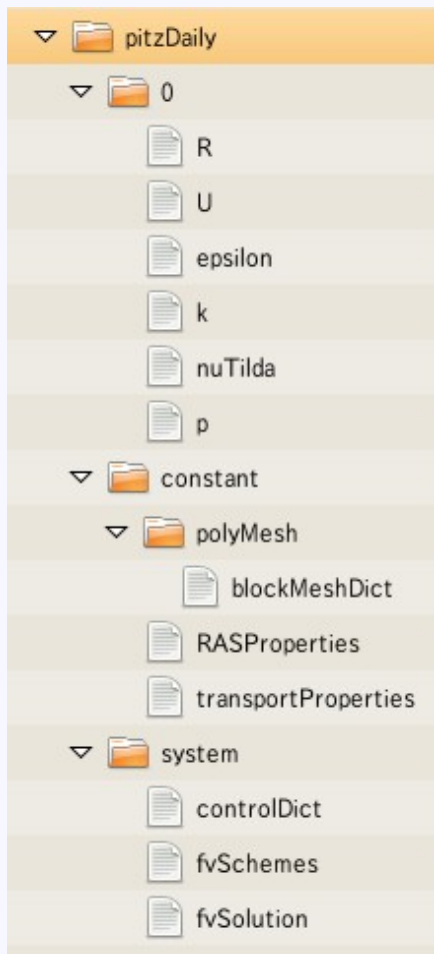
ソルバーに応じて

ファイル名称や定義内容が異なる

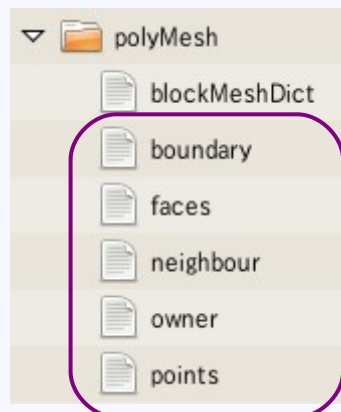
OpenFOAMのファイル構成

<case>

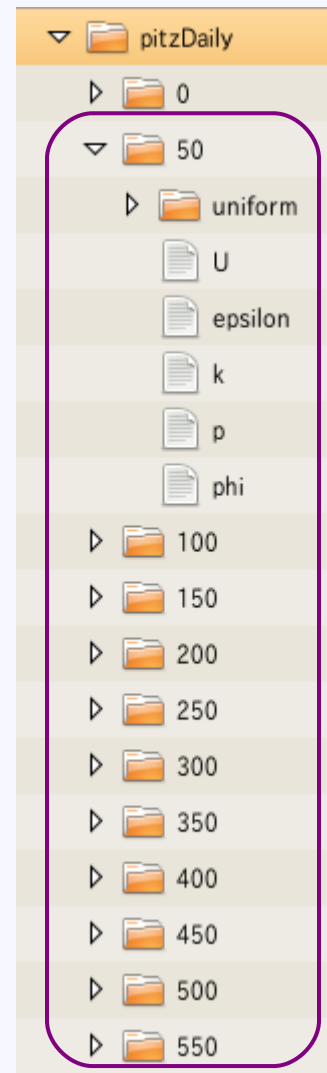
計算前



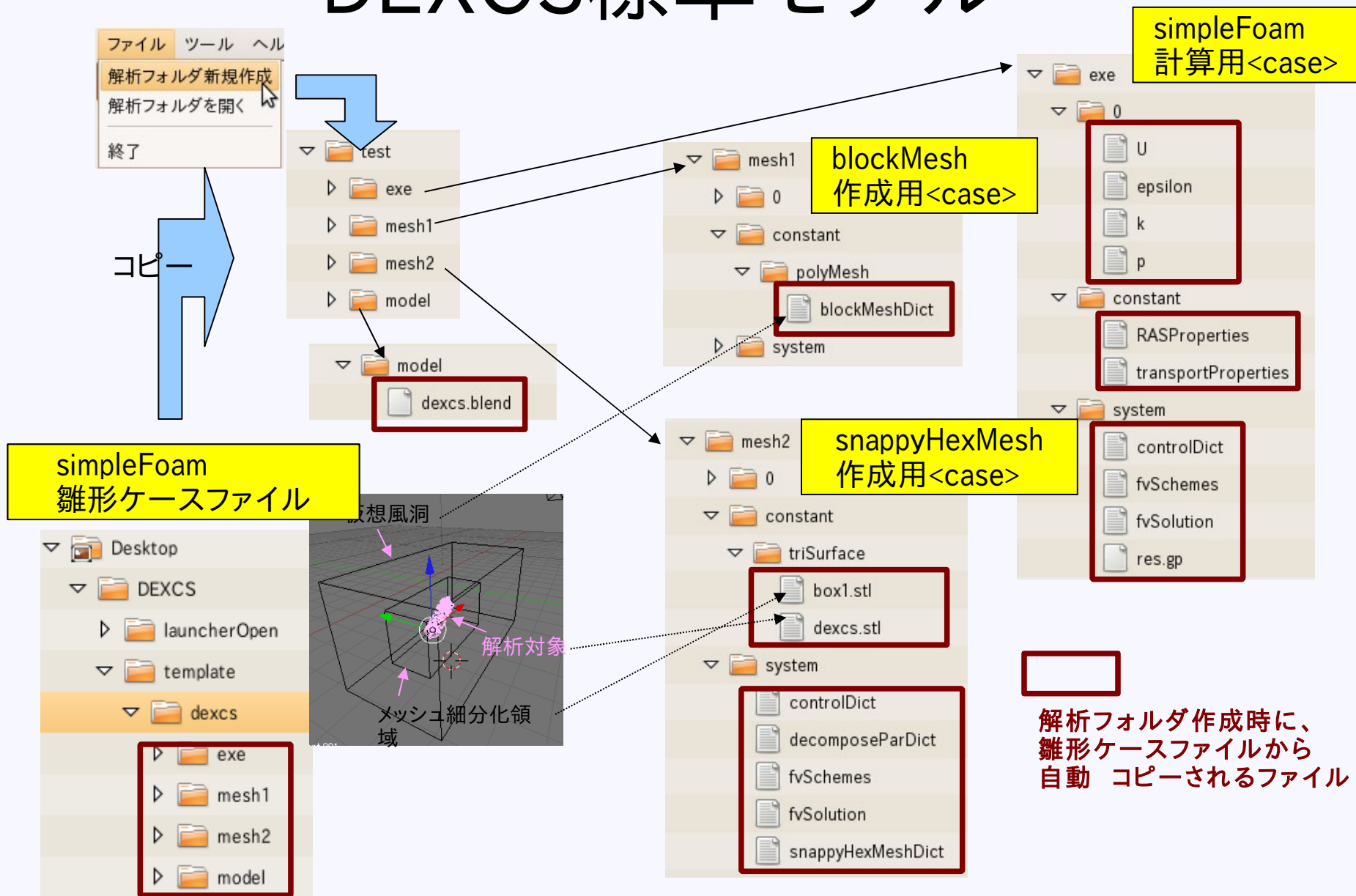
メッシュ作成後



計算後



DEXCS標準モデル



DEXCSの組み込みツール

OpenFOAMを上手に使いこなす為のツール

OpenFOAMの固有ユーティリティ

blockMesh, autoPatch, createPatch, snappyHexMesh, ...

pyFoam (http://openfoamwiki.net/index.php/Contrib_PyFoam)

pyFoamPlotRunner.py, pyFoamClearCase.py

pyFoamPotentialFoam.py, pyFoamCreateBoundaryPatches.py

オール・イン・ワン組み込み
⇒すぐに使える

blender用Script

BlockMeshExporter GUI

(<http://nikwik.webfactional.com/public.html>)

NamedASCII Export STL

煩雑なコマンド & パラメタ入力
⇒ボタン1発起動

DEXCS固有ユーティリティ

snappyDictExporter.py

....

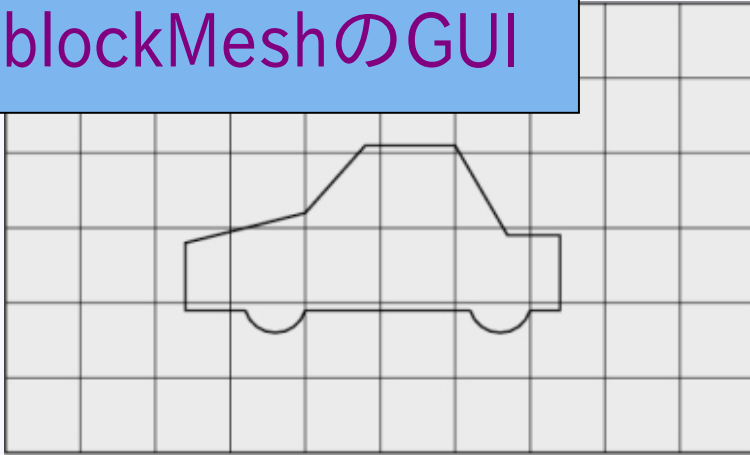
ユーザーの好みに応じて
組み込みや組み換えが可能

主眼はsnappyHexMeshによる自動メッシュ作成

snappyHexMesh

(OpenFOAM固有ユーティリティ)

blockMeshのGUI

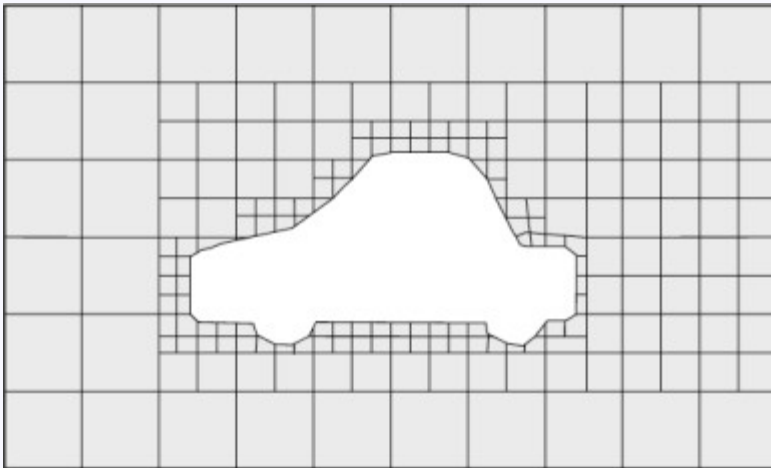


1.基礎メッシュ (blockMesh)

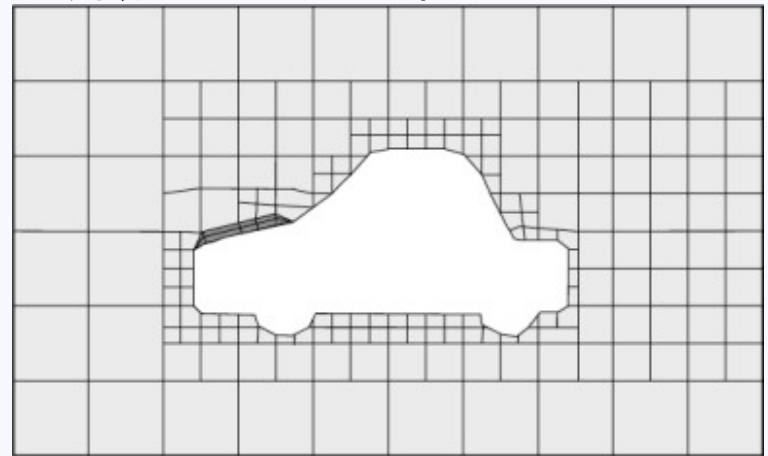
形状データのGUI



2.外形のトレース (castellatedMesh)



3.外形に合わせてセル変形 (snappedMesh)



4.レイヤーの挿入 (addLayers)

BlockMesh Exporter GUI

情報出所

<http://www.cfd-online.com/Forums/openfoam-meshing-blockmesh/61765-tool-block-generation-blender.html>

The screenshot shows a forum thread on the CFD Online website. The thread title is "Tool Block generation with blender". The user "wikstrom" has posted two messages. The first message, dated December 29, 2008, at 21:05, contains the text "Some enhancements and changes." and a link to "blockMeshExporterGUI.py". The second message, dated December 29, 2008, at 23:50, contains the text "Later updates can be fetched at" and a link to "http://nikwik.webfactional.com/public.html". The user profile for "wikstrom" is visible on the left, showing a join date of March 2009, 73 posts, and a reputation power of 3. The forum interface includes a navigation bar with links like Home, News, Forums, Wiki, Links, Jobs, Books, Events, About, and Search. A welcome message for "E.Mogura" is also present.

CFD Online
www.cfd-online.com

EnSight CFD

Home News Forums Wiki Links Jobs Books Events About Search

Home > Forums > OpenFOAM Meshing: blockMesh

Tool Block generation with blender

Welcome, E.Mogura.
You last visited: Yesterday at 04:48
Private Messages: Unread 0, Total 1.

USER PANEL BLOGS ▼ FAQ COMMUNITY ▼ NEW POSTS ▼ UPDATED THREADS ▼ SEARCH ▼ QUICK LINKS ▼ LOG OUT

Post Reply

December 19, 2008, 22:19

wikstrom
Member
Niklas Wikstrom
Join Date: Mar 2009
Posts: 73
Rep Power: 3

Hi,
sinc
writ
stru
The
follc

December 29, 2008, 21:05

Some enhancements and changes.

wikstrom
Member
Niklas Wikstrom
Join Date: Mar 2009
Posts: 73
Rep Power: 3

blockMeshExporterGUI.py

/N

Quote

December 29, 2008, 23:50

Later updates can be fetched a

wikstrom
Member
Niklas Wikstrom
Join Date: Mar 2009
Posts: 73
Rep Power: 3

Later updates can be fetched at

<http://nikwik.webfactional.com/public.html>

/n

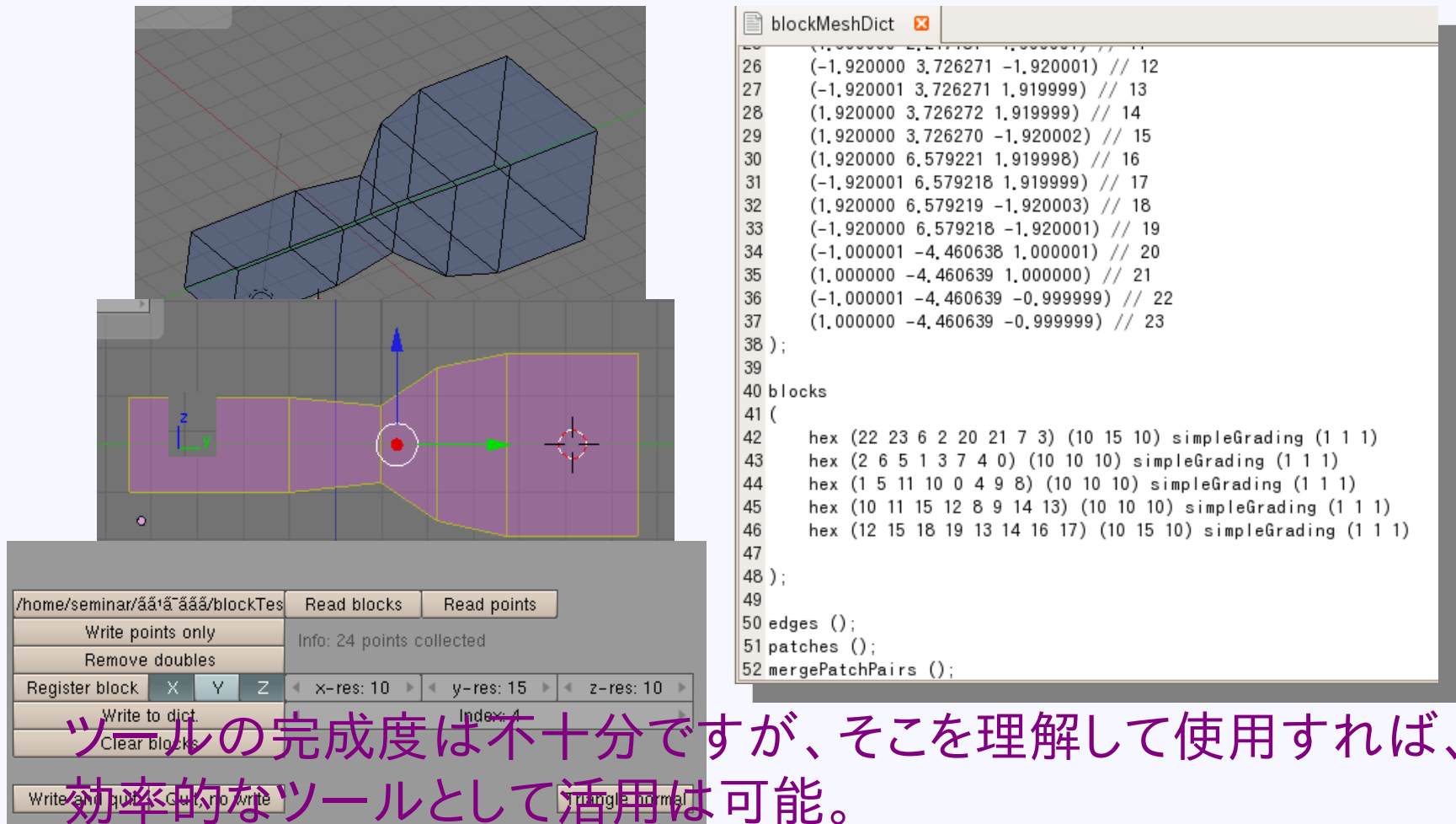
Quote

Script開発元 <http://nikwik.webfactional.com/public.html>

BlockMesh Exporter GUI

使用法解説

<http://mogura7.zenno.info/~et/xoops/modules/wordpress/index.php?p=164>

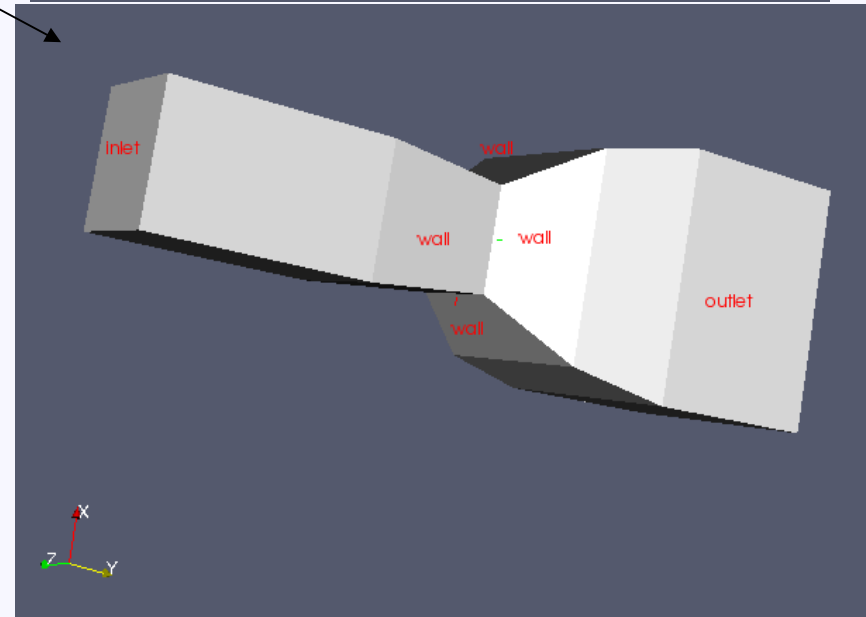
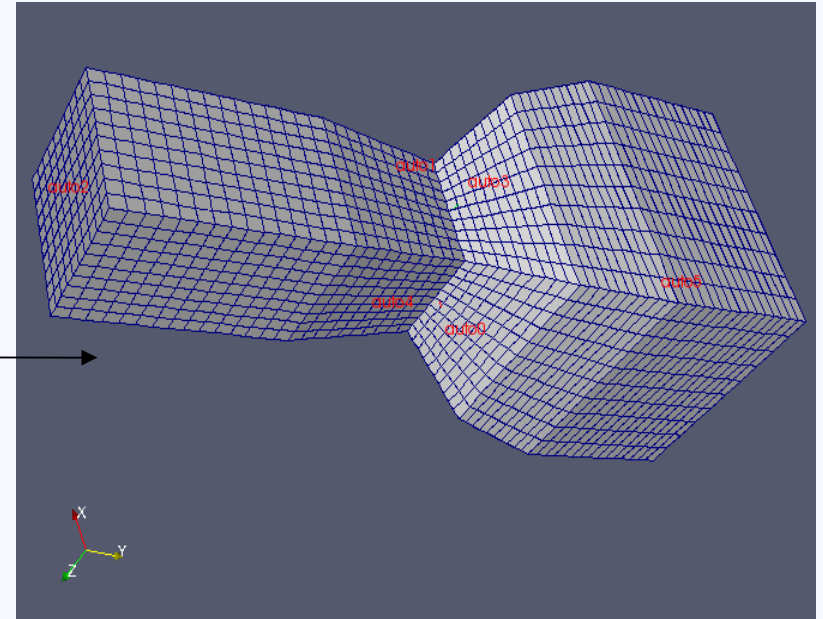


The screenshot displays the BlockMesh Exporter GUI. The top panel shows a 3D visualization of a block mesh structure. The bottom panel contains a control interface with buttons for 'Read blocks', 'Read points', 'Write points only', 'Remove doubles', 'Register block', 'Write to dict', 'Clear blocks', and 'Write and quit'. A status bar indicates 'Info: 24 points collected' and 'Index: 4'. To the right, a terminal window titled 'blockMeshDict' shows the generated code, which includes a list of points and a series of hexagonal blocks with simple grading.

```
blockMeshDict
26 (-1.920000 3.726271 -1.920001) // 12
27 (-1.920001 3.726271 1.919999) // 13
28 (1.920000 3.726272 1.919999) // 14
29 (1.920000 3.726270 -1.920002) // 15
30 (1.920000 6.579221 1.919998) // 16
31 (-1.920001 6.579218 1.919999) // 17
32 (1.920000 6.579219 -1.920003) // 18
33 (-1.920000 6.579218 -1.920001) // 19
34 (-1.000001 -4.460638 1.000001) // 20
35 (1.000000 -4.460639 1.000000) // 21
36 (-1.000001 -4.460639 -0.999999) // 22
37 (1.000000 -4.460639 -0.999999) // 23
38 );
39
40 blocks
41 (
42   hex (22 23 6 2 20 21 7 3) (10 15 10) simpleGrading (1 1 1)
43   hex (2 6 5 1 3 7 4 0) (10 10 10) simpleGrading (1 1 1)
44   hex (1 5 11 10 0 4 9 8) (10 10 10) simpleGrading (1 1 1)
45   hex (10 11 15 12 8 9 14 13) (10 10 10) simpleGrading (1 1 1)
46   hex (12 15 18 19 13 14 16 17) (10 15 10) simpleGrading (1 1 1)
47
48 );
49
50 edges ();
51 patches ();
52 mergePatchPairs ();
```

ツールの完成度は不十分ですが、そこを理解して使用すれば、
効率的なツールとして活用は可能。
(本講習では取り扱いません)

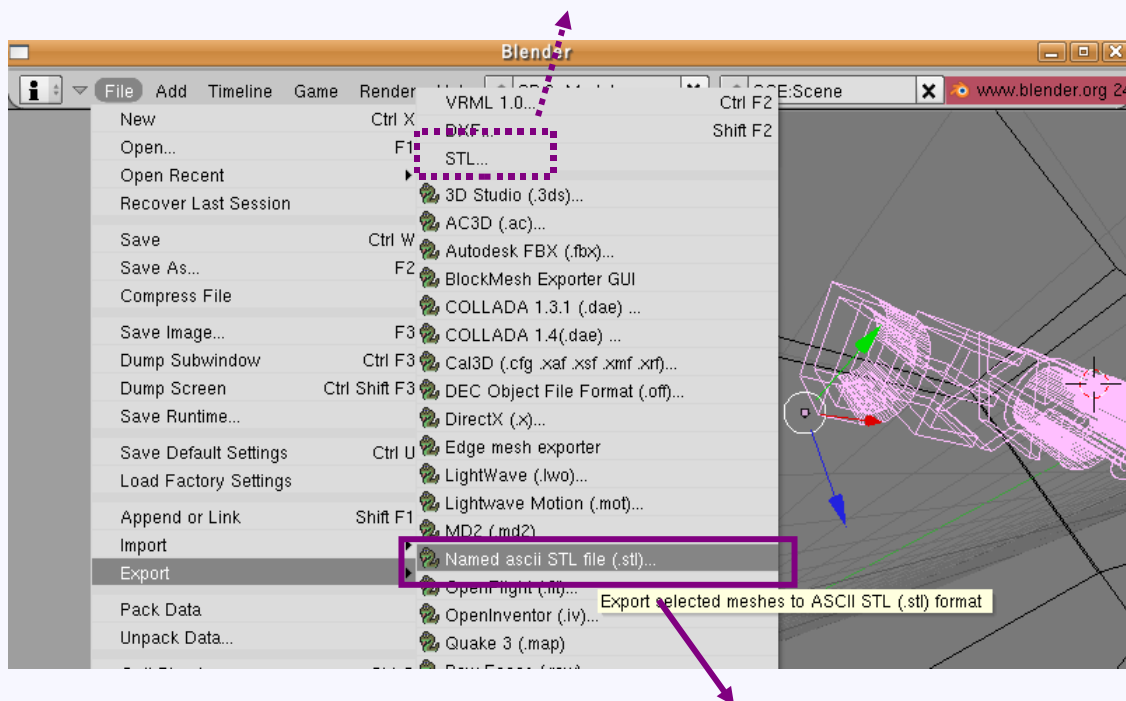
Patch名の変更 (OpenFOAM固有ユーティリティ)



詳細使用方法は後述します

blenderのSTLインタフェース

標準出力はバイナリ形式
(OpenFOAMでは読み取り不可)



アドオンscriptにて、
ascii形式を出力

Named ascii Export STL

情報出所

<http://www.cfd-online.com/Forums/openfoam-meshing-other/61601-blender-export-script-named-ascii-stlbs.html>

**CFD Online**
www.cfd-online.com



Home News Forums Wiki Links Jobs Books Events About Search

Home > Forums > OpenFOAM Meshing: Other

Blender export script for named ASCII STLbs

Welcome, **E.Mogura**.
You last visited: Yesterday at 04:48
Private Messages: Unread 0, Total 1.

USER PANEL BLOGS ▼ FAQ COMMUNITY ▼ NEW POSTS ▼ UPDATED THREADS ▼ SEARCH ▼ QUICK LINKS ▼ LOG OUT

Post Reply

LINKBACK ▼ THREAD TOOLS ▼ SEARCH THIS THREAD ▼ RATE THREAD ▼ DISPLAY MODES ▼

July 28, 2008, 12:18

Hi All, I have written an

#1

andersking
Member

Andrew King
Join Date: Mar 2009
Location: Perth, Western Australia, Australia
Posts: 77
Rep Power: 3

Hi All,

I have written an export script for blender, which exports selected objects (with names) to an ASCII STL file. This should be all you need to get a suitable stl for using in snappyHexMesh.

Improvements/bug fixes (hopefully none) are welcome. Requests for improvements not so much (I don't have time).

To use the script, save it in ~/.blender/scripts. It should then give you a menu item in File->Export->Named ascii STL file...

You can give your blender objects useful names and these will be kept in the exported stl. These can be used for creating patches in snappyHexMesh.

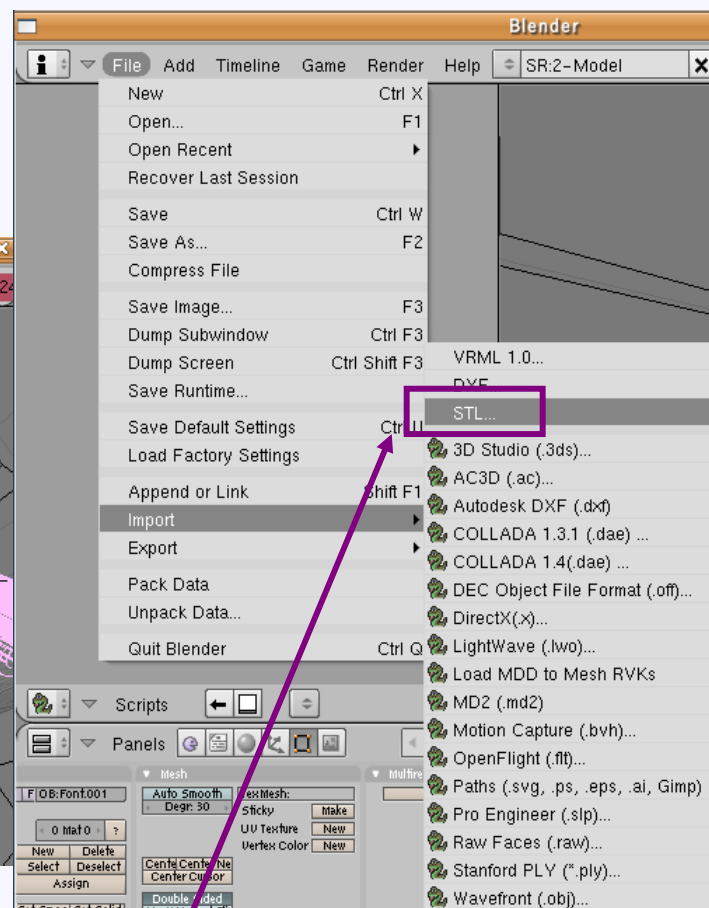
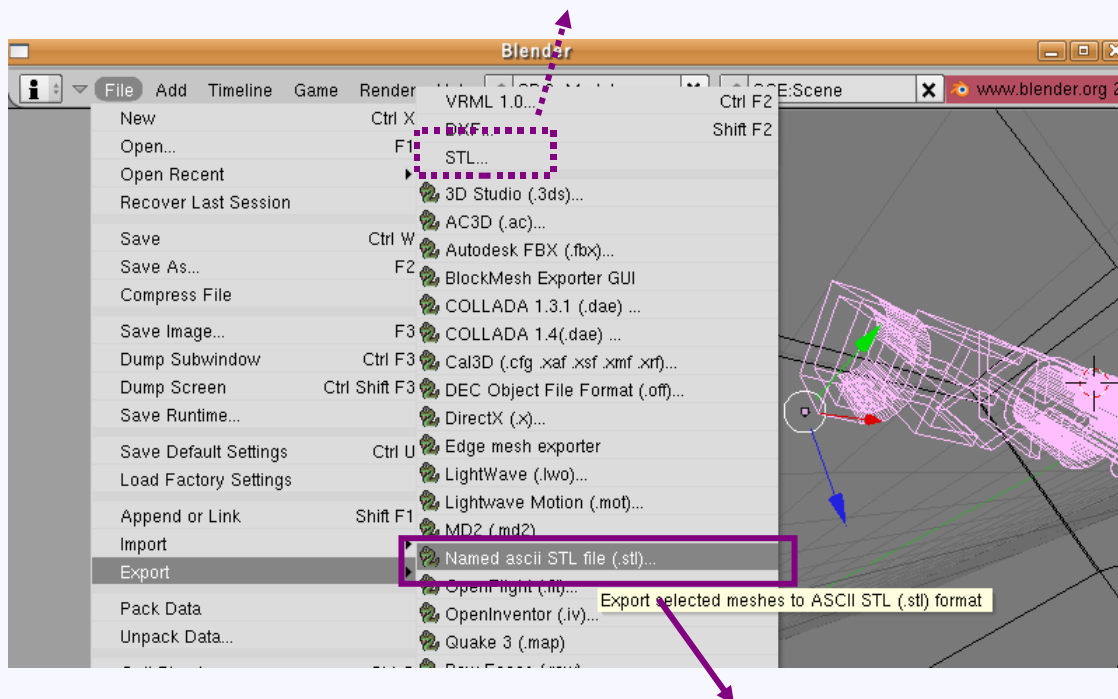
 **stlasciiexport.py**

Regards,
Andrew

Dr Andrew King
Fluid Dynamics Research Group
Curtin University

blenderのSTLインタフェース

標準出力はバイナリ形式
(OpenFOAMでは読み取り不可)



アドオンscriptにて、
ascii形式を出力

入力はバイナリ/ascii 形式
どちらもOK, 但し...

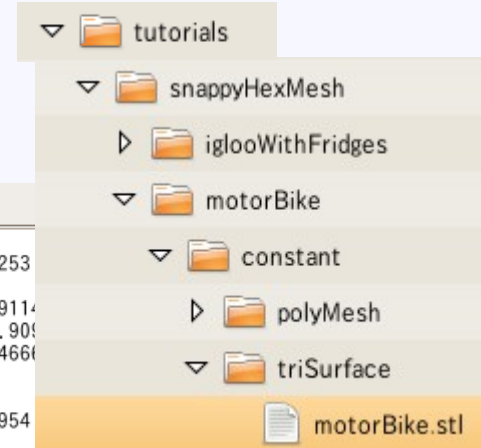
Named ascii STL

blender, DEXCS

```
dexcs.stl
solid Font
facet normal 0.295607298613 -0.344685822725 -0.890959024429
  outer loop
    vertex -3.29274535179 -3.27592134476 0.58639138937
    vertex -3.28169870377 -3.27366447449 0.589183390141
    vertex -3.27007651329 -3.27175045013 0.592298984528
  endloop
endfacet
facet normal 0.29560059309 -0.344598650932 -0.89099496603
  outer loop
    vertex -3.30336380005 -3.27854084969 0.583881676197
    vertex -3.29274535179 -3.27592134476 0.58639138937
    vertex -3.27007651329 -3.27175045013 0.592298984528
  endloop
endfacet
facet normal 0.295599162579 -0.344583690166 -0.891001224518
  outer loop
    vertex -3.30336380005 -3.27854084969 0.583881676197
    vertex -3.27007651329 -3.27175045013 0.592298984528
    vertex -4.83896541595 -4.12005519867 -0.0213044807315
  endloop
endfacet
facet normal -0.295582473278 0.344375908375 0.891087114811
  outer loop
    vertex -4.68074941635 -4.09089136124 0.0199064724147
    vertex -4.66805648804 -4.08817720413 0.0230679102242
    vertex -4.83896541595 -4.12005519867 -0.0213044807315
  endloop
endfacet
endsolid
```

OpenFOAM

```
motorBike.stl
solid frt-fairing:001%1
  facet normal -0.5249406 -0.58519253
    outer loop
      vertex 0.103075 -0.25693 0.9114
      vertex 0.106083 -0.262068 0.909
      vertex 0.1095 -0.25928 0.914666
    endloop
  endfacet
  facet normal -0.5087473 -0.51952954
    outer loop
      vertex 0.106322 -0.253072 0.917009
      vertex 0.1095 -0.25928 0.914666
      vertex 0.113314 -0.25619 0.919831
    endloop
  endfacet
  facet normal -0.55355344 -0.71416028 0.42843166
    outer loop
      vertex 0.100684 -0.266434 0.896745
      vertex 0.1102694 0.0068240141 -0.99379397
    endloop
  endfacet
endsolid frt-fairing:001%1
solid windshield:002%2
  facet normal 0.74104179 0.28539924 -0.60778642
    outer loop
      vertex 0.0404717 -0.0409035 0.969627
      vertex 0.0434768 -0.0367081 0.975261
      vertex 0.0449475 -0.0435998 0.973818
    endloop
  endfacet
endsolid
```



solidブロック定義は1つだけ
境界面ごとに、ファイルを分割定義
solidの名前はツールが自動識別

複数のsolidブロック定義可能
⇒境界面に応じて名前を変更
solidの名前はユーザーが定義

SnappyHexMeshDict Exporter GUI

The image illustrates the workflow of the SnappyHexMeshDict Exporter GUI through five numbered steps:

- 形状選択 (Shape Selection):** Selecting a shape from the left sidebar.
- 形状選択細分化パラメタ指定 (Shape Selection Refinement Parameter Specification):** Specifying refinement parameters in the 'snappyHexMeshDict+ Exporter' dialog.
- Export...** Clicking the 'Export...' button in the 'snappyHexMeshDict+ Exporter' dialog.
- Export...** Clicking the 'Export...' button in the 'snappyHexMeshDict+ Exporter' dialog (repeated).
- はい(Y) (Yes):** Confirming the export in the 'Question' dialog.

snappyHexMeshDict+ Exporter Dialog (Step 2):

shapeName	Type	Apply	Smin	Smax	Region	Layers
dexcs.stl	Surface	<input type="checkbox"/>	1	1	0	1
box1.stl	Region	<input type="checkbox"/>	0	0	3	0
inlet	Patch	<input type="checkbox"/>	0	0	0	1
outlet	Patch	<input type="checkbox"/>	0	0	0	1
wall	Patch	<input type="checkbox"/>	0	0	0	1
		<input type="checkbox"/>	0	0	0	0

snappyHexMeshDict File Content (Step 2):

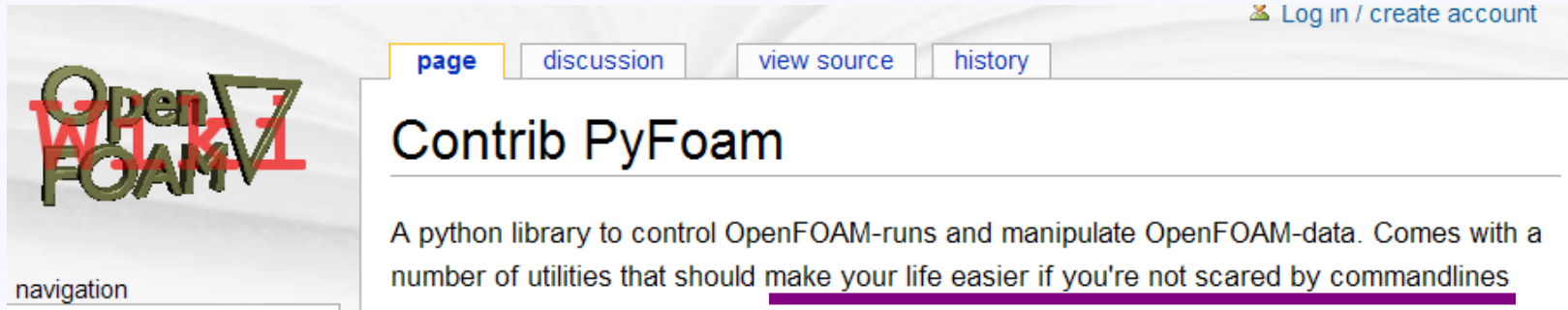
```
refinementSurfaces
{
  dexcs
  // Surface-wise
  level (3 4);
}

refinementRegions
{
  box1
  mode inside;
  levels ((1 1 3));
}
```

詳細使用方法は後述します

pyFoam

http://openfoamwiki.net/index.php/Contrib_PyFoam



OpenFOAM Wiki

Log in / create account

page discussion view source history

Contrib PyFoam

A python library to control OpenFOAM-runs and manipulate OpenFOAM-data. Comes with a number of utilities that should make your life easier if you're not scared by commandlines

navigation

4 Installed Utilities

4.1 Runner-Utilities

- 4.1.1 pyFoamRunner.py
- 4.1.2 pyFoamUtilityRunner.py
- 4.1.3 pyFoamSteadyRunner.py
- 4.1.4 pyFoamPlotRunner.py
- 4.1.5 pyFoamMeshUtilityRunner.py
- 4.1.6 pyFoamPotentialRunner.py
- 4.1.7 pyFoamRunAtMultipleTimes.py

4.2 Utilities for Logfiles

- 4.2.1 pyFoamPlotWatcher.py
- 4.2.2 pyFoamStandardLogAnalyzer.py

4.3 Networking Utilities

- 4.3.1 pyFoamNetList.py
 - 4.3.1.1 pyFoamMetaServer.py
- 4.3.2 pyFoamNetShell.py

4.4 Utilities for Manipulating case data

- 4.4.1 pyFoamAddEmptyBoundary.py
- 4.4.2 pyFoamChangeBoundaryType.py
- 4.4.3 pyFoamCreateBoundaryPatches.py
- 4.4.4 pyFoamClearCase.py
- 4.4.5 pyFoamPackCase.py
- 4.4.6 pyFoamCloneCase.py
- 4.4.7 pyFoamCopyLastToFirst.py
- 4.4.8 pyFoamClearInternalField.py
- 4.4.9 pyFoamClearBoundaryValue.py

4.5 Manipulating dictionaries (from scripts)

- 4.5.1 pyFoamReadDictionary.py
- 4.5.2 pyFoamWriteDictionary.py
- 4.5.3 pyFoamFromTemplate.py
- 4.5.4 pyFoamCompareDictionary.py
- 4.5.5 pyFoamUpdateDictionary.py

4.6 Paraview related utilities

- 4.6.1 pyFoamPVSnapShot.py
- 4.6.2 pyFoamPVLoadState.py

4.7 Other

- 4.7.1 pyFoamListCases.py
- 4.7.2 pyFoamDecompose.py
- 4.7.3 pyFoamComparator.py
- 4.7.4 pyFoamVersion.py
- 4.7.5 pyFoamExecute.py
- 4.7.6 pyFoamDumpConfiguration.py
- 4.7.7 pyFoamSamplePlot.py
- 4.7.8 pyFoamCaseReport.py
- 4.7.9 pyFoamEchoDictionary.py
- 4.7.10 pyFoamCaseBuilder.py

4.8 GUI-Tools

- 4.8.1 pyFoamDisplayBlockMesh.py
- 4.8.2 pyFoamAPoMaFoX.py: A Poor Man's FoamX

4.9 Special utilities

- 4.9.1 pyFoamModifyGGIBoundary.py

pyFoam on DEXCS2009

4 Installed Utilities

4.1 Runner-Utilities

4.1.1 pyFoamRunner.py

4.1.2 pyFoamUtilityRunner.py

4.1.3 pyFoamSteadyRunner.py

4.1.4 pyFoamPlotRunner.py

4.1.5 pyFoamMeshUtilityRunner.py

4.1.6 pyFoamPotentialRunner.py

4.1.7 pyFoamRunAtMultipleTimes.py

4.2 Utilities for Logfiles

4.2.1 pyFoamPlotWatcher.py

4.2.2 pyFoamStandardLogAnalyzer.py

4.3 Networking Utilities

4.3.1 pyFoamNetList.py

4.3.1.1 pyFoamMetaServer.py

4.3.2 pyFoamNetShell.py

4.4 Utilities for Manipulating case data

4.4.1 pyFoamAddEmptyBoundary.py

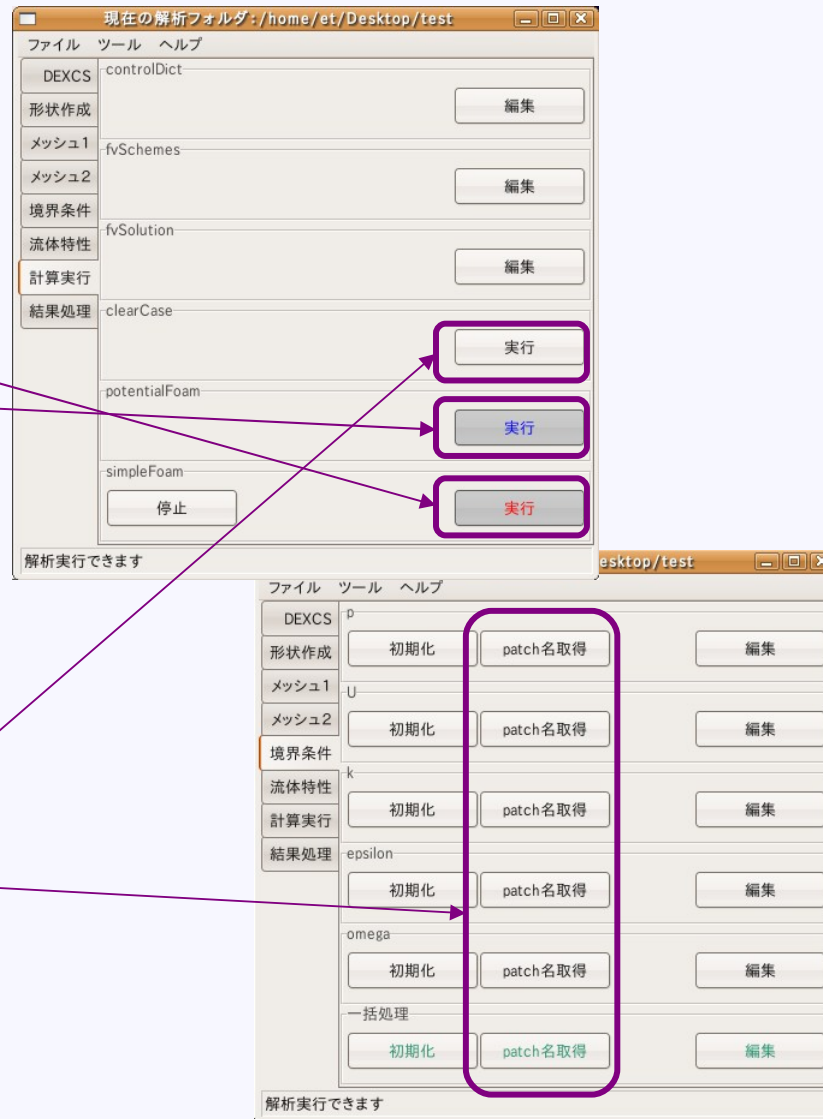
4.4.2 pyFoamChangeBoundaryType.py

4.4.3 pyFoamCreateBoundaryPatches.py

4.4.4 pyFoamClearCase.py

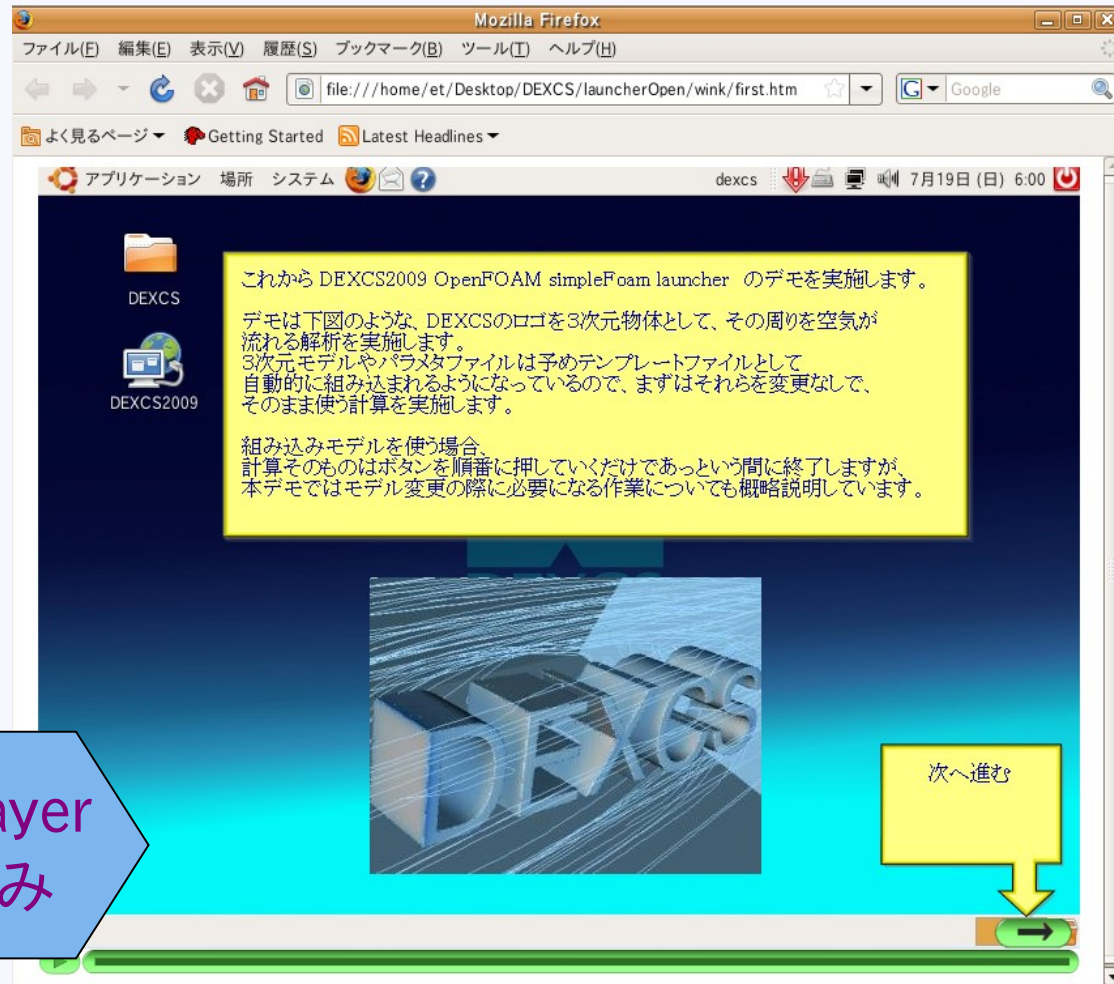
4.4.5 pyFoamPackCase.py

4.4.6 pyFoamCloneCase.py



ランチャーの使い方

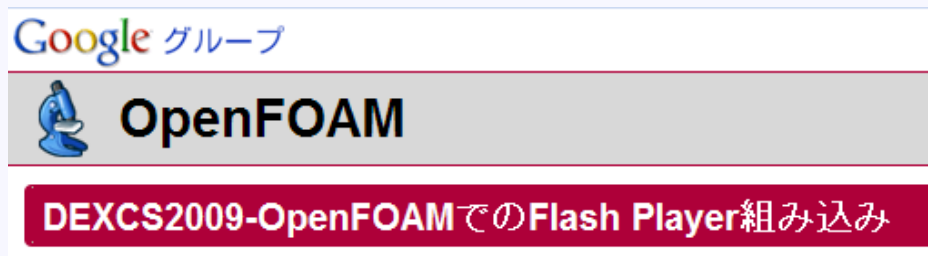
Winkチュートリアル



Flash Player
組み込み

Flash Player組み込み方法

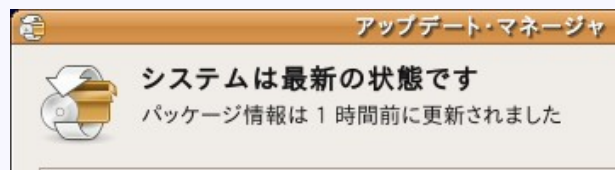
http://groups.google.com/group/openfoam/browse_thread/thread/c946e00edeac7d03



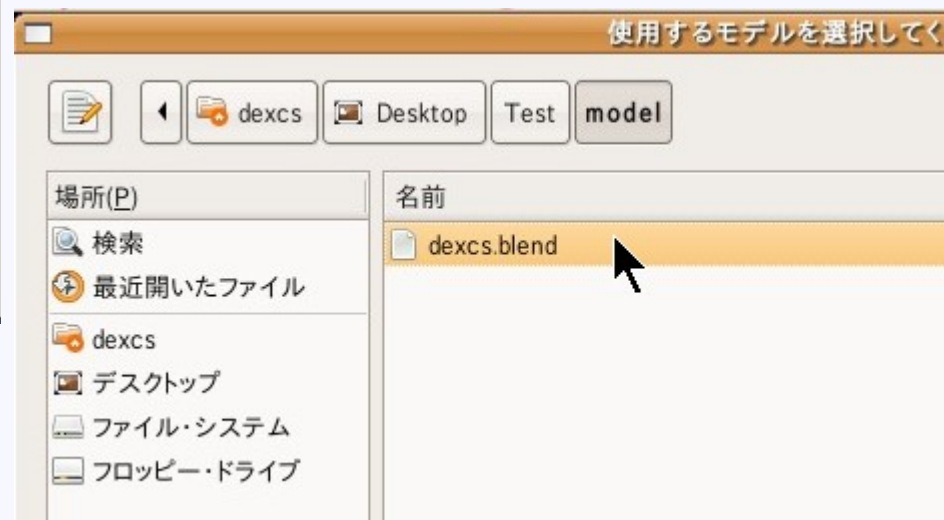
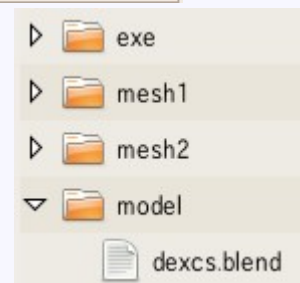
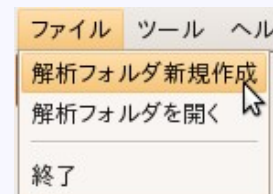
詳細手順

<http://mogura7.zenno.info/~et/xoops/modules/wordpress/index.php?p=300>

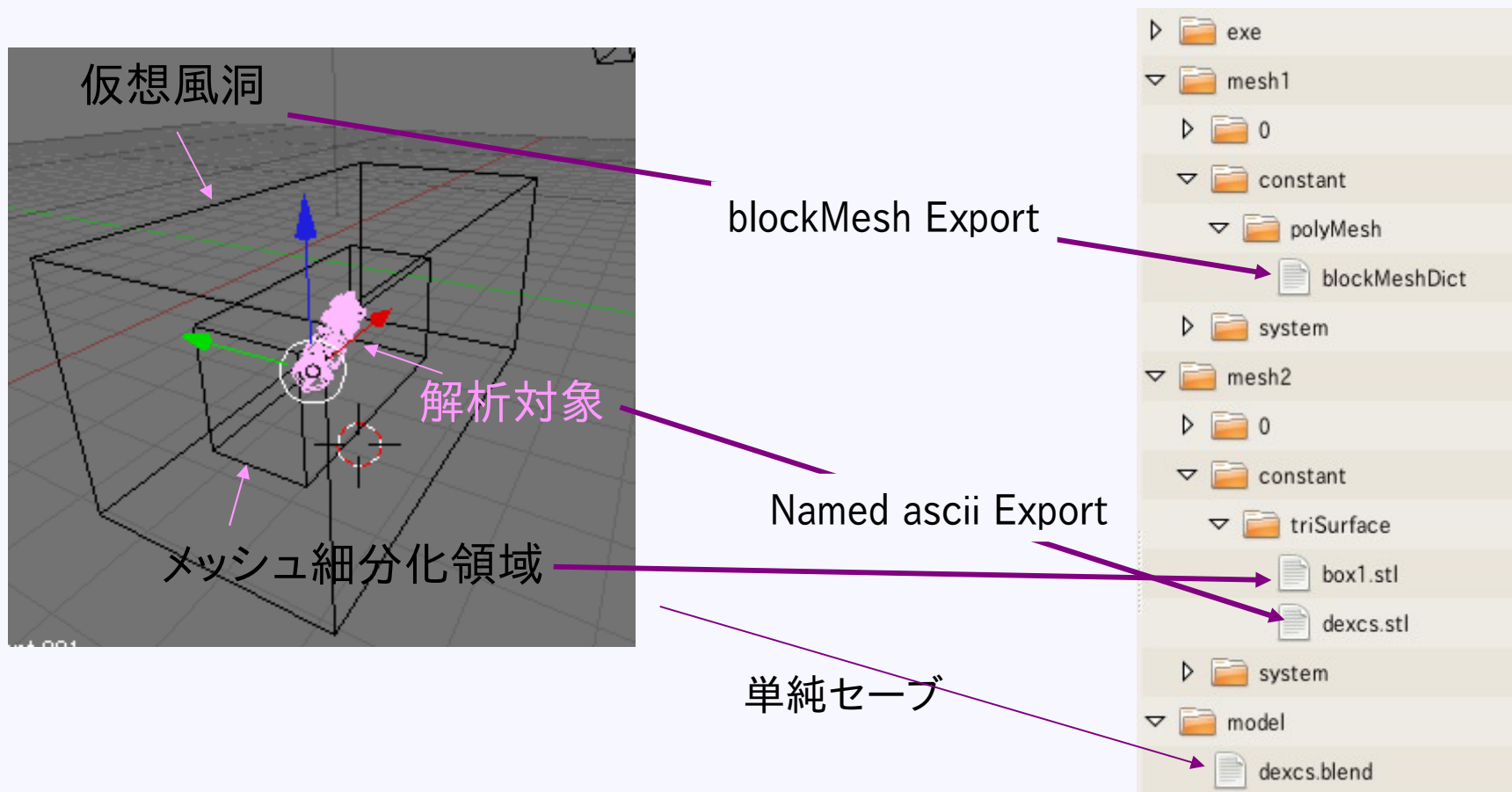
注意事項: 要システムアップデート



形状作成

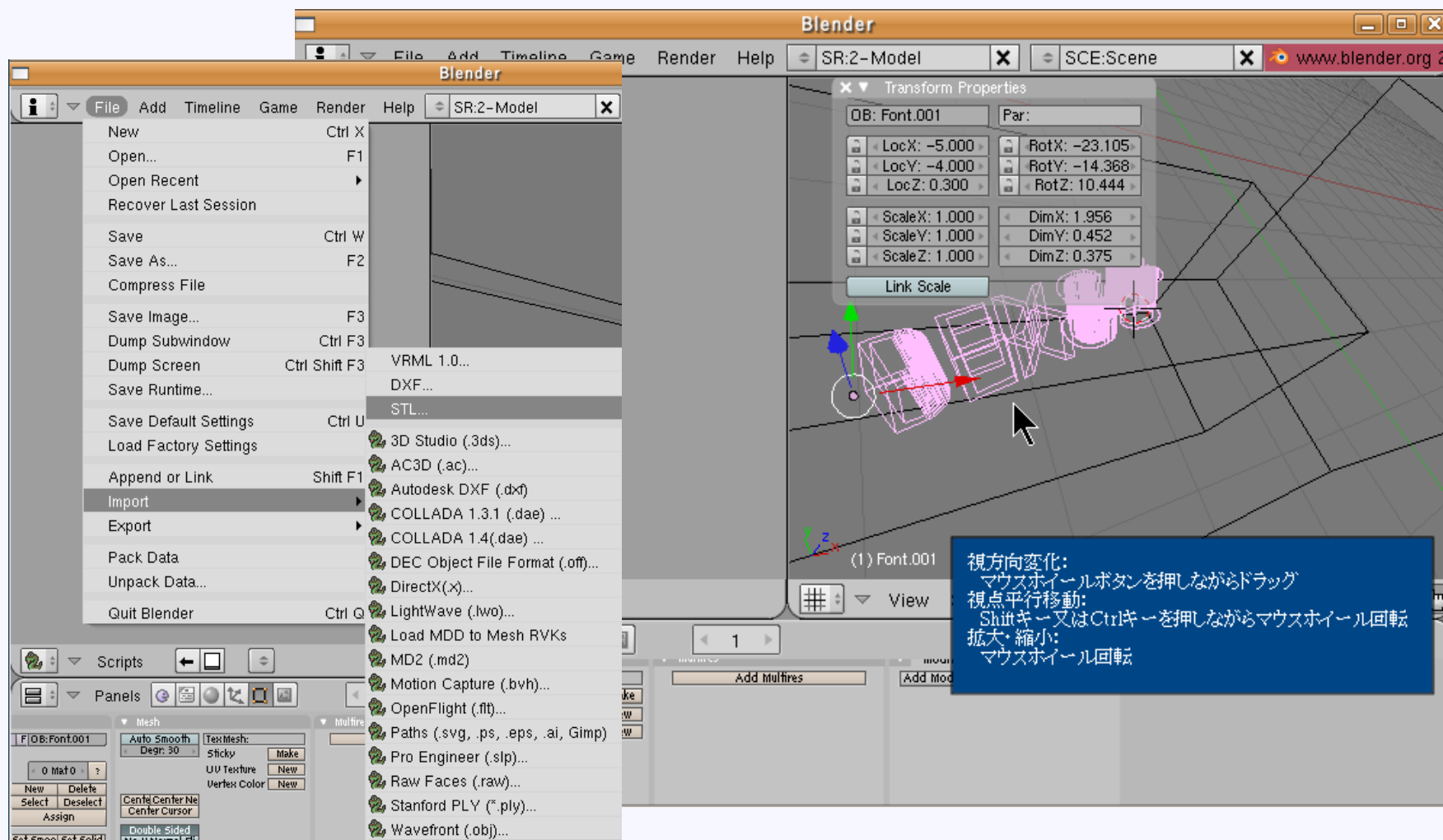


形状変更の反映方法



単純セーブだけでは解析モデルに反映されません

DEXCSにおけるblenderの位置づけ



講習A-2

CAD形状に対するfilterツールとして利用
ごく簡単な形状は作成可能

メッシュ作成1 (blockMesh)

M1-1

M1-2

M1-3

M1-4

M1-5

M1-6

M1-7

M1-8

現在の解析ファイル: /home/et/Desktop/test

ファイル ツール ヘルプ

DEXCS blockMesh

形状作成 blockMeshDict編集 初期化(やり直し) 実行

メッシュ1 importMesh 実行

メッシュ2

境界条件

流体特性 autoPatch

計算実行 二面挟角(deg) 60 実行

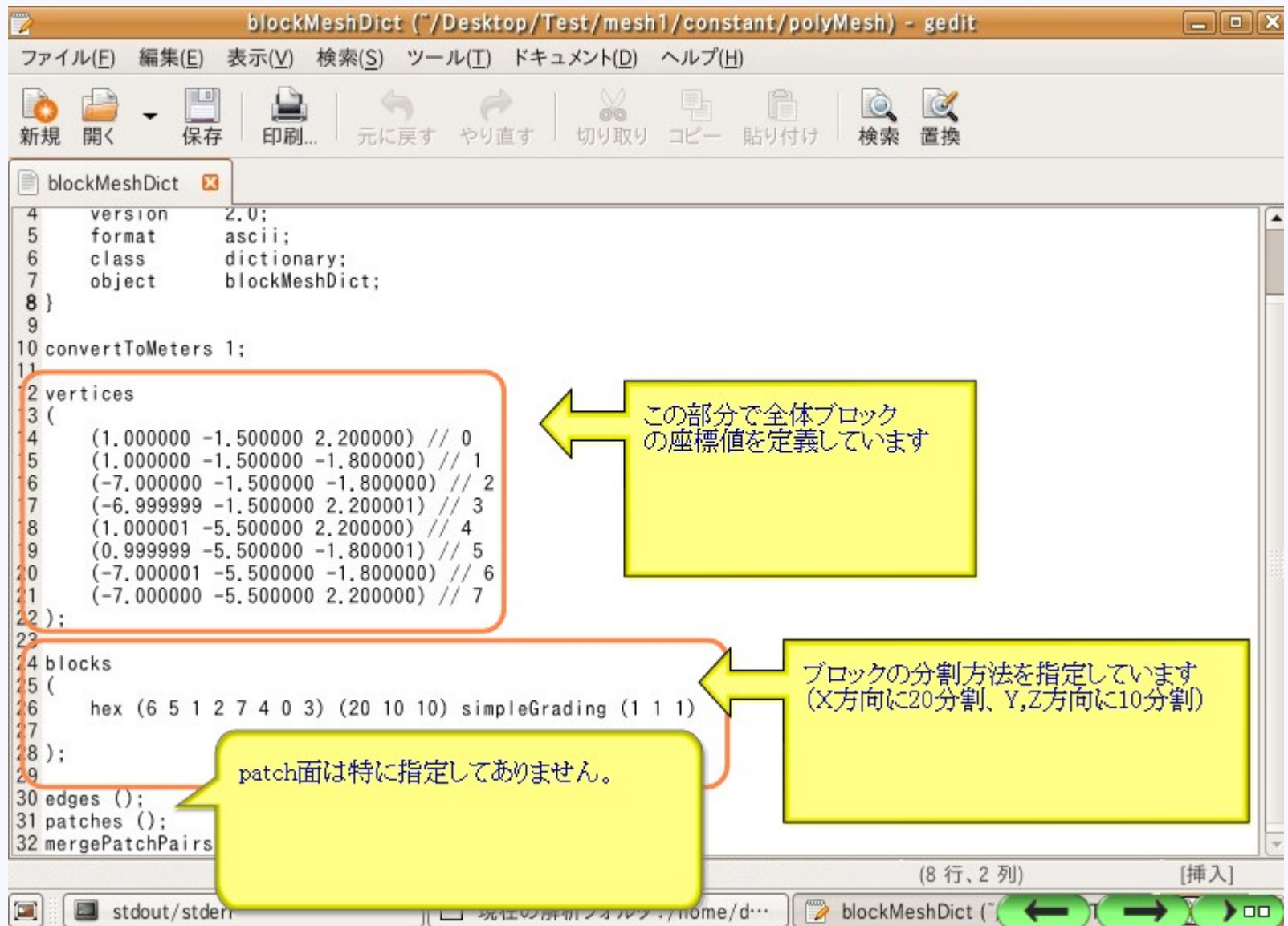
結果処理 createPatch

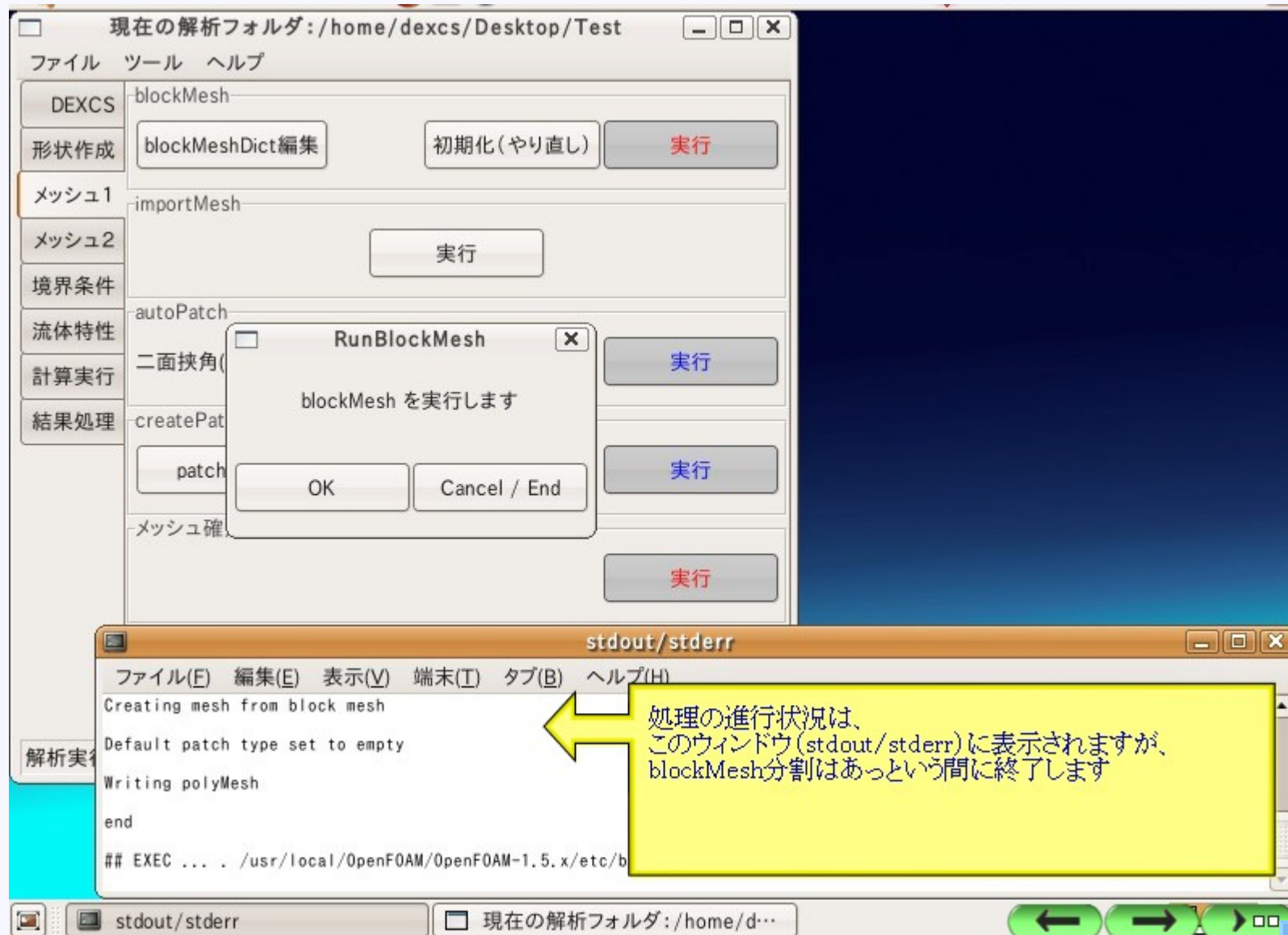
patch確認 createPatchDict編集 実行

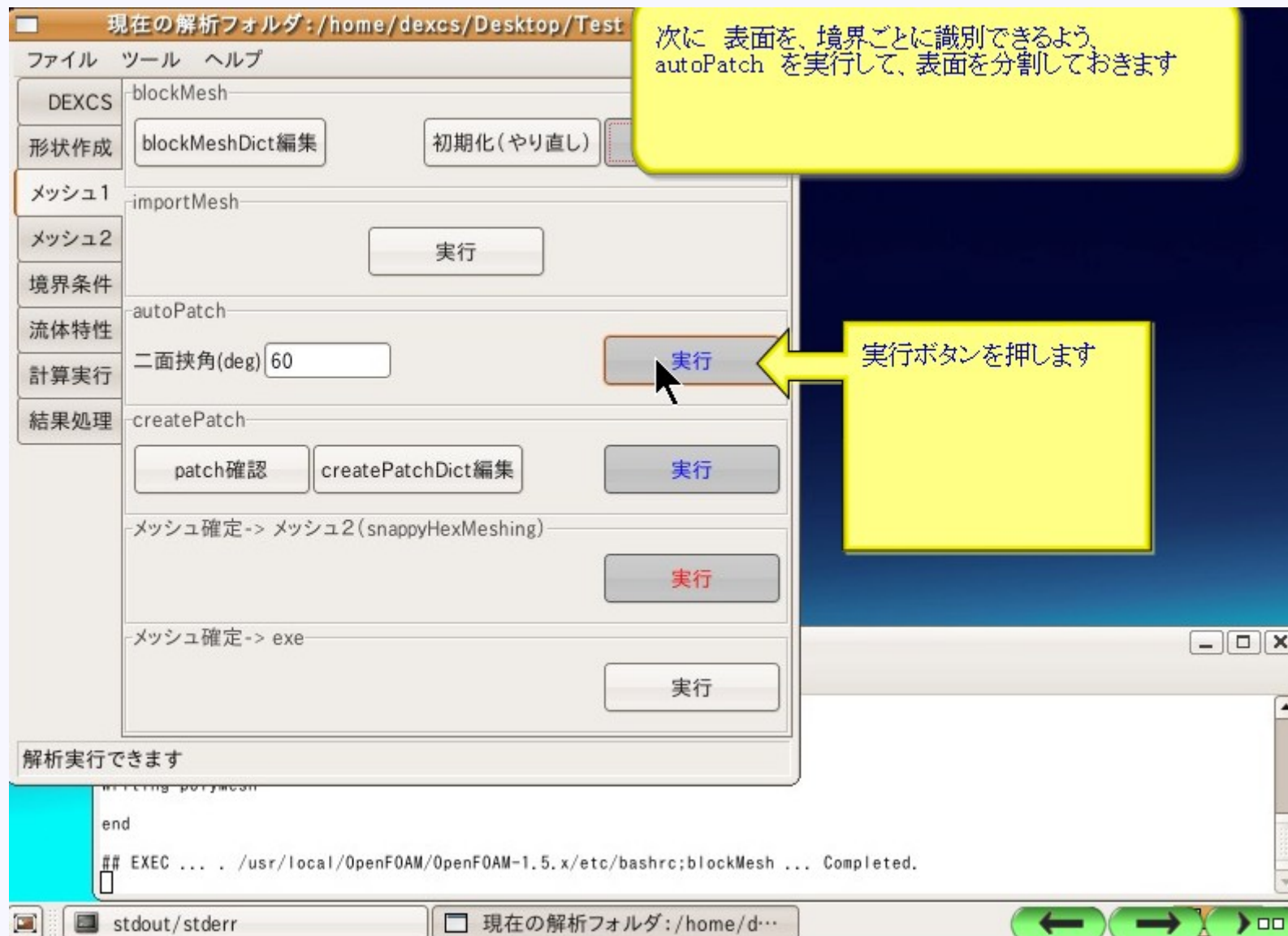
メッシュ確定-> メッシュ2 (snappyHexMeshing) 実行

メッシュ確定-> exe 実行

解析実行できます







二面挟角とは

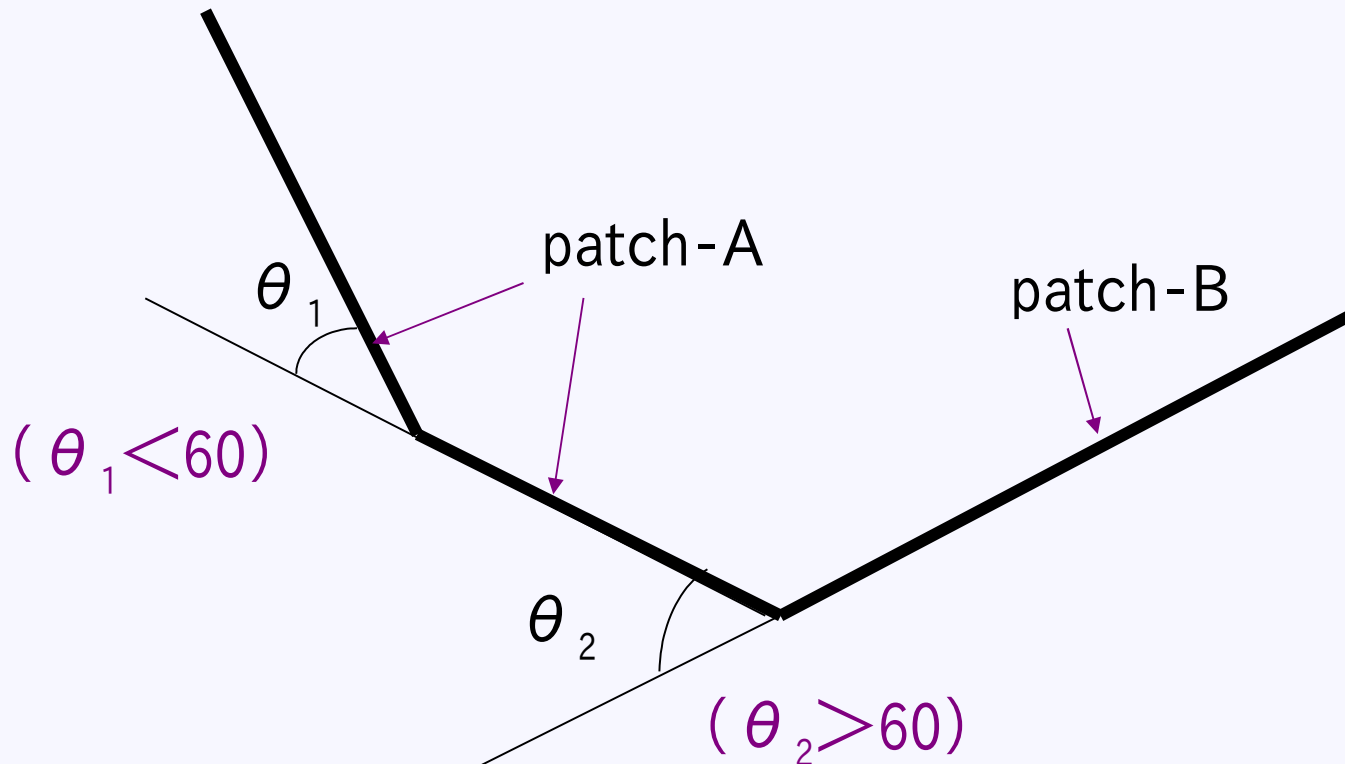


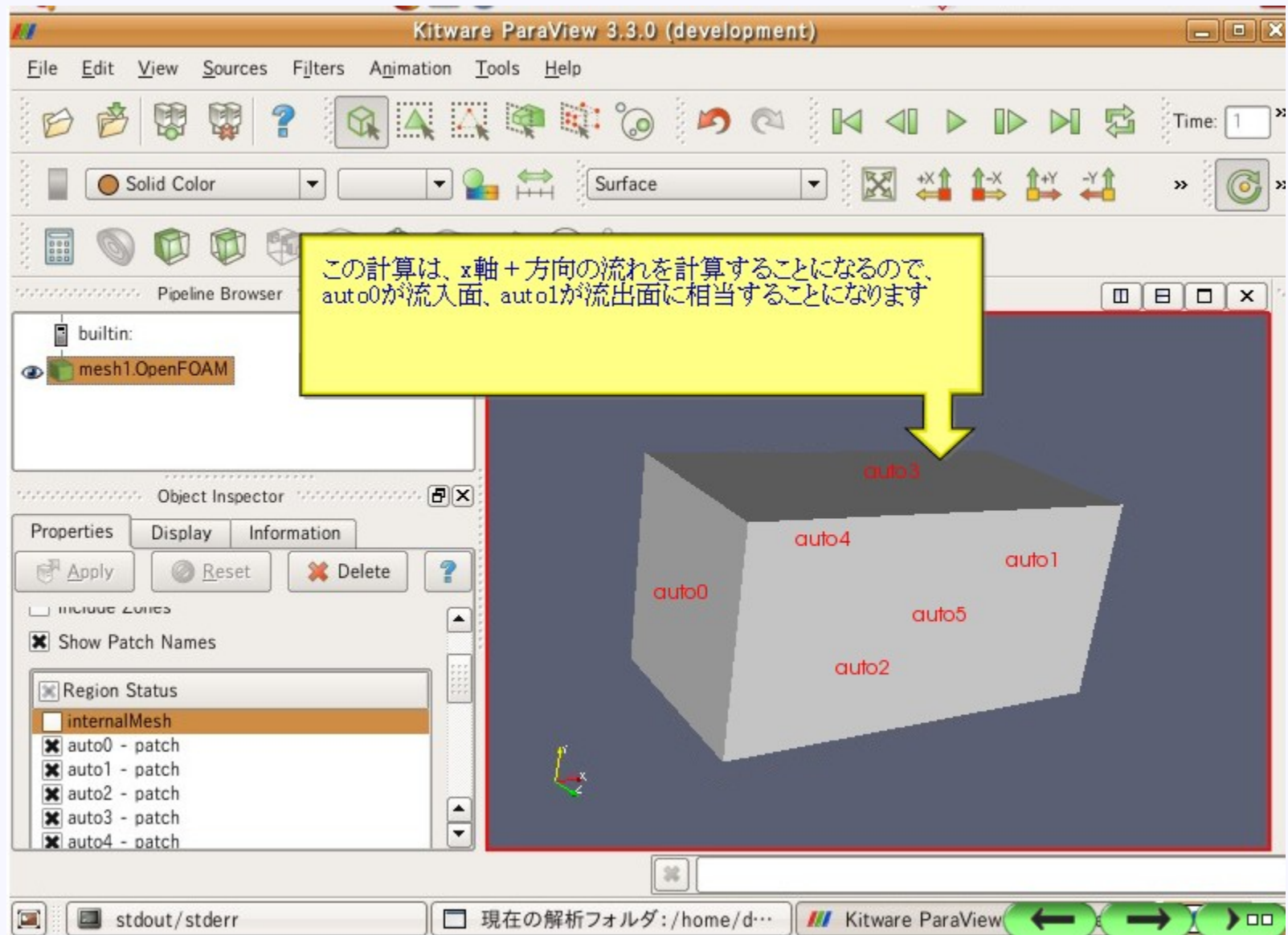
$$\theta > 60$$

patch-Aとpatch-Bを区別

$$\theta < 60$$

patch-Aとpatch-Bは同一面





createPatchDict (/Desktop/Test/mesh1/system) - gedit

ファイル(E) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

新規 開く 保存 印刷... 元に戻す やり直す 切り取り コピー 貼り付け

createPatchDict

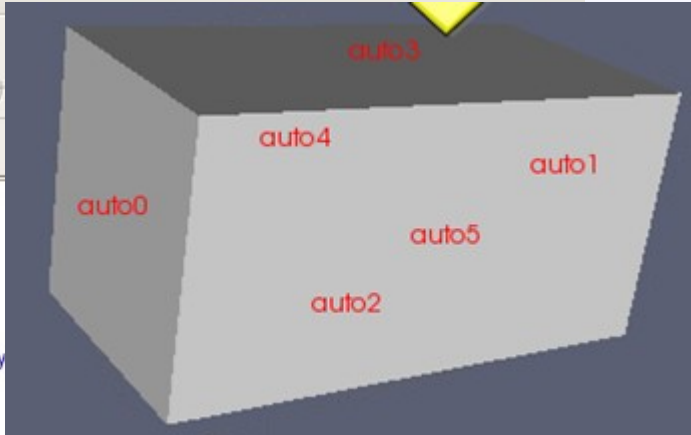
```

28
29 // Do a synchronisation of coupled points.
30 pointSync true;
31
32
33 // Patches to create.
34 // If no patches does a coupled point and face synchronisation anyway
35 patches
36 (
37     {
38         // Name of new patch
39         name inlet;
40
41         // Type of new patch
42         type patch;
43
44         // How to construct: either 'patches' or 'set'
45         constructFrom patches;
46
47         // If constructFrom = patches : names of
48         patches (auto0) set fv;
49     }
50 // If constructFrom = set : name of faceSet
51 // set fv;
52 }
53
54 {
55     // Name of new patch
56     name wall;
57
58     // Type of new patch
59     type patch;
60
61     // How to construct: either 'patches' or 'set'
62     constructFrom patches;
63
64     // If constructFrom = patches : names of patch
65     patches (auto2 auto3 auto4 auto5);
66
67     // If constructFrom = set : name of faceSet
68     set fv;
69 }

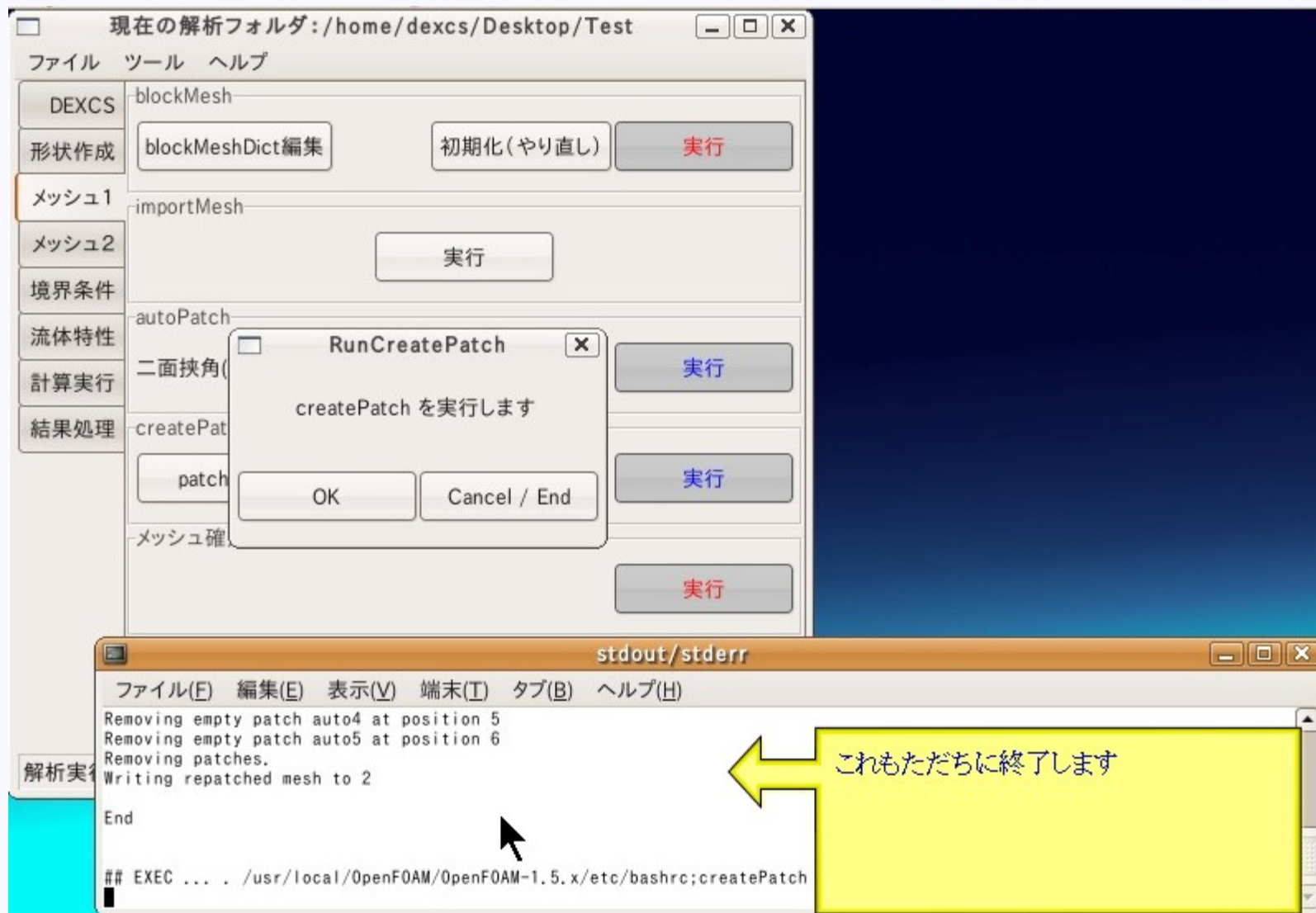
```

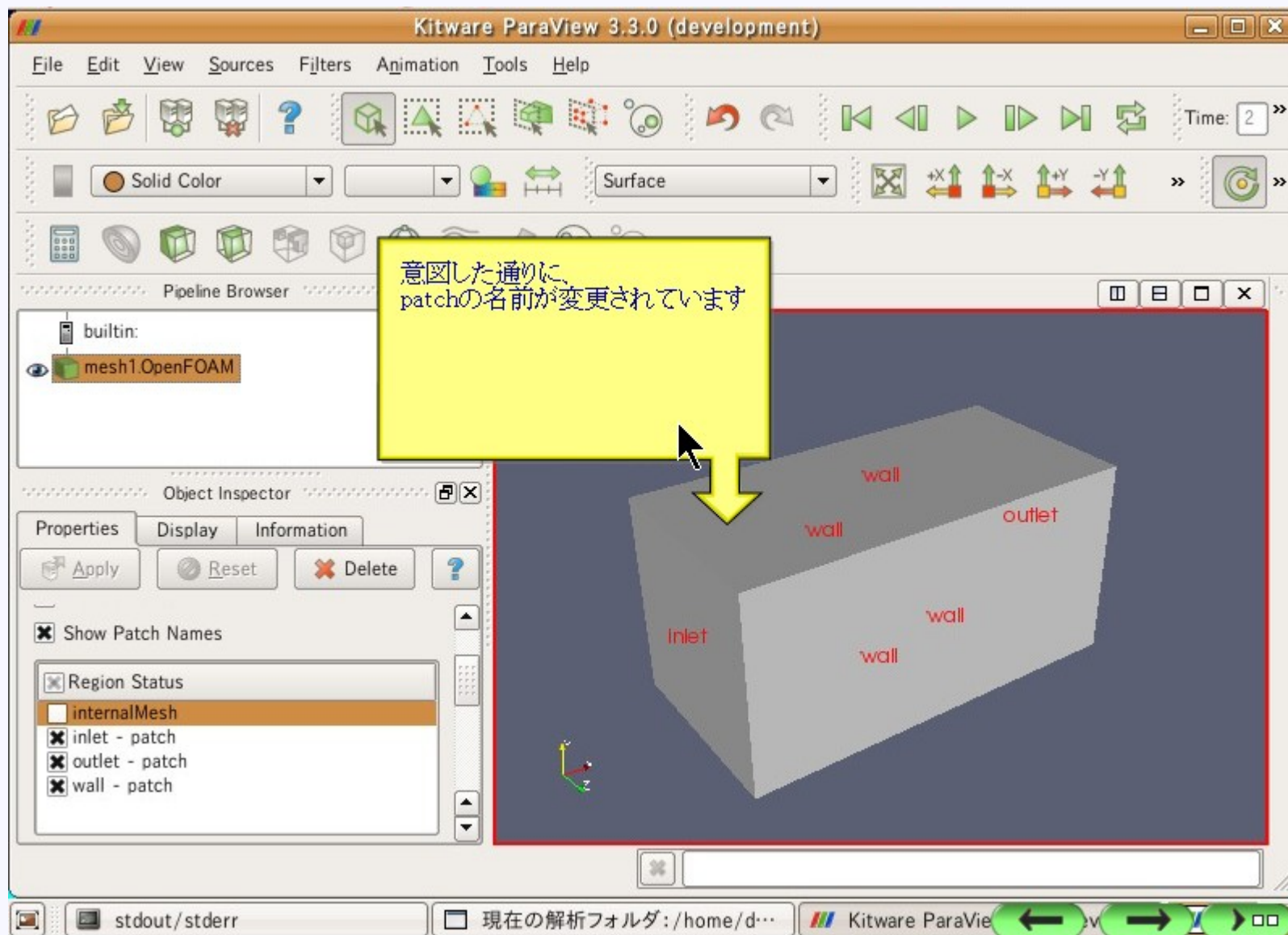
この部分で、auto0のpatchを、inletという名前に変更しています

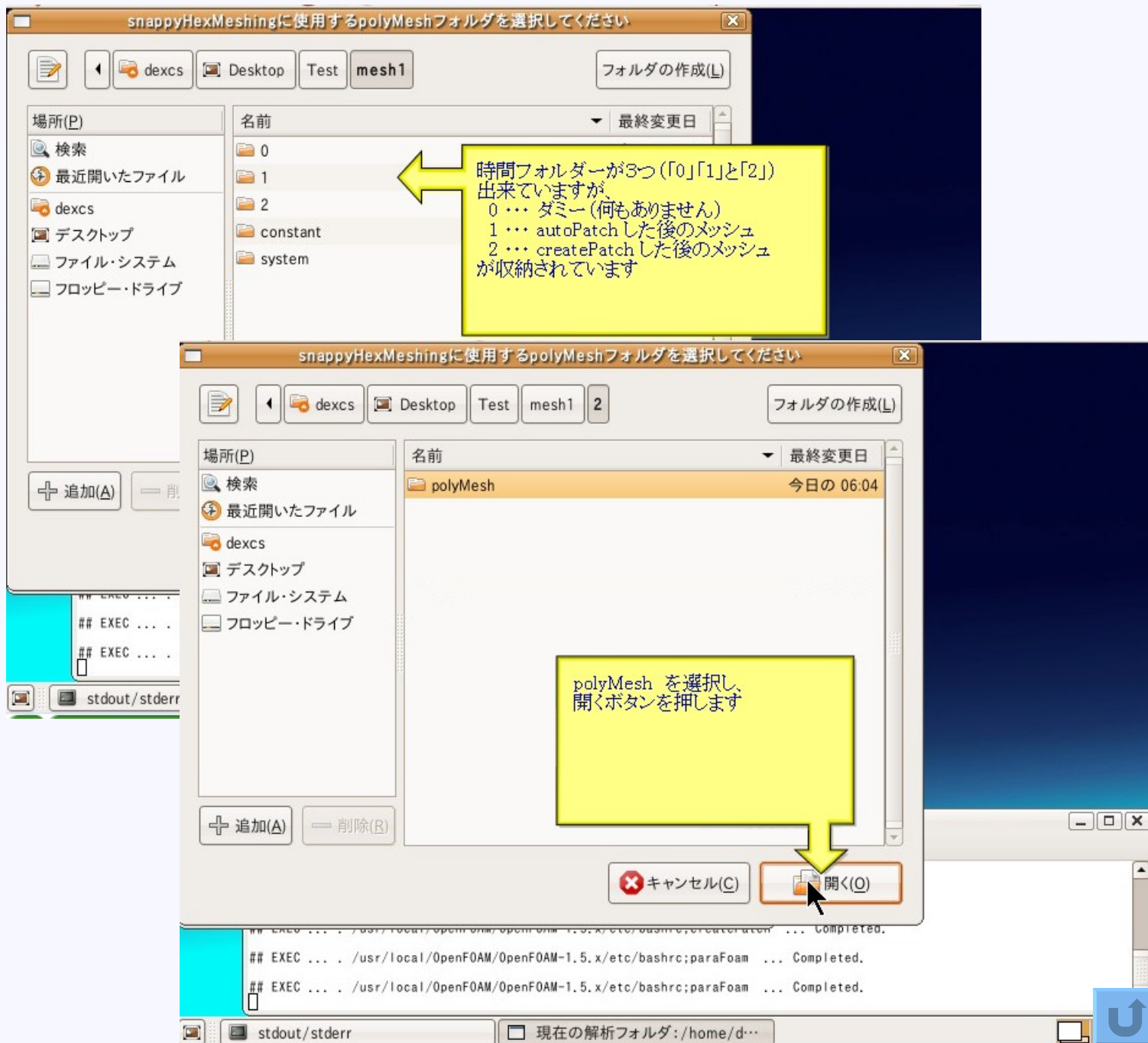
残りの面(auto2,auto3,auto4,auto5)は、wallという名前の境界にしています(地面境界にしたい場合などは、この部分を変更することになります)



stdout/stderr







メッシュ作成2 (snappyHexMesh)



M2-1

M2-2

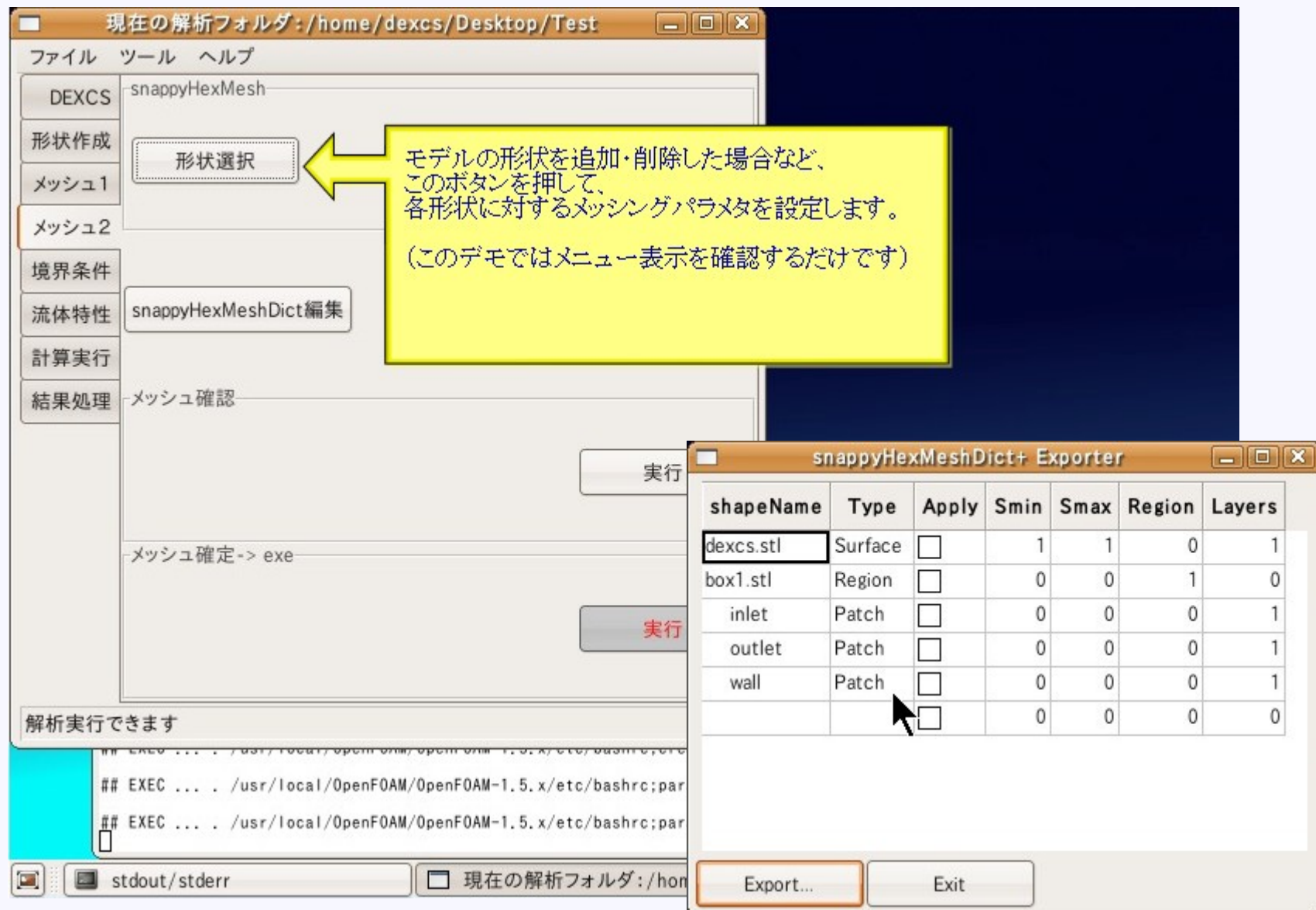
M2-6

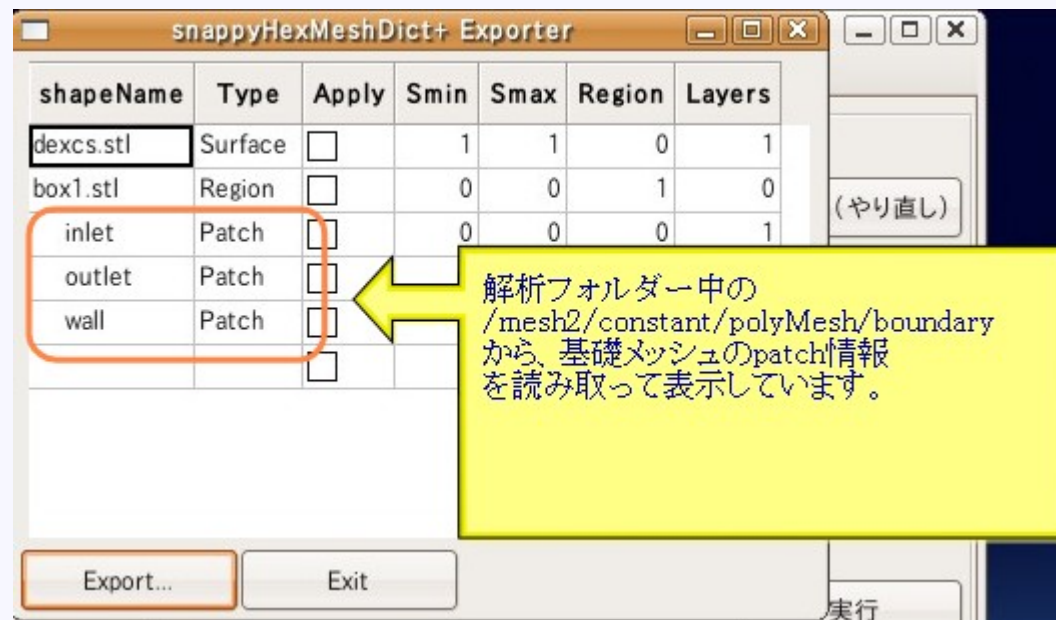
M2-3

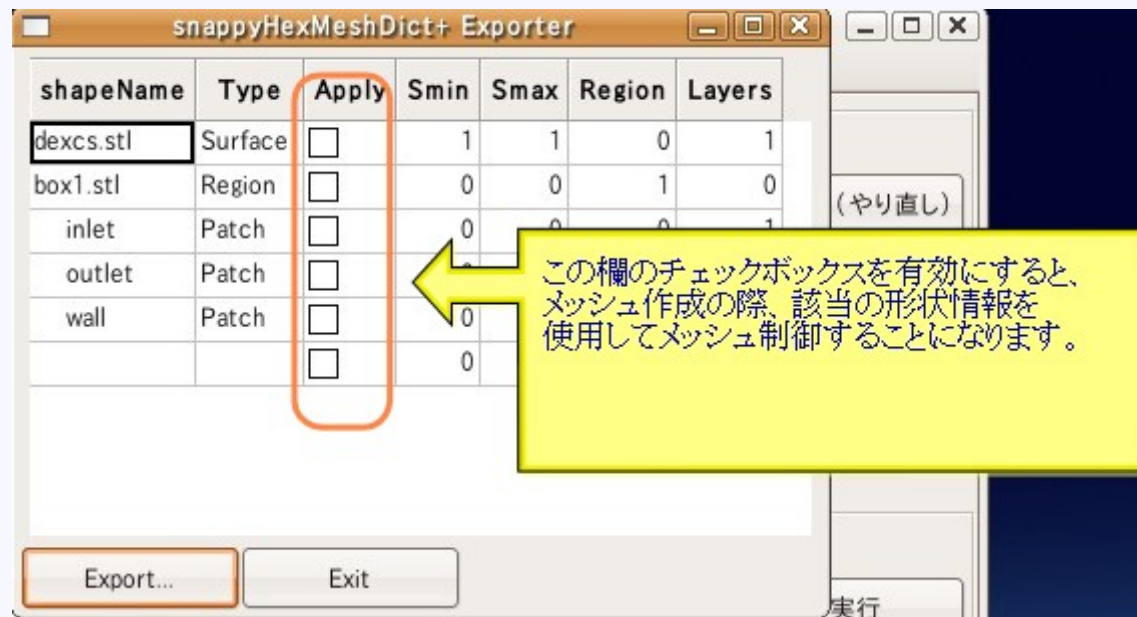
M2-4

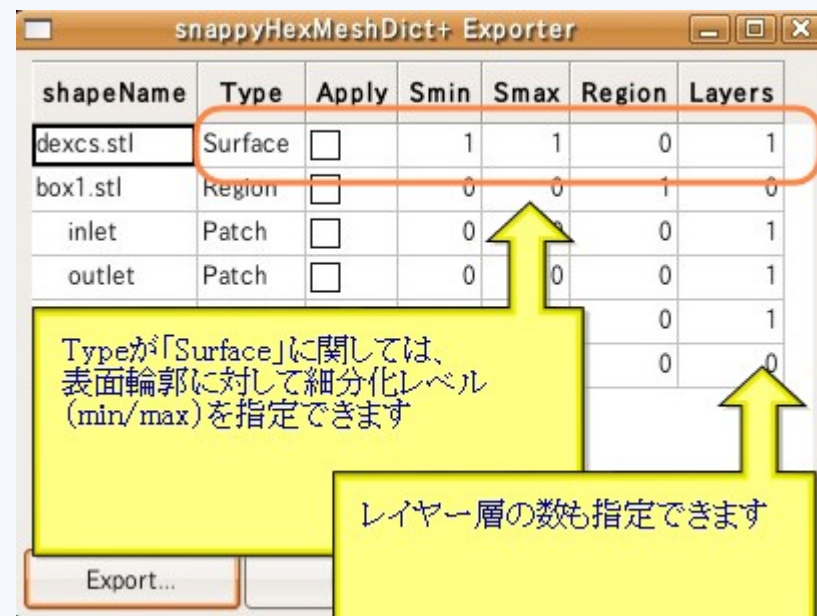
M2-5

M2-7

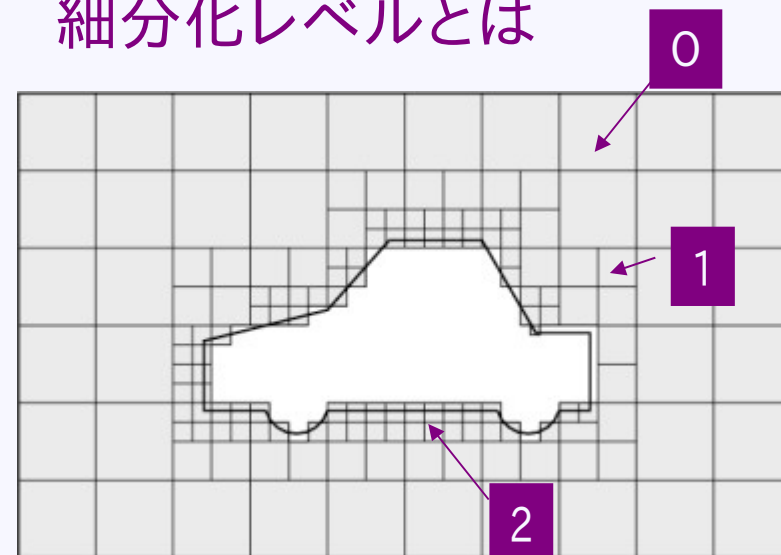








細分化レベルとは



snappyHexMeshDict+ Exporter

shapeName	Type	Apply	Smin	Smax	Region	Layers
dexcs.stl	Surface	<input type="checkbox"/>	1	1	0	1
box1.stl	Region	<input type="checkbox"/>	0	0	1	0
inlet	Patch	<input type="checkbox"/>	0	0	0	1
outlet	Patch	<input type="checkbox"/>	0	0	0	1
wall	Patch	<input type="checkbox"/>	0	0	0	1
		<input type="checkbox"/>	0	0	0	0

(やり直し)

実行

Export... Exit

メッシュ確定-> exe

解析実行できます

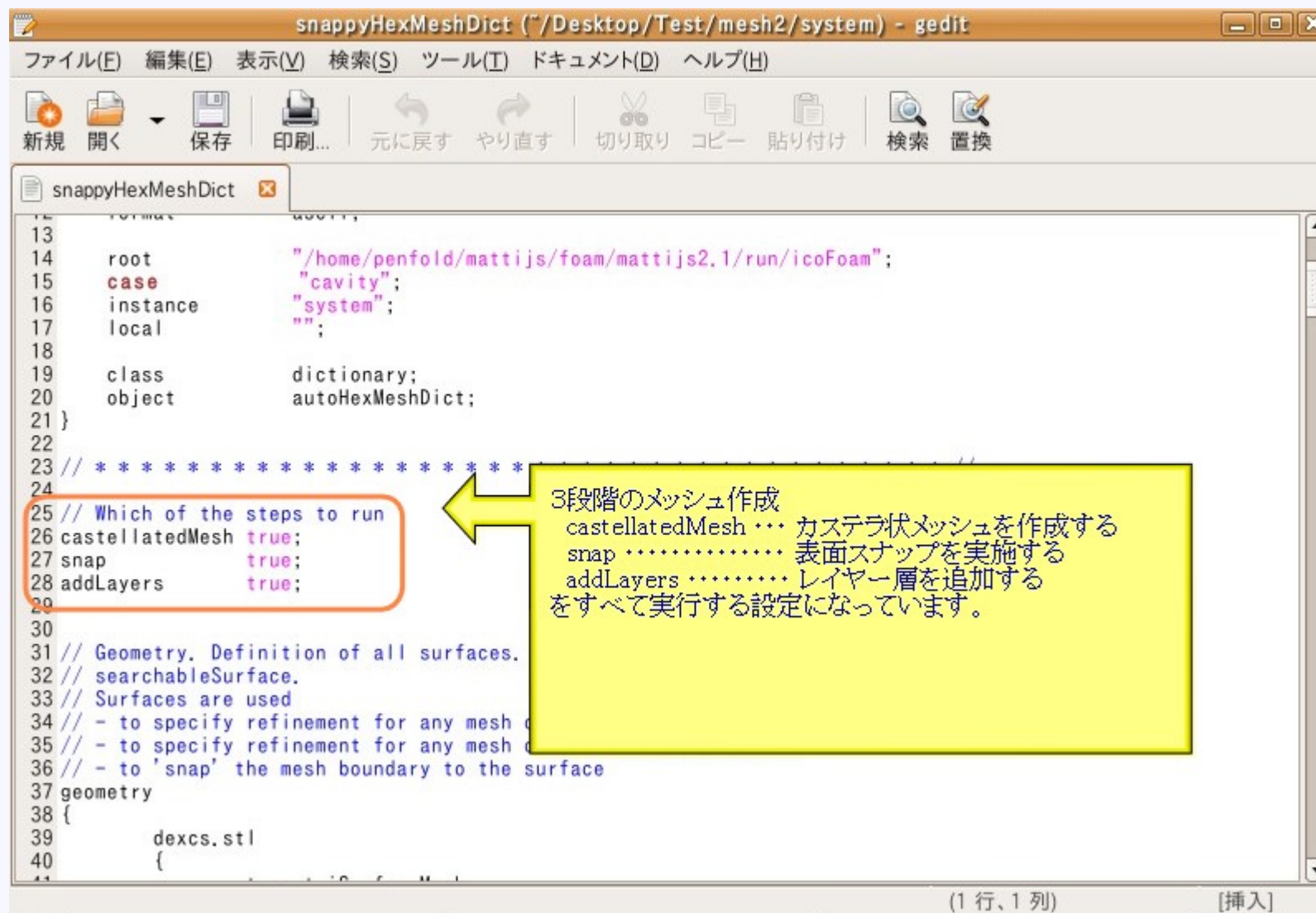
```
## EXEC ... /usr/local/OpenFOAM/OpenFO
dexcs.stl 21087
box1.stl 87
```

現在の解析フォルダ: /home/d... snappyHexMeshD

Exportボタンを押せば
この画面で指定した値が反映された
snappyHexMeshDict (暫定版) を自動作成し、
ついでExitボタンを押せば、
書き換えるかどうか聞いてきます。

ここでは何もしないでExitボタンを押します。

(現画面は何もチェックしていないので、
何もしないし、聞かれることもありません)



形状情報を定義してあります

aces. All surfaces are of class

```
34 // to specify refinement for any mesh cell intersecting it
35 // - to specify refinement for any mesh cell inside/outside/near
36 // - to 'snap' the mesh boundary to the surface
37 geometry
38 {
39     dexcs.stl
40     {
41         type triSurfaceMesh;
42         name dexcs;
43     }
44     box1.stl
45     {
46         type triSurfaceMesh;
47         name box1;
48     }
49 };
50
51
```

dexcs.stlファイルという
stlファイルから読み取って、
以下、dexcsという名前で定義し、
処理方法を記述していくことになります

矩形領域です

```
103 // angle > resolveFeatureAngle get refined up to the maximum level.
104
105 refinementSurfaces
106 {
107     dexcs
108     {
109         // Surface-wise min and max
110         level (3 4);
111     }
112 }
113
114 // Resolve sharp angles
115 resolveFeatureAngle 30;
116
```

指定表面に対する細分化レベルを指定しています


```

130
131 refinementRegions
132 {
133     box1
134     {
135         mode inside;
136         levels ((1E15 3));
137     }
138 }
139
140
141 // Mesh selection
142 // -----

```

特定領域内部の細分化レベルを指定しています

```

140
141 // Mesh selection
142 // -----
143
144 // After refinement patches get added for all refinementSurfaces and
145 // all cells intersecting the surfaces get put into these patches. The
146 // section reachable from the locationInMesh is kept.
147 // NOTE: This point should never be on a face, always inside a cell, even
148 // after refinement.
149 // locationInMesh (0.1 0.1 3.5)
150 // locationInMesh (-6 -4.5 -0.8);
151 }
152
153
154
155 // Settings for the snapping.
156 snapControls

```

この座標値を含んだ領域にて
メッシュ作成されることになります

形状変更した場合などは、必要に応じて、
この部分は手作業で変更することになります

```

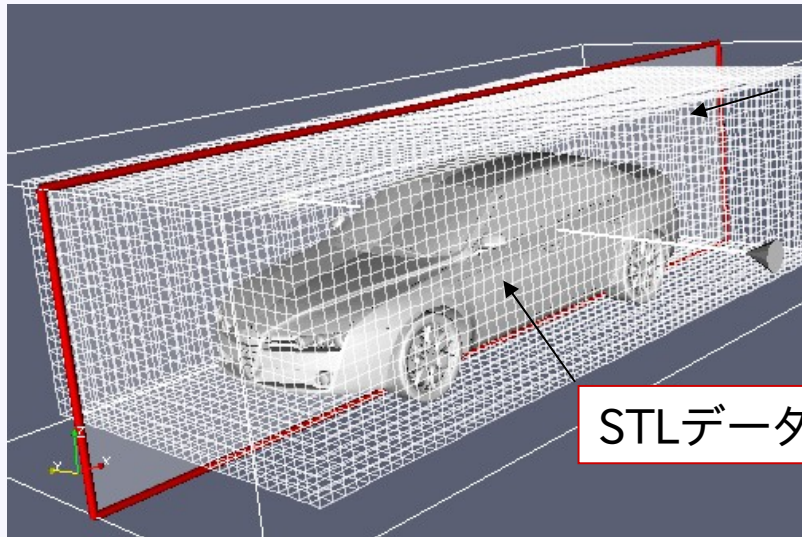
176
177 // Settings for the layer addition.
178 addLayersControls
179 {
180     // Per final patch (so not geometry!) the layer information
181     layers
182     {
183         dexcs_Font
184         {
185             nSurfaceLayers 1;
186         }
187     }
188
189 // Expansion factor for layer mesh
190 expansionRatio 1.0;
191
192 //- Wanted thickness of final added
193 // is the

```

レイヤー層を指定しています。

ここでは、dexcs_Font
(dexcs.stl中の、fontというパッチ)
に対して、レイヤー層を1

locationInMesh

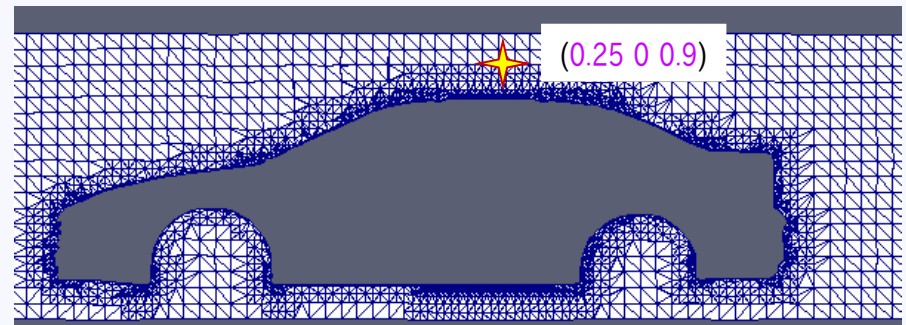
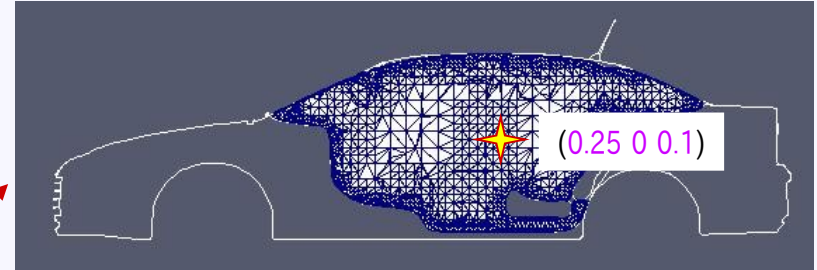


デフォルト設定値 (0.1 0.1 3.5)・・・領域外部点

⇒snappyHexMesh実行時エラー

```
snappyHexMeshDict
129
130
131 // Mesh selection
132 // -----
133
134 // After refinement patches get
135 // all cells intersecting the su
136 // section reachable from the lo
137 // NOTE: This point should never
138 // after refinement.
139 // locationInMesh (0.1 0.1 3.5)
140 locationInMesh (0.25 0 0.1);
141 // locationInMesh (0.25 0 0.9);
142 }
```

要手修正



snappyHexMeshDict (/Desktop/Test/mesh2/)

ファイル(E) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

新規 開く 保存 印刷... 元に戻す やり直す 切り取り コピー 貼り付け

snappyHexMeshDict

```
192 // - Wanted thickness of final added cell layer. If multiple
193 // is the
194 // thickness of the layer furthest away from the wall.
195 // Relative to undistorted size of cell outside layer.
196 finalLayerRatio 0.3;
197
198 // - Minimum thickness of
199 // cannot be above min
200 // Relative to undistorted
201 minThickness 0.1;
202
203 // - If points get not ex
204 // also not grown. This
205 // close to features.
206 nGrow 1;
207
208
209 // Advanced settings
210
211 // - When not to extrude surface. 0 is flat surface, 90
212 // make straight angle.
213 featureAngle 30;
214
215 // - Maximum number of snapping relaxation iterations. Should stop
216 // before upon reaching a correct mesh.
217 nRelaxIter 3;
218
219 // Number of smoothing iterations of surface normals
```

(1 行, 1 列) [挿入]

stdout/stderr 現在の解析フォルダ: /home/d... snappyHexMeshD

第5章 メッシュの生成と変換

5.1 メッシュの記法

5.2 境界

5.3 blockMeshユーティリティを使ったメッシュ生成

5.4 snappyHexMeshユーティリティを使ったメッシュ生成

5.4.1 snappyHexMeshによるメッシュ生成の過程

5.4.2 六面体基礎メッシュの作成

5.4.3 面と輪郭に合わせたセルの分割

5.4.4 セルの除去

5.4.5 特定領域内のセルの分割

5.4.6 面へのスナップ

5.4.7 メッシュレイヤ

5.4.8 メッシュの水準

5.5 メッシュの変換

5.6 異なるジオメトリ間のフィールドマッピング

snappyHexMeshDictには、その他にも多くのメッシュ制御パラメタが定義してありますが、通常は、ほとんどデフォルト値でそのまま使えます。

それぞれのパラメタの意味は、注釈の内容から、ある程度推定できると思いますが、詳細は OpenFoam User's Guide の5.4を参照して下さい。

OpenFoam User's Guide(または和訳版)は、ランチャーのヘルプメニューから参照できます

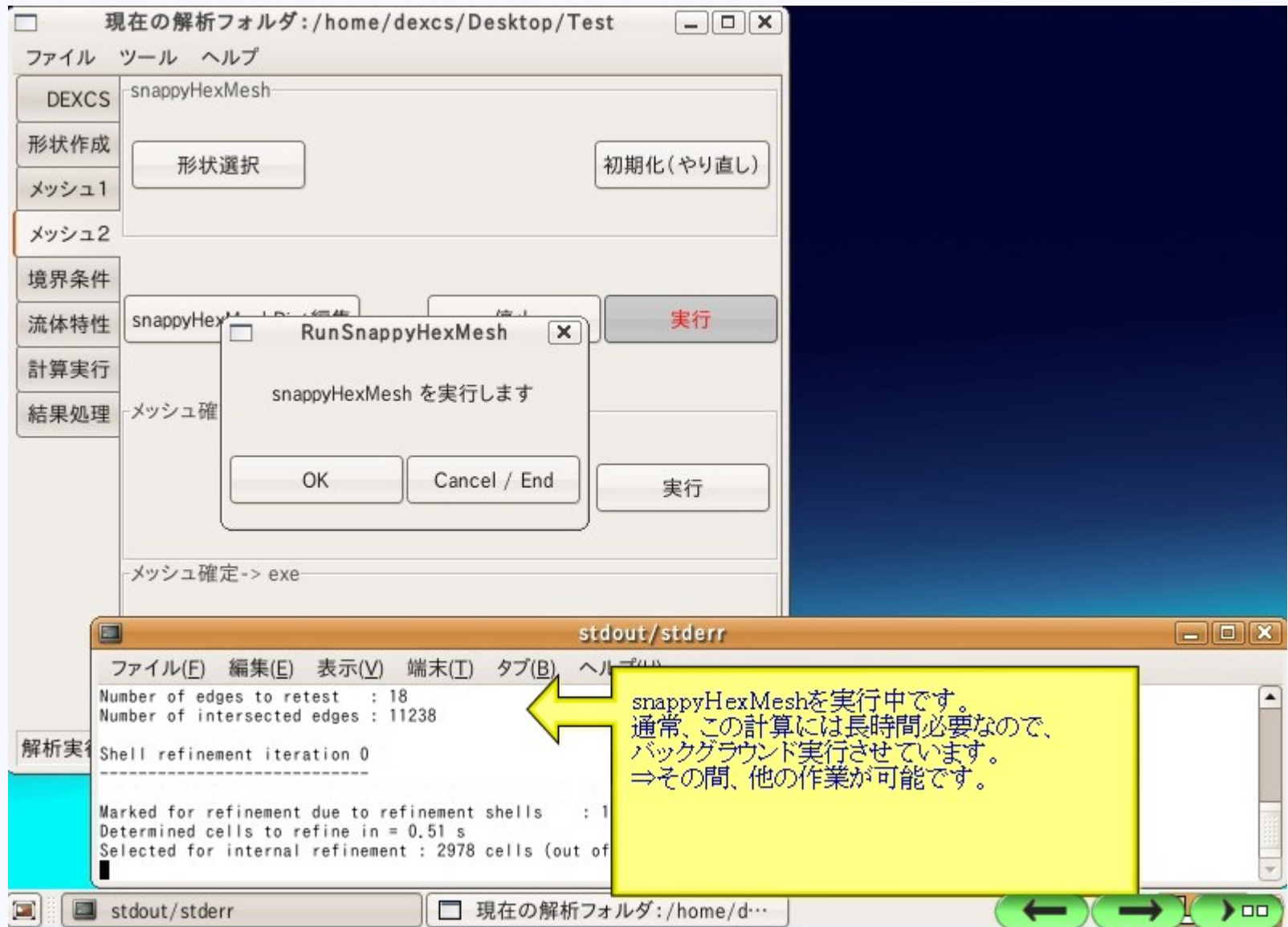
OpenFOAM
オープンソース CFD ツールボックス

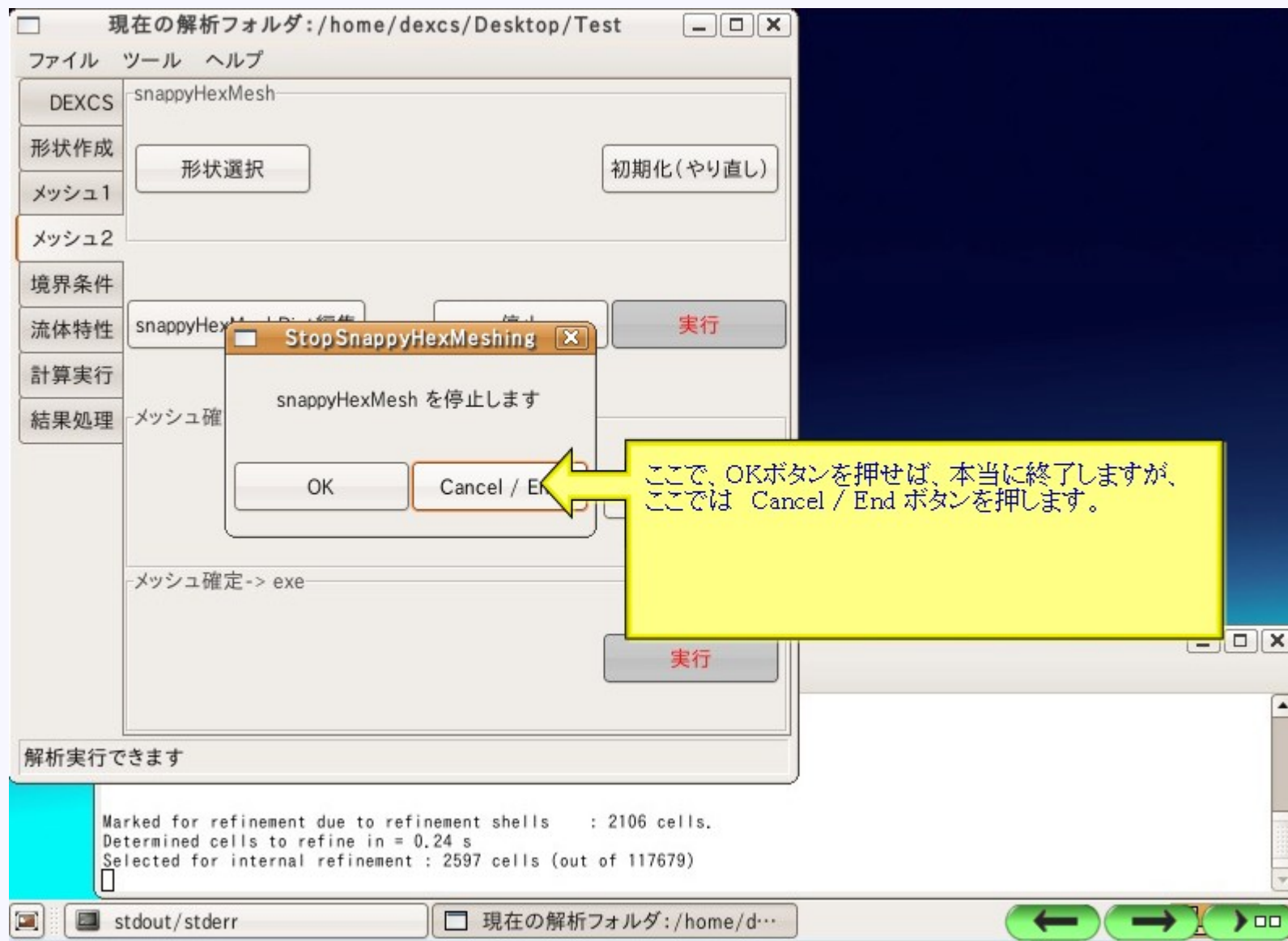
ユーザガイド和訳

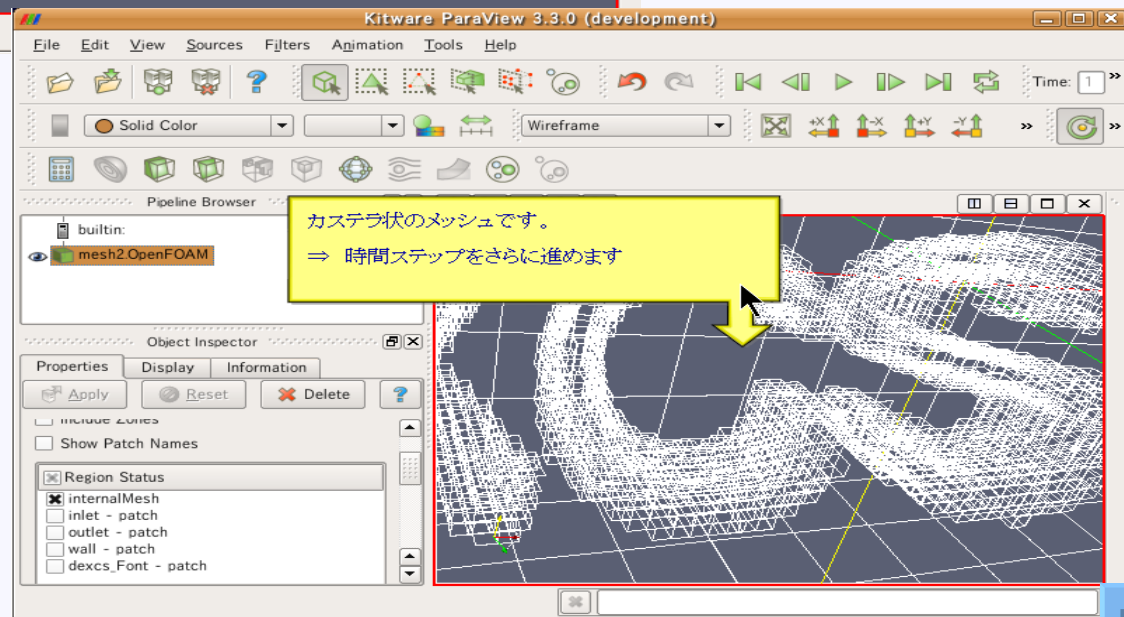
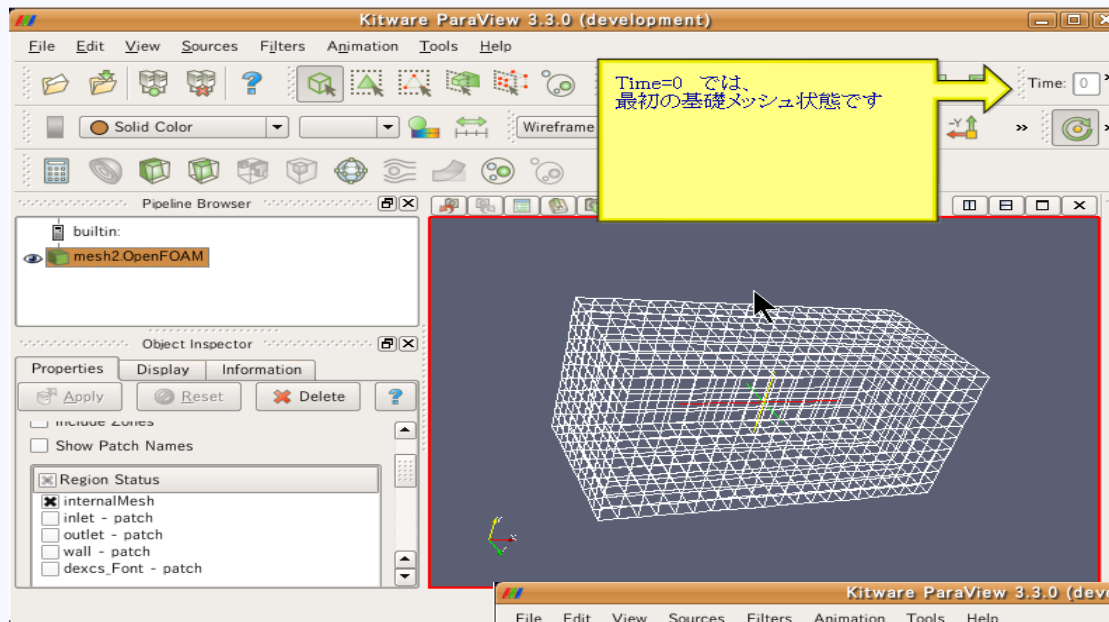
Version 1.5
2010 年 3 月 1 日

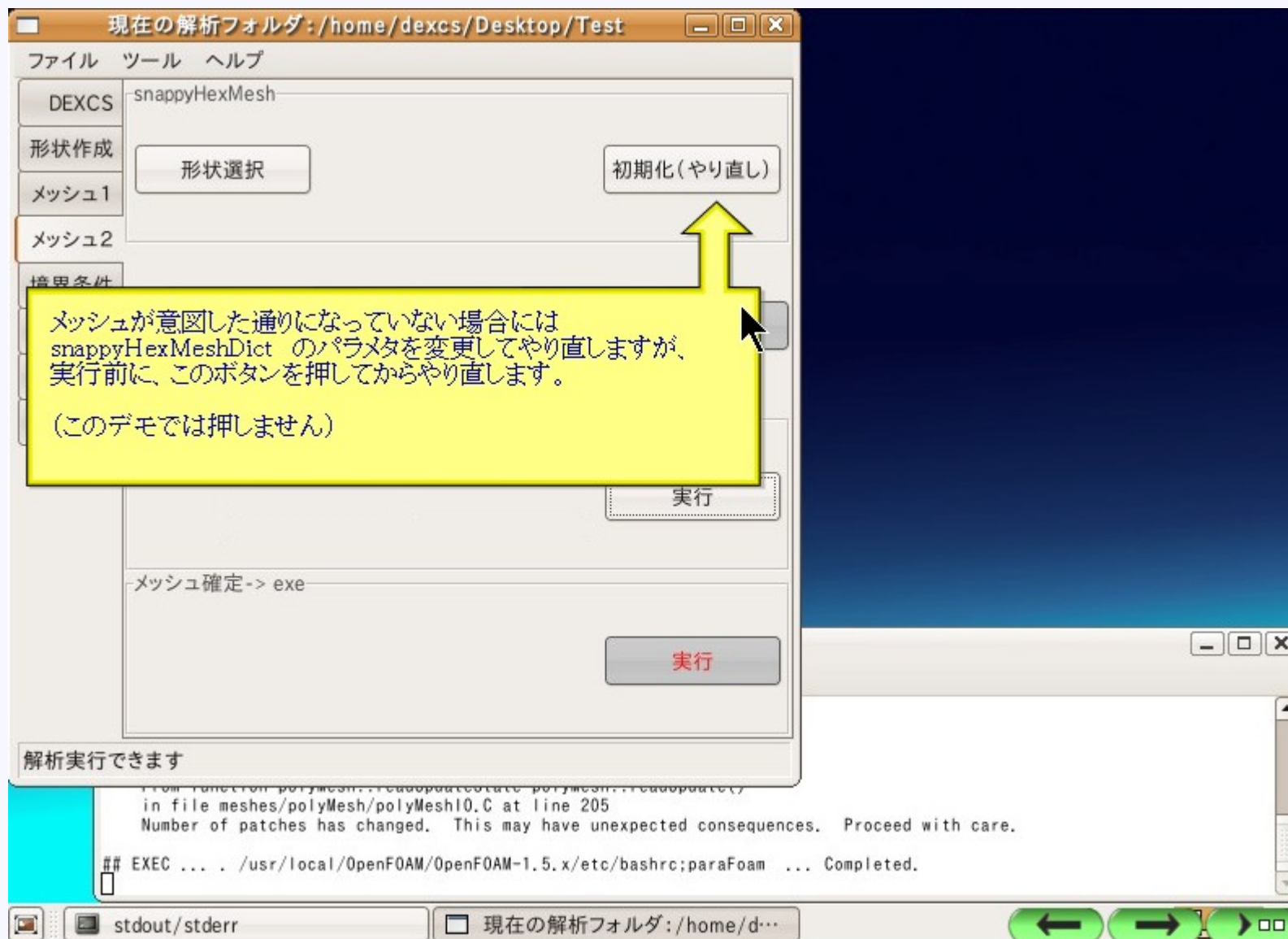
OpenFOAM ユーザー会
一般社団法人 オープン CAE 学会

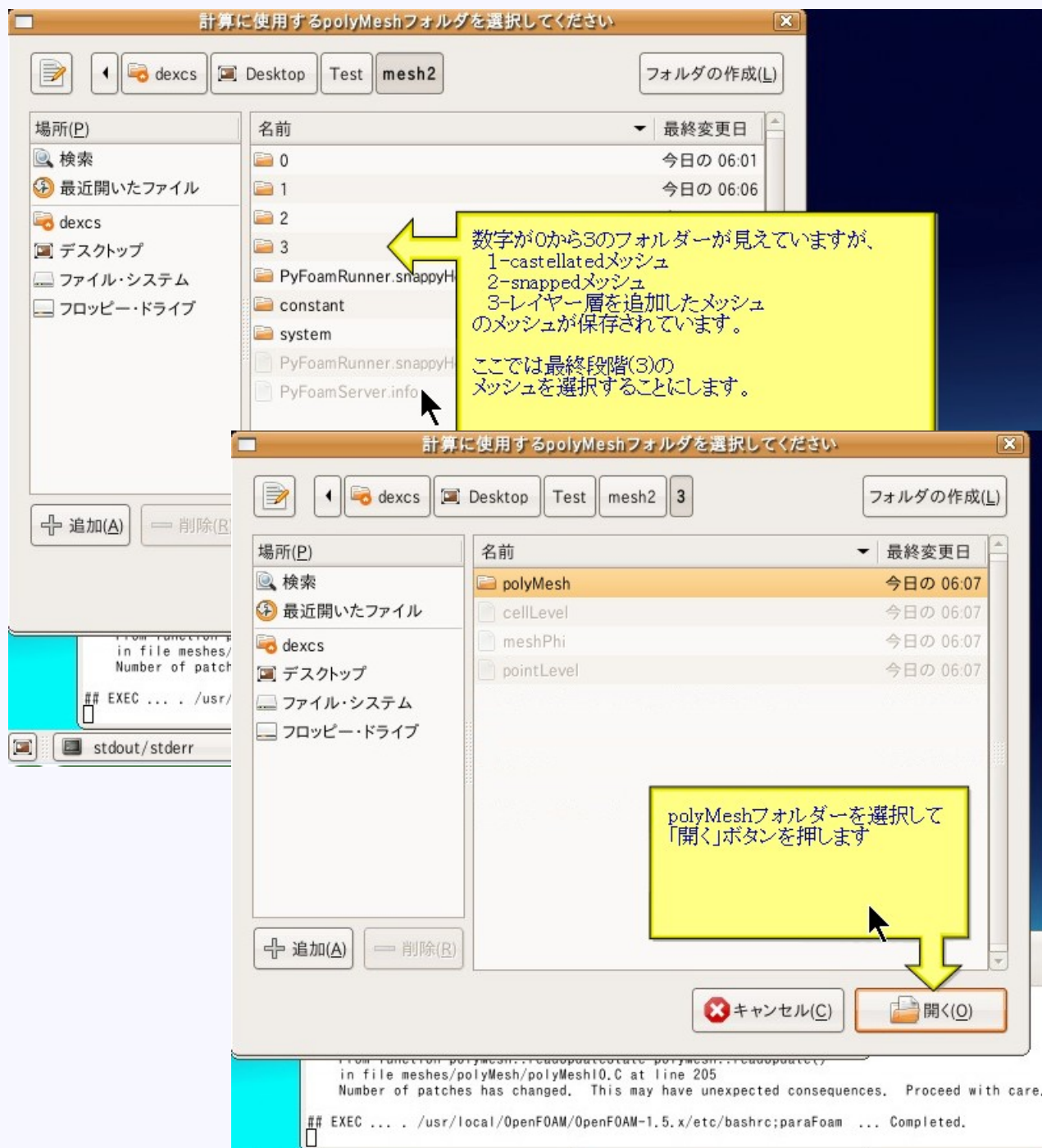












境界条件

現在の解析フォルダ: /home/dexcs/Desktop/Test

ファイル ツール ヘルプ

DEXCS

形状作成

メッシュ1

メッシュ2

境界条件

流体特性

計算実行

結果処理

p

初期化 patch名取得 編集

U

初期化 patch名取得 編集

k

初期化 patch名取得 編集

epsilon

初期化 patch名取得 編集

omega

初期化 patch名取得 編集

境界条件と初期条件の設定です。

モデルを何も変更しなければ、デフォルト設定でそのまま計算開始可能ですが、計算条件などを変更したい場合には、このパネルを使います。

ここではボタンの説明と条件内容を確認しておくのにとどめます。

simpleFoamでは、4つの変数

- ・p (圧力)
- ・U (流速)
- ・k (乱流エネルギー)
- ・epsilon (散逸率)

または

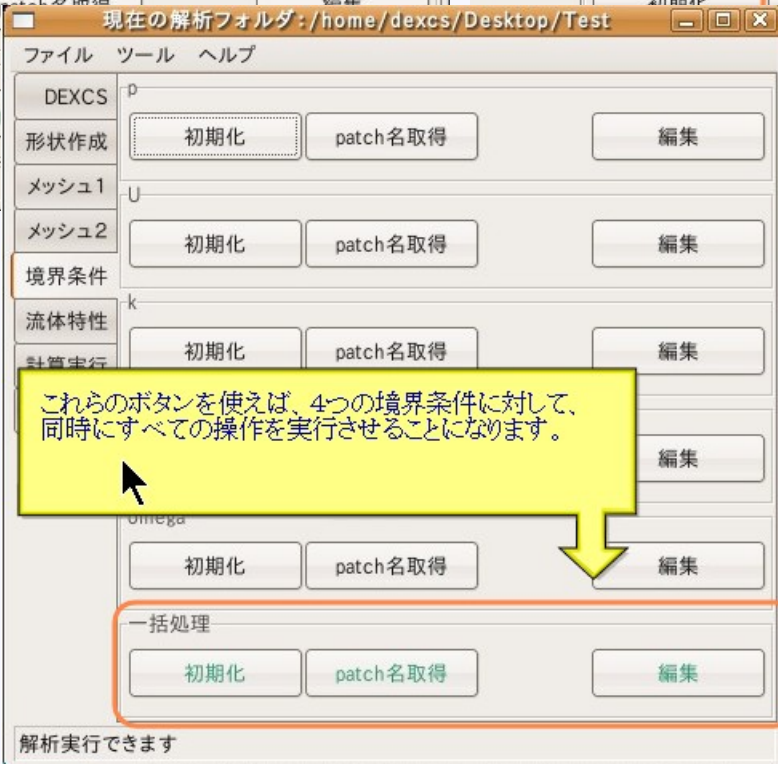
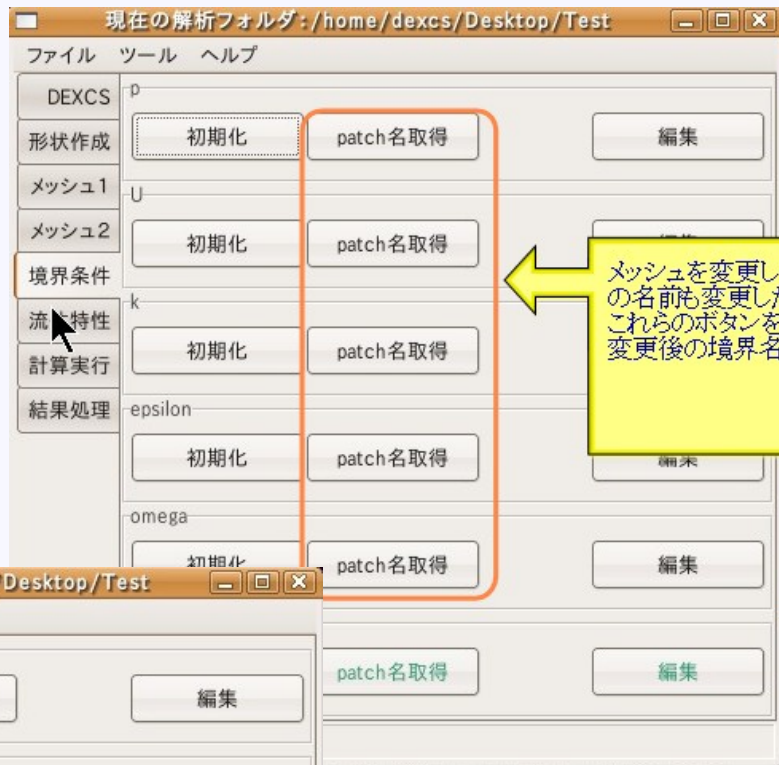
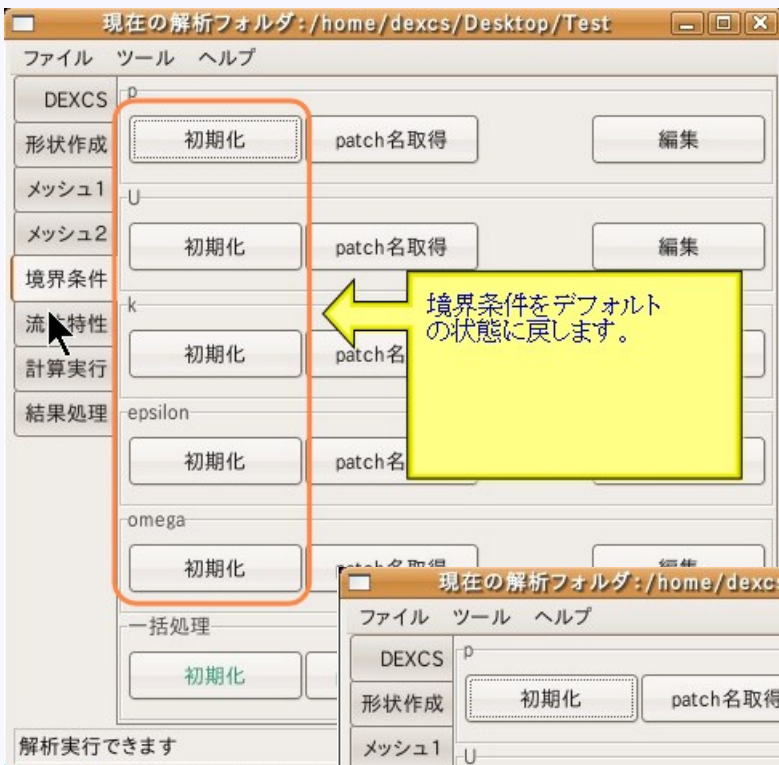
- ・omega (比散逸率)

について定義しておく必要があり、これらのボタンを押して個別に直接編集します。

解析実行でき

stdout/stderr

現在の解析フォルダ: /home/d...



p (圧力) の境界&初期条件です

```

5 {
6  version 2.0;
7  format ascii;
8  class volScalarField;
9  object p;
10 }
11
12 dimensions [ 0 2 -2 0 0 0 ];
13 internalField uniform 0;
14 boundaryField
15 {
16  wall|
17  {
18    type slip;
19  }
20  inlet
21  {
22    type zeroGradient;
23  }
24  dexcs_Font
25  {
26    type zeroGradient;
27  }
28  outlet
29  {
30    type fixedValue;
31    value uniform 0.0;
32  }
33 }

```

外周はスリップ壁

出口圧力を規定

U (流速) の境界&初期条件です

```

6  version 2.0;
7  format ascii;
8  class volVectorField;
9  object U;
10 }
11
12 dimensions [ 0 1 -1 0 0 0 ];
13 internalField uniform (0 0 0);
14 boundaryField
15 {
16  wall
17  {
18    type slip;
19  }
20  inlet
21  {
22    type value
23    {
24      type value
25      {
26        type value
27        {
28          type value
29          {
30            type value
31            {
32              type value
33              {
34                type value

```

外周はスリップ壁

流入条件を規定

固体壁(すべり無)

- 第5章 メッシュの生成と変換
 - 5.1 メッシュの記法
 - 5.2 境界
 - 5.2.1 パッチの形式の類型化
 - 5.2.2 基底型
 - 5.2.3 基本型
 - 5.2.4 派生型

OpenFOAM
オープンソース CFD ツールボックス

ユーザガイド和訳

Version 1.5
2010年3月1日
OpenFOAM ユーザー会
一般社団法人 オープン CAE 学会

ファイル(E) 編集(E) 表示(V) k (~/Desktop/Test/exe/0) - ge

新規 開く

p x U x k x epsilon x omega x

```

5 {
6  version 2.0;
7  format ascii;
8  class volScalarField;
9  object k;
10 }
11
12 dimensions [ 0 2 -2 0 0 0 ];
13 internalField uniform 0.375;
14 boundaryField
15 {
16     wall
17     {
18         type slip;
19     }
20     inlet
21     {
22         type fixedValue;
23         value uniform 0.375;
24     }
25     dexcs_Front
26     {
27         type zeroGradient;
28     }
29     outlet
30     {
31         type zeroGradient;
32     }
33 }

```

epsilon(散逸率)の境界&初期条件です

外周はスリップ壁

流入条件を規定

ファイル(E) 編集(E) 表示(V) epsilon (~/Desktop/Test/exe/0) - edit

新規 開く 保存 印刷

p x U x k x epsilon x omega x

```

5 {
6  version 2.0;
7  format ascii;
8  class volScalarField;
9  object epsilon;
10 }
11
12 dimensions [ 0 2 -3 0 0 0 ];
13 internalField uniform 14.855;
14 boundaryField
15 {
16     wall
17     {
18         type slip;
19     }
20     inlet
21     {
22         type fixedValue;
23         value uniform 14.855;
24     }
25     dexcs_Front
26     {
27         type zeroGradient;
28     }
29 }

```

epsilon(散逸率)の境界&初期条件です

外周はスリップ壁

流入条件を規定

ファイル(E) 編集(E) 表示(V) 検索(S) ツール omega (~/Desktop/Test/exe/0) - edit

新規 開く 保存 印刷... 元に戻す

p x U x k x epsilon x omega x

```

16 // *****
17
18 dimensions [ 0 0 -1 0 0 0 ];
19
20 internalField uniform 167;
21
22 boundaryField
23 {
24     wall
25     {
26         type slip;
27     }
28     inlet
29     {
30         type fixedValue;
31         value uniform 167;
32     }
33     outlet
34     {
35         type zeroGradient;
36     }
37     dexcs_Front
38 }

```

omega (比散逸率)の境界&初期条件です

外周はスリップ壁

流入条件を規定

epsilonの値に
対応の値になっています

乱流変数について

http://www.cfd-online.com/Wiki/Turbulence_free-stream_boundary_conditions



Turbulence free-stream boundary conditions

In most CFD simulations it is necessary to specify values of the turbulence variables at the inlets. For example, if you are using a $k - \epsilon$ model you have to specify values of k and ϵ at the inlets. This is often difficult and a source of uncertainty since the incoming turbulence is rarely known exactly. Most often you are forced to make a more or less educated guess of the incoming turbulence.

Estimating the turbulence model variables, like turbulent energy, dissipation or Reynolds numbers of variables like the density ratio. These variables can be estimated

Turbulent energy

The turbulent energy, k , can be computed as:

$$k = \frac{3}{2} (U I)^2$$

Where U is the mean flow velocity and I is the [turbulence intensity](#).

Dissipation rate

The turbulent dissipation rate, ϵ , can be computed using the following formulas:

From the turbulence length scale

$$\epsilon = C_\mu^{\frac{3}{4}} \frac{k^{\frac{3}{2}}}{l}$$

Where C_μ is a turbulence model constant which usually has a value of 0.09, k is the turbulent energy and l is the [turbulent length scale](#)

Specific dissipation rate

The specific turbulent dissipation rate, ω , can be computed using the following formulas:

From the turbulence length scale

$$\omega = C_\mu^{-\frac{1}{4}} \frac{\sqrt{k}}{l}$$

流体特性

現在の解析フォルダ: /home/

ファイル ツール ヘルプ

DEXCS transportProperties

形状作成

メッシュ1

メッシュ2

境界条件

流体特性

計算実行

結果処理 RASProperties

動粘性係数

編集

FP-1

乱流モデル

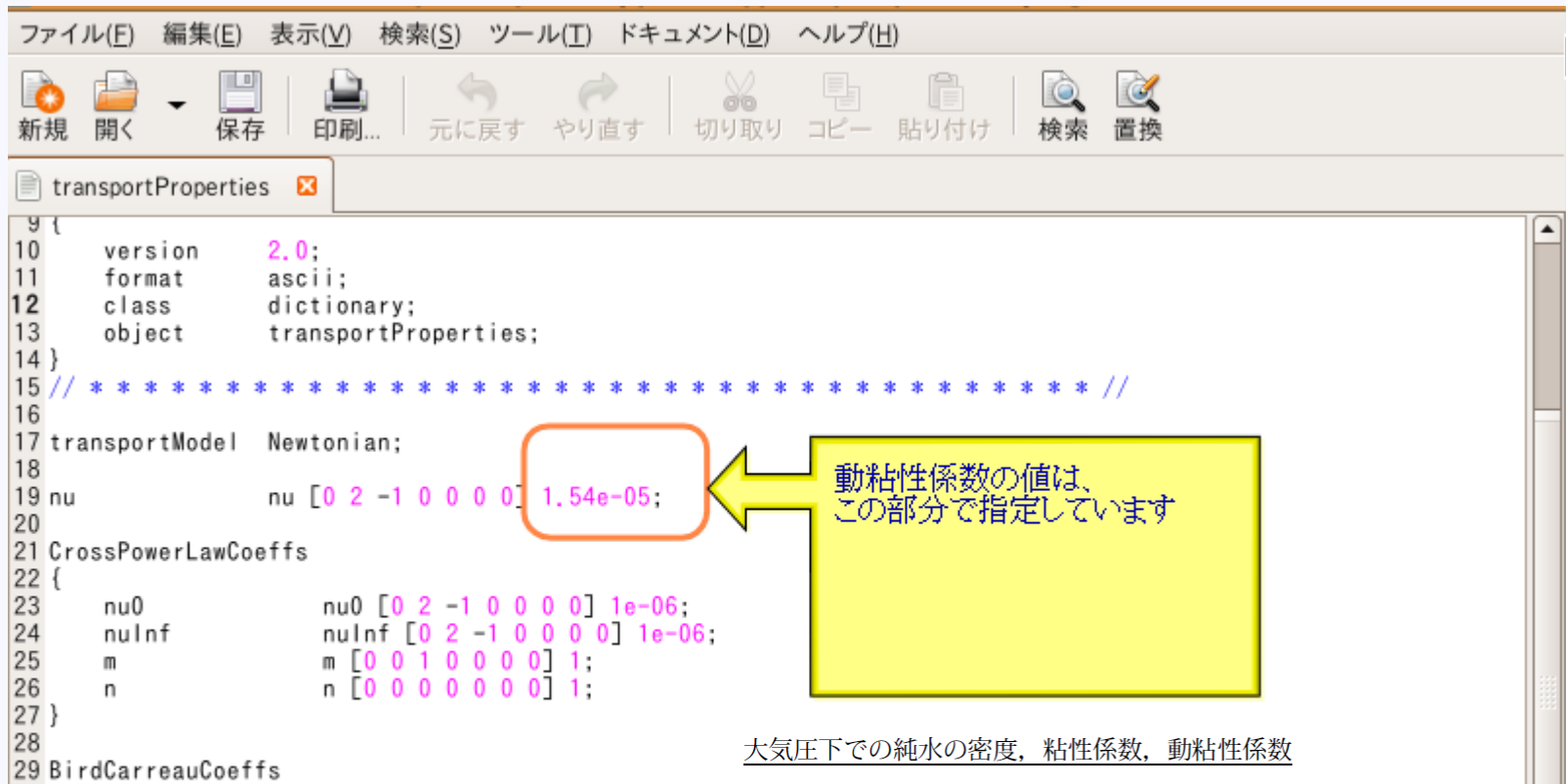
編集

FP-2

解析実行できます

number of patches has changed. This may have unexpected consequences. Proceed with care.

```
## EXEC ... /usr/local/OpenFOAM/OpenFOAM-1.5.x/etc/bashrc;paraFoam ... Completed.  
## EXEC ... gedit ./0/p ./0/U ./0/k ./0/epsilon ./0/omega & ... Completed.
```



大気圧下での空気の数値係数、粘性係数、動粘性係数

温度 T ℃	密度 ρ kg/m ³	粘性係数 μ × 10 ⁻³ Pa·s	動粘性係数 ν × 10 ⁻⁶ m ² /s
-50	1.584	0.0146	9.2
-25	1.424	0.0159	11.2
0	1.293	0.0171	13.2
25	1.184	0.0182	15.4
50	1.093	0.0193	17.7
75	1.014	0.0205	20.2
100	0.946	0.0216	22.8

大気圧下での純水の密度、粘性係数、動粘性係数

温度 T ℃	密度 ρ × 10 ³ kg/m ³	粘性係数 μ × 10 ⁻³ Pa·s	動粘性係数 ν × 10 ⁻⁶ m ² /s
0	0.9998	1.792	1.792
5	1.0000	1.520	1.520
10	0.9997	1.307	1.307
15	0.9991	1.138	1.139
20	0.9982	1.002	1.004
25	0.9970	0.890	0.893
30	0.9957	0.797	0.800
40	0.9922	0.653	0.658
50	0.9880	0.548	0.555
60	0.9832	0.467	0.475
70	0.9778	0.404	0.413
80	0.9718	0.355	0.365
90	0.9653	0.315	0.326

(1989年版理科年表 (国立天文台編, 丸善) より)

RASProperties (~/Desktop/Test/exe/constant) - gedit

ファイル(E) 編集(E) 表示(V) 検索(S) ツール(T) ドキュメント(D) ヘルプ(H)

新規 開く 保存 印刷... 元に戻す やり直す 切り取り コピー 貼り付け 検索 置換

RASProperties

```

13 object RASProperties;
14 }
15 // *****
16
17 RASModel
18 turbulence
19 printCoeffs on;
20
21 laminarCoeffs
22 {
23 }
24
25 kEpsilonCoeffs
26 {
27     Cmu 0.09;
28     C1 1.44;
29     C2 1.92;
30     alphaEps 0.76923;
31 }
32
33 RNGkEpsilonCoeffs
34 {
35     Cmu 0.0845;
36     C1 1.42;
37     C2 1.68;
38     alphak 1.39;
39 }
40

```

乱流モデルで、 $k-\varepsilon$ モデルを使っている

モデルパラメタの値です

OpenFOAM
オープンソース CFD ツールボックス

ユーザガイド和訳

Version 1.5
2010 年 3 月 1 日
OpenFOAM ユーザー会
一般社団法人 オープン CAE 学会

- 第3章 アプリケーションとライブラリ
 - 3.1 OpenFOAMのプログラミング言語
 - 3.2 アプリケーションやライブラリのコンパイル
 - 3.3 アプリケーションの実行
 - 3.4 アプリケーションの並列実行
 - 3.5 標準のソルバ
 - 3.6 標準のアプリケーション
 - 3.7 標準のライブラリ
- 第7章 モデルと物性値
 - 7.1 熱物理モデル
 - 7.2 乱流モデル

計算実行



C-1

C-2

C-3

C-7

C-4

C-5

C-6

第4章 OpenFOAMのケース

4.1 OpenFOAMのケースのファイル構造

4.2 基本的な入出力ファイルのフォーマット

4.3 時間とデータの入出力制御

4.4 数値スキーム

4.5 解法とアルゴリズム制御

```
controlDict
17 application simpleFoam;
18
19 startFrom      startTime;
20
21 startTime      0;
22
23 stopAt         endTime;
24
25 endTime        1000;
26
27 deltaT         1;
28
29 writeControl    timeStep;
30
31 writeInterval   50;
32
33 purgeWrite      0;
34
35 writeFormat     ascii;
36
37 writePrecision  6;
38
39 writeCompression uncompressed;
40
41 timeFormat      general;
42
43 timePrecision   6;
44
45 runTimeModifiable yes;
```

controlDict には、
時間とデータの入出力制御パラメタを記述してあります。

それぞれのパラメタの意味は、英単語の内容から、
ある程度推定できると思いますが、
詳細は OpenFoam User's Guide の4.3を参照して下さい。

OpenFoam User's Guide(または和訳版)は、
ランチャーのヘルプメニューから参照できます

ファイル '/home/dexcs/Desktop/Test/exe/system/controlDict' を読み込んでいます...

(21 行, 19 列)

[挿入]

第4章 OpenFOAMのケース
4.1 OpenFOAMのケースのファイル構造
4.2 基本的な入出力ファイルのフォーマット
4.3 時間とデータの入出力制御
4.4 数値スキーム
4.5 解法とアルゴリズム制御

```
fvSchemes
20 /
21
22 gradSchemes
23 {
24     default          Gauss linear;
25     grad(p)          Gauss linear;
26     grad(U)          Gauss linear;
27 }
28
29 divSchemes
30 {
31     default          none;
32     div(phi,U)       Gauss upwind;
33     div(phi,k)       Gauss upwind;
34     div(phi,epsilon) Gauss upwind;
35     div(phi,R)       Gauss upwind;
36     div(R)           Gauss linear;
37     div(phi,nuTilda) Gauss upwind;
38     div((nuEff*dev(grad(U).T()))) Gauss linear;
39 }
40
41 laplacianSchemes
42 {
43     default          none;
44     laplacian(nuEff,U) Gauss linear corrected;
45     laplacian((1/A(U)),p) Gauss linear corrected;
46     laplacian(DkEff,k) Gauss linear corrected;
47     laplacian(DepsilonEff,epsilon) Gauss linear corrected;
48     laplacian(DREff,R) Gauss linear corrected;
```

fvSchemes では、
離散化の際の数値スキームを記述してあります。

それぞれのパラメタの意味は、英単語の内容から、
ある程度推定できると思いますが、
詳細は OpenFoam User's Guide の ~~4.3~~ を参照して下さい。

4.4

OpenFoam User's Guide(または和訳版)は、
ランチャーのヘルプメニューから参照できます

(12 行、28 列)

[挿入]

第4章 OpenFOAMのケース

- 4.1 OpenFOAMのケースのファイル構造
- 4.2 基本的な入出力ファイルのフォーマット
- 4.3 時間とデータの入出力制御
- 4.4 数値スキーム
- 4.5 解法とアルゴリズム制御

```

fvSolution (/De
ファイル(E) 編集(E) 表示(V) 検索(S) ツール(T) ド
新規 開く 保存 印刷... 元に戻す やり直す
fvSolution
45 preconditioner DILU;
46 tolerance 1e-05;
47 relTol 0.1;
48 };
49 nuTilda PBiCG
50 {
51 preconditioner DILU;
52 tolerance 1e-05;
53 relTol 0.1;
54 };
55 }
56
57 SIMPLE
58 {
59 nNonOrthogonalCorrectors 0;
60 convergence 1e-3;
61 }
62
63 relaxationFactors
64 {
65 p 0.3;
66 U 0.7;
67 k 0.7;
68 epsilon 0.7;
69 R 0.7;
70 nuTilda 0.7;
71 }
72
73 // *****

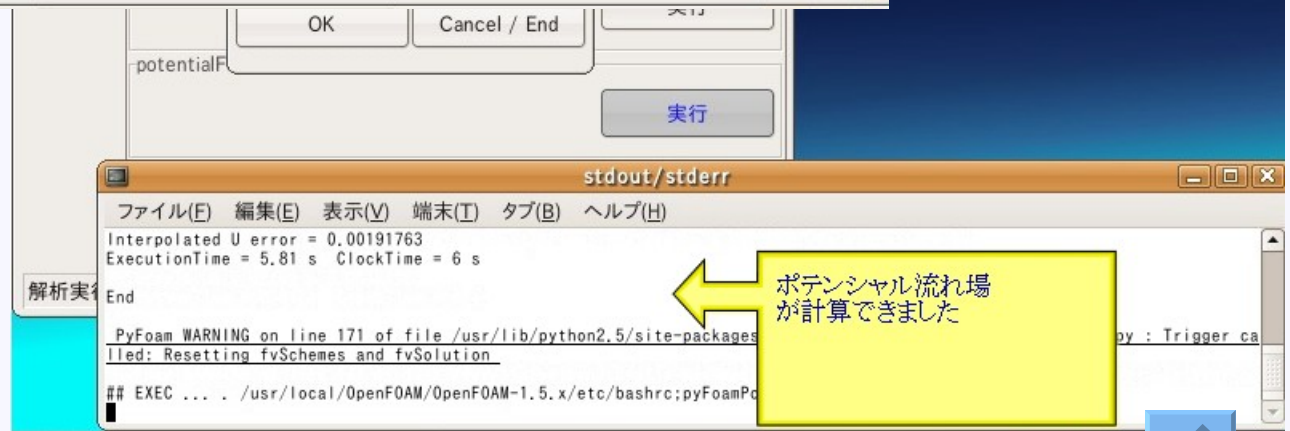
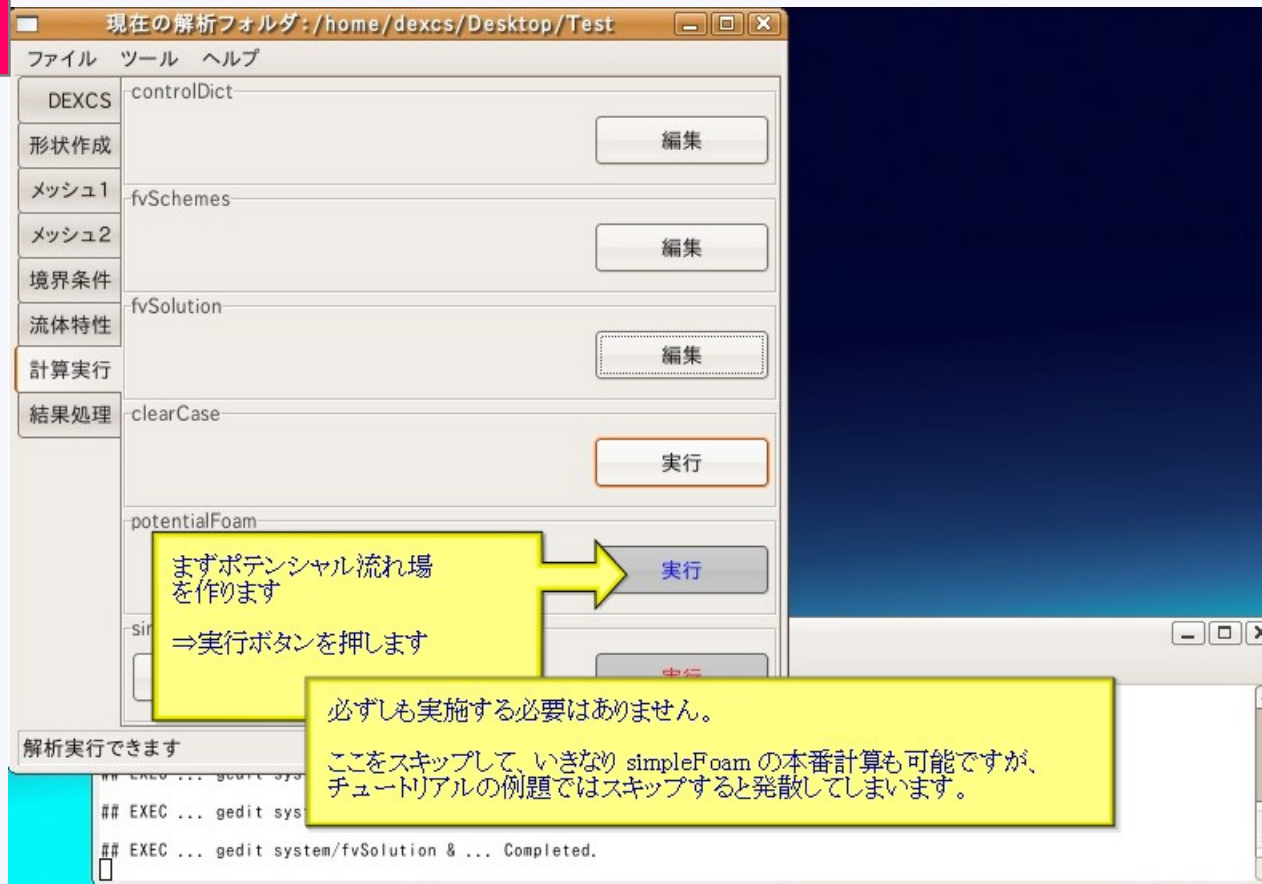
```

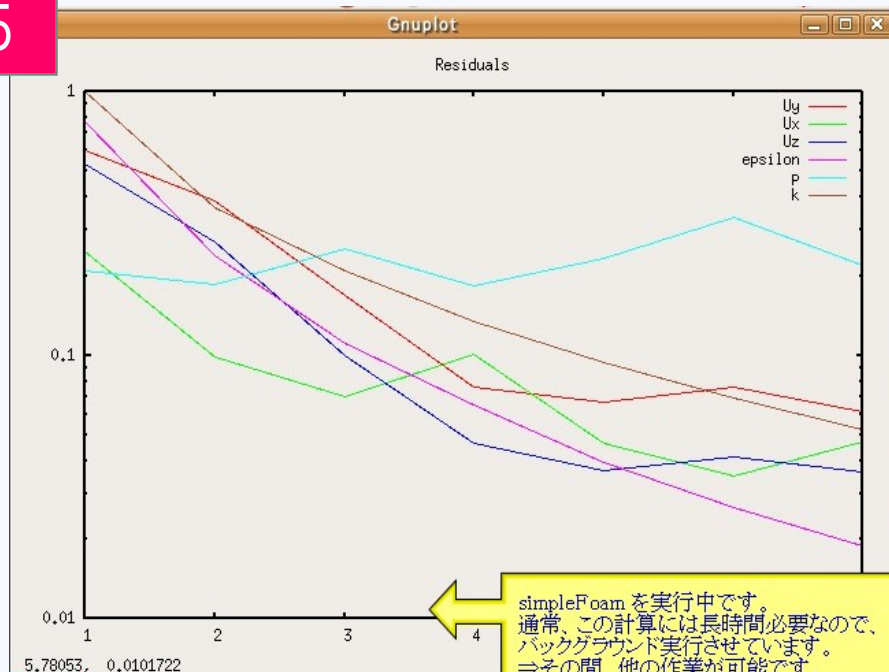
fvSolution には、
解法とアルゴリズム制御に関するパラメタを記述してあります。

それぞれのパラメタの意味は、英単語の内容から、
ある程度推定できると思いますが、
詳細は OpenFoam User's Guide の4.5を参照して下さい。

4.5

OpenFoam User's Guide(または和訳版)は、
ランチャーのヘルプメニューから参照できます





simpleFoam を実行中です。
通常、この計算には長時間必要なので、
バックグラウンド実行させています。
⇒その間、他の作業が可能です。

DILUPBiCG: Solving for Ux, Initial residual = 0.0
DILUPBiCG: Solving for Uy, Initial residual = 0.0
DILUPBiCG: Solving for Uz, Initial residual = 0.0



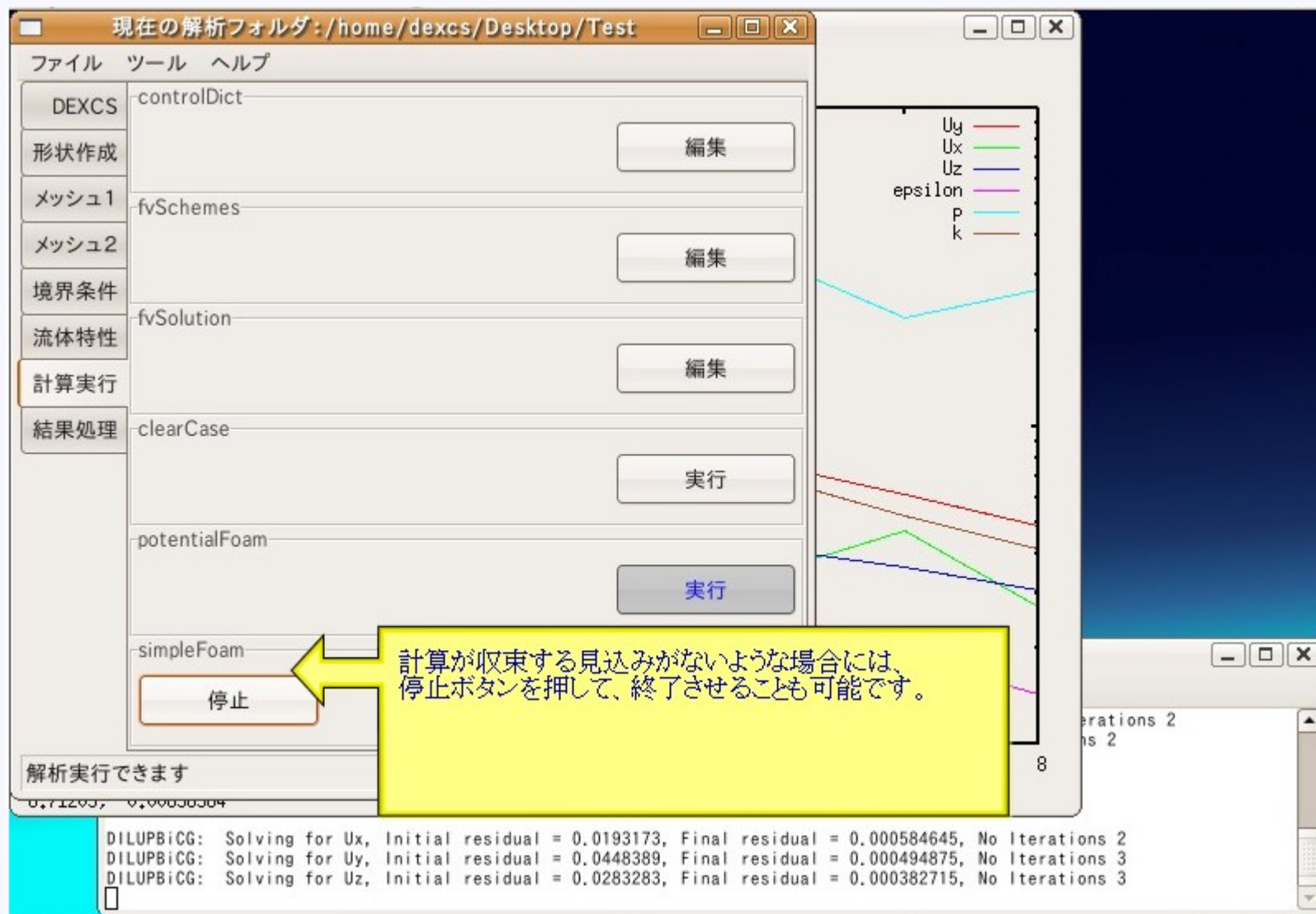
計算は正常に終了しているようです

計算は終了していても、何やらエラーメッセージが出て、
制御が戻らない場合があります。
⇒その場合、Ctrl+C キーを押して復帰させて下さい

ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)

time step continuity errors : sum local = 2.30182e-06, global = -2.75715e-07, cumulative = 7.82212e-06
DILUPBiCG: Solving for epsilon, Initial residual = 1.587e-05
DILUPBiCG: Solving for k, Initial residual = 0.0001
ExecutionTime = 332.98 s ClockTime = 385 s

reached convergence criterion: 0.001
latestTime = 100
End

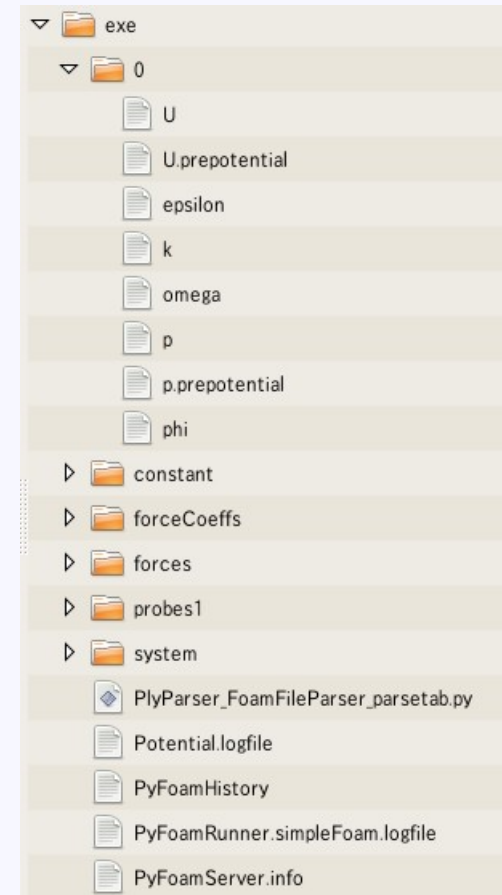
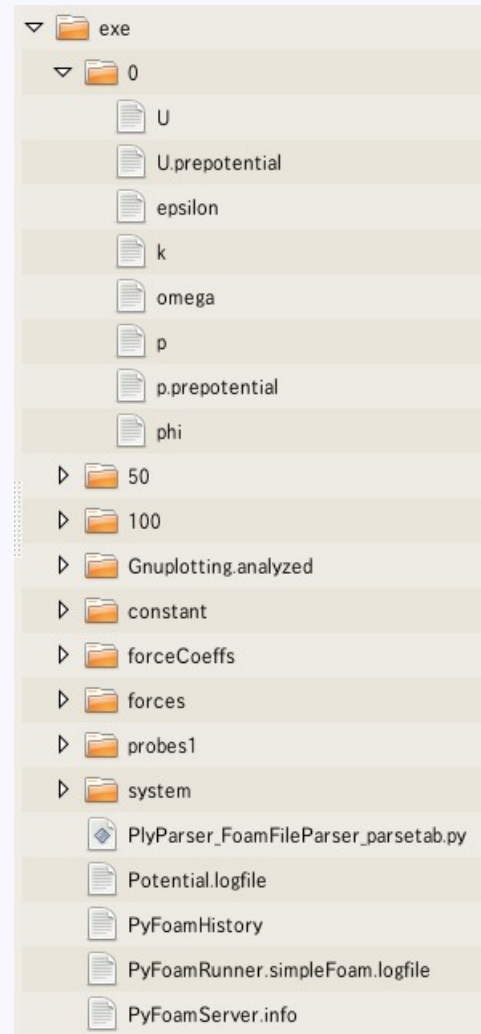
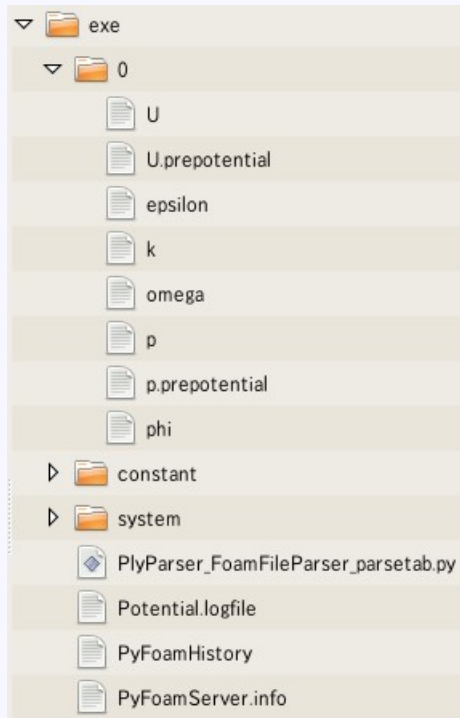
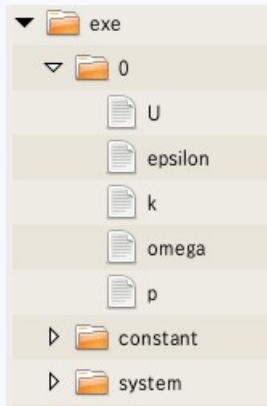


新規作成

potentialFoam

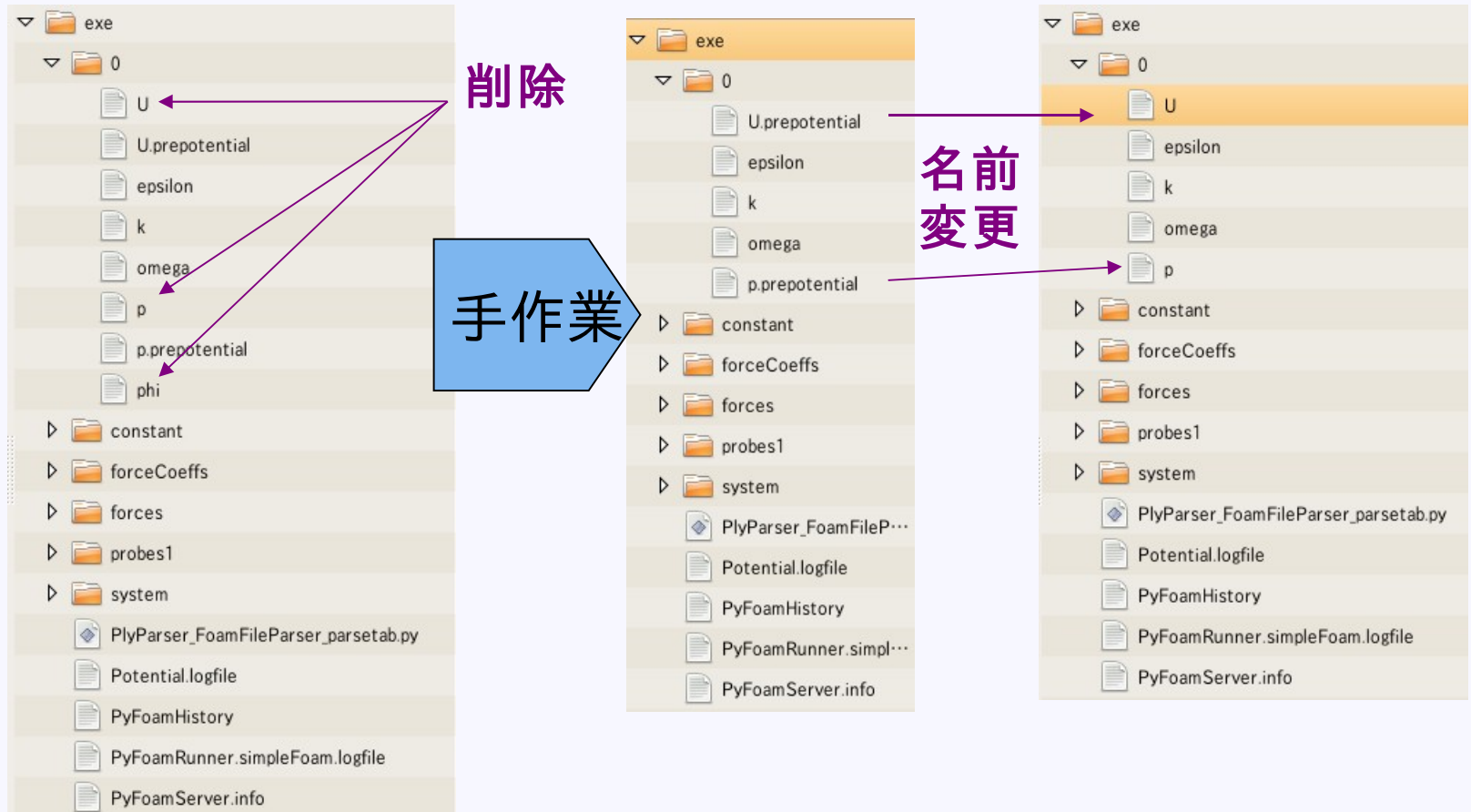
simpleFoam

→ clearCase



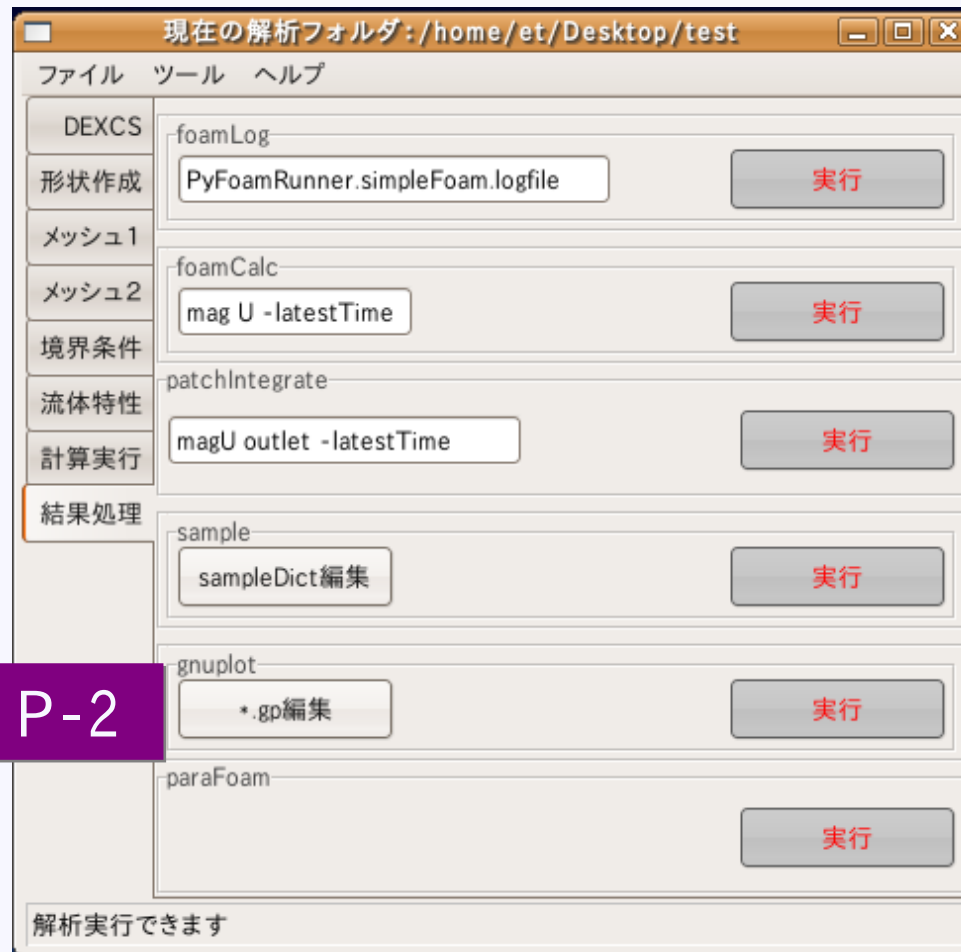
メッシュ変更なし、初期 (potential) 流れ場計算後の再計算はOK

clearCase実行後



メッシュを変更した場合には、U,pファイルの初期化(手作業)が必要

結果処理



P-1

P-4

P-5

P-2

P-3

P-6

このデモでは、残渣プロットを、
改めてやり直すことにします

実行ボタンを押します

DEXCS

形状作成

メッシュ1

メッシュ2

境界条件

流体特性

計算実行

結果処理

PyFoamRunner.simpleFoam.logfile

実行

foamCalc

mag U -latestTime

実行

patchIntegrate

magU outlet -latestTime

実行

sample

sampleDict編集

gnuplot

*.gp編集

paraFoam

解析実行できます

reached convergence criterion: 0.001
latestTime = 100
End

現在の解析フォルダ: /home/dexcs/Desktop/Test

ファイル ツール ヘルプ

DEXCS

形状作成

メッシュ1

メッシュ2

境界条件

流体特性

計算実行

結果処理

foamLog

PyFoamRunner.simpleFoam.logfile

実行

foamCalc

mag U -latestTime

実行

patchIntegrate

magU outlet

実行

sample

sampleDict編集

gnuplot

*.gp編集

実行

RunFoamLog

foamLog を実行します

OK Cancel / End

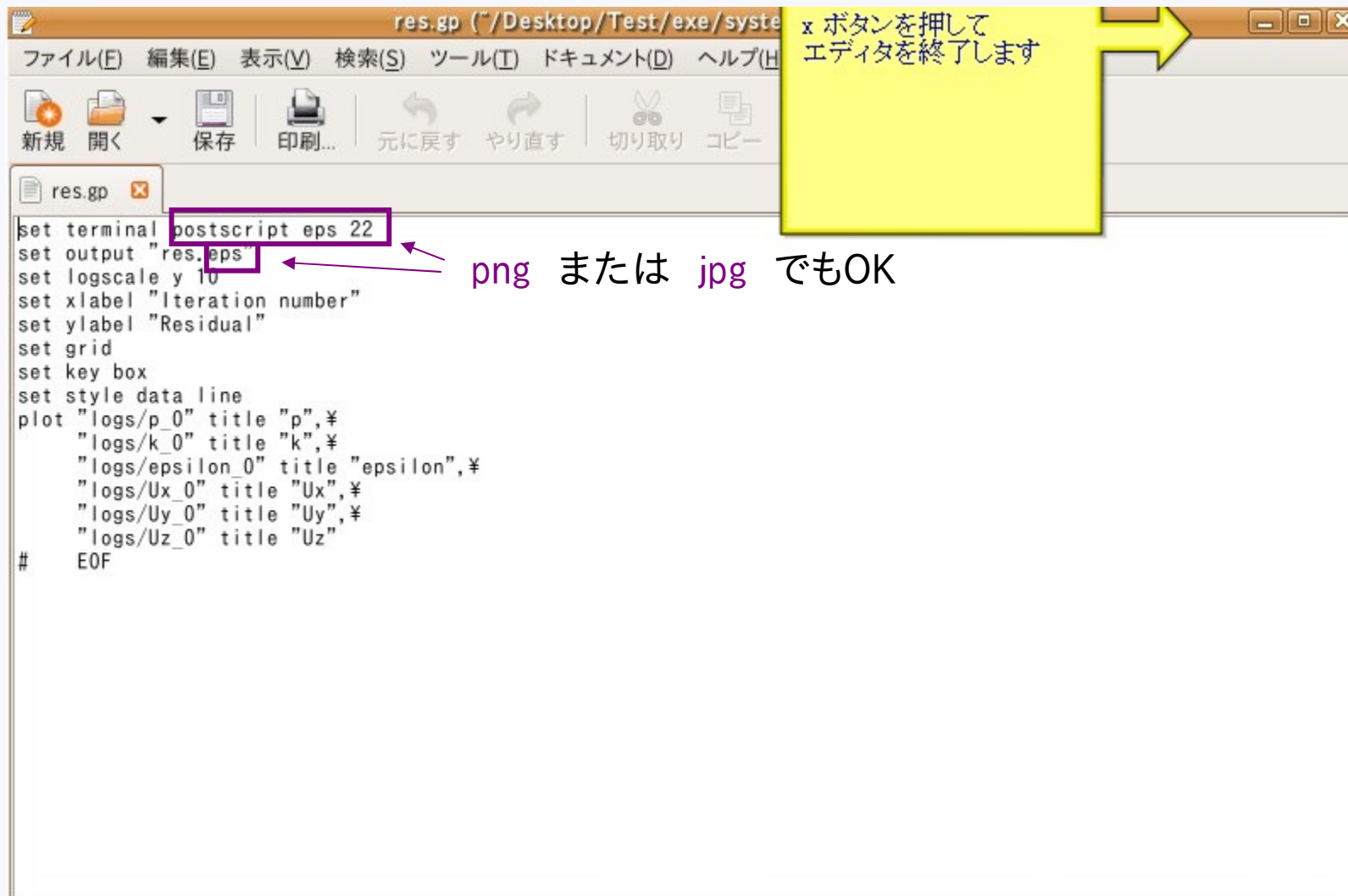
stdout/stderr

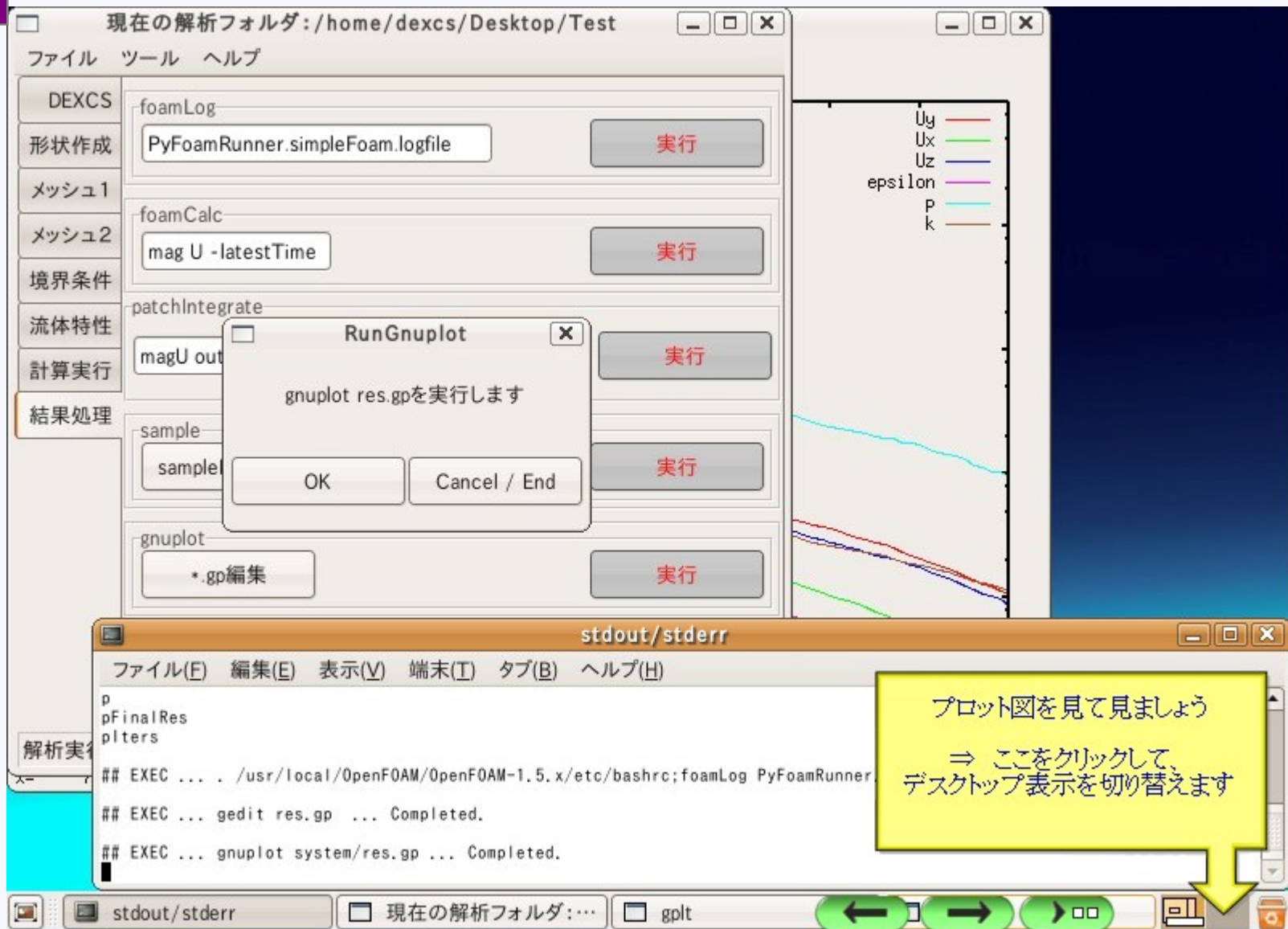
ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)

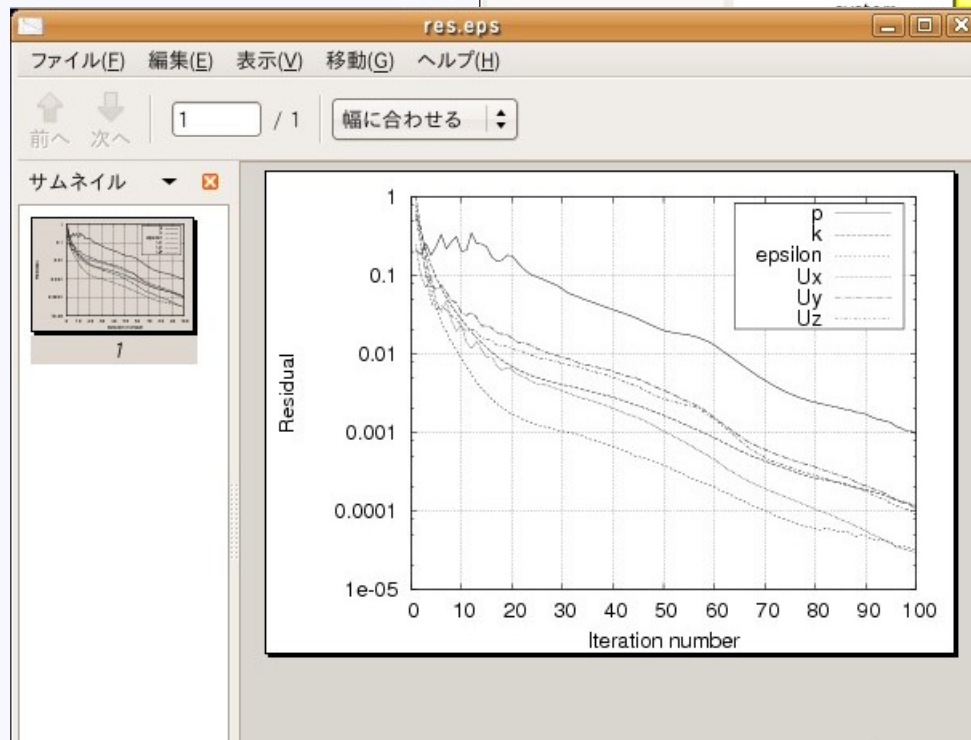
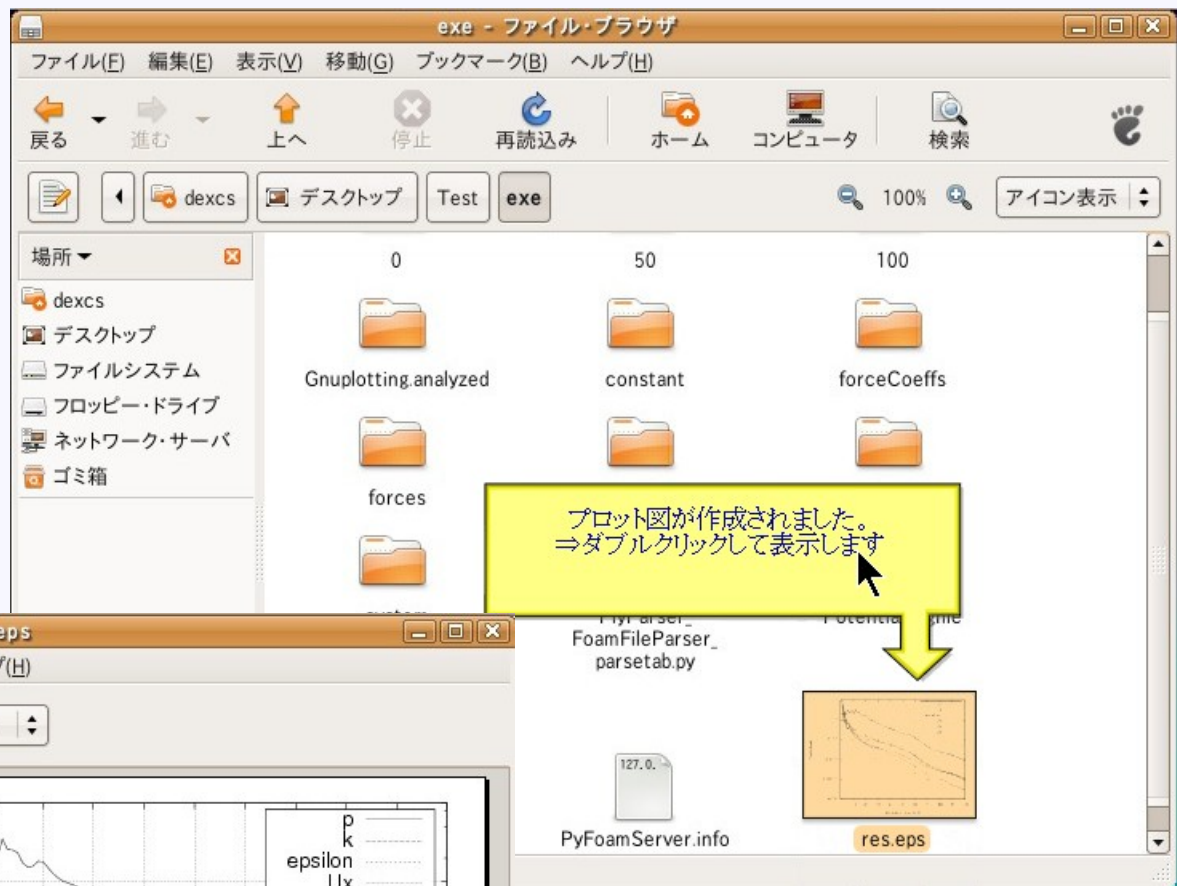
executionTime
k
kFinalRes
klters
p
pFinalRes
plters
EXEC ... /usr/local/OpenFOAM/OpenFOAM-1.5.x/etc/bashrc; foamLog PyFoamRunner.simpleFoam.logfile ... Completed.

foamLogが終了しました

Uy
Ux
Uz
epsilon
p
k







形状作成

PyFoamRunner.simpleFoam.logfile

実行

メッシュ1

メッシュ2

foamCalc

mag U -latestTime

実行

実行ボタンを押します

境界条件

流体特性

patchIntegrate

magU outlet -latestTime

実行

計算実行

結果処理

sample

sampleD

次に、
foamCalc
↓
patchIntegrate
を使って、流量をチェックしてみましょ

gnuplot

*.gp

paraFoam

解析実行できます

```
## EXEC ... gedit res.gp ... Completed.
## EXEC ... gnuplot system/res.gp ... Completed.
```

現在の解析フォルダ: /home/dexcs/Desktop/Test

ファイル ツール ヘルプ

DEXCS

形状作成

メッシュ1

メッシュ2

境界条件

流体特性

計算実行

結果処理

foamLog

PyFoamRunner.simpleFoam.logfile

実行

foamCalc

mag U -latestTime

実行

patchIntegrate

magU out

実行

sample

sampleD

実行

gnuplot

*.gp編集

RunFoamCalc

foamCalc を実行します

OK

Cancel / End

foamCalcが終了しました。

(各速度のベクトル成分から
速度の大きさ(magU)を出力しました)

解析実行

Create mesh for time = 100

Time = 100

Reading U

Calculating magU

EXEC ... /usr/local/OpenFOAM/OpenFOAM-1.5.x/etc/bashrc;foamCalc mag U -latestTime ... Completed.

現在の解析フォルダ: /home/dexcs/Desktop/Test

次に、patchIntegrateを実行します。

とりあえずパラメータはデフォルト
magU outlet -latestTime
のまま実行します。

RunPatchIntegrate
patchIntegrate を実行します

magU(速度)をoutlet(流出面)
にて積分した結果が、ここに出力されています。
⇒160.021 となっています。

```

ファイル(F) 編集(E) 表示(V) 端末(T) タブ(B) ヘルプ(H)
Area magnitude of patch outlet[1] = 16
Reading volScalarField magU
Integral of magU over vector area of patch outlet[1] = (160.021 1.19863e-07 -4.2084e-05)
Integral of magU over area magnitude of patch outlet[1] = 160.021
End

## EXEC ... /usr/local/OpenFOAM/OpenFOAM-1.5.x/etc/bashrc;patchIntegrate magU outlet -latestTime ... Completed.

```


現在の解析フォルダ: /home/dexcs/Desktop/Test

ファイル ツール ヘルプ

DEXCS

形状作成 PyFoamRunner.simpleFoam.logfile 実行

メッシュ1

メッシュ2

境界条件

流体特性

計算実行 magU inlet 実行

結果処理

sampleDict編集

gnuplot *gp編集

paraFoam

解析実行できます

End

EXEC ... /usr/local/OpenFOAM/OpenFOAM-1.5.x/etc/bashrc; patchIntegrate magU inlet -latestTime ... Completed.

outlet を inlet に変更しました

次にパラメータを変えた計算です

Uy
Ux
Uz
epsilon
p
k

現在の解析フォルダ: /home/dexcs/Desktop/Test

ファイル ツール ヘルプ

DEXCS

形状作成 PyFoamRunner.simpleFoam.logfile 実行

メッシュ1

メッシュ2

境界条件

流体特性

計算実行 magU inlet 実行

結果処理

sampleDict編集

gnuplot *gp編集

paraFoam

解析実行できます

End

EXEC ... /usr/local/OpenFOAM/OpenFOAM-1.5.x/etc/bashrc; patchIntegrate magU inlet -latestTime ... Completed.

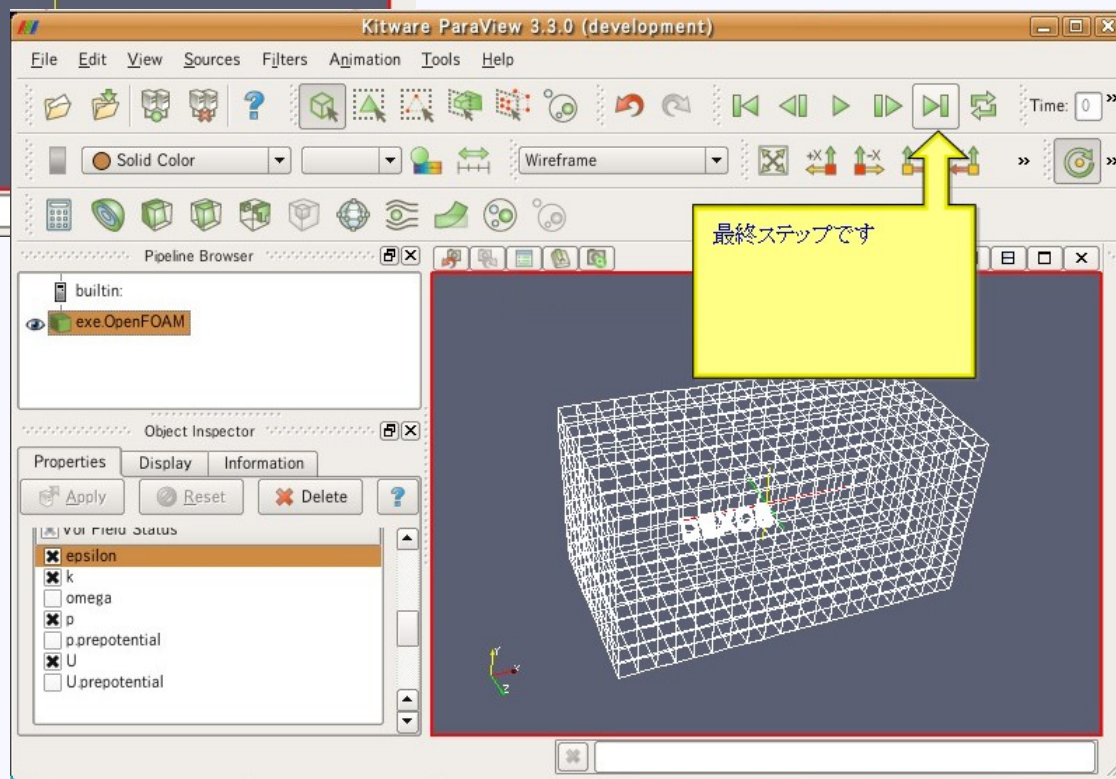
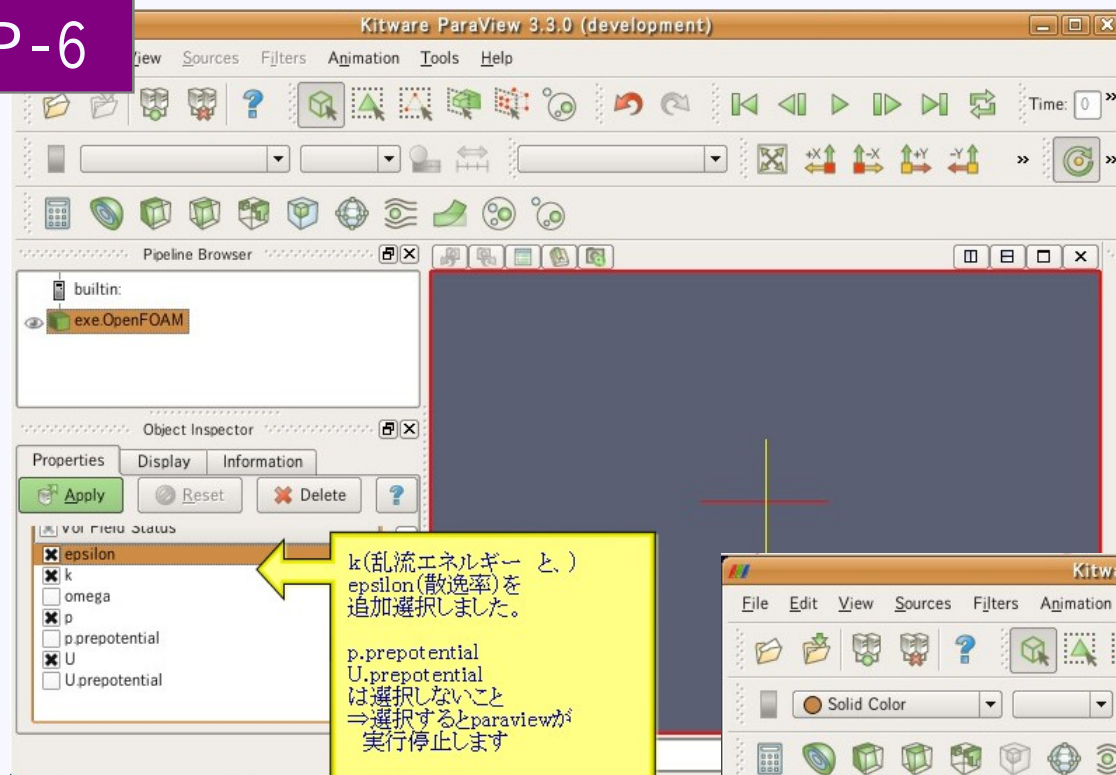
RunPatchIntegrate

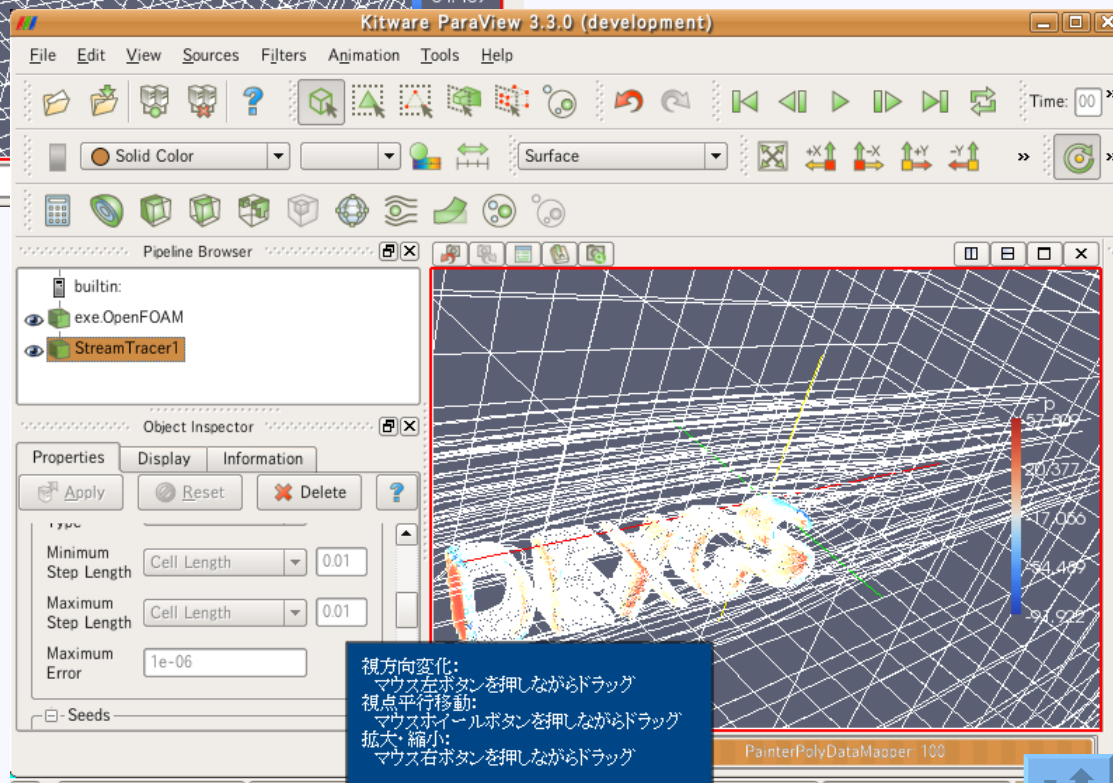
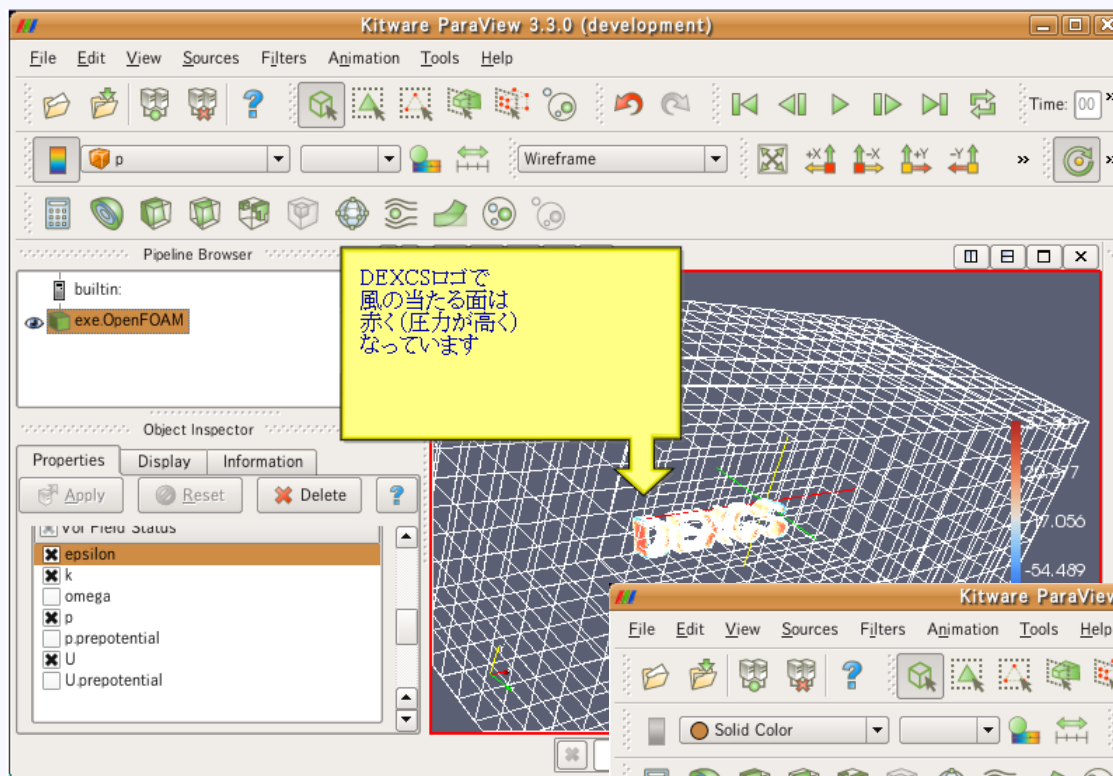
patchIntegrate を実行します

結果がここに出ています。
⇒160 となっています。
outletは160.021だったので
ほとんど同じ値になっているという事、当然ですね。。

Area magnitude of patch inlet[0] = 16
Reading volScalarField magU
Integral of magU over vector area of patch inlet[0] = (-160 3.6e-05 3.6e-05)
Integral of magU over area magnitude of patch inlet[0] = 160



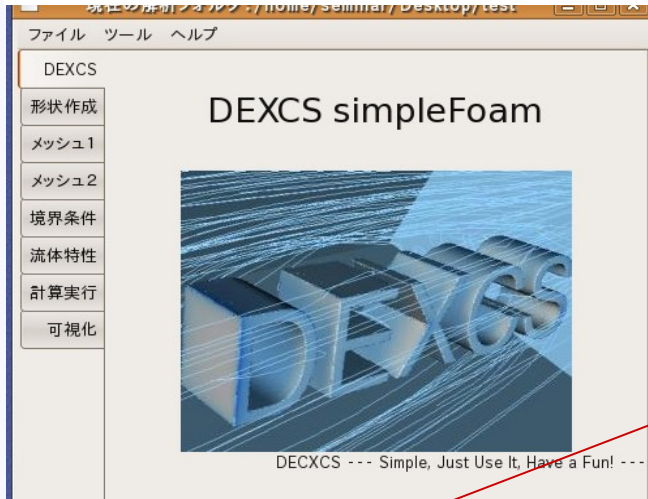




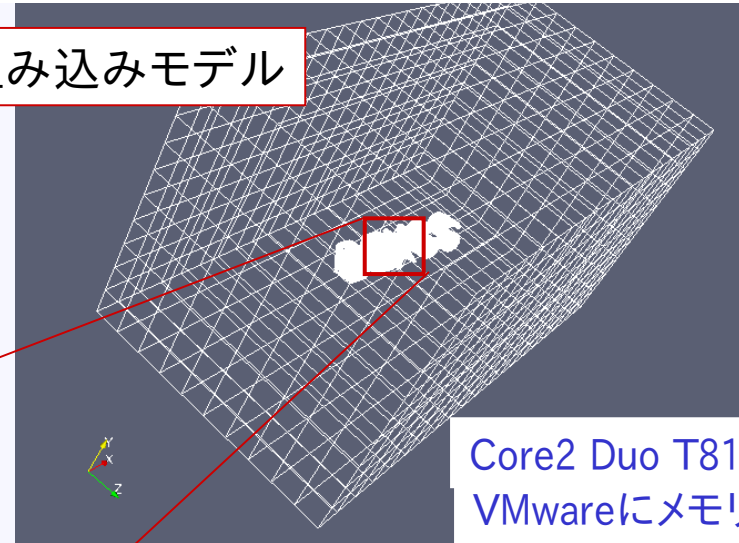
標準モデルを使ったパラメタスタディ

メッシュサイズ変更例

■ 第二回オープンソースCAEワークショップ発表資料「DEXCS-OpenFOAMの紹介」(PDF)

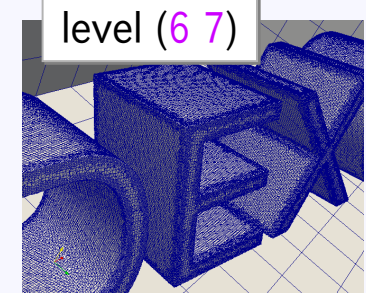
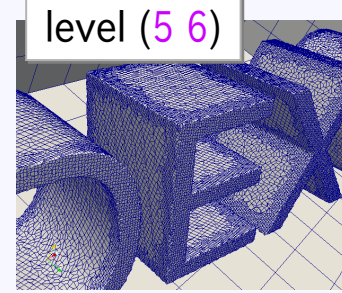
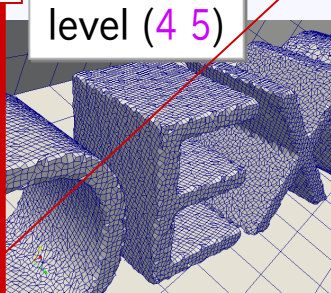
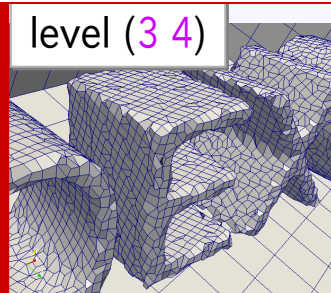


標準組み込みモデル



Core2 Duo T8100(2.1GHz)
VMwareにメモリー2.6G占有

標準組み込みの細分化パラメタ



メッシュ	要素数	137,456	229,815	426,552	1,085,734
	節点数	153,783	278,594	545,530	1,442,646
	作成時間 (sec)	77	152	332	1,064
simple Foam	収束回数	100	145	200	800
	計算時間 (sec)	457	796	2,626	32,167

パラメタスタディ例 サマリ

case	メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数	
	blockMesh	ref_Surfaces			divSchemes	relaxation	動粘性係数	流速
1	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05	10
2	20x10x10	(4,5)	kEpsilon	実行	upwind	0.7	1.54E-05	10
3	40x20x20	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05	10
4	20x10x10	(3,4)	k- ω SST	-	upwind	0.7	1.54E-05	10
5	20x10x10	(3,4)	k- ω SST	実行	upwind	0.3	1.54E-05	10
6	20x10x10	(3,4)	k- ω SST	実行	lim_linear 0.2	0.3	1.54E-05	10
7	20x10x10	(3,4)	k- ω SST	実行	upwind	0.7	1.54E-05	10
8	20x10x10	(3,4)	k- ω SST	-	SFCD	0.3	1.54E-05	10
9	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-06	1
10	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-03	10
11	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-07	10

パラメタ

デフォルト値

変更値

case	節点数	要素数	収束回数	計算時間 sec	圧力		流速 max	乱流(k) max	Cd	Cl
					max	min				
1	153,783	137,456	100	114	49.976	-48.044	11.111	3.5105	0.01515	0.00212
2	278,594	229,815	145	926	50.849	-69.142	11.485	2.8292	0.01597	0.00234
3	955,519	1,022,764	259	3,283	50.911	-70.391	11.582	3.1313	0.01568	0.0022
4	153,783	137,456	447	618	50.077	-49.043	11.075	3.1981	0.01526	0.00204
5	153,783	137,456	275	339	50.081	-49.039	11.064	3.2003	0.01527	0.00199
6	153,783	137,456	1,000	1,144	47.83	-46.23	12.216	5.8232	0.01498	0.00259
7	153,783	137,456	94	119	50.069	-49.056	11.075	3.1986	0.01527	0.00204
8	153,783	137,456	1,000	1,359	46.586	-48.021	11.553	4.3069	0.01469	0.00256
9	153,783	137,456	103	278	0.49976	-0.48047	1.1112	0.0035	0.00015	2.11E-05
10	153,783	137,456	87	293	50.499	-60.176	11.003	1.4597	0.01476	0.00127
11	153,783	137,456	108	317	50.017	-47.937	11.109	3.6648	0.01523	0.00214

評価項目

パラメタスタディ要領(1)

メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数
blockMesh	ref_Surfaces	乱流モデル	初期流れ場	divSchemes	relaxation	動粘性係数
20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05
						流速
						10

```
blockMeshDict
23
24 blocks
25 (
26     hex (6 5 1 2 7 4 0 3) (20 10 10) simpleGrading (1 1 1)
27
28 );
```

```
snappyHexMeshDict
105     refinementSurfaces
106     {
107         dexcs
108         {
109             // Surface wise min and
110             level (3 4);
111         }
112     }
```

現在: /home/et/Desktop/test5

ファイル ツール

DEXCS blockMesh

形状作成 blockMeshDict編集 初期化(やり直し) 実行

メッシュ1 importMesh

メッシュ2 実行

境界条件 autoPatch

流体特性 二面挟角(deg) 60 実行

計算実行 createPatch

結果処理 patch確認 createPatchDict編集 実行

メッシュ確定-> メッシュ2 (snappyHexMeshing) 実行

メッシュ確定-> exe

現在の解 /home/et/Desktop/test5

ファイル ツール

DEXCS snappyHexM

形状作成 形状選択 初期化(やり直し)

メッシュ1

メッシュ2

境界条件

流体特性 snappyHexMeshDict編集 停止 実行

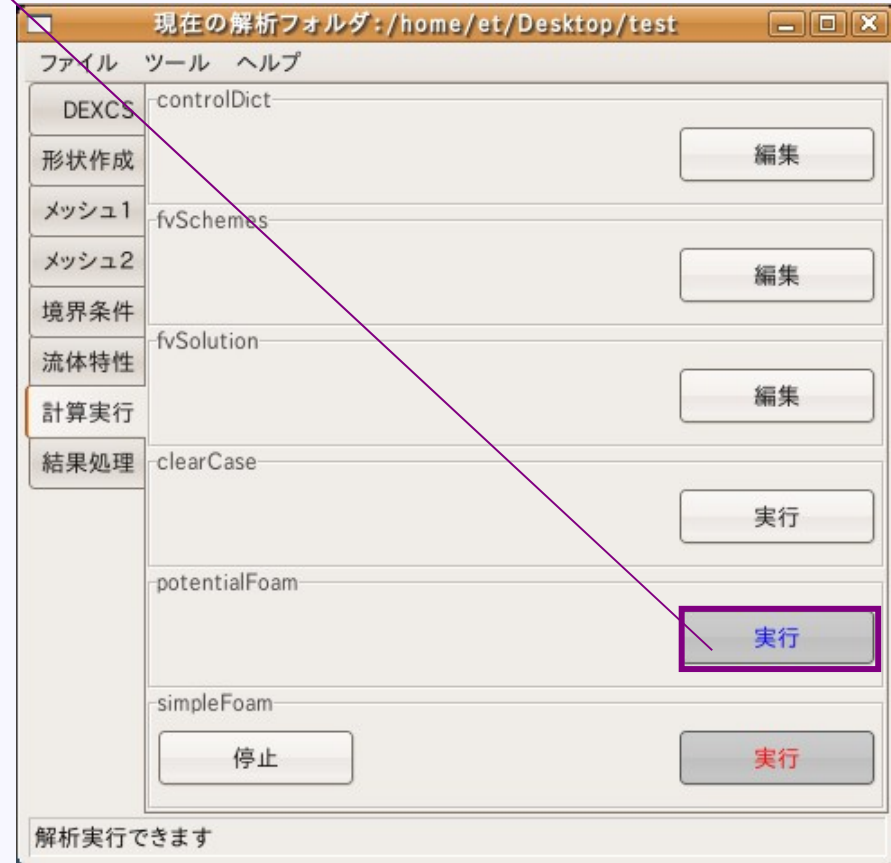
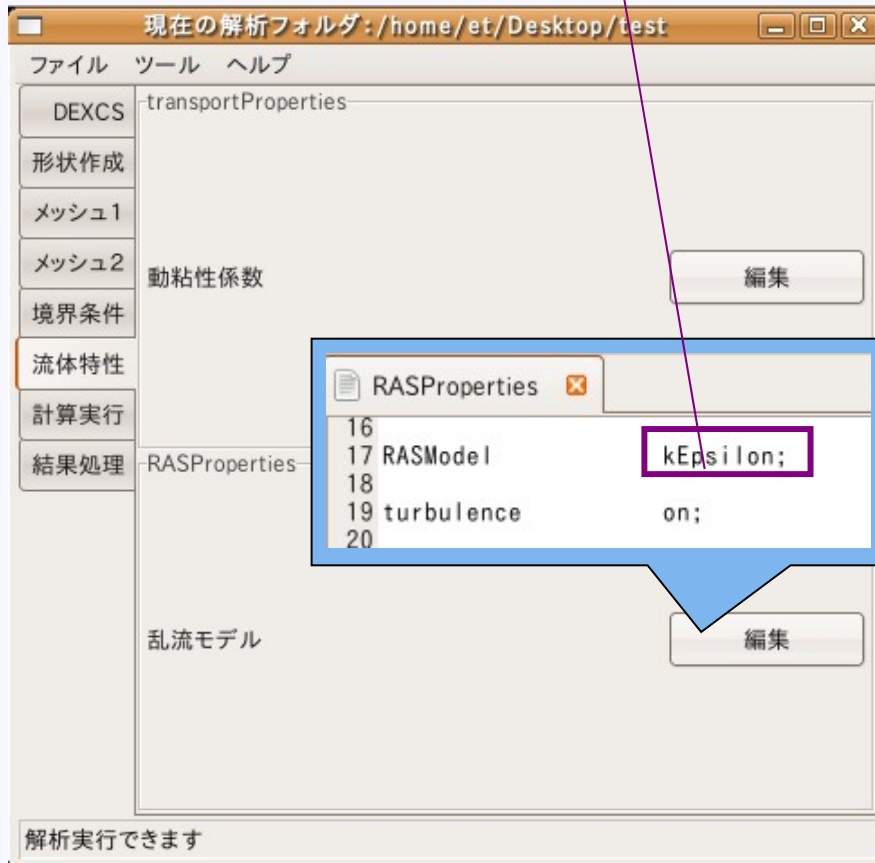
計算実行

結果処理 メッシュ確認 実行

メッシュ確定-> exe

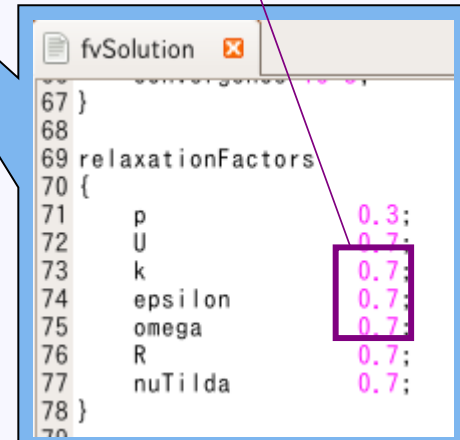
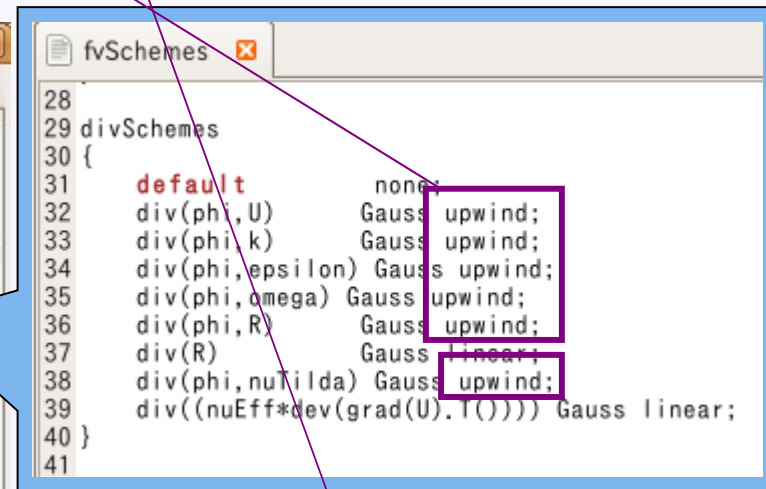
パラメタスタディ要領(2)

メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数	
blockMesh	ref_Surfaces	kEpsilon	実行	divSchemes	relaxation	動粘性係数	流速
20x10x10	(3,4)			upwind	0.7	1.54E-05	10



パラメタスタディ要領(3)

メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数
blockMesh	ref_Surfaces			divSchemes	relaxation	動粘性係数
20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05
						流速
						10



パラメタスタディ要領(4)

メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数
blockMesh	ref_Surfaces			divSchemes	relaxation	動粘性係数
20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05
						流速
						10

transportProperties

```

16
17 transportModel Newtonian;
18
19 nu
20   nu [0 2 -1 0 0 0] 1.54e-05;

```

現在0

ファイル ツール

DEXCS

形状作成

メッシュ1

メッシュ2

境界条件

流体特性

計算実行

結果処理

動粘性係数

編集

RASProperties

乱流モデル

編集

解析実行できます

U

```

19 }
20 inlet
21 {
22   type
23   value
24 }
25

```

現在0

ファイル ツール

DEXCS

形状作成

メッシュ1

メッシュ2

境界条件

流体特性

計算実行

結果処理

初期化

patch名取得

編集

初期化

patch名取得

編集

初期化

patch名取得

編集

一括処理

初期化

patch名取得

編集

解析実行できます

評価項目について(1)

節点数	要素数	収束回数	計算時間	圧力		流速	乱流(k)	Cd	Cl
				max	min	max	max		
153783	137456	100	114	49.976	-48.044	11.111	3.5105	0.0151484	0.0021172

The screenshot displays the PyFoamRunner software interface. On the left, the 'neighbour' window shows the 'FoamFile' configuration with the following values: version 2.0, format ascii, class labelList, note 'nPoints: 153783 nCells: 137456 nFaces: 427435', location '3/polyMesh', and object neighbour. The 'PyFoamRunner.simpleFoam.logfile' window shows the execution log, including the time step continuity errors, solving for epsilon and k, and the execution time of 113.78 seconds. The 'constant' directory is expanded, showing the 'polyMesh' subdirectory, which contains the 'sets' subdirectory and the 'neighbour' file. The 'neighbour' file is highlighted in orange. The 'PyFoamRunner.simpleFoam.logfile' file is also highlighted in orange in the file explorer on the right.

neighbour

```

5 | ¥¥ / And
6 | ¥¥ / Manipulation
7 | **-----
8 | FoamFile
9 | {
10 |     version      2.0;
11 |     format        ascii;
12 |     class         labelList;
13 |     note          "nPoints: 153783 nCells: 137456 nFaces: 427435";
14 |     location      "3/polyMesh";
15 |     object        neighbour;

```

PyFoamRunner.simpleFoam.logfile

```

1135 time step continuity errors : sum local = 2.30182e-06,
1136 DILUPBiCG: Solving for epsilon, Initial residual = 3.
1137 DILUPBiCG: Solving for k, Initial residual = 0.000115
1138 ExecutionTime = 113.78 s ClockTime = 384 s
1139
1140 reached convergence criterion: 0.001
1141 latestTime = 100
1142 End
1143

```

constant

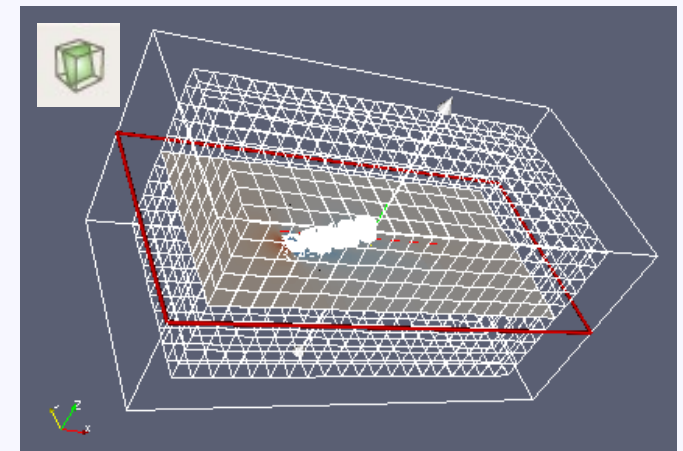
- polyMesh
 - sets
 - boundary
 - cellLevel
 - cellZones
 - faceZones
 - faces
 - neighbour
 - owner
 - pointLevel

exe

- 0
- 50
- 100
- Gnuplotting.analyzed
- constant
- forceCoeffs
- forces
- probes1
- system
- PlyParser_FoamFileParser_parsetab.py
- Potential.logfile
- PyFoamHistory
- PyFoamRunner.simpleFoam.logfile
- PyFoamServer.info

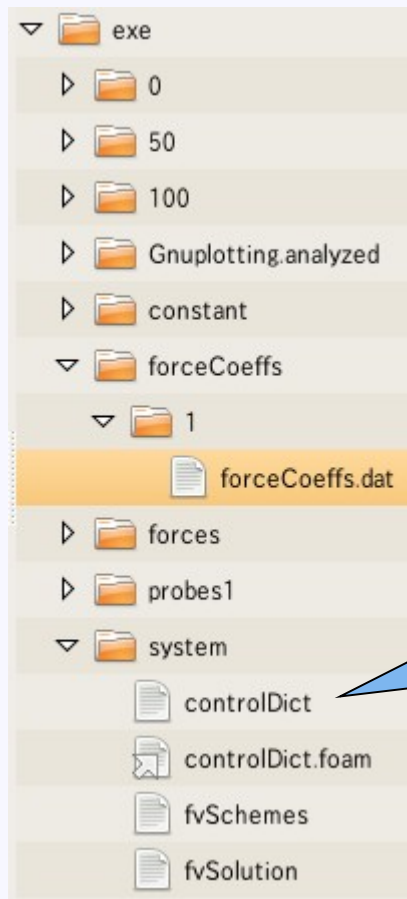
評価項目について(2)

節点数	要素数	収束回数	計算時間	圧力		流速	乱流(k)	Cd	Cl
				max	min	max	max		
153783	137456	100	114	49.976	-48.044	11.111	3.5105	0.0151484	0.0021172



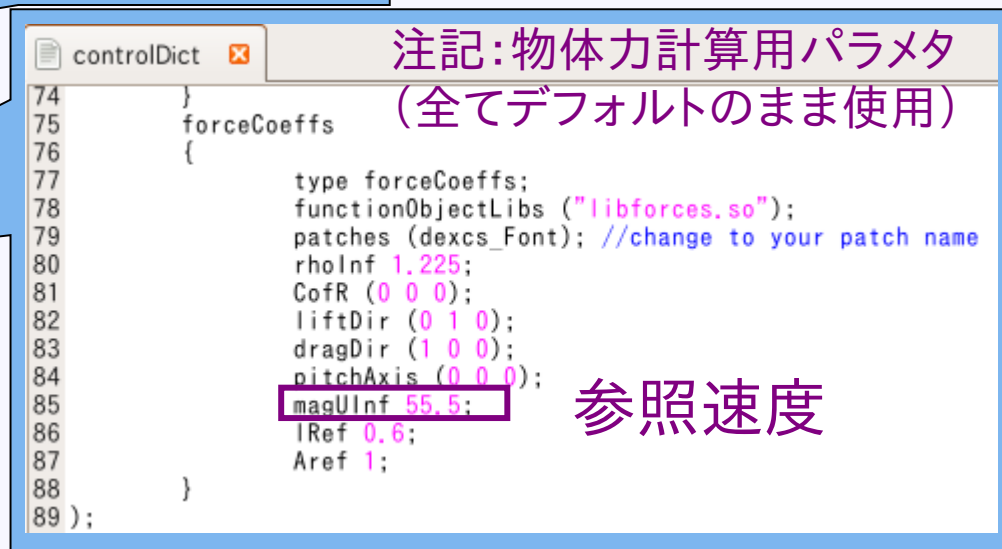
評価項目について(3)

節点数	要素数	収束回数	計算時間	圧力		流速	乱流(k)	Cd	Cl
				max	min	max	max		
153783	137456	100	114	49.976	-48.044	11.111	3.5105	0.0151484	0.0021172



A screenshot of the 'forceCoeffs.dat' file. The file contains a table of data for iterations 90 to 100. The values for iteration 100 are highlighted with red boxes: 0.0151484 for Cd and 0.0021172 for Cl.

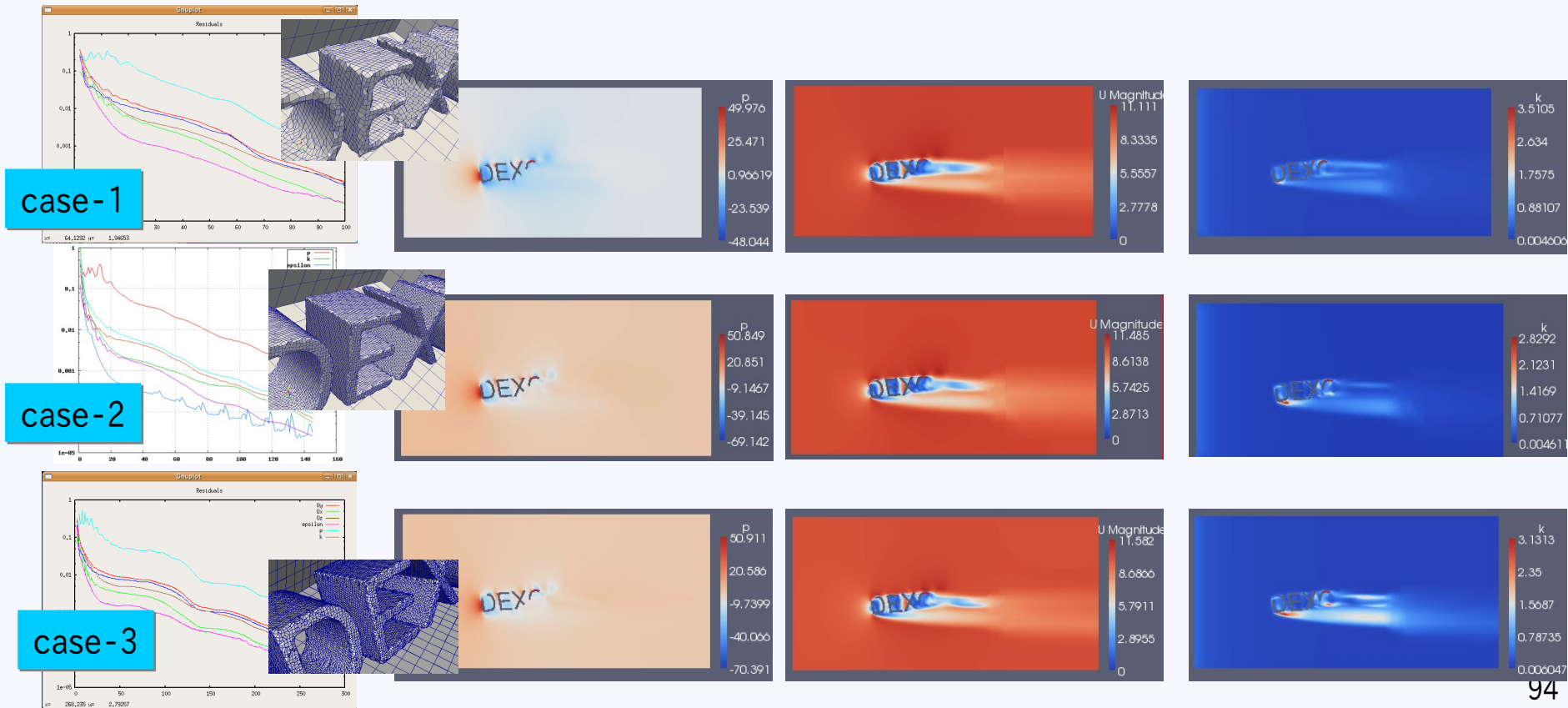
Iteration	Cd	Cl
90	0.0151531	0.00210983
91	0.015152	0.00211083
92	0.0151513	0.0021113
93	0.0151504	0.00211187
94	0.0151507	0.00211157
95	0.0151498	0.00211317
96	0.0151501	0.00211402
97	0.015149	0.0021157
98	0.0151487	0.00211679
99	0.0151478	0.00211689
100	0.0151484	0.0021172



メッシュサイズの影響

case	メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数	
	blockMesh	ref_Surfaces			divSchemes	relaxation	動粘性係数	流速
1	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05	10
2	20x10x10	(4,5)	kEpsilon	実行	upwind	0.7	1.54E-05	10
3	40x20x20	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05	10

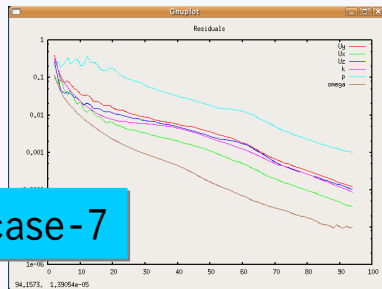
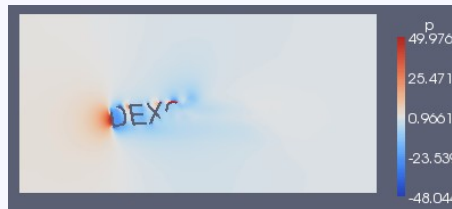
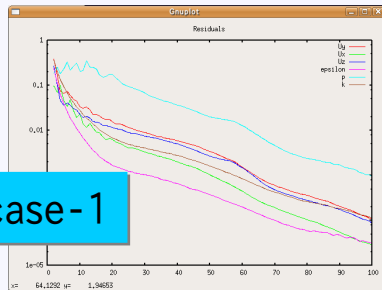
case	節点数	要素数	収束回数	計算時間 sec	圧力		流速 max	乱流(k) max	Cd	Cl
					max	min				
1	153,783	137,456	100	114	49.976	-48.044	11.111	3.5105	0.01515	0.00212
2	278,594	229,815	145	926	50.849	-69.142	11.485	2.8292	0.01597	0.00234
3	955,519	1,022,764	259	3,283	50.911	-70.391	11.582	3.1313	0.01568	0.0022



乱流モデルの影響

case	メッシュ1 blockMesh	メッシュ2 ref_Surfaces	乱流モデル	初期流れ場	計算スキーム divSchemes	計算解法 relaxation	レイノルズ数 動粘性係数	流速
1	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05	10
7	20x10x10	(3,4)	k- ω SST	実行	upwind	0.7	1.54E-05	10

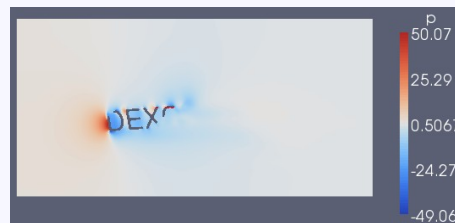
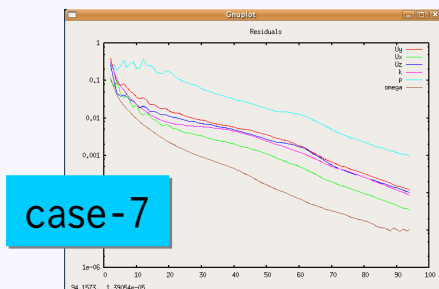
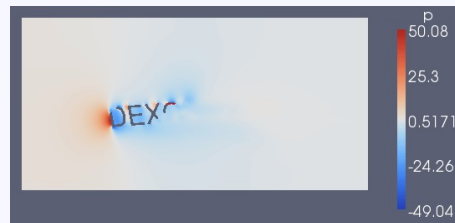
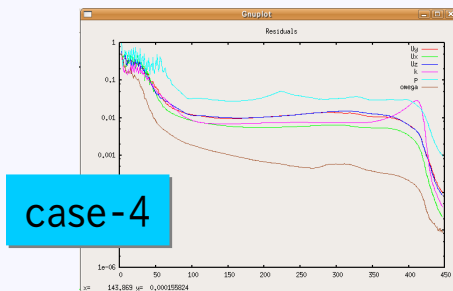
case	節点数	要素数	収束回数	計算時間 sec	圧力		流速	乱流(k)	Cd	Cl
					max	min	max	max		
1	153,783	137,456	100	114	49.976	-48.044	11.111	3.5105	0.01515	0.00212
7	153,783	137,456	94	119	50.069	-49.056	11.075	3.1986	0.01527	0.00204



初期流れ場の影響

case	メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数	
	blockMesh	ref_Surfaces			divSchemes	relaxzation	動粘性係数	流速
4	20x10x10	(3,4)	k- ω SST	-	upwind	0.7	1.54E-05	10
7	20x10x10	(3,4)	k- ω SST	実行	upwind	0.7	1.54E-05	10

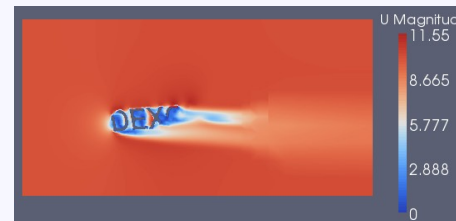
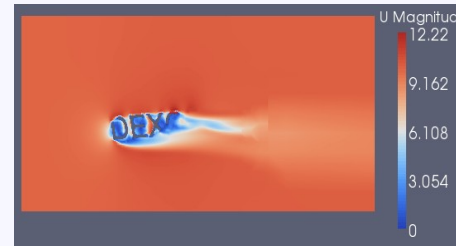
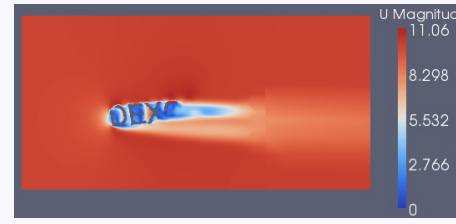
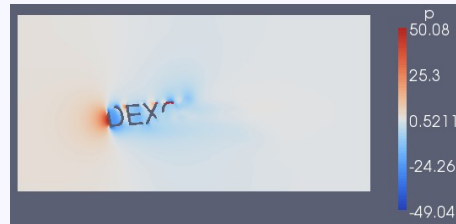
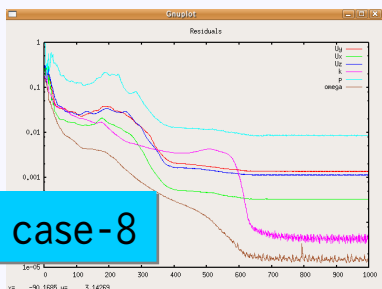
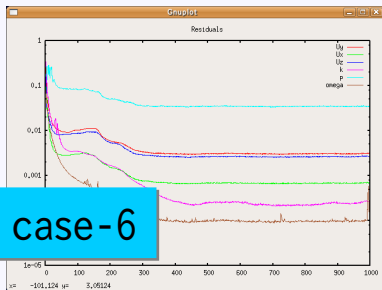
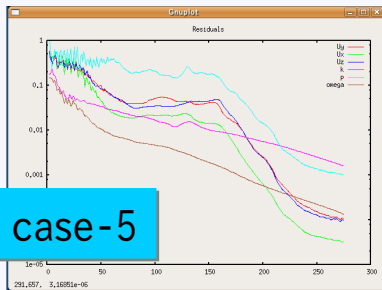
case	節点数	要素数	収束回数	計算時間 sec	圧力		流速 max	乱流(k) max	Cd	Cl
					max	min				
4	153,783	137,456	447	618	50.077	-49.043	11.075	3.1981	0.01526	0.00204
7	153,783	137,456	94	119	50.069	-49.056	11.075	3.1986	0.01527	0.00204



計算スキームの影響

case	メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数	
	blockMesh	ref_Surfaces			divSchemes	relaxation	動粘性係数	流速
5	20x10x10	(3,4)	k- ω SST	実行	upwind	0.3	1.54E-05	10
6	20x10x10	(3,4)	k- ω SST	実行	lim_linear 0.2	0.3	1.54E-05	10
8	20x10x10	(3,4)	k- ω SST	-	SFCD	0.3	1.54E-05	10

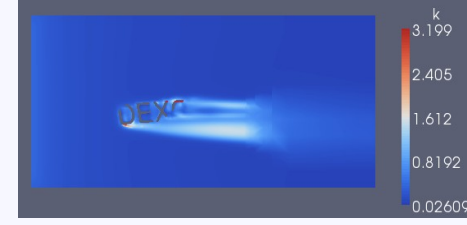
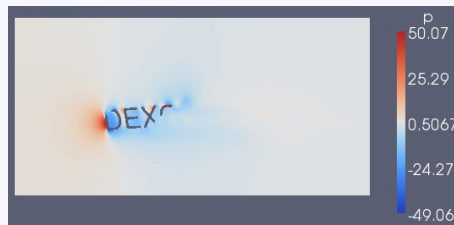
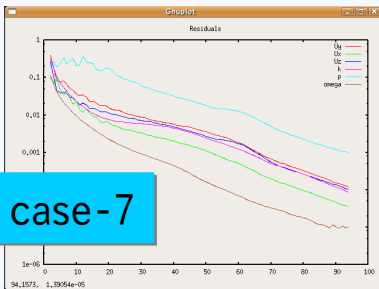
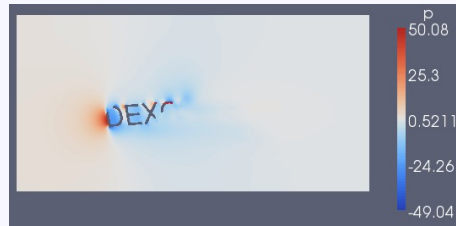
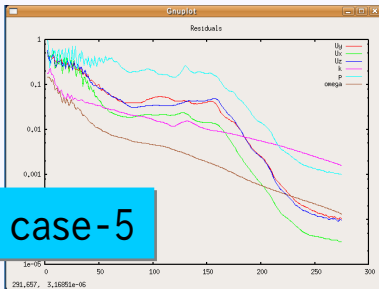
case	節点数	要素数	収束回数	計算時間 sec	圧力		流速	乱流(k)	Cd	Cl
					max	min	max	max		
5	153,783	137,456	275	339	50.081	-49.039	11.064	3.2003	0.01527	0.00199
6	153,783	137,456	1,000	1,144	47.83	-46.23	12.216	5.8232	0.01498	0.00259
8	153,783	137,456	1,000	1,359	46.586	-48.021	11.553	4.3069	0.01469	0.00256



解法(緩和係数)の影響

case	メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数	
	blockMesh	ref_Surfaces			divSchemes	relaxation	動粘性係数	流速
5	20x10x10	(3,4)	k- ω SST	実行	upwind	0.3	1.54E-05	10
7	20x10x10	(3,4)	k- ω SST	実行	upwind	0.7	1.54E-05	10

case	節点数	要素数	収束回数	計算時間 sec	圧力		流速	乱流(k)	Cd	Cl
					max	min	max	max		
5	153,783	137,456	275	339	50.081	-49.039	11.064	3.2003	0.01527	0.00199
7	153,783	137,456	94	119	50.069	-49.056	11.075	3.1986	0.01527	0.00204



レイノルズ数(Re)

$$Re = \frac{U \cdot D}{\nu}$$

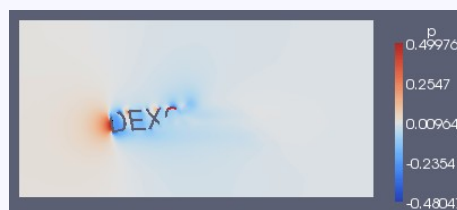
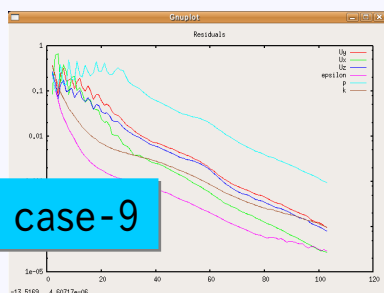
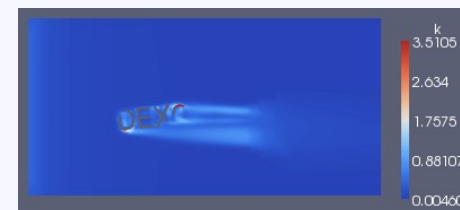
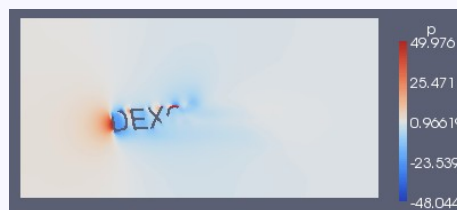
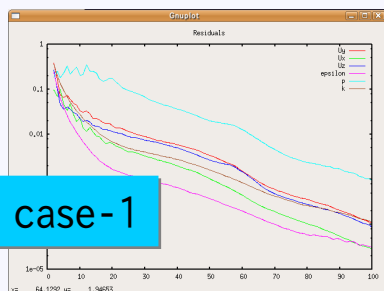
U:流速

D:代表寸法

ν :動粘性係数

case	メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数	
	blockMesh	ref_Surfaces			divSchemes	relaxation	動粘性係数	流速
1	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-05	10
9	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-06	1

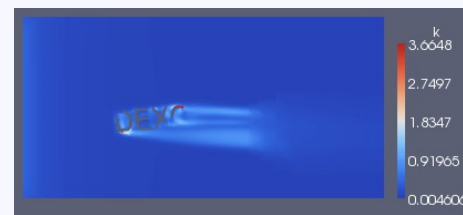
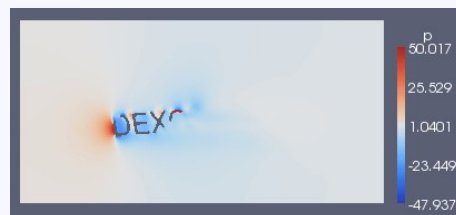
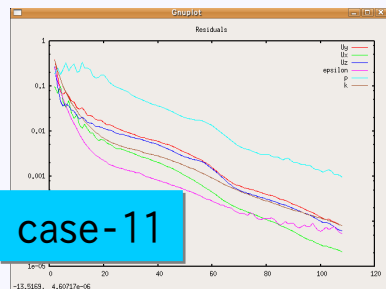
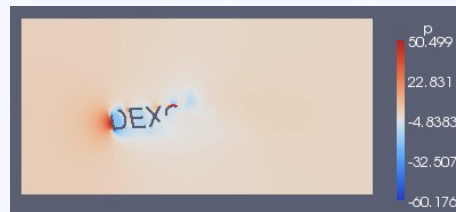
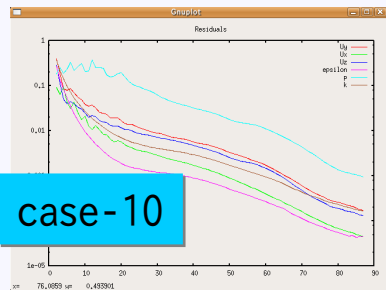
case	節点数	要素数	収束回数	計算時間 sec	圧力		流速 max	乱流(k) max	Cd	Cl
					max	min				
1	153,783	137,456	100	114	49.976	-48.044	11.111	3.5105	0.01515	0.00212
9	153,783	137,456	103	278	0.49976	-0.48047	1.1112	0.0035	0.00015	2.11E-05



レイノルズ数(動粘性係数)

case	メッシュ1	メッシュ2	乱流モデル	初期流れ場	計算スキーム	計算解法	レイノルズ数	
	blockMesh	ref_Surfaces			divSchemes	relaxation	動粘性係数	流速
10	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-03	10
11	20x10x10	(3,4)	kEpsilon	実行	upwind	0.7	1.54E-07	10

case	節点数	要素数	収束回数	計算時間 sec	圧力		流速	乱流(k)	Cd	Cl
					max	min	max	max		
10	153,783	137,456	87	293	50.499	-60.176	11.003	1.4597	0.01476	0.00127
11	153,783	137,456	108	317	50.017	-47.937	11.109	3.6648	0.01523	0.00214



おわり