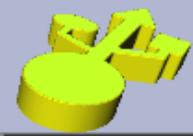


Open ∇ FOAM®初級体験コース

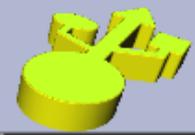
オープンCAEコンサルタント OCSE²
代表 野村 悦治



OpenFOAM®初級体験コースの狙い

OpenFOAMを含めてオープンCAEはコストフリーで導入できますが、活用するには使用者の自助努力が欠かせません。本コースを受講することにより、

- OpenFOAMの機能の概要が理解できる
- 実用レベルの解析が簡単にできることを体験できる
- より高度な活用に向けての勉強方法、情報ソースがわかることで、自助努力をやりてみよう!・・・という気持ちになってもらうのが狙いです。



OpenFOAM®初級体験コース 必要環境

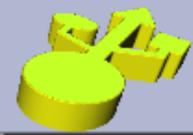
【実習に必要なWindowsマシン】

CPUはPentium4以上、メモリー2GB以上

(メモリー1GBでも動作しますがスペース不足で十分な実習が出来ません)

DVDドライブがあり、ブート順序がDVD優先になっていること。

(既存のOS、HDDは使用しません)



OpenFOAM®初級体験コース カリキュラム(予定)

1. OpenFOAMの概要(20分)

歴史、機能

2. DEXCS for OpenFOAM の体験1(30分)

ランチャーを使った仮想風洞試験の演習体験

3. DEXCS for OpenFOAMの概要(30分)

DEXCSの狙い、DEXCSとOpenFOAMの関係、DEXCS for OpenFOAMの仕組み

4. OpenFOAMの基礎知識概要(30分)

計算手順、メッシュ、パラメタファイルの構成、チュートリアルケースの仕組み

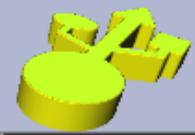
5. DEXCS for OpenFOAM の体験2(1.5時間)

ランチャー演習の振り返り

dexcsSWAKを使ったチュートリアルケースの演習

6. 今後の活用に向けて(30分)

DEXCS for OpenFOAMの実践的な活用法と、OpenFOAM 関連の情報ソース紹介



OpenFOAM®の概要

オープンCAE学会 平成22年度OpenFOAM®非圧縮性流体解析演習シリーズ 第1回
「Bashシェルの基礎とOpenFOAM概要および主要な非圧縮性ソルバー・チュートリアルの説明」
より 抜粋、改変(最新情報で更新)

OpenFOAM®の歴史

- 1989年—2000年:研究室のハウスコード
 - 開発元:インペリアルカレッジ・ゴスマン研
 - (Star-CDの開発元)のHenry、Hrvoje、Niklas
- 1999年—2004年:商用コード化(FOAM: Field Operation And Manipulation)、開発元: ∇ Nabla
- 2004年12月:オープンソース化(OpenFOAM)、
 - 開発元: OpenCFD
- 2011年8月: SGIがOpenCFDを買収



OpenFOAM®の特徴

- 有限体積法の採用
- 3次元非構造格子に対応
- ポリヘドラル(任意多角形)格子に対応
- 領域分割型の並列計算が容易
- 化学反応、燃焼を含めた複雑な流れ場、乱流、熱伝達、固体力学、電磁場解析などを解析可能

OpenFOAM®の特徴

- オブジェクト指向型言語C++言語で記述されたCFD、固体の応力解析等の連続体力学の分野で使用可能な汎用の数値計算クラス・ライブラリ
- オブジェクト指向型言語C++の特徴が活かされており、支配方程式の解法の実装が非常に簡潔

OpenFOAM®の特徴

非圧縮性のNavier-Stokesの式

$$\frac{\partial U}{\partial t} + \nabla \cdot \phi U - \nabla \cdot \nu \nabla U = -\nabla p$$

陽解法で解くコード

```
solve(  
    fvm::ddt(U) + fvm::div(phi,U) - fvm::laplacian(nu,U) == -fvc::grad(p)  
);
```

OpenFOAM®の特徴

■ フリーソフトウェア

- 学生や研究者がCFD解析をトライするきっかけ
- 超並列計算が低コストで可能

■ オープンソース

- 既存のソースを参照可能
- 新しい計算モデルや離散スキームの実装が容易

■ ライセンス :GPL(General Public License)

OpenFOAM®の構成

形状作成

メッシュ作成

計算

後処理

blockMesh

snappyHexMesh

blender

Adventure-tetmesh

Google SketchUp

Netgen

gmesh

enGrid

CAD tools

STAR-CD(ProSTAR)

others

OpenFOAM

- ・ ソルバ
- ・ 乱流モデル
- ・ 境界条件
- ・ 物性
- ・ メッシュ

Utilities

ParaView

VisIt

enSight

FieldView

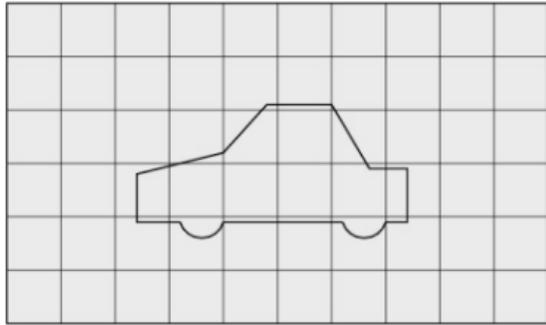
others

OpenSource

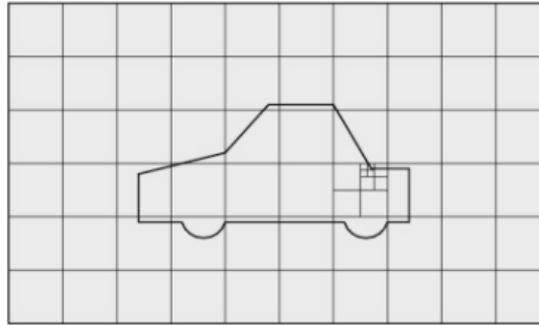
OpenFOAM

Commercial

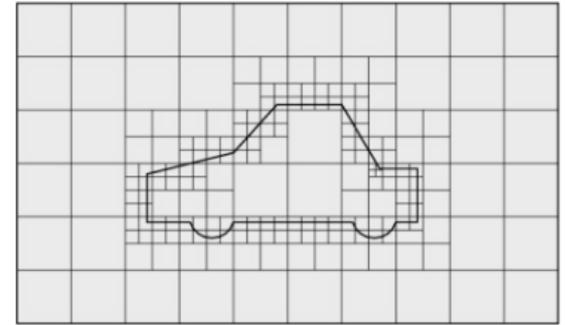
境界適合六面体格子生成ユーティリティ snappyHexMesh



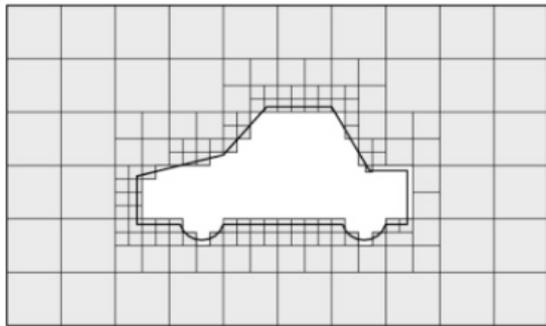
ベース格子



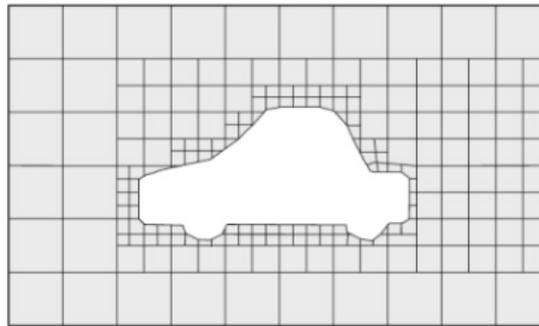
STL表面の細分割



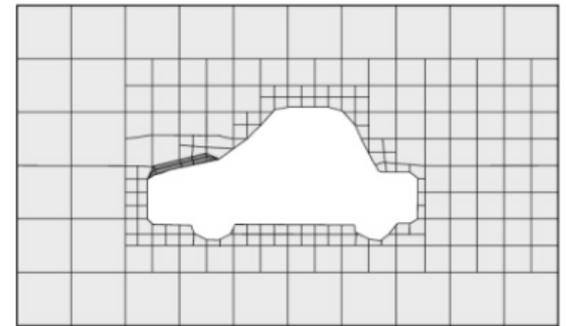
スムージング



階段状格子



境界適合(snap)



レイヤ付加

OpenFOAM®の特徴

basic	基礎的なCFDコード
incompressible	非圧縮性流れ
compressible	圧縮性流れ
multiphase	多相流
DNS	直接数値シミュレーション
combustion	燃焼
heatTransfer	熱輸送
electromagnetics	電磁流体
stressAnalysis	固体応力解析
discreteMethod	離散解析
lagrangian	粒子追跡
financial	金融工学

標準ソルバ
78個

チュートリアルケース
157個

OpenFOAMのインストール

対応OS

- Linux : Ver. 1.6まではバイナリ配布、
Ver. 1.7からは Ubuntu/Debianのパッケージ配布
Ver.2.0からはSuSE/Fedraのパッケージ配布
- UNIX: ソースからコンパイルで概ね動く
- Mac OS X : ユーザによる非公式ポーティング
- Windows : ユーザによる非公式ポーティング
 - ネイティブ+GUI版は様々なサードパーティで開発中

OpenFOAMのインストール

- Linux機がある場合、本格運用したい場合
 - バイナリパッケージをダウンロードし、READMEに従って、展開と設定
- Windows機やMac機で試用したい場合
 - VMWare等でLinuxの仮想環境を作り、Linux機と同様にインストール
- LinuxにOpenFOAMを組み込んだ仮想環境パッケージをインストール
 - CAELinux
 - DEXCS for OpenFOAM

DEXCS for OpenFOAM(R)

- 岐阜高専・(株)デンソーによる共同開発
- ベースOS: Ubuntu Linux 10.04 32bit版、64bit版
- インストール済の主なソフトウェア
 - OpenFOAM-2.0.x
 - Paraview-3.10.1 (可視化ツール)
 - blender-2.49 (3次元モデラー、CAD)
 - 上記ソフトウェアの一部を設定、起動するためのGUIランチャー&動画マニュアル

DEXCS2011 for OpenFOAM(R)

動画チュートリアル

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

DEXCS2011 for OpenFOAM(R) 2011/12

ヘルプ

形状作成
メッシュ1
メッシュ2
計算実行
結果処理

shapeName	Type	Apply	Smin	Smax	Region	Layers
box1.stl	Region	<input checked="" type="checkbox"/>	0	1	0	0
dexcs.stl	Surface	<input checked="" type="checkbox"/>	2	8	0	1
dexcs.aMesh	Feature	<input checked="" type="checkbox"/>	0	0	0	0
inlet	Patch	<input type="checkbox"/>	0	0	0	1
outlet	Patch	<input type="checkbox"/>	0	0	0	1
			0	0	0	1
			0	0	0	0

レイヤー層の数も指定できます

Typeが「Surface」に関しては、表面輪郭に対して細分化レベル (min/max) を指定できます

メッシュ確定-> exe
time: 3
実行

case
simpleFoam solve/application
実行

caseフォルダを対象に「便利ツール」の簡単起動

体験環境 の説明



www.opencae.jp/wiki/オープンCAEワークショップ2012

OpenCAE

Wiki: オープンCAEワークショップ2012 ツイート 0

オープンCAEワークショップ2012 / OpenCAE Workshop 2012

講習会 A

- 題目: 「SwiftツールによるOpenFOAM®用メッシュ作成」
- 講師: 野村 悦治(オープンCAE学会)

DEXCS2011 for OpenFOAM(R) を改変

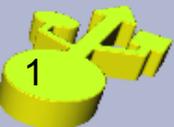
blenderを2.49から2.62に変更。併せて、**Swiftツール** ([SwiftSnap](#), [SwiftBlock](#)) を組み込み済

- **ランチャーの標準チュートリアル問題も、Swiftツールを使う方法**で出来るようにした。
 - ただし、**Winkチュートリアルは未対応**
- **Swiftツールの検証用サンプル**を各種用意した
 - 公開されている3つの例題も含めて、OpenFoamのケースファイル付き(その場でメッシュ確認できるように構成)
- OpenFOAMの**最新マニュアル和訳版**を同梱
 - UserGuideJa-2.1.0_beta.pdf (索引未完成のため β 版)
 - ProgrammersGuideJa-2.1.0.pdf [ソース]
- ランチャーと十徳ナイフの一部バグフィックス

朱字: 本体験コースで活用

DEXCS for OpenFOAM®の体験1

オープンCAEコンサルタント OCSE²
代表 野村 悦治



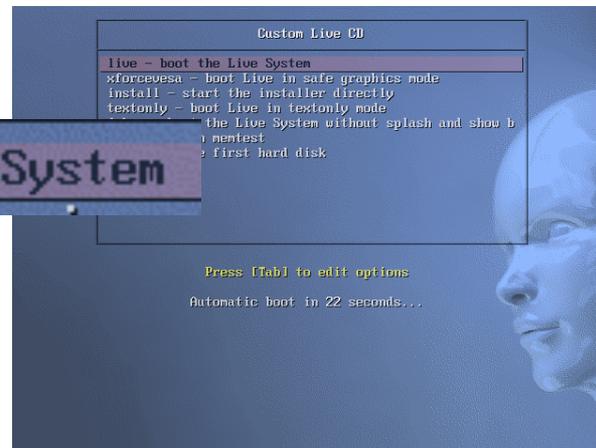
起動～インストール



ライブDVD起動画面

BIOS設定(起動順をDVD優先)
パソコンをDVDから起動

DVD



live - boot the Live System

アプリケーション 場所 システム 5月27日 (日) 午前 9:02 custom

DEXCS

setupDEXCS.sh

Custom Live のインストール

SwiftExample

1 ダブルクリック

2 クリック

3

いくつかのプログラムが未だ実行中です:

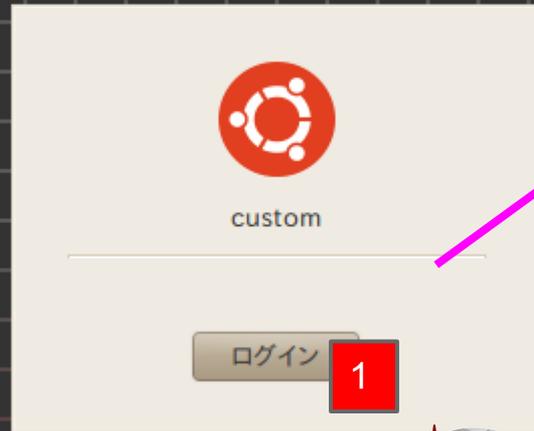
- gnome-keyring-daemon 応答なし
- Power Manager 応答なし

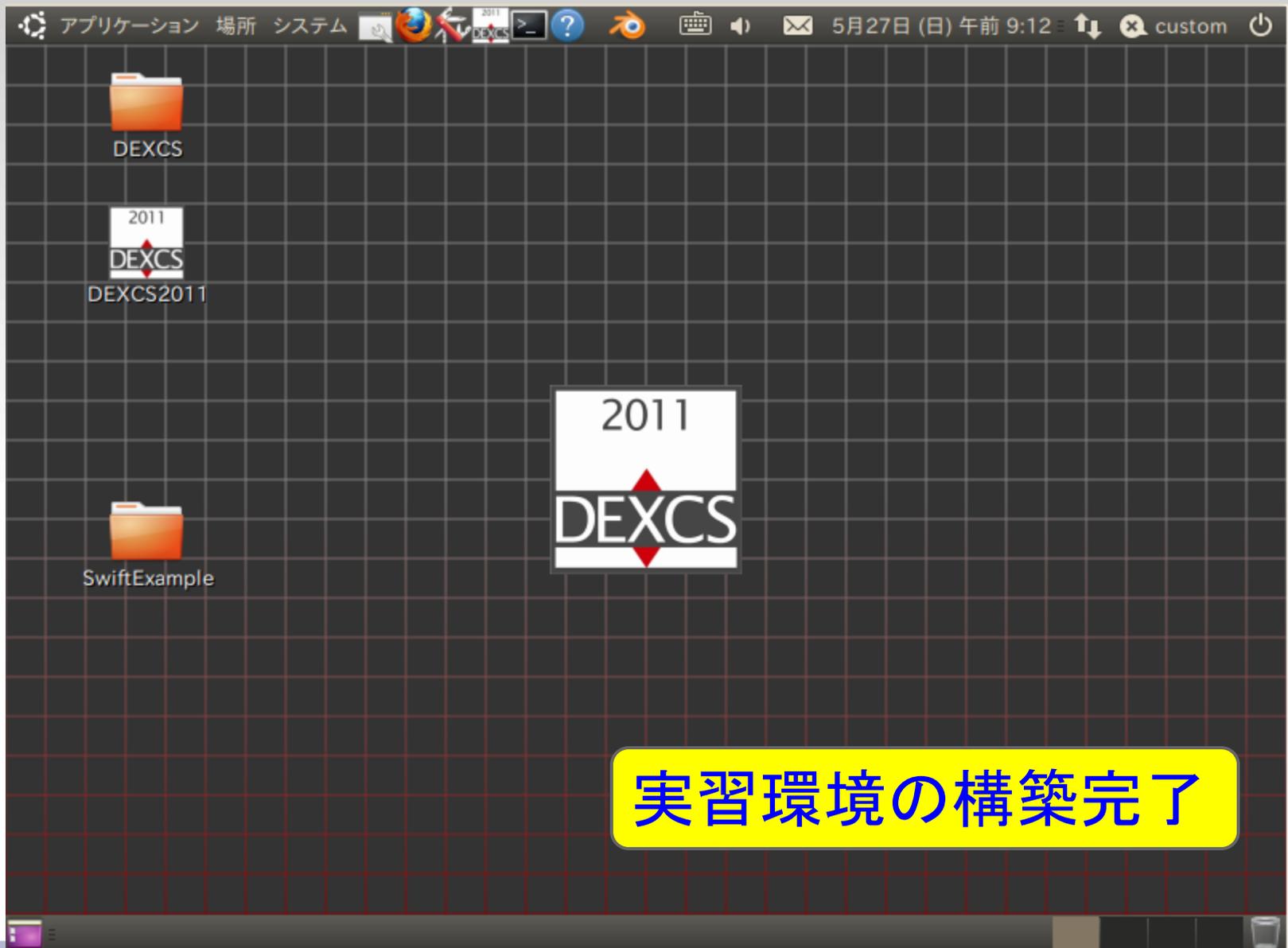
プログラムが終了するのを待っています。実行中のプログラムを途中で強制終了させてしまうと、これまでの処理結果を失ってしまう可能性があります。

画面をロックする キャンセル とにかくログアウトする

このダイアログは出ない場合もある







実習環境の構築完了

DEXCSランチャーの起動と操作方法

まずは、使ってみる



DEXCSランチャーの起動

アプリケーション 場所 システム 5月27日(日) 午前 9:12 custom

DEXCS

2011
DEXCS
DEXCS 2011

SwiftExample **ダブルクリック**

stdout/stderr
ファイル(F) 編集(E) 表示(V) 端末(T) へ

DEXCS2011 for OpenFOAM(R) 2011/12
ファイル ツール ヘルプ

DEXCS

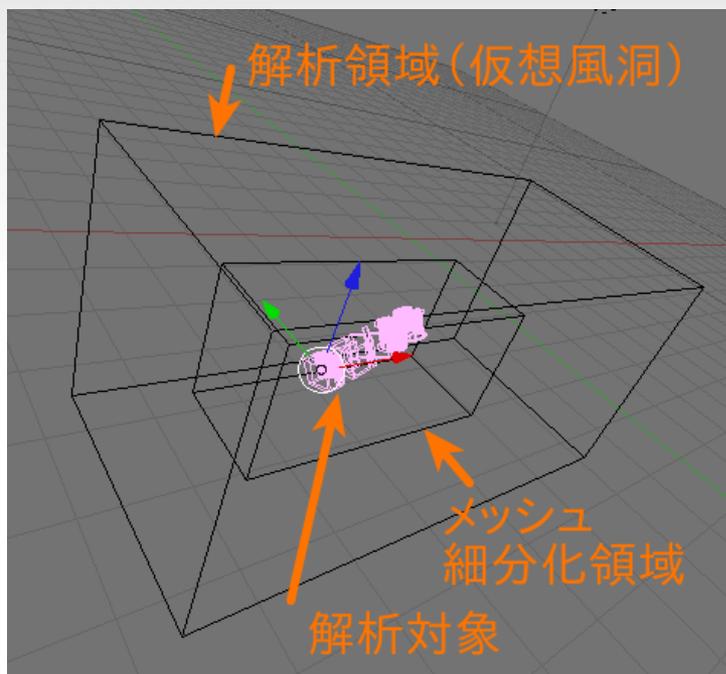
- 形状作成
- メッシュ1
- メッシュ2
- 計算実行
- 結果処理

DEXCS for OpenFOAM(R)

DEXCS --- Simple, Just Use It, Have Fun! ---
オープンCAEワークショップ2012(特別版)
OPENFOAM and OpenCFD are
registered trademarks of SGI Corp.

DEXCSランチャーの嬉しさ

仮想風洞試験が、ボタンを順番に押して
いだけで実行可能



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・名前を知らなくても
解析シーン毎にボタンを順番に押していけばよい)

解析フォルダ

DEXCS2011 for OpenFOAM(R) 2011/11

ファイル ツール ヘルプ

解析フォルダ新規作成

解析フォルダを開く

終了

メッシュ1

メッシュ2

計算実行

結果処理

Change Directory

解析フォルダを /home/custom/Desktop/test に設定しました。

OK(O)

新規解析フォルダを選択してください

custom フォルダの作成(L)

場所(P)	名前	サイズ	最終変更日
検索	Desktop		20:51
最近開いた...	OpenFOAM		昨日の 22:33

custom

デスクトップ

ファイル・シス...

フロッピーード...

追加(A) 削除(R)

test + Enterキーを押す

test

DEXCS

追加(A) 削除(R)

キャンセル(C) 開く(O)

新規解析フォルダを選択してください

custom Desktop test フォルダの作成(L)

場所(P)	名前	サイズ	最終変更日
-------	----	-----	-------

検索

最近開いた...

custom

デスクトップ

ファイル・シス...

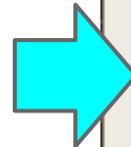
フロッピーード...

追加(A) 削除(R)

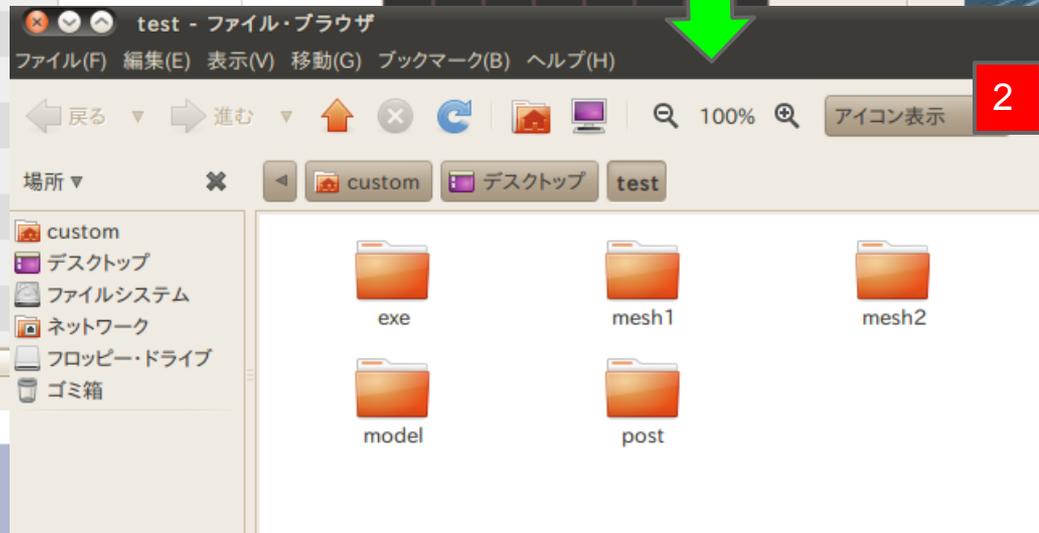
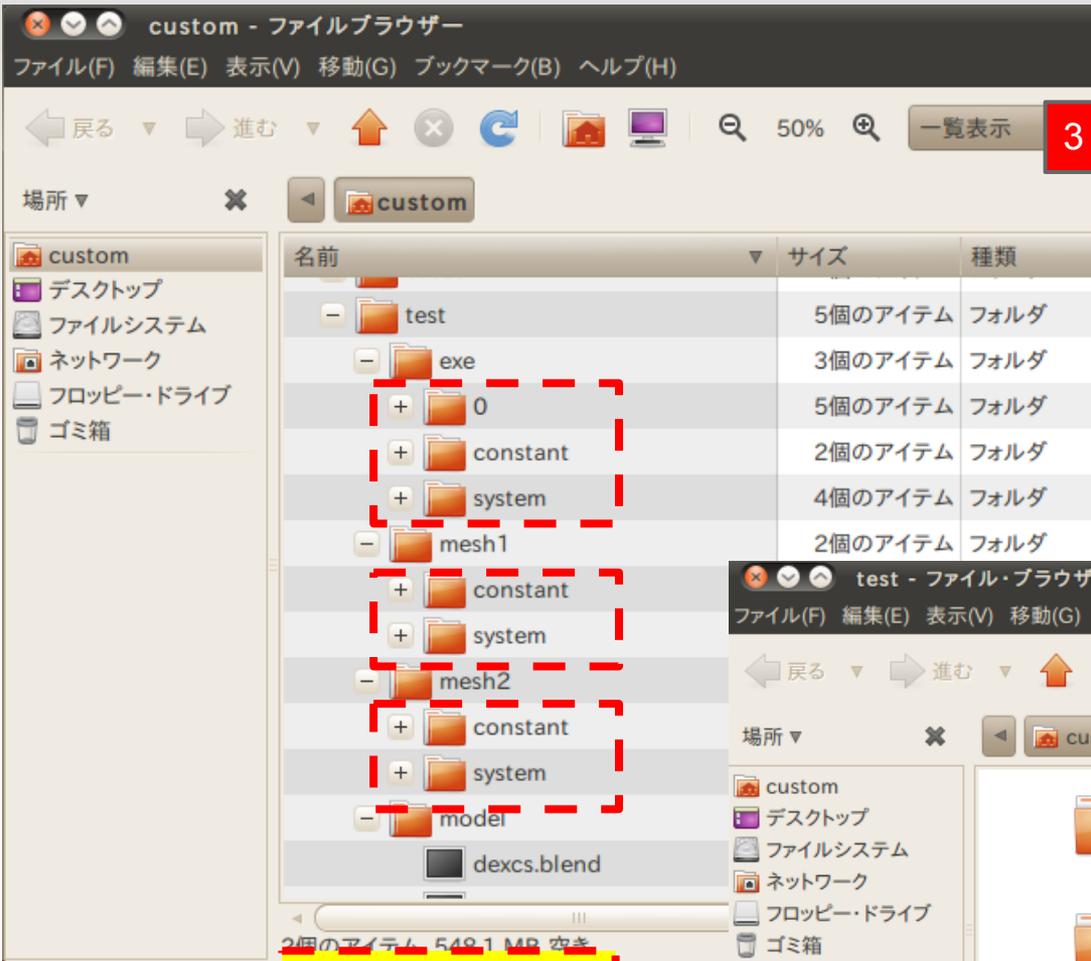
キャンセル(C) 開く(O)

解析フォルダ

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



解析フォルダ



OpenFOAMの
caseファイル構造

形状確認 (blender)

現在の解析フォルダ: /home/custom/Desktop/test

ファイル ツール ヘルプ

DEXCS

形状作成 **1**

メッシュ1

メッシュ2

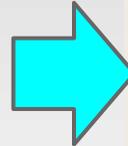
計算実行

結果処理

blender

2 実行

解析実行できます



使用するモデルを選択してください

custom Desktop test model

場所(P)	名前	サイズ	最終変更日
検索	dexcs.blend	533.4 KB	18:13
最近開いた...	dexcsSwitt.blend	511.9 KB	18:13

追加(A) 削除(R)

Files (*)

キャンセル(C) 開く(O) **4**



Blender [/home/custom/Desktop/test/model/dexcs.blend]

オブジェクトツール

変形:

移動

回転

拡大縮小

原点

オブジェクト:

オブジェクトの複製

操作

レンダ

レンダ

レン

解像度: フレー

19 開

10 終

50 フ

アスペ フレー

1.0 24

1.0 タイ

アンチエイ

ビュー 選択 オブジェクト オブジェクトモ

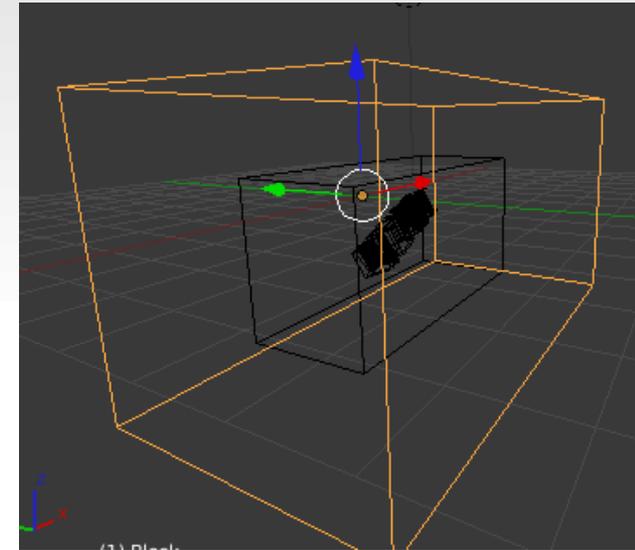
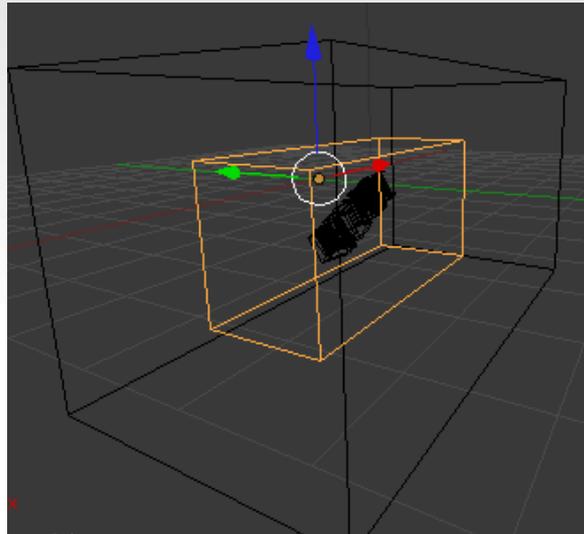
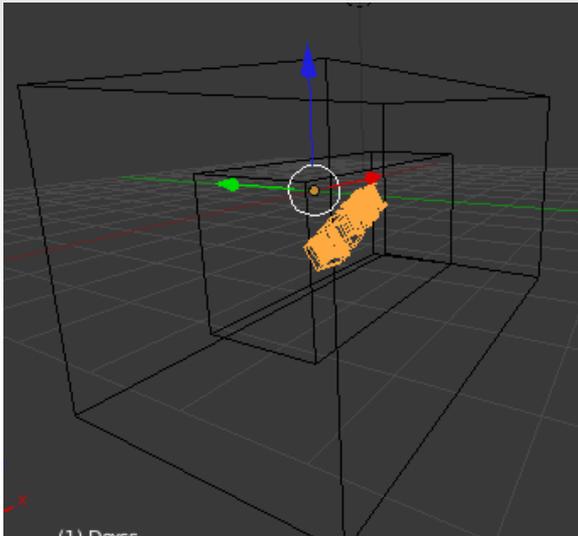
開始: 1 終了: 250

モデル確認 (blender)

解析対象

メッシュ細分化領域

解析領域
(仮想風洞)



視方向回転:
マウスホイールボタンを押しながらドラッグ
視方向平行移動:
ShiftまたはCtrlキーを押しながらマウスホイール回転
拡大・縮小:
マウスホイール回転

オブジェクトの選択切替は、
マウス右ボタンクリックで

メッシュDict作成

現在の解析フォルダ: /home/custom/Desktop/test

ファイル ツール ヘルプ

DEXCS

形状作成 **1**

メッシュ1

メッシュ2

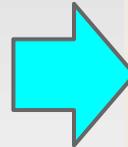
計算実行

結果処理

blender

2 実行

解析実行できます



使用するモデルを選択してください

custom Desktop test model

場所(P)	名前	サイズ	最終変更日
	dexcs.blend	533.4 KB	18:13
	dexcsSwift.blend	511.9 KB	18:13

追加(A) 削除(R) Files (*)

キャンセル(C) 開く(O) **4**

Dict作成用パラメタ
設定済モデル



Blender [/home/custom/Desktop/test/model/dexcsSwift.blend]

ファイル 追加 レンダー ヘルプ Default Scene Blenderレンダー

User Persp

レンズ: 35.000

開始: 0.100 終了: 1000.000

カメラ

3Dカーソル

ls, bgl, blf, mathutils
Convenience Imports: from mathutils import *; from math import *

>>> |

コンソール オートコンプリート

SwiftSnap確認 (blender)

The image displays two screenshots of the Blender 2.79 interface. The top screenshot shows a 3D scene with a red 'DEXS' model. A yellow callout box labeled '2' points to the 'Box1' object in the Outliner panel, with the text 'Box1を表示状態にしておく' (Keep Box1 in visible state). A red dashed arrow points from this box to the bottom screenshot. The bottom screenshot shows the same scene with a wireframe box overlaid on the model. A yellow callout box labeled '3' points to the 'SwiftSnap settings' panel, which is circled in red. The 'SwiftSnap settings' panel includes options for 'Write', 'convertToMeters: 1.00', 'Make base me' (set to 'resoluti: 0.28'), 'Snap mesh', and 'Add layers'. A yellow callout box labeled '1' points to the mouse cursor in the 3D view, with the text 'Zキー Nキー' (Z key N key) and 'マウスマウスカーソルを、モデル表示画面の上に置いてキー操作' (Place the mouse cursor on the model display screen and perform key operations). A yellow callout box labeled '2' also points to the mouse cursor area with the text 'Zキー Nキー' (Z key N key). A yellow callout box labeled '3' points to the 'SwiftSnap の設定確認' (Check SwiftSnap settings) text. The bottom left corner features a yellow callout box with the number '16' and a small 3D model of the 'DEXS' object.

2 Box1を表示状態にしておく

1 Zキー Nキー
マウスマウスカーソルを、モデル表示画面の上に置いてキー操作

3 SwiftSnap の設定確認

16

SwiftSnap⇒snappyHexMeshDict作成

The image shows a Blender 2.62 interface with several panels and windows. A red box labeled '1' highlights the 'Write' button in the SwiftSnap settings panel. A blue arrow points from this button to a file browser window. In the file browser, a red box labeled '2' highlights the path '/home/custom/Desktop/test/mesh2/system/' and a red box labeled '3' highlights the 'Write' button. A red arrow points from the 'Write' button in the file browser to a file explorer window showing a directory structure. In this explorer, the 'system' folder is highlighted with a red box. A yellow callout box with a red border contains the text '出力先を間違いないように注意' (Be careful not to get the output destination wrong).

SwiftSnap settings

Write

convertToMeters: 1.00

Make base mesh

Snap mesh Add layers

Set locationInMesh

Feature settings

Detect Non-Manifold

featureAngle: 150.00

エミッター遮蔽

Feat. level: 0

Patch settings

名前: Dexcs

Min level: 3 Max level: 4

レイヤー: 2

Color, min, max, name, 0, 0, de 3

Refinement settings

Refine region: Box1

レベル: 3 距離: 0.00

Refine inside object

Set Refinement Region

Refine 3 level 0 m inside Box1

Blender* [/home/custom/Desktop/test/model/dexcsSwift.blend]

ファイル 追加 レンダー ヘルプ 戻る Blenderレンダー blender.org 262 | Ve:1530 | Fa:3030

新しいディレクトリを作成

システム

cdrom rofs

Bookmarks

追加

custom Desktop

最近利用したファイル

Write

Distribute

Reset

test

exe

mesh1

mesh2

constant

system

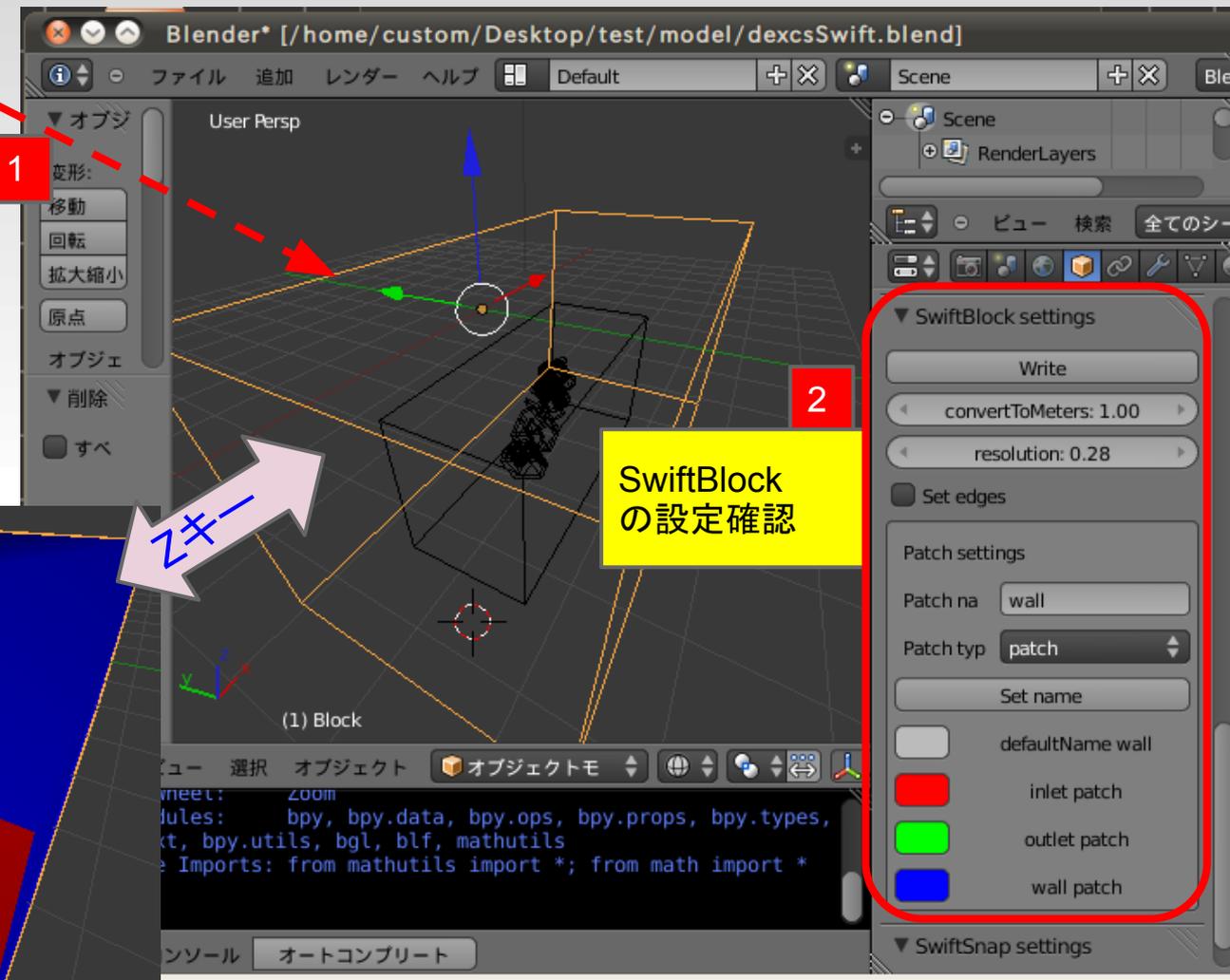
model

post

出力先を間違いないように注意

SwiftBlock確認 (blender)

マウス右クリック
にて解析領域
(Block)を選択



SwiftBlock
の設定確認

Zキー

SwiftBlock⇒blockMeshDict作成

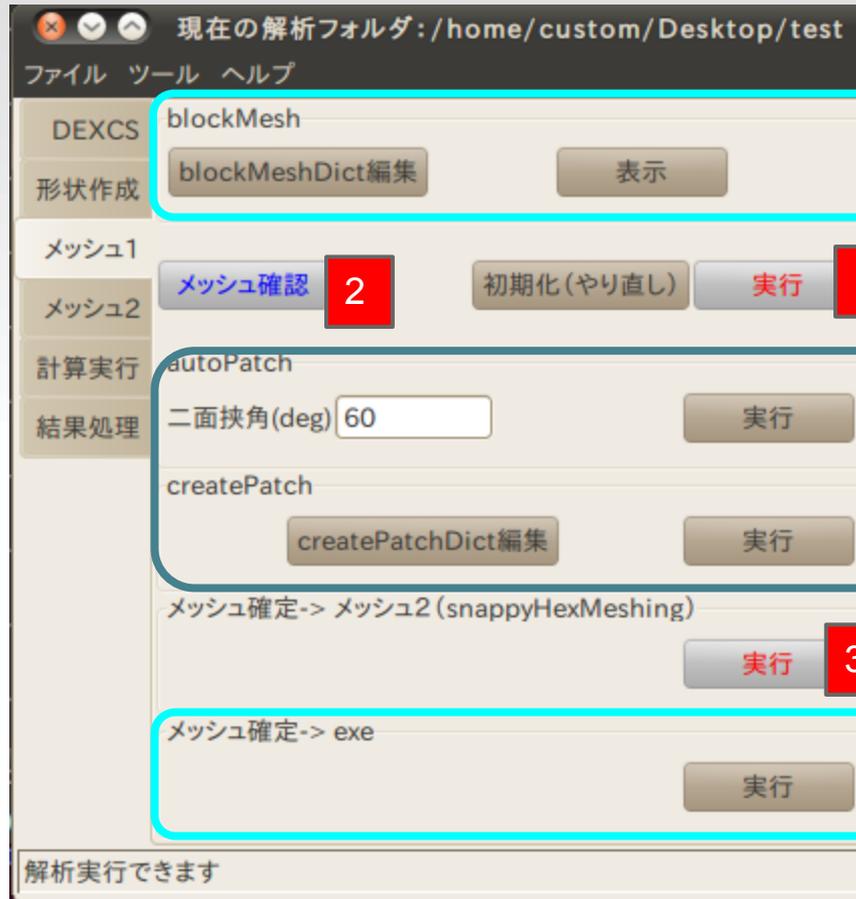
1

2

3

出力先を間違いないよう注意

メッシュ1 (基礎メッシュ作成)



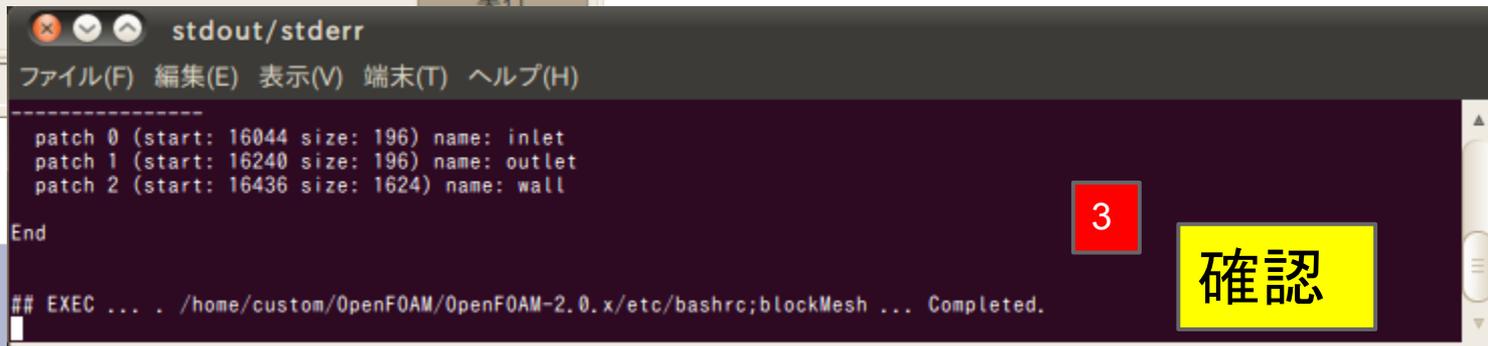
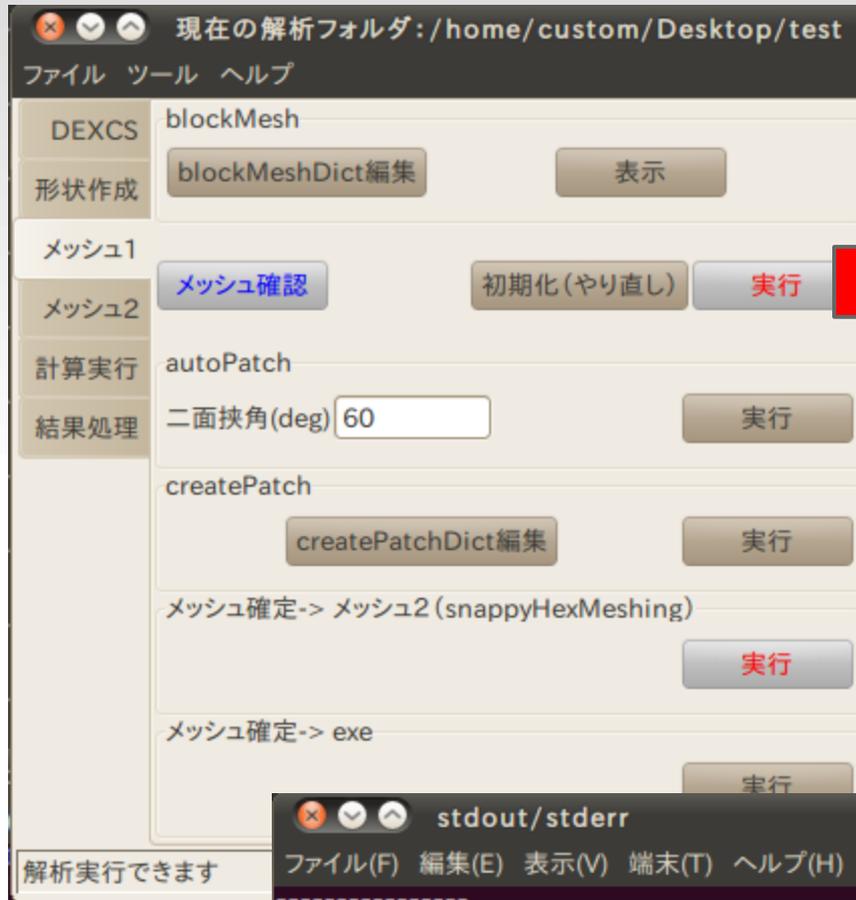
blockMeshDict を手修正
する場合に使用

使用しない
(次期Ver.で廃止予定)

snappyHexMeshを実行せず計
算実行

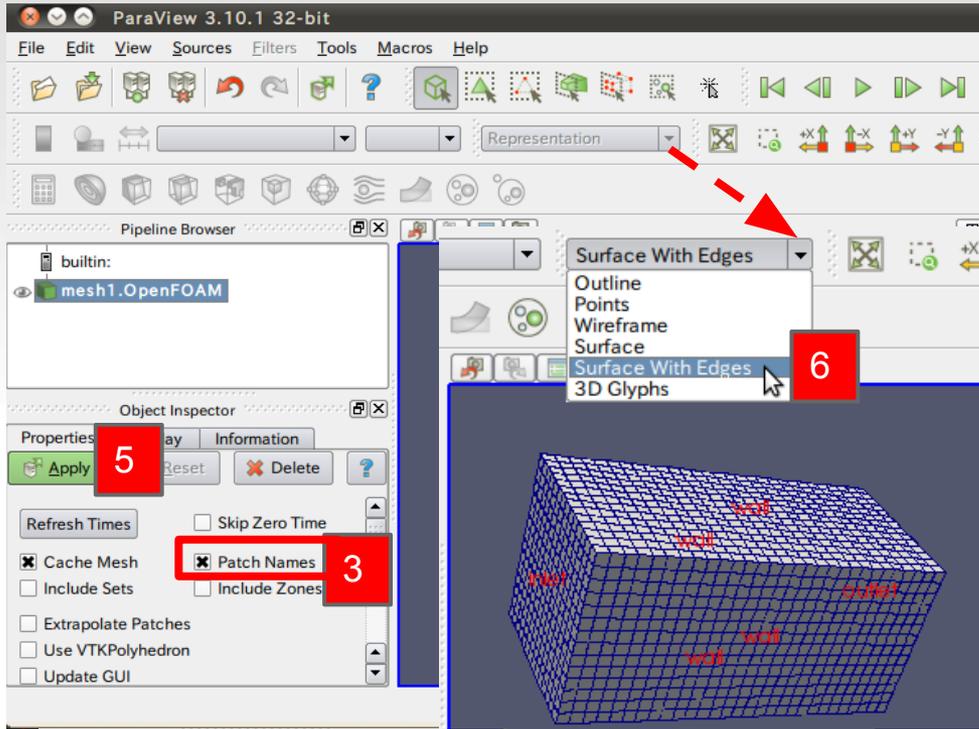
- blockMeshのみを使用
- 別途作成したメッシュファ
イルを使用

メッシュ1 (基礎メッシュ作成)



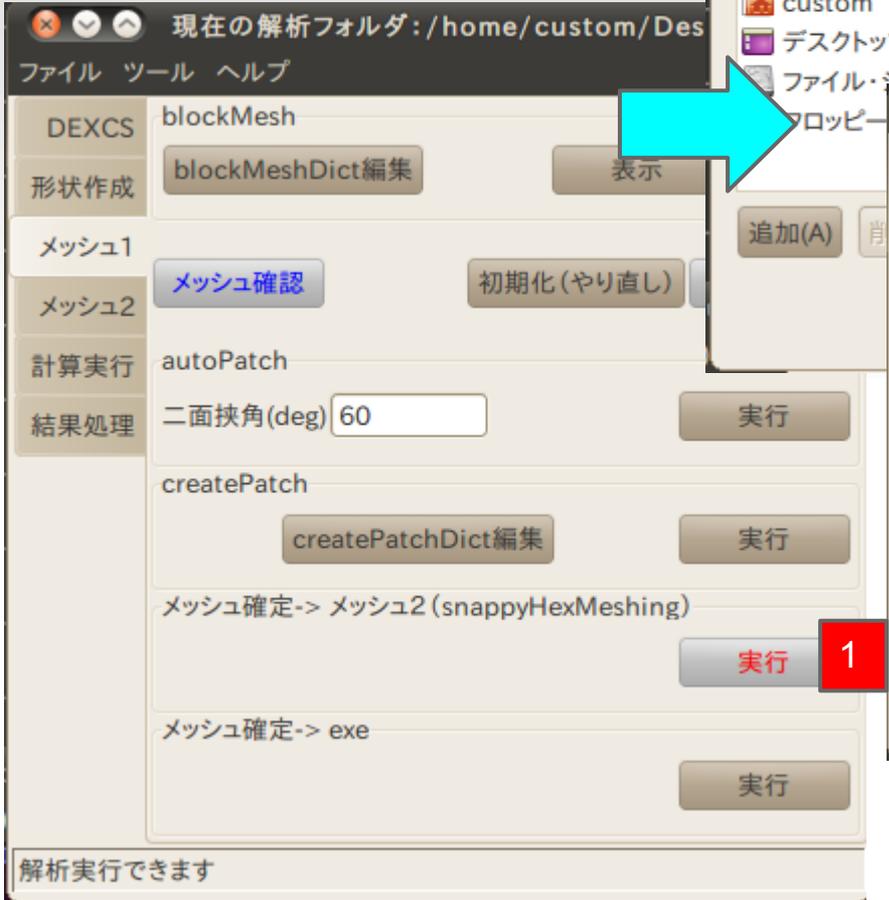
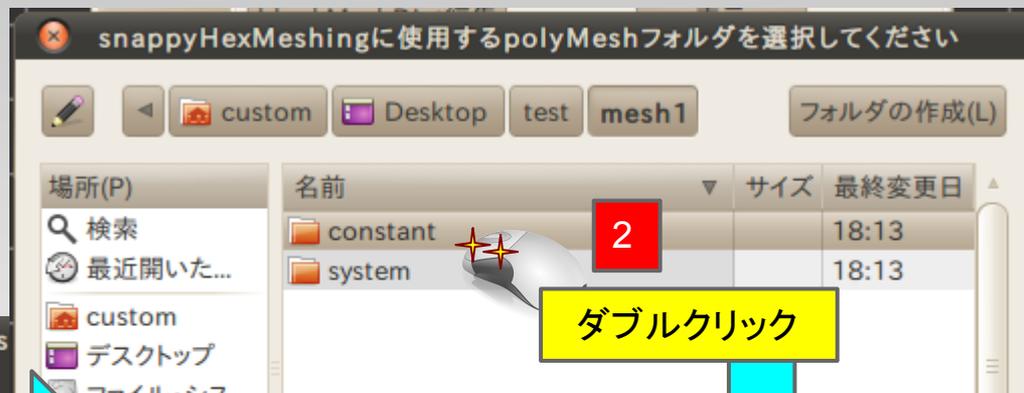
メッシュ1

メッシュ確認(paraFoam)



- 視方向回転:
マウス左ボタンを押しながらドラッグ
- 視方向移動:
マウスホイールボタンを押しながらドラッグ
- 拡大・縮小:
マウスホイール回転

メッシュ1 メッシュ確定



メッシュ2 (snappyHexMesh作成)

The screenshot shows a software window titled "現在の解析フォルダ: /home/custom/Desktop/test". The interface is divided into several sections:

- surfaceFeatureExtract**: Includes a text input field for "includedAngle" with the value "150" and a "選択実行" button. A callout box points to this section with the text "使用しない (次期Ver.で廃止予定)".
- snappyHexMeshDict**: Includes buttons for "形状選択", "Dict編集", and "初期化(やり直し)". A callout box points to this section with the text "snappyHexMeshDict を手修正する場合に使用".
- 並列分割**: Includes a text input field for "並列分割数" with the value "2". A callout box points to this section with the text "並列計算のコア数を変更可能".
- snappyhexmesn**: Includes a "停止" button and a red "実行" button labeled "1".
- メッシュ確認**: Includes a checked checkbox for "native" and a blue "実行" button labeled "2".
- メッシュ確定-> exe**: Includes a text input field for "time" with the value "3" and a red "実行" button labeled "3".

At the bottom of the window, it says "解析実行できます".

メッシュ2

snappyHexMesh

現在の解析フォルダ: /home/custom/Desktop/test
ファイル ツール ヘルプ

DEXCS surfaceFeatureExtract
includedAngle 150 選択実行

形状作成

メッシュ1 snappyHexMeshDict
形状選択 Dict編集 初期化(やり直し)

メッシュ2 形状選択 Dict編集 初期化(やり直し)

計算実行 並列分割
並列分割数 2

結果処理 snappyHexMesh
停止 実行 1

メッシュ確認
 native 実行

メッシュ確定-> exe
time 3 実行

解析実行できます



RunSnappyParalle
SnappyParallel を実行します
2 OK Cancel / End

```
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
3 98811
4 26200
Writing mesh to time 3
Wrote mesh in = 28.2 s.
Layers added in = 28.2 s.
Finished meshing in = 80.5 s.
End
Finalising parallel run
```

3

確認
(数分~10分)

メッシュ2

snappyHexMesh 異常時の処置

現在の解析フォルダ: /home/custom/Desktop/test

ファイル ツール ヘルプ

DEXCS surfaceFeatureExtract

形状作成 includedAngle 150 選択実行

メッシュ1 snappyHexMeshDict

メッシュ2 形状選択 Dict編集 初期化(やり直し)

計算実行 並列分割

結果処理 並列分割数 2

snappyHexMesh 3 停止 実行

メッシュ確認 native 実行

メッシュ確定-> exe

StopSnappyHexMe 実行

解析実行できます snappyHexMesh を停止します

異常判定?

システム プロセス リソース ファイルシステム

CPU 使用率の履歴

100% 50% 0%

60秒 50 40 30 20 10 0

CPU1 100.0% CPU2 100.0%

メモリとスワップの履歴

100% 50% 0%

60秒 50 40 30 20 10 0

メモリ 378.3 MiB (18.8%) / 2.0 GiB 2

スワップ 0 byte (0.0%) / 0 byte

ネットワークの履歴

2.0 KiB/秒 1.0 KiB/秒 0.0 KiB/秒

60秒 50 40

受信中 0 byte/秒 送信中 0 byte/秒

受信の合計 2.2 MiB 送信の合計 75.4 KiB

メモリとスワップ消費量に注目

OK 4 Cancel / End

メッシュ2 メッシュ確認

現在の解析フォルダ: /home/custom/Desktop/t

ファイル ツール ヘルプ

DEXCS surfaceFeatureExtract

形状作成 includedAngle 150 選択実行

メッシュ1 snappyHexMeshDict

メッシュ2 形状選択 Dict編集 初期化(やり直)

計算実行 並列分割

結果処理 並列分割数 2

snappyHexMesh

停止 実行

メッシュ確認

native 実行 1

メッシュ確定-> exe

time 3 実行

解析実行できます

ParaView 3.10.1 32-bit

File Edit View Sources Filters Tools Macros Help

Pipeline Browser

- builtin:
- exe.foam

Object Inspector

Properties Display Information

4

File Name /n/Desktop/test/mesh2/exe.foam

Case Type Reconstructed Case

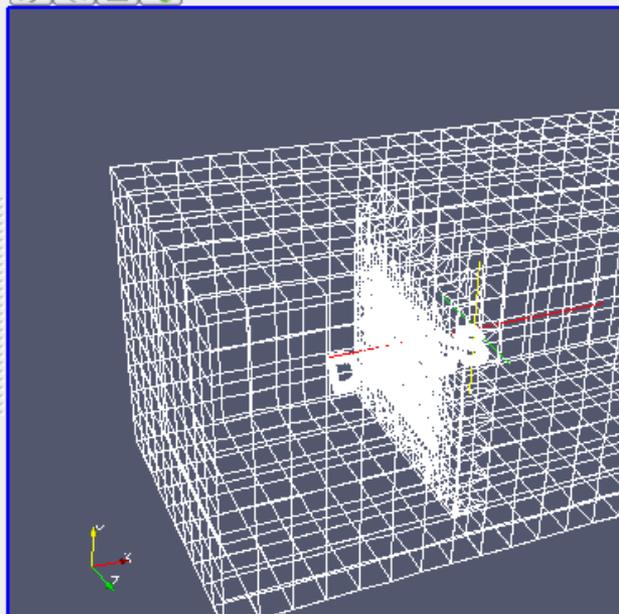
3

- Decomposed Case
- Reconstructed Case

Add dimensional units to array names

Mesh Regions

Wireframe 5



RunParaFoamS

paraview(native) を実行します

OK 2 Cancel / End

メッシュ2 メッシュ確認(内部断面)

ParaView 3.10.1 32-bit

File Edit View Sources Filters Tools Macros Help

Solid Color Wireframe

Pipeline browser: builtin: exe.foam

Object Inspector: Properties Display Information

Apply Reset Delete

Show Plane

Origin: -3, 0.5, 0.199999

Normal: 0, 0, 1

X Normal Y Normal Z Normal

Reset Bounds

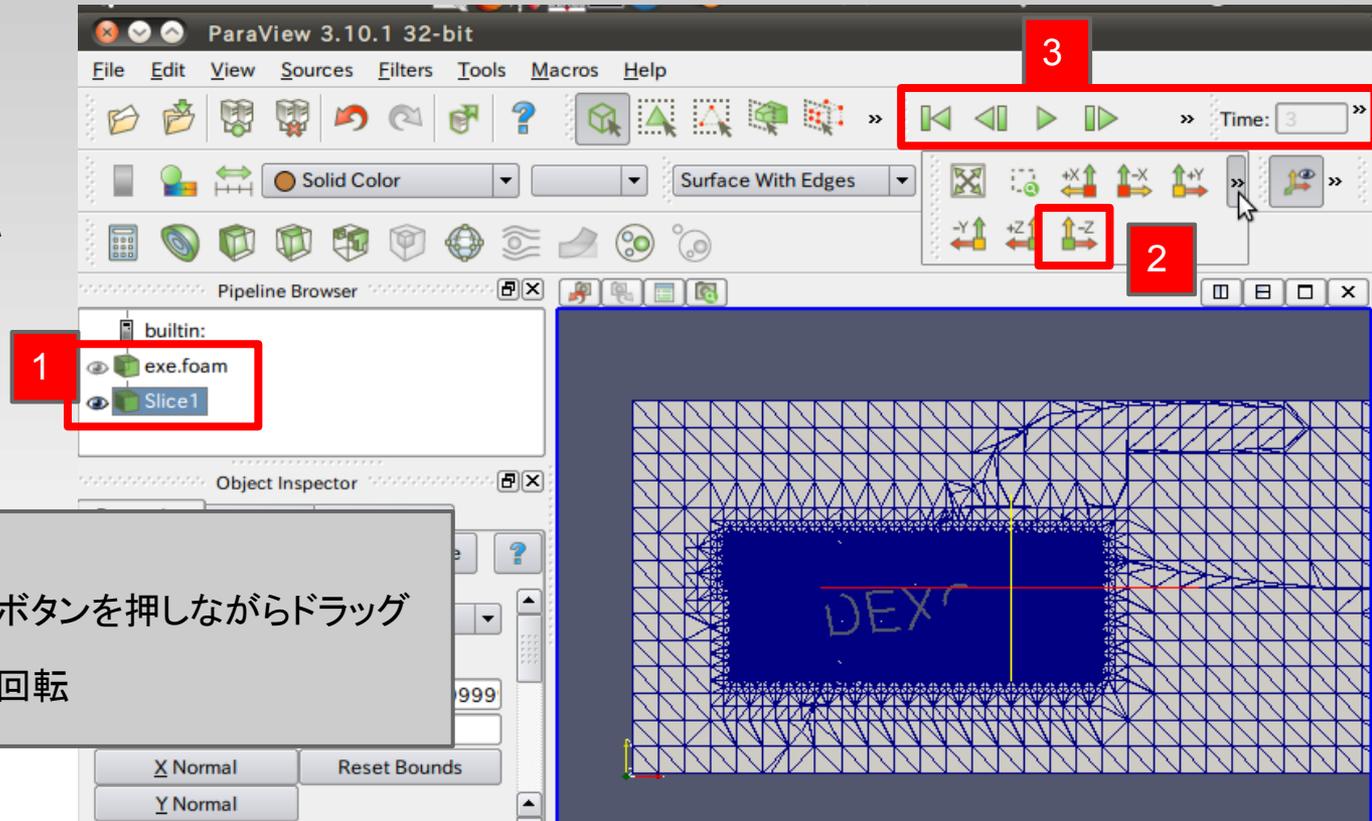
Camera Normal Center on Bounds

Surface With Edges

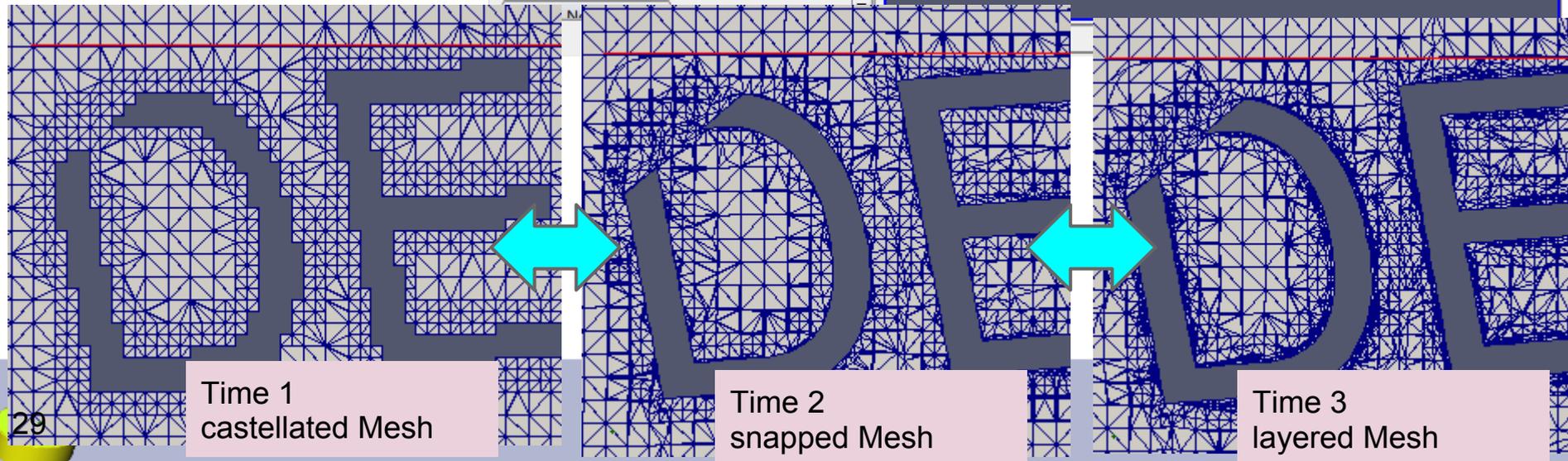
28

OCSE^2

メッシュ2 メッシュ確認 (断面詳細)



視方向移動:
マウスホイールボタンを押しながらドラッグ
拡大・縮小:
マウスホイール回転



メッシュ2 メッシュ確定

現在の解析フォルダ: /home/custom/Desktop

ファイル ツール ヘルプ

DEXCS surfaceFeatureExtract

形状作成 includedAngle 150

メッシュ1 snappyHexMeshDict

メッシュ2 形状選択 Dict編集 初期化(やり直し)

計算実行 並列分割

結果処理 並列分割数 2

snappyHexMesh

停止 実行

メッシュ確認

native 実行

メッシュ確定-> exe

time 3 実行 1

解析実行できます

並列計算時には、少々時間がかかる

計算に使用するpolyMeshフォルダを選択してください

custom Desktop test mesh2 フォルダの作成(L)

場所(P)	名前	サイズ	最終変更日
	3 2		00:37
	Decompose.analyzed		00:26
	PyFoamRunner.snappyH...		00:32
	constant		00:24
	processor0		00:29
	processor1		00:29
	system		昨日の 16:15

計算に使用するpolyMeshフォルダを選択してください

custom Desktop test mesh2 3 フォルダの作成(L)

場所(P)	名前	サイズ	最終変更日
	polyMesh 3		10:24

追加(A) 削除(R)

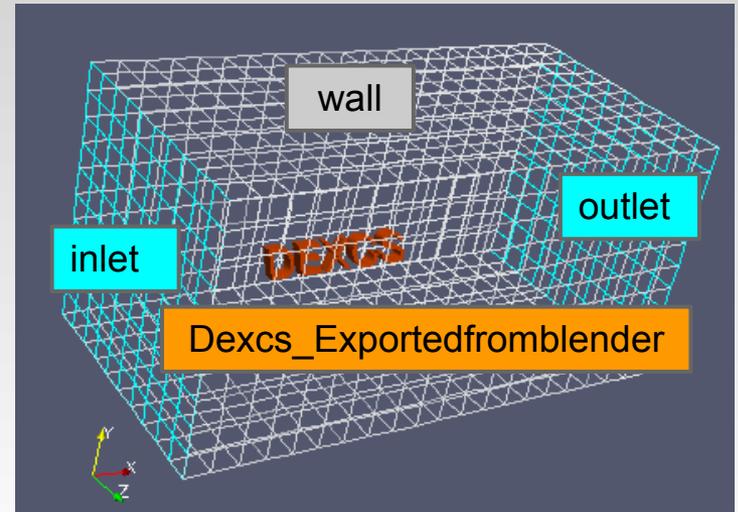
キャンセル(C) 開く(O) 4

Set Mesh

/home/custom/Desktop/test/mesh2/3/polyMesh を
計算用メッシュとして設定しました。

5 OK(O)

境界条件



デフォルト設定

inlet : 流入境界
 $U_{in}=10\text{m/s}$

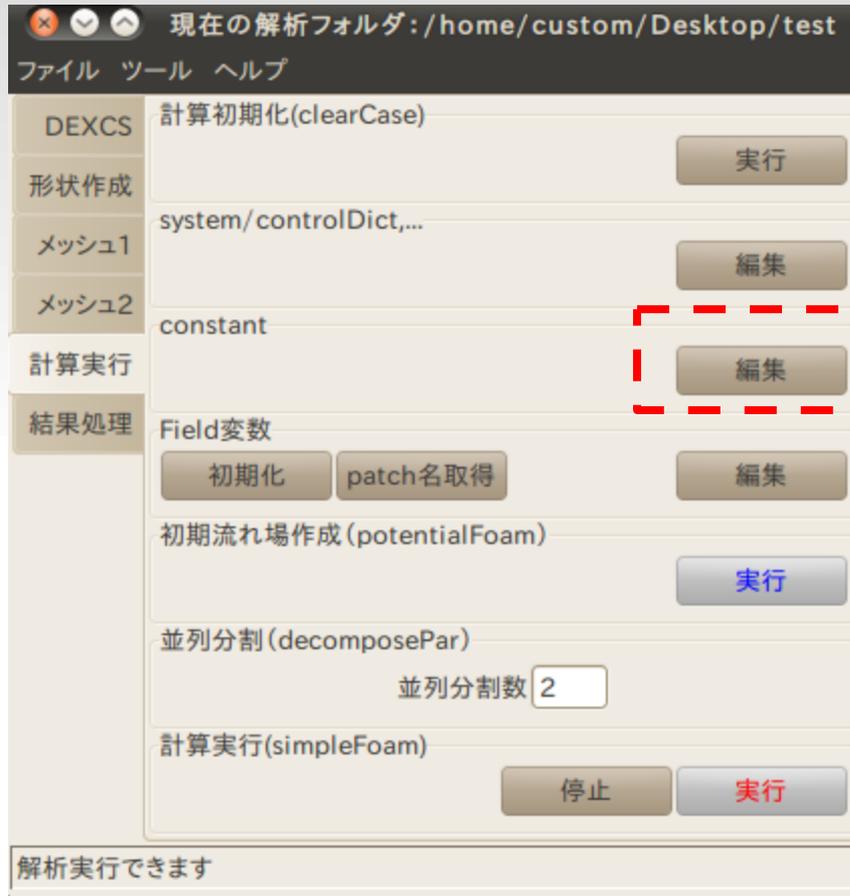
outlet : 流出境界
 $P=0$

wall : スリップ壁

dexcs_Font : 壁
すべり無し、壁法則

確認・変更は可能(後述)

流体特性



デフォルト設定

nu(動粘性係数)
= 1.54e-5 m²/s
kEpsilon乱流モデル

確認・変更は可能(後述)

計算実行(初期流れ場作成)

現在の解析フォルダ: /home/custom/Desktop/test
ファイル ツール ヘルプ

DEXCS 計算初期化(clearCase) 実行

形状作成 system/controlDict,... 編集

メッシュ1 constant 編集

メッシュ2 constant 編集

計算実行 編集

結果処理 Field変数 初期化 patch名取得 編集

初期流れ場作成 (potentialFoam) 実行 **1**

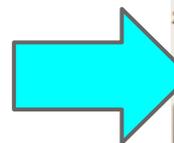
並列分割 (decomposePar) 並列分割数 2

計算実行(simpleFoam)

解析実行できます

非粘性のポテンシャル流れ場計算を実行。

本例では必須だが、必要ない場合や、却って悪化する場合もある。



RunPotentialFoam

初期流れ場作成(potentialFoam)を実行します

OK **2** Cancel / End

```
stdout/stderr
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
in "/home/custom/OpenFOAM/OpenFOAM-2.0.x/platforms/linuxGccDP0pt/bin/potentialFoam"
#9 __libc_start_main in "/lib/tls/i686/cmov/libc.so.6"
#10
in "/home/custom/OpenFOAM/OpenFOAM-2.0.x/platforms/linuxGccDP0pt/bin/potentialFoam"
Segmentation fault (core dumped)
PyFoam WARNING on line 196 of file /usr/local/lib/python2.6/dist-packages/PyFoam/Applications/Pote
ger called: Resetting fvSchemes and fvSolution
## EXEC ... . /home/custom/OpenFOAM/OpenFOAM-2.0.x/etc/bashrc;pyFoamPotentialRunner.py . ... Completed.
```

3 **確認**
(1分程度)

計算実行

The image shows the OpenFOAM GUI workflow for running a simulation. The 'simpleFoam' control dictionary is visible, with the 'decomposePar' section set to 'parallel' and 'n' set to 2. The 'RunsimpleFoam' dialog box prompts the user to execute 'simpleFoam'. The 'Residuals' plot shows the convergence of variables U_x , U_y , U_z , ϵ , p , and k over 45 iterations. The 'Continuity' plot shows the convergence of the continuity error. The 'stdout/stderr' terminal window displays the simulation results, including the inlet and outlet mass flows.

1 実行

2 simpleFoam を実行します

3

4

5

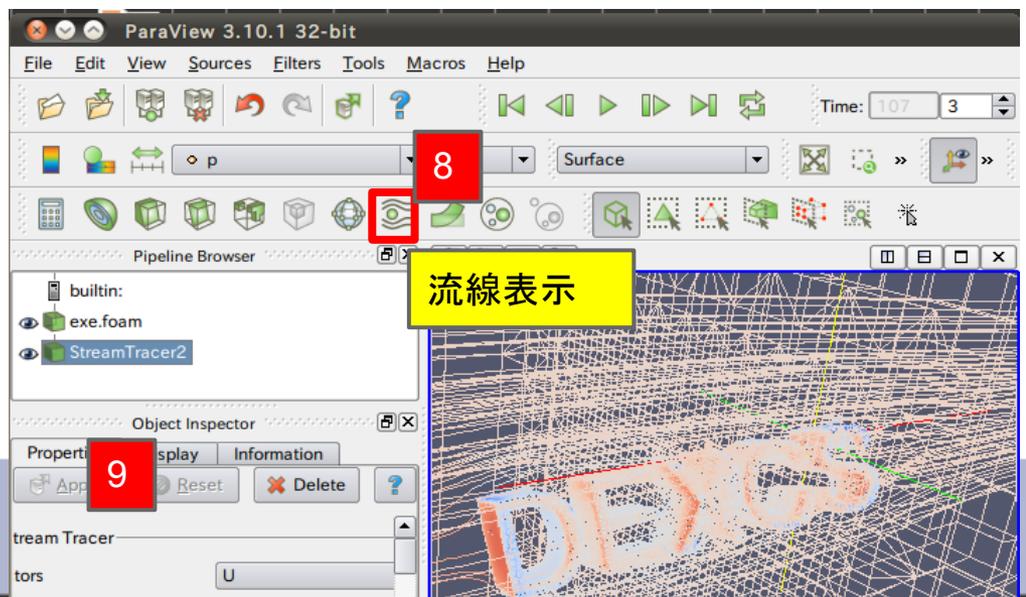
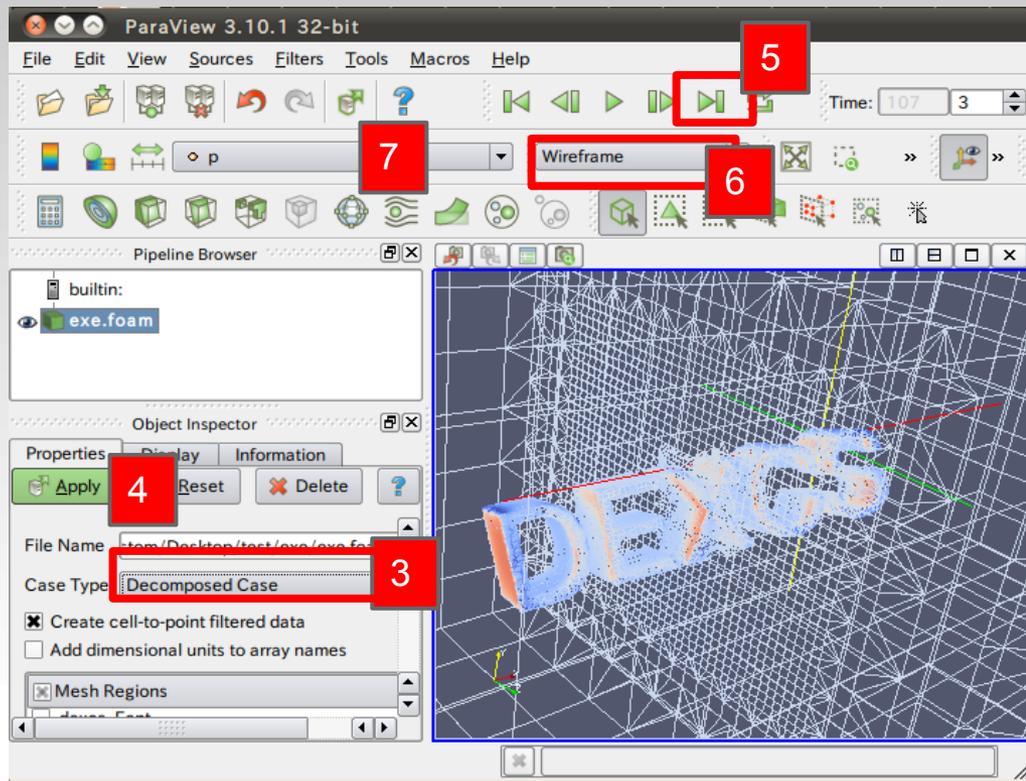
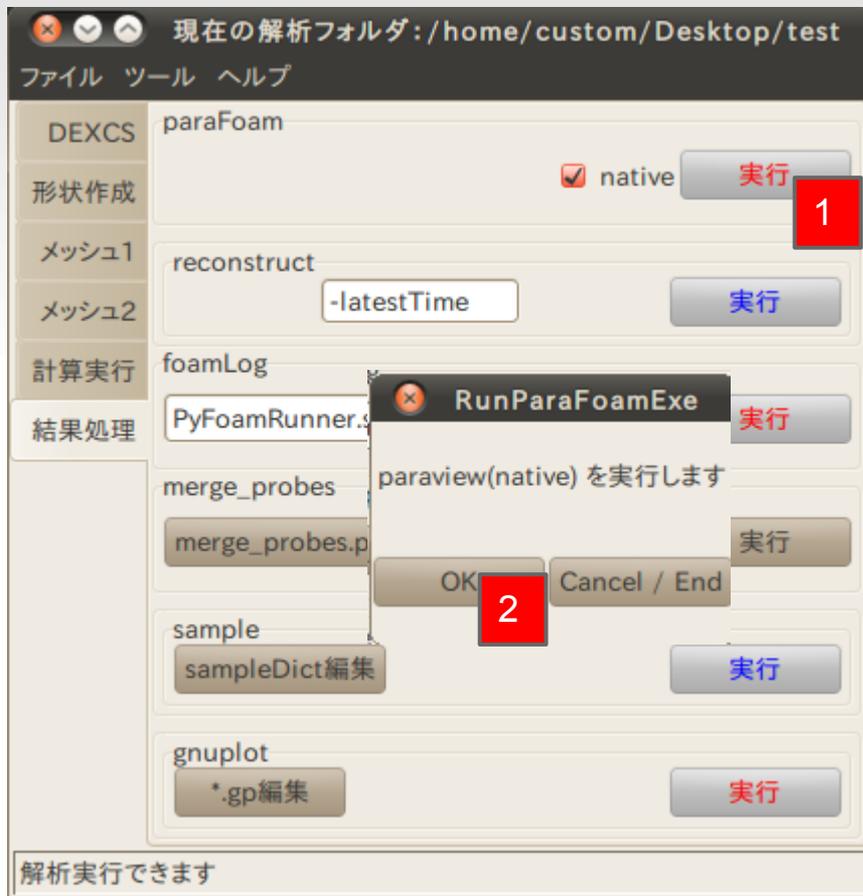
並列計算のコア数を変更可能

並列分割数 2

途中停止は可能

計算終了 (数分)

流れ場の可視化 (ParaView)



reconstruct (領域結合)

現在の解析フォルダ: /home/custom/Desktop/test
ファイル ツール ヘルプ

DEXCS	paraFoam	<input checked="" type="checkbox"/> native	実行
形状作成			
メッシュ1	reconstruct	-latestTime	実行 1
メッシュ2			
解実行			
結果処理	foamLog	PyFoamRunner.simpleFoam.logfile	実行
	merge_probes	merge_probes.py 編集	実行
	sample	sampleDict 編集	実行
	gnuplot	*.gp 編集	実行

解析実行できます

RunReconstructPar

reconstructPar -latestTime を実行し

OK Cancel / End

test

- exe
 - 0
 - 170
 - Decomposer.analyzed
 - Gnuplotting.analyzed
 - constant
 - dynamicCode
 - forceCoeffs
 - forces
 - patchExpression_massFlowSwak
 - patchMassFlows_massFlowSimple
 - probes
 - processor0
 - processor1
 - system

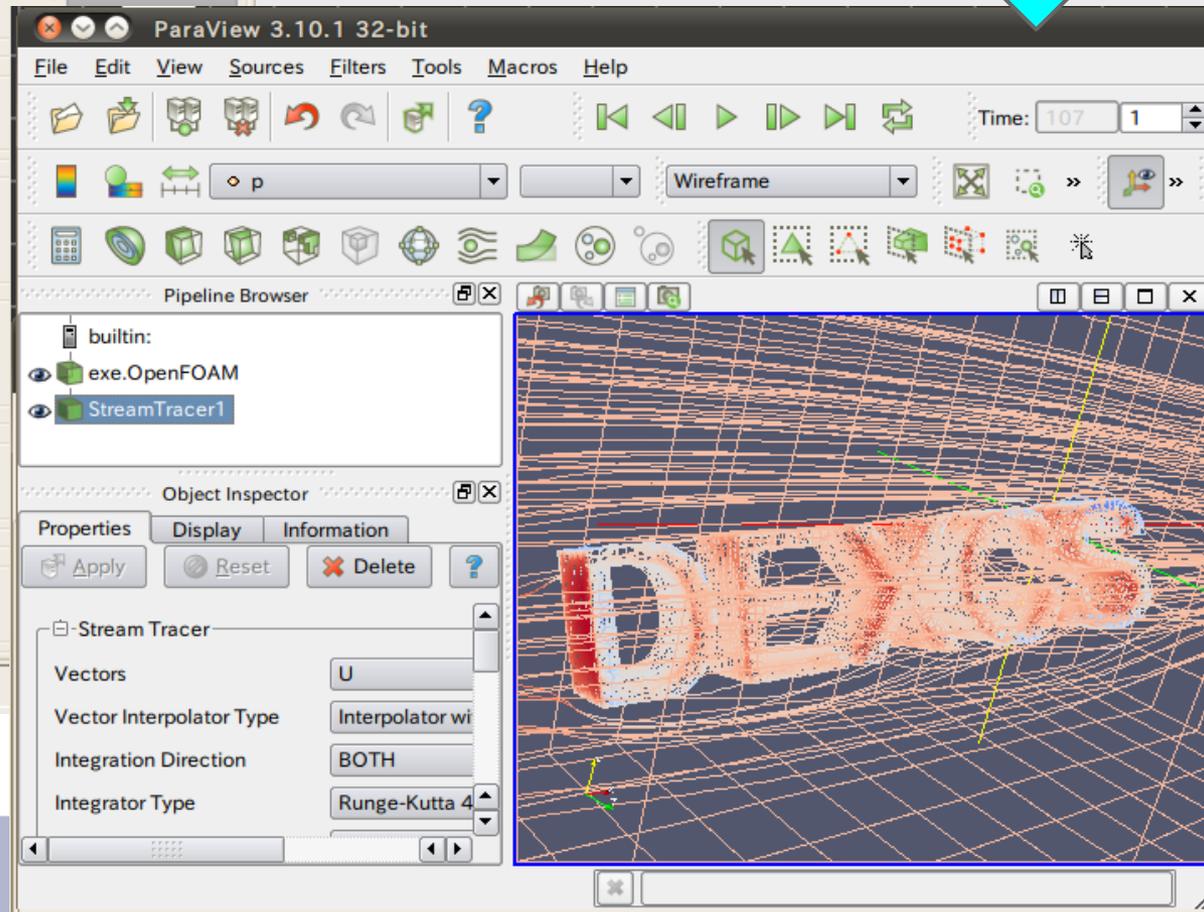
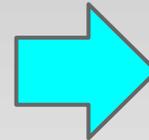
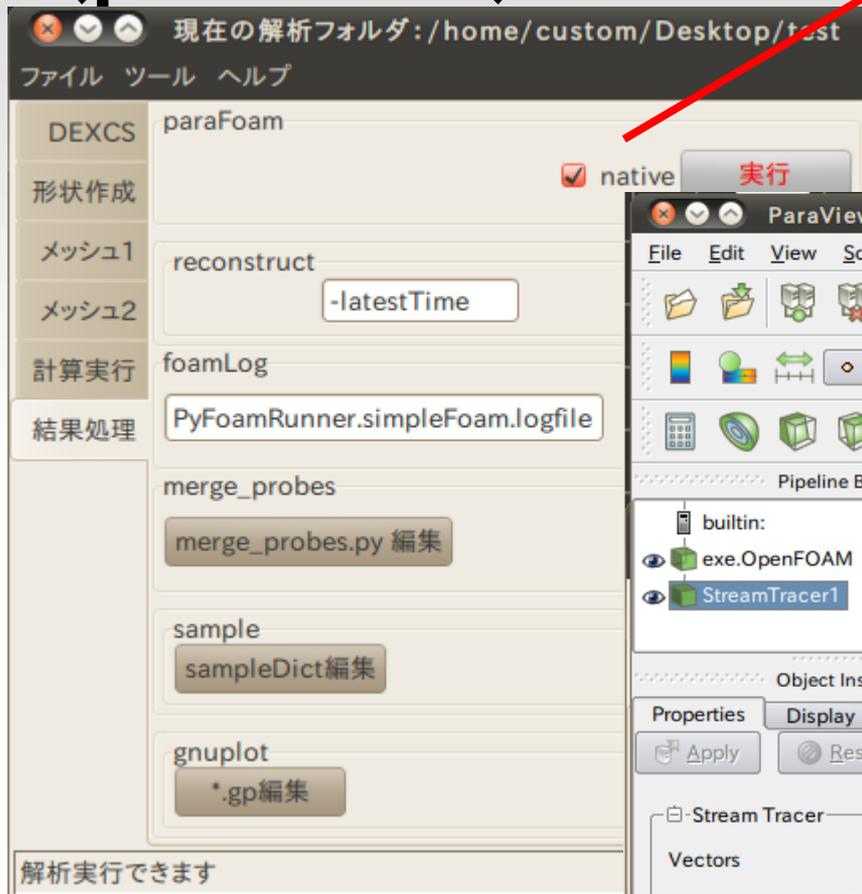
test

- exe
 - 0
 - Decomposer.analy
 - Gnuplotting.analy
 - constant
 - dynamicCode
 - forceCoeffs
 - forces
 - patchExpression
 - patchMassFlows
 - probes
 - processor0
 - processor1
 - system
- Decomposer.logfile

processor0

- 0
- 50
- 100
- 150
- 170
- constant

流れ場の可視化 (paraFoam)



計算結果の後処理



計算ログ解析
⇒残差プロット
用データの作成

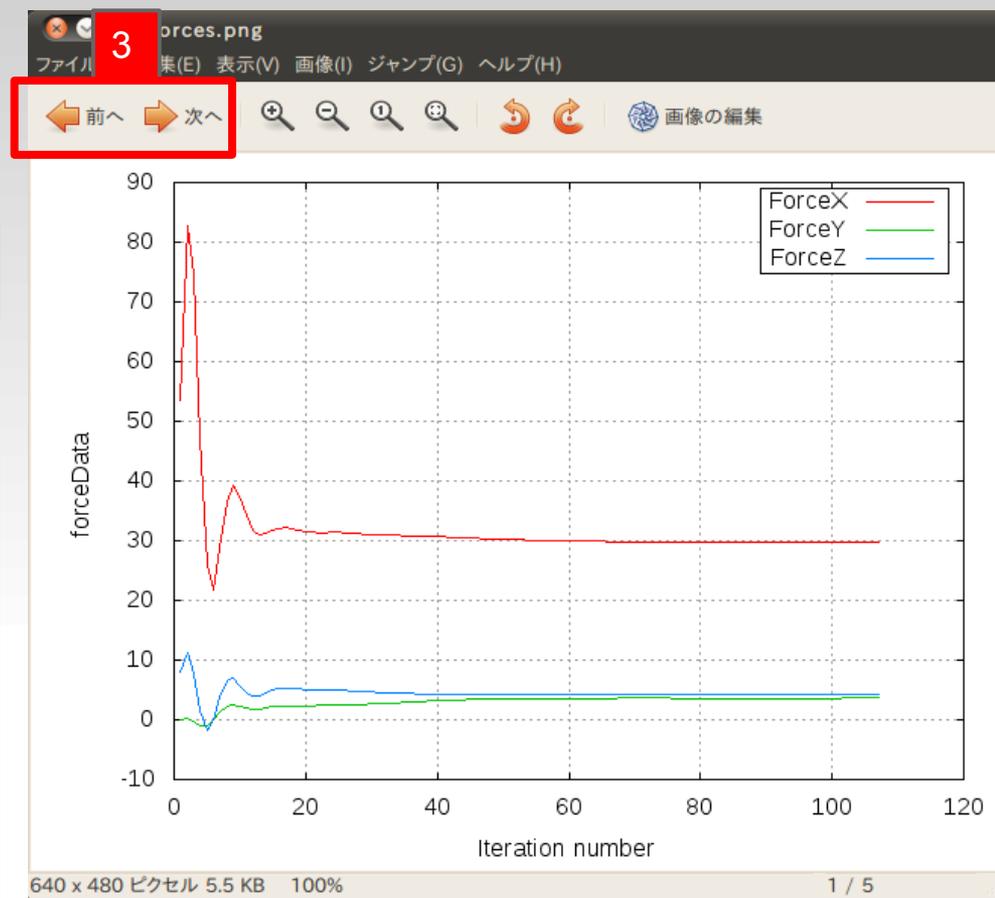


指定サンプル線
(面)上で、プロット
用データの作成

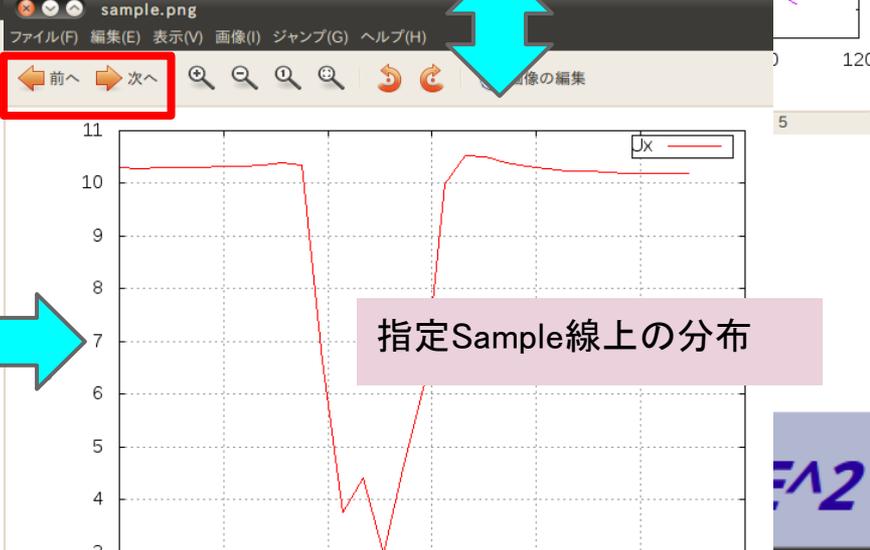
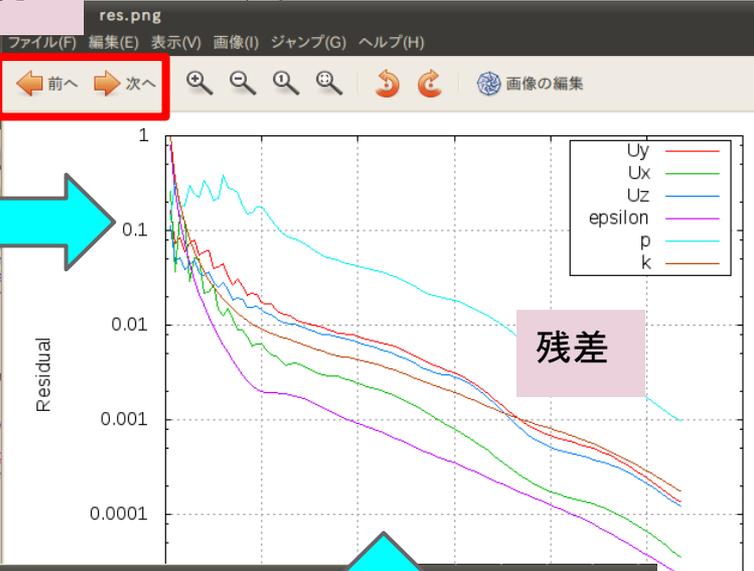
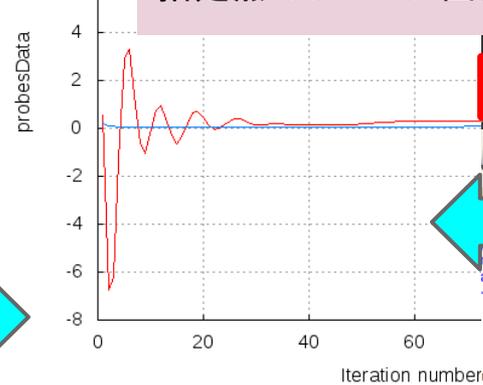
プロット図の 作成と表示



グラフスケール
など変更したい
場合に使用



プロット図 の表示



終了



以上、お疲れ様でした



ランチャー演習の振り返り

狙い: CAEの要諦
を簡単体験

実演習に
必要な基礎・
ツール知識

- モデル変更
 - 風洞
 - 対象物体
- メッシュ変更
 - blockMesh
 - snappyHexMesh
- 計算方法の変更
 - 境界条件
 - 流体特性
 - 計算、解法スキーム

- blender
/ Swiftツール

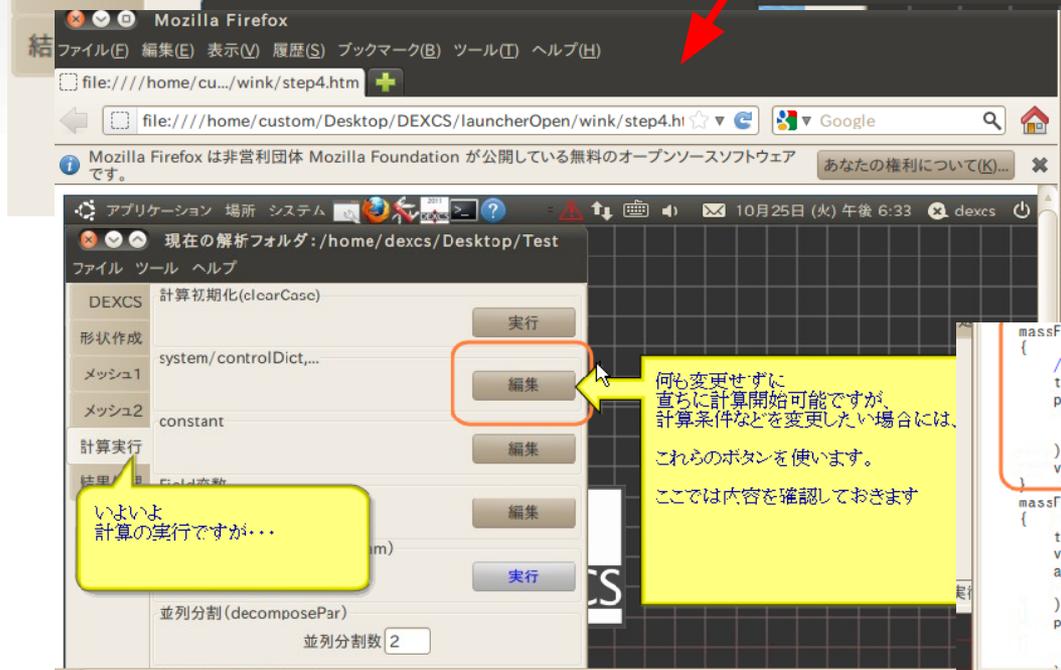
体験2-1で説明

- OpenFOAM基礎
(Winkチュートリアル
で補完)

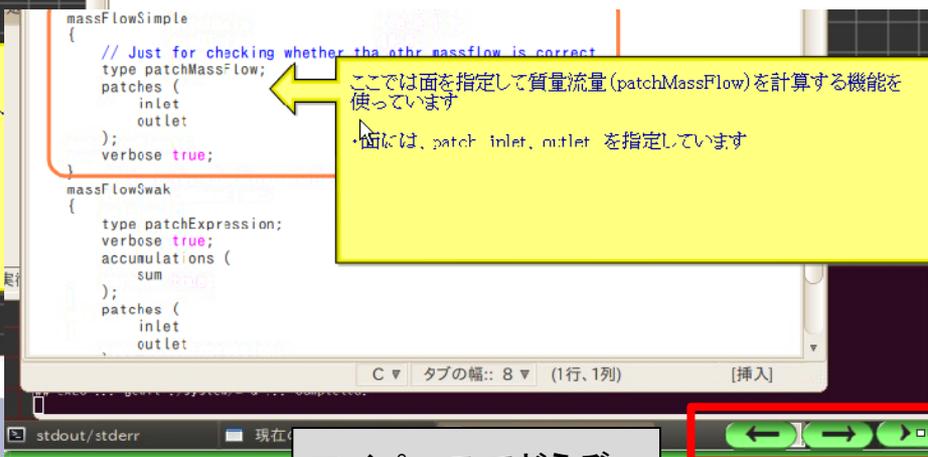
winkチュートリアルについて



公開版ベースのチュートリアル
(本実習のやり方とは異なっている)



ボタンの意味や、パラメタファイルの中味に関する説明があります。



マイペースでどうぞ

winkチュートリアルについて

DEXCS2011 for OpenFOAM(R) 2011/12

ヘルプ

- ランチャの使い方
- OpenFoam UsersGuide(V-2.0)
- 同上(和訳版, V-2.1.0_beta)
- OpenFoam ProgrammersGuide(V-2.0)
- 同上(和訳版, V-2.1.0)
- Blenderの使い方
- Paraviewの使い方
- DEXCS HP
- DEXCS2011 for OpenFOAM(R)について
- フラッシュプレーヤーの変更方法**

変更前

ヘルプファイル中、現在ご覧になっているような動画チュートリアルですが、これを表示するのにDEXCSの標準はgnashというフリーのソフトを使用しています。

しかし正直、あまり表示品質がよろしくありません。一方、Adobe社のフラッシュプレーヤーがお勧めですが、これもフリー製品ですが、ライセンス上、再配布できないものなので、ユーザー自身の手でインストールして使っていただく必要があります。

ここでは、そのやり方について説明します。

インターネットに接続できていることが前提となっていることをご承知おきください

DEXCS2011 for OpenFOAM(R) 2011/12

ヘルプ

- ランチャの使い方
- OpenFoam UsersGuide(V-2.0)
- 同上(和訳版, V-2.1.0_beta)
- OpenFoam ProgrammersGuide(V-2.0)
- 同上(和訳版, V-2.1.0)
- Blenderの使い方
- Paraviewの使い方
- DEXCS HP
- DEXCS2011 for OpenFOAM(R)について
- フラッシュプレーヤーの変更方法

変更後

ヘルプファイル中、現在ご覧になっているような動画チュートリアルですが、これを表示するのにDEXCSの標準はgnashというフリーのソフトを使用しています。

しかし正直、あまり表示品質がよろしくありません。一方、Adobe社のフラッシュプレーヤーがお勧めですが、これもフリー製品ですが、ライセンス上、再配布できないものなので、ユーザー自身の手でインストールして使っていただく必要があります。

ここでは、そのやり方について説明します。

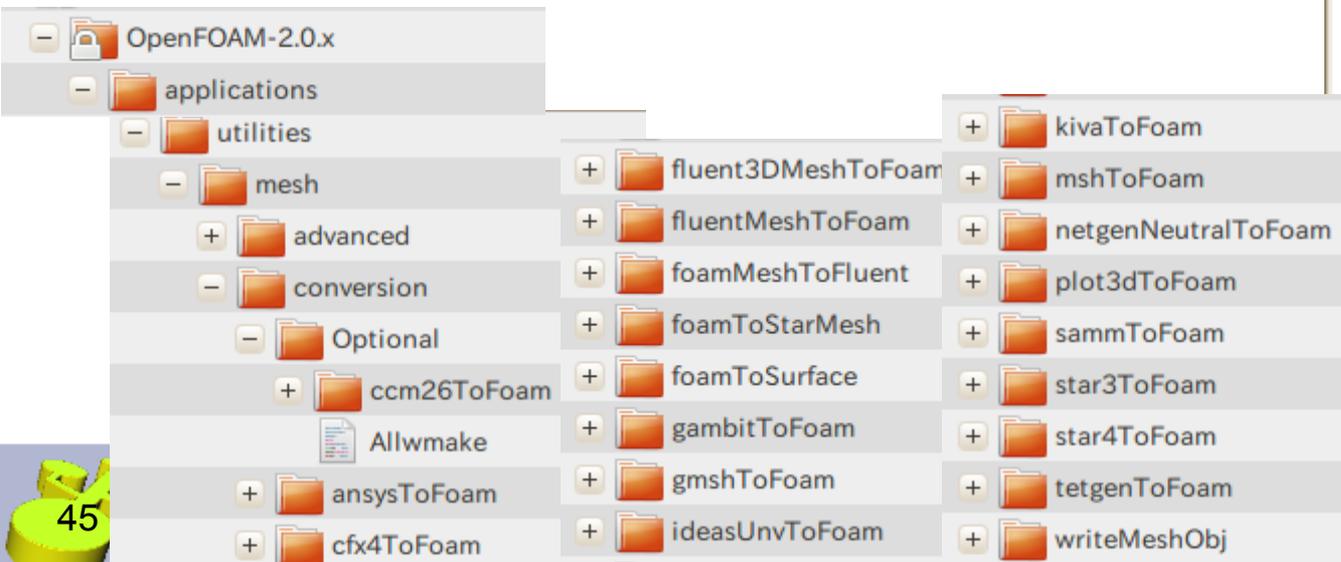
フラッシュプレーヤーを
変更して使用することを推奨。
(ライセンス上の問題)
但し、変更方法として上記ヘル
プの方法が現在では使えない
⇒後述

形状、メッシュデータについて

snappy
HexMesh
作成法は
同一

- 簡単な形状・・・blenderで作成(実習予定)
- 複雑な形状・・・通常利用の3D-CADで作成
 - ⇒STL形式でエクスポート
 - ⇒blenderへインポート
- 他のソフトで作成したメッシュが存在
 - ⇒###ToFoam ユーティリティを利用

+ GNU Free Documentation License	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
+ 第3章 アプリケーションとライブラリ	U-73
+ 第4章 OpenFOAMのケース	U-10
- 第5章 メッシュの生成と変換	U-12
+ 5.1 メッシュの記法	U-12
+ 5.2 境界	U-13
+ 5.3 blockMeshユーティリティを使...	U-13
+ 5.4 snappyHexMeshユーティリティ...	U-14
- 5.5 メッシュの変換	U-15
5.5.1 fluentMeshToFoam	U-15
5.5.2 starToFoam	U-15
5.5.3 gambitToFoam	U-16
5.5.4 ideasToFoam	U-16
5.5.5 cfx4ToFoam	U-16
+ 5.6 異なるジオメトリ間のフィールドマ...	U-16
第6章 後処理	U-16
第7章 モデルと物性値	U-18
索引	U-18



OpenFOAMの実践的活用法

(標準ソルバーが使える場合)

1. メッシュ作成は前頁記載の方法
 - a. Swiftツール ⇒ snappyHexMesh
 - b. (他のソフトで作成) ⇒ ###ToFoam
2. 標準チュートリアルケースを精査し、自分が解きたい現象・モデルに近いものを探し出す。
3. 上記チュートリアルケースのメッシュを自前で作成したメッシュに置き換える。
4. モデルパラメタの細部詳細を整合

体験
2-1
で実践

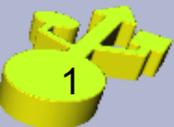
体験
2-2
で実践

memo



DEXCS for OpenFOAM®の概要

オープンCAEコンサルタント OCSE²
代表 野村 悦治



COMPUTATIONAL MECHANICS

計算力学部門ニュースレター No.46

April, 2011



オープンCAE：DEXCSの可能性

柴田良一 岐阜工業高等専門学校 建築学科 (左)
野村悦治 株式会社デンソー 技術管理部 (右)

7. DEXCS for OpenFOAMの解析機能

OpenFOAMは、日本機械学会誌2008年12月号にてトピックス紹介されたように、近年、劇的にユーザー数を増やしているオープンソースCFD統合ソフトウェアです。DEXCS for OpenFOAMでは、**手軽に本格的なCFDを体験**できることを目標に、ボタンを順番に押していくだけで、**図4**に示すような複雑な物体周りの流れ解析(定常非圧縮性乱流計算)が可能です。

これにはOpenFOAMでVer.1.5以降、SnappyHexMesh という六面体をベースにした自動メッシュ作成ツールが使えるようになったことが大きく寄与しています。但し、この設定ファイルは形状データに応じて書き換えが必要になるので、この**主要パラメタをGUIパネルで簡単に編集できるようにしたツールも組み込む**ことにしました。これでSTLなど3次元形状のポリゴン系のモデルデータさえあれば、GUI操作で簡単な仮想風洞試験ができることを実感してもらえそうです。

動画チュートリアルも同梱しており、DEXCSランチャーの操作方法を覚えるかたわらで、OpenFOAMの計算に必要なパラメータファイルの概要や構成についても学習できるようになっています。**これからOpenFOAMの本格的な勉強や活用を考える際の足掛かりとして利用**してもらえればよく、本格的な活用では、ランチャーをカスタマイズして利用するのがベストとして、ランチャーのカスタマイズツールも同梱しました。なお、最新版ではマルチコア対応の並列計算が可能です、64bit版もあります。

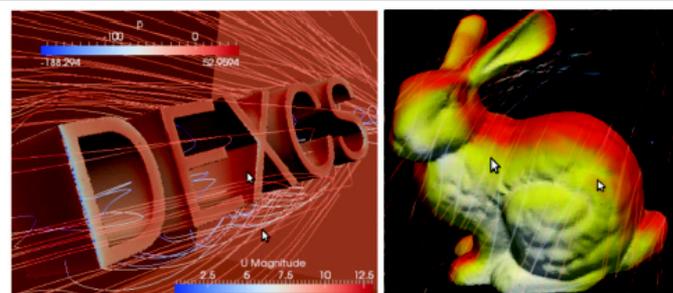


図4 DEXCS for OpenFOAMの解析例

2011年1月1日(土曜日)

新年明けましておめでとうございます

カテゴリ: General CAE OpenFOAM DEXCS ParaView - et @ 05時42分42秒
今年もオープンCAEで頑張ります

2010年12月30日(木曜日)

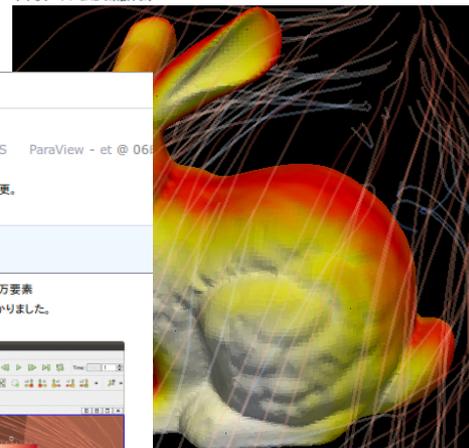
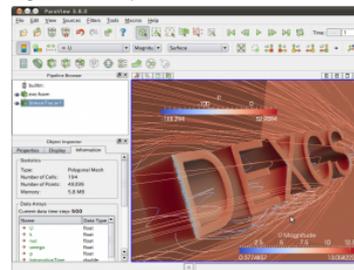
新マシンでは実用計算も十分可能

カテゴリ: General CAE OpenFOAM DEXCS ParaView - et @ 06時42分42秒

DEXCS2010 for OpenFOAM(R) の標準問題を使用。
基礎メッシュ分割 (20 x 10 x 10) → (100 x 50 x 50) に変更。
snappyHexMeshDictの変更点

```
maxLocalCells 5000000;
maxGlobalCells 20000000;
```

snappyHexMeshの実行は4並列で計算(約40分) → 約1500万要素
simpleFOAMで500回のイテレーションに、2並列で約8時間かかりました。
乱流モデルはk-omegaSSTを使用(k-epsilonだと発散)



DEXCS公開版



Salome-Meca →

構造解析 Adventure
線形弾性

弾塑性、大変形

完
(終了)



Ver-1.4.1
'09/2

流体解析

OpenFOAM

Ver-1.5.x
'09/7

Ver-1.7.x
'10/10

Ver-2.0.x
'10/12

DEXCSの目標

オープンCAE: **DEXCS** (デックス)



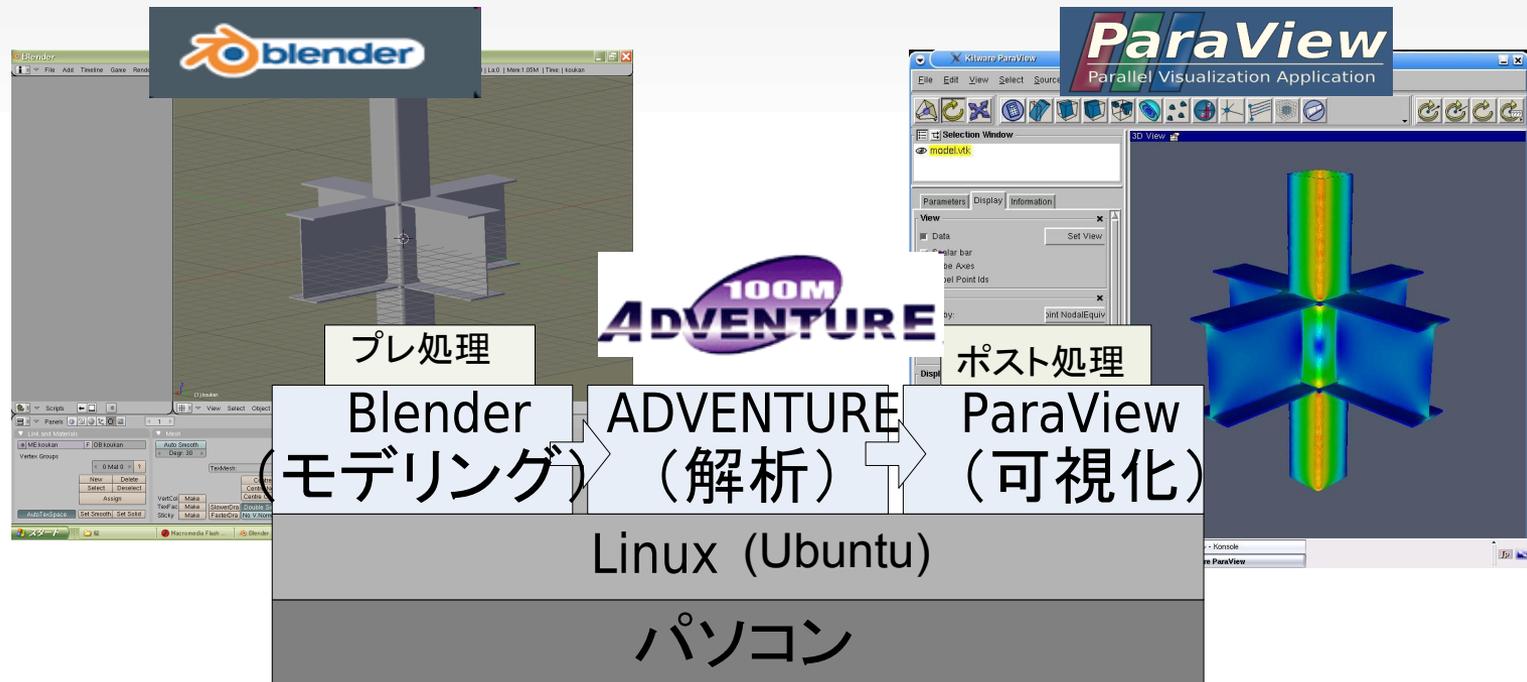
拡張性を持つ設計支援用解析システム

Digital **E**ngineering on e**X**tensible **C**omputing **S**ystem

- CD起動や仮想PC上に**オールインワン**のCAEを実現
様々な起動方法に対応する
- 数値解析を中心に高機能のプリポストを備える
構造解析や流体解析に対応する
- 教育研修を対象のCAEとして基本機能を実現する
大規模弾性解析の並列処理に対応する
- 企業内実務での適用・拡張も可能

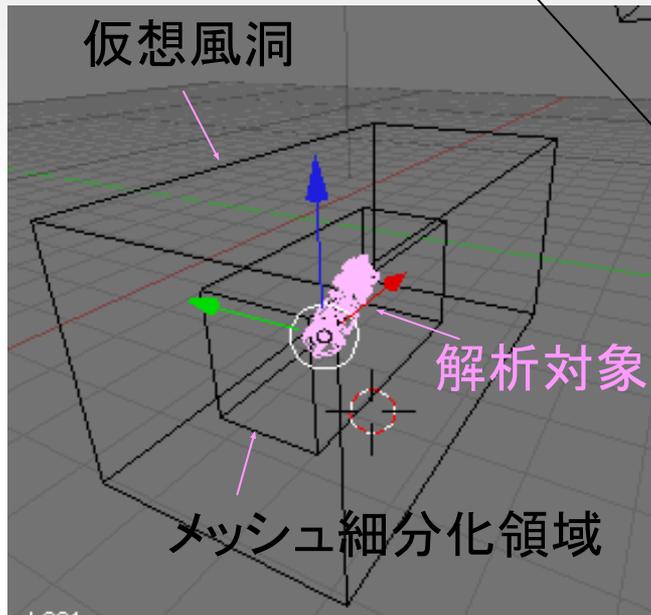
DEXCSの構成

Linux(Ubuntu)上に、構造解析にADVENTUREを活用し
プリポスト機能を統合したCAE環境を構築する。
これを、**CD起動形式**や**仮想PC起動形式**で公開



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
($k-\epsilon$ 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)

DEXCS2011-OpenFOAMの嬉しさ

DEXCS2009-OpenFOAMの嬉しさを踏襲

上記をブラッシュアップ

- 並列計算可能、64 bit版もリリース
- 操作性改善、結果処理サンプルの充実
- ライブDVDでの動作を前提としたテスト、操作説明

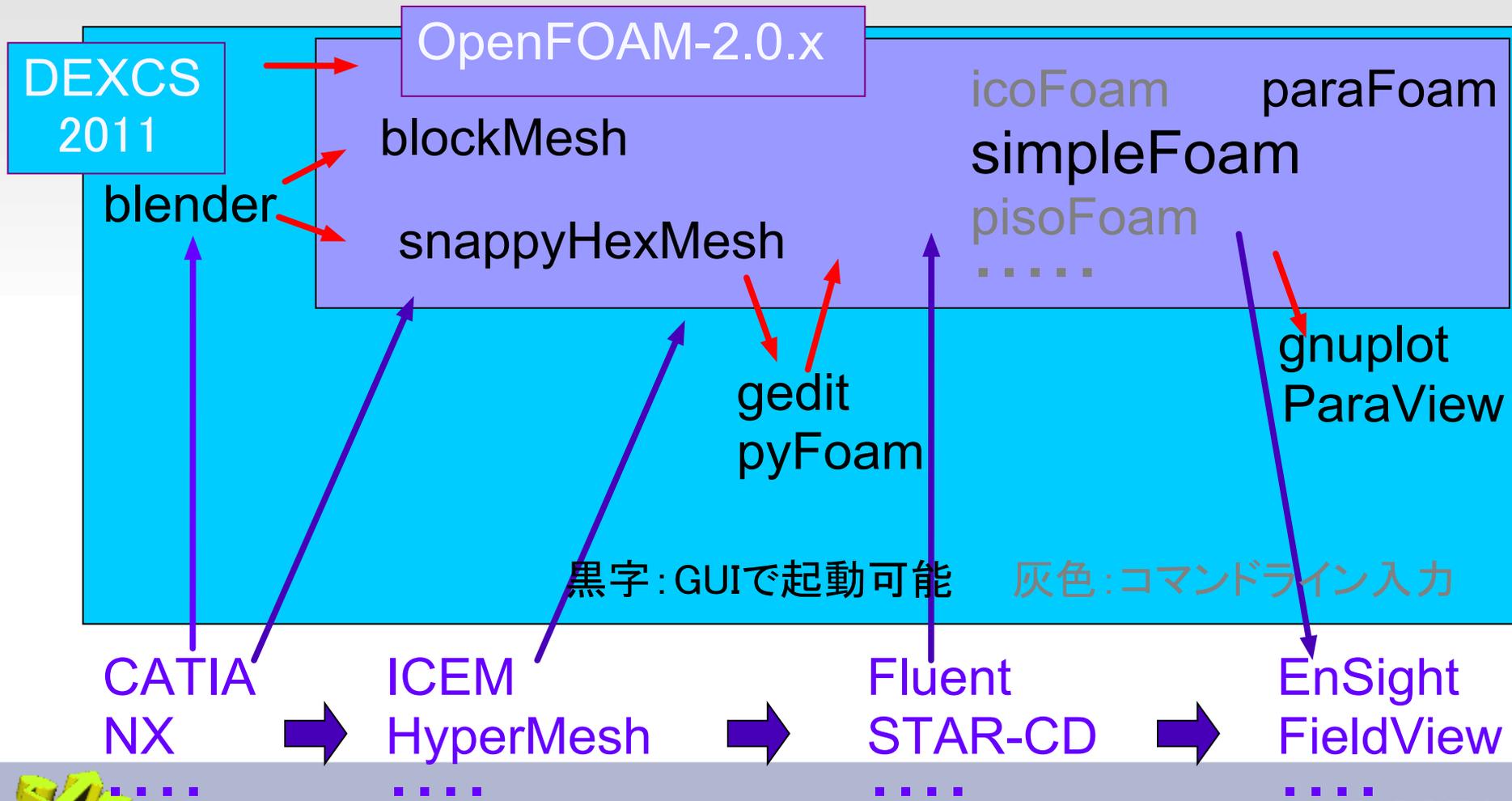
新機能

- OpenFOAM 中級者を想定した簡単GUI操作



解析の流れとプログラム

形状作成 メッシュ作成 境界条件 計算実行 後処理



DEXCS2011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) **オール・イン・ワン**組み込み

pyFoamPlotRunner.py, pyFoamClearCase.py

pyFoamPotentialFoam.py, pyFoamCreateBoundaryPatches.py

pyFoamDisplayBlockMesh.py

⇒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, bcFilesCoverter.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

+ 標準チュートリアルベースの簡単メッシュ変更

DEXCS2011 for OpenFOAM(R) リリースノートより

<http://dexcs.gifu-nct.ac.jp/pukiwiki/index.php?DEXCS-OpenFOAM-ReleaseNote>

同梱プログラム(ベースOSはubuntu-10.04 LTS)†

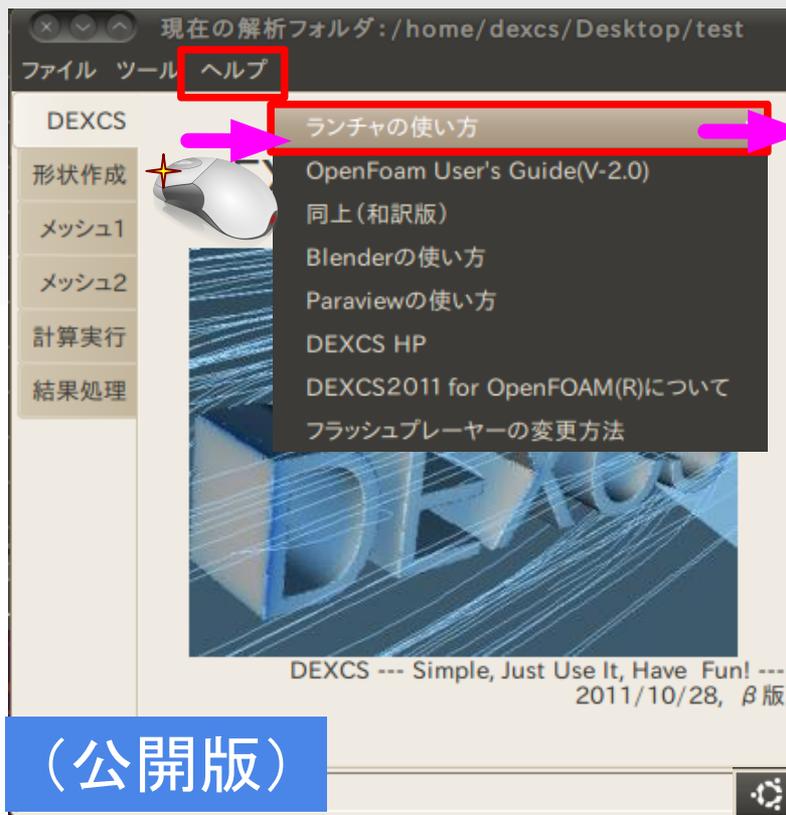
- OpenFOAM-2.0.x (2011年8月時点入手版)(*注1)(<http://www.openfoam.com/>)
- (*注1)本リリース時点(2011年12月)で最新版が入手可能ですが、最新版だと一部のプログラムに動作しないものがある為、あえて少々古いものを同梱しました。
- paraview-3.10.1 (<http://www.paraview.org/>) 並列可視化, Pythonバッチ機能有効版
- PyFoam-0.5.6 (http://openfoamwiki.net/index.php/Contrib_PyFoam)
- swak4Foam (<http://openfoamwiki.net/index.php/Contrib/swak4Foam>)
- blender-2.49 (<http://www.blender.org/>)
- remastersys-2.0.17-1 (<http://www.geekconnection.org/remastersys/>)
- wx-glade(python-wxglade-0.6.3-0.1ubuntu3) (<http://wxglade.sourceforge.net/>)
- SPE (Stani's Python Editor) 0.8.4.h (<http://pythonide.blogspot.com/2008/02/spe-084c-python-ide-editor-released.html>) emacs23-lucid
- gnuplot-4.2.6-1
- DEXCSオリジナルツールとして
 - DEXCSランチャー(dexcs.py)
 - DEXCS十徳ナイフ(dexcsSWAK.py)
 - snappyHexMeshDict簡易エディタ(snappyDictExporter.py) <http://mogura7.zenno.info/~et/xoops/modules/wordpress/index.php?p=345>)
 - 境界条件名の一括変更ツール(bcFilesConverter.py)<http://mogura7.zenno.info/~et/xoops/modules/wordpress/index.php?p=348>)
 - 時刻歴データのプロットツール(timeLinePlotter.py)<http://mogura7.zenno.info/~et/xoops/modules/wordpress/index.php?p=390>)
 - ソースコード検索ツール(dexcsTextSearcher.py)<http://mogura7.zenno.info/~et/xoops/modules/wordpress/index.php?p=413>)

推奨利用方法1

OpenFOAM初学者向け



Linux (ubuntu) のコマンド入力が不得手な人

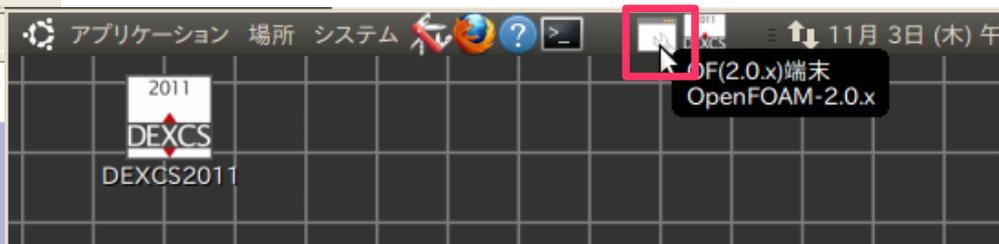


1 ランチャーの動作確認と全体概要の理解

2 OpenFOAMの動作とファイル構成の概要理解

(公開版)

コマンド入力に不便を感じない人

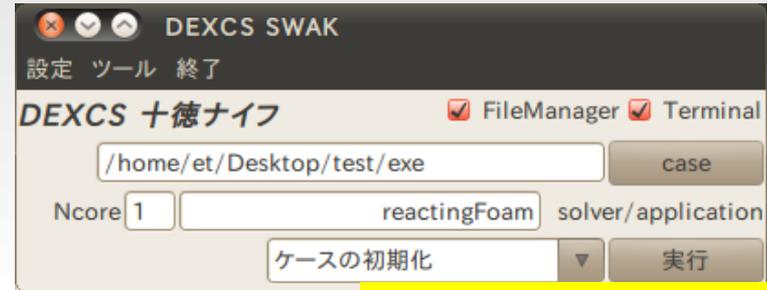


推奨利用方法2

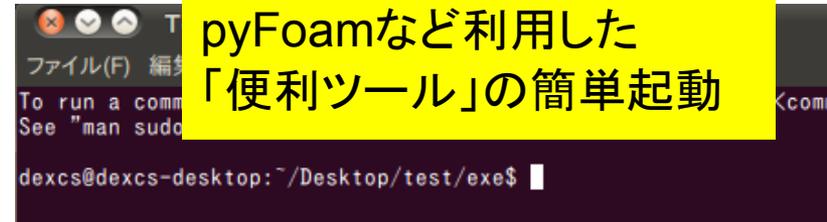
OpenFOAM利用経験者

OpenFOAMの動作と
ファイル構成を理解している

Linux(ubuntu)のコマンド入力が不得手な人



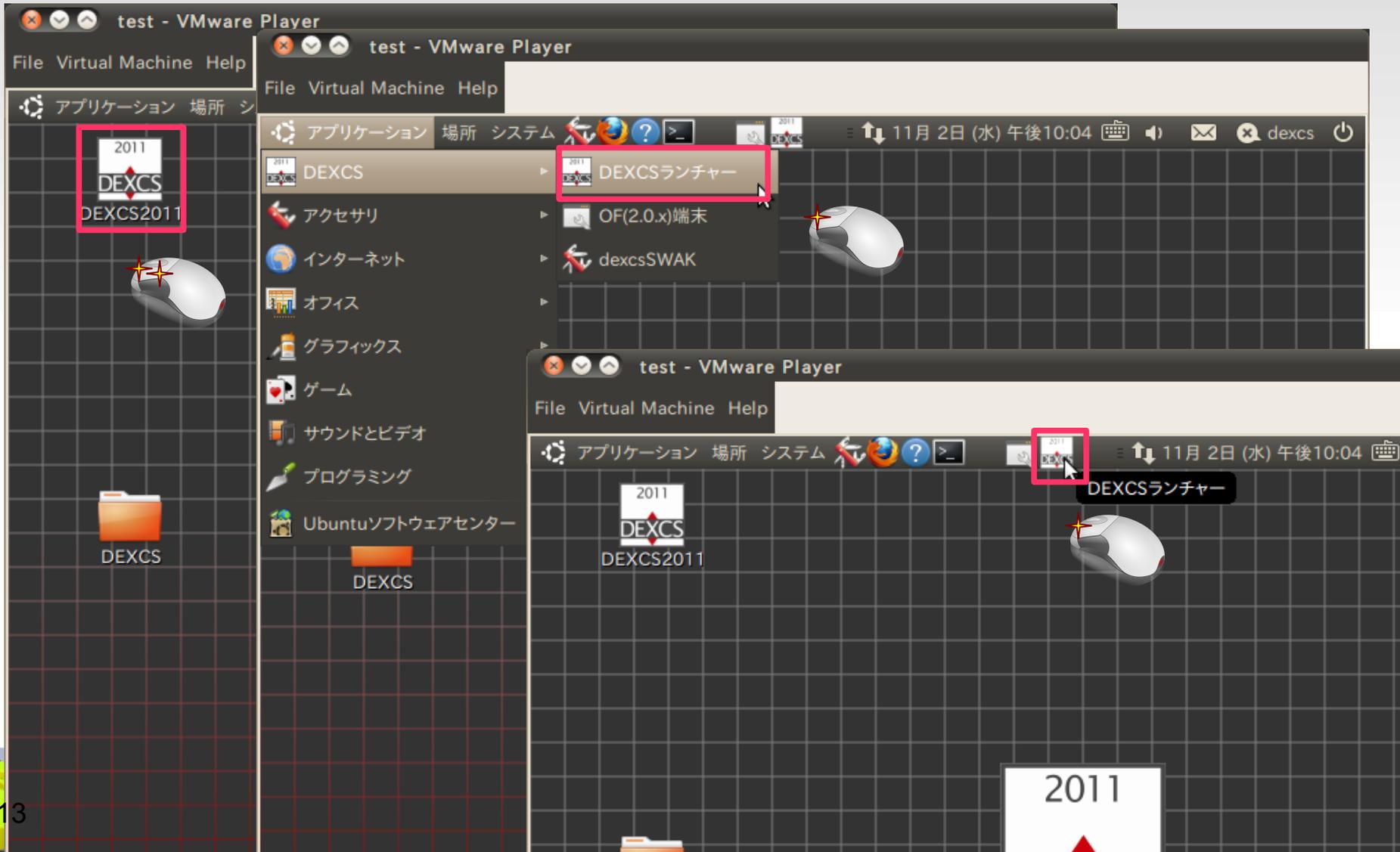
caseフォルダを対象に
OpenFOAM専用端末や
pyFoamなど利用した
「便利ツール」の簡単起動



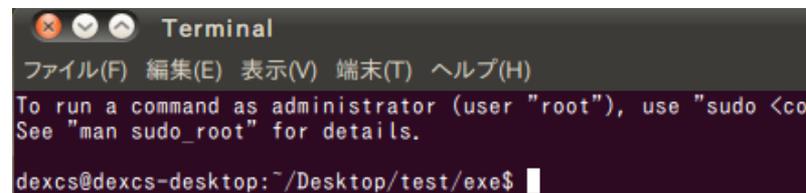
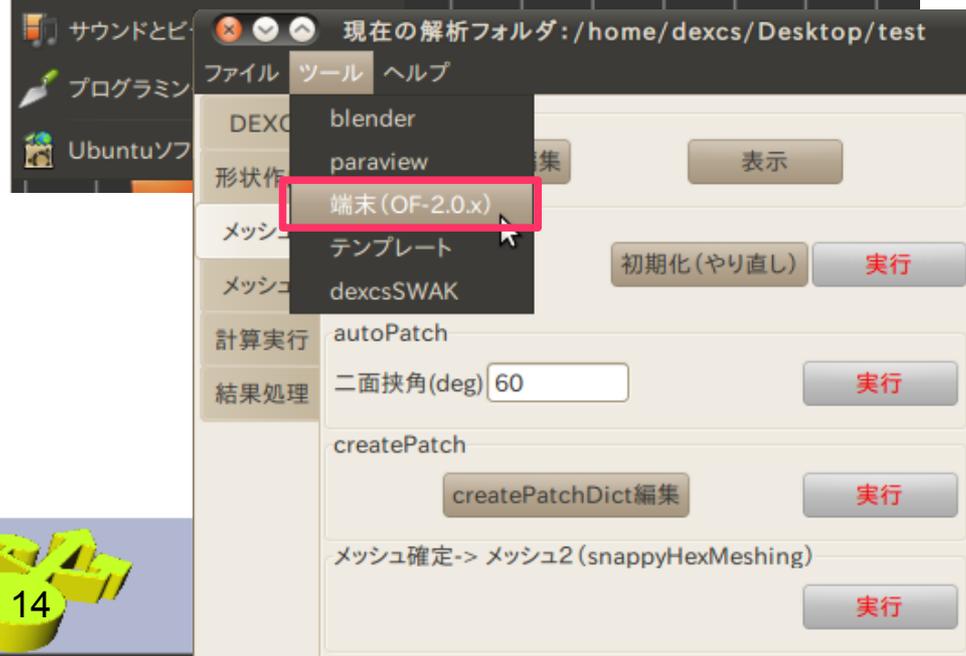
コマンド入りに不便を感じない人
pyFoam などのコマンドも熟知している人



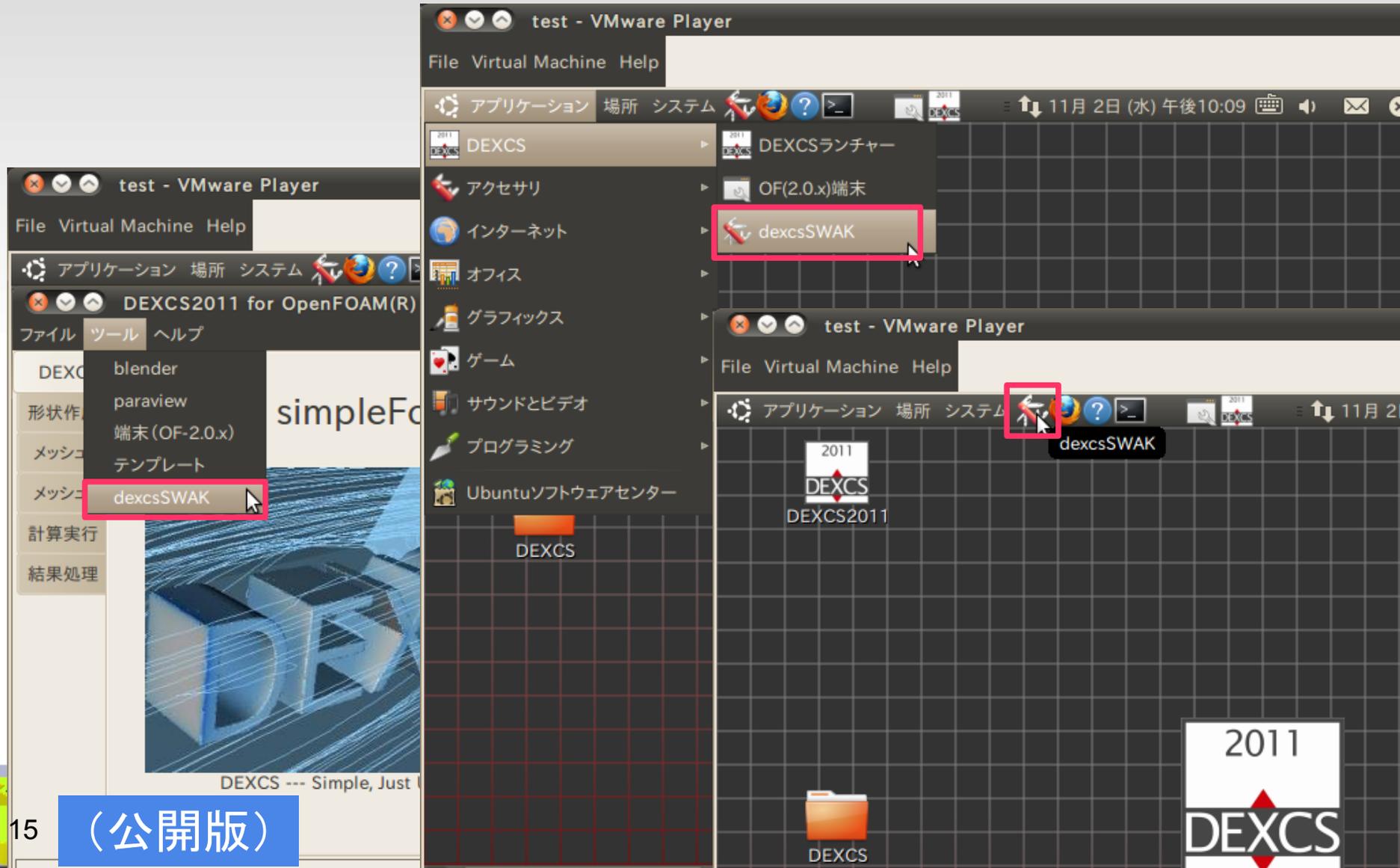
ランチャー起動方法(3通り)



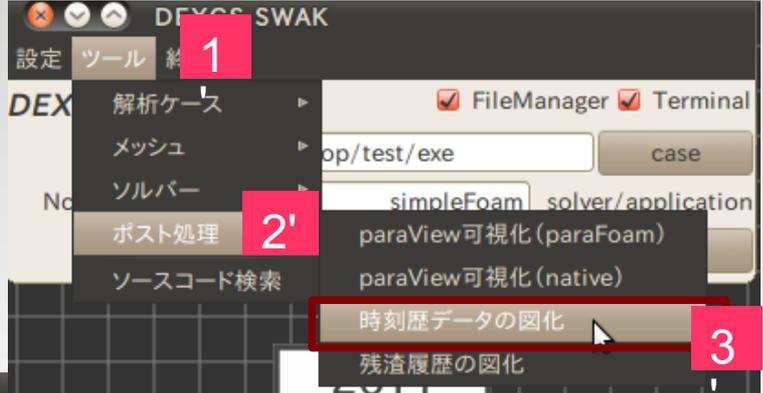
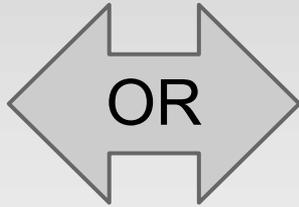
OpenFOAM専用端末の起動方法(4通り)



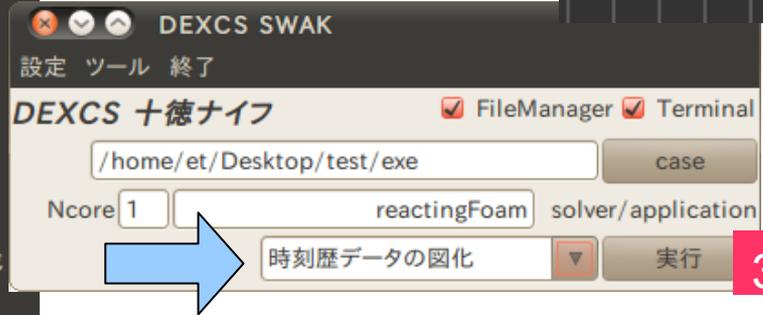
十徳ナイフ起動方法(3通り)



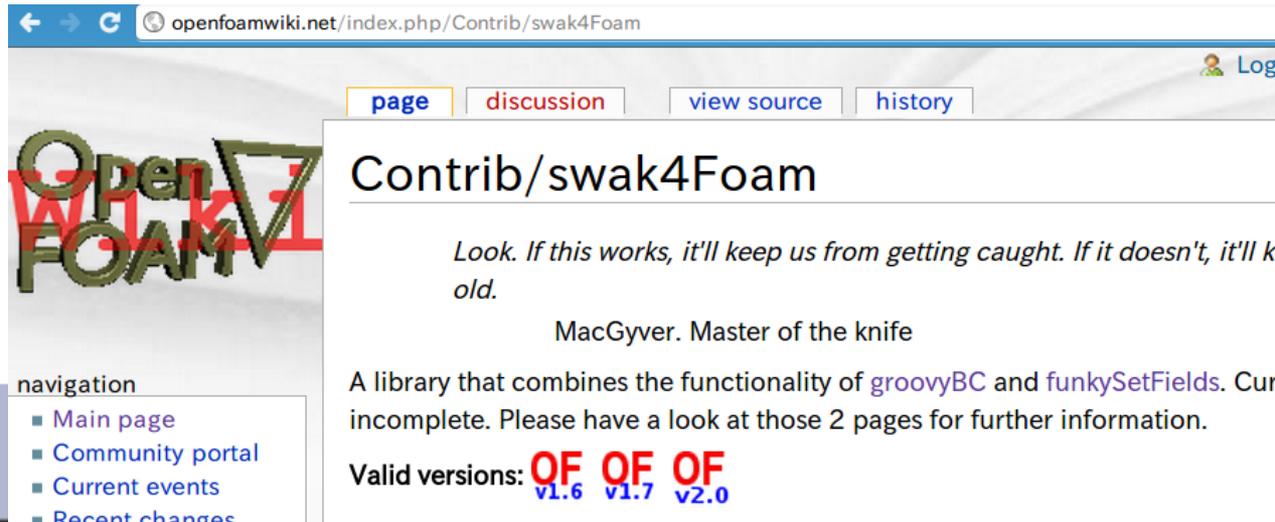
dexcsSWAK(Swiss Army Knife)



- ケースの初期化 1
- テンプレートケースの選択・変更
- blockMeshの表示・編集
- blockMeshの実行
- patch名の一括整合
- patch名の一括変更
- featureデータの作成
- snappyHexDictの簡易作成
- snappyHexMeshの実行
- 初期条件の一括参照・変更
- constデータの一括参照・変更
- systemデータの一括参照・変更
- ポテンシャル流れ場作成
- ソルバーの実行
- ソルバーの実行(プロット付)
- ソルバーの停止
- paraView可視化 (paraFoam)
- paraView可視化 (native) 2
- 時刻歴データの図化
- 残渣履歴の図化
- ソースコード検索



caseフォルダを対象に
pyFoamなど利用した
「便利ツール」の簡単起動



DEXCS timeLinePlotter



1



2



3

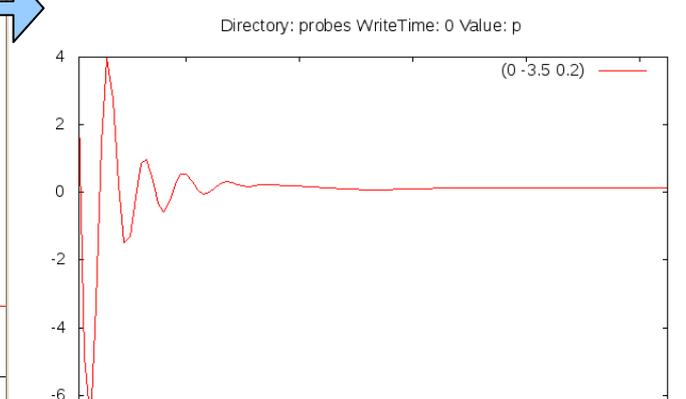
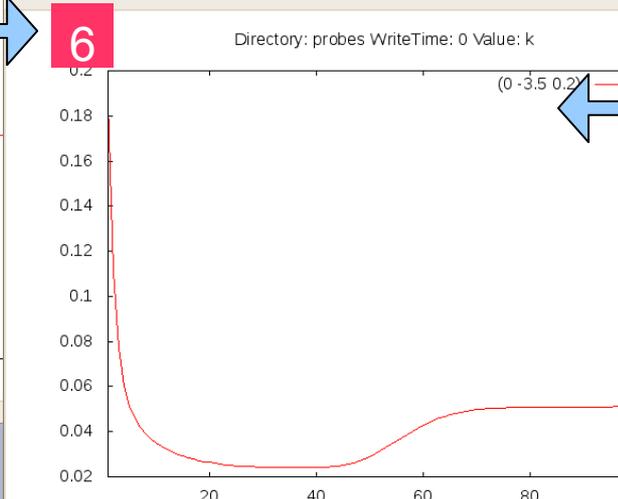
4



5



6



snappyHexMesh Feature Edge

次期バージョンでは廃止予定

ファイル ツー

DEXCS surfaceFeatureExtract

形状作成 includedAngle 150 選択実行 1

メッシュ1 snappyHexMeshDict

メッシュ2 形状選択 Dict編集 初期化(やり直し)

計算実行 並列分割 2

結果処理 並列分割数 2

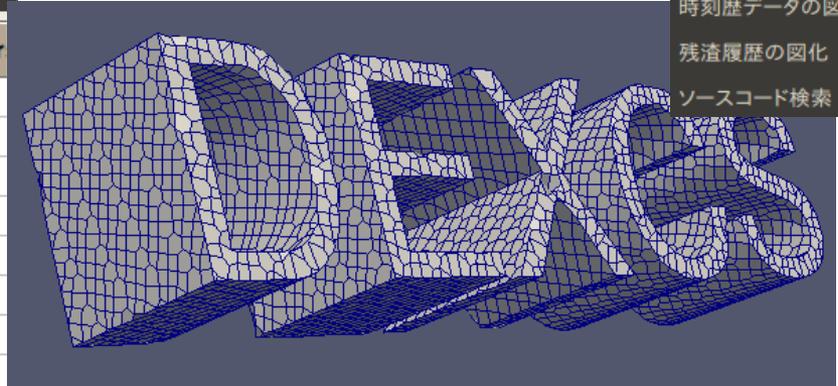
計算実行

snappyHexMeshDict+ Exporter

メッシュ	shapeName	Type	Apply	Smin	Smax	Region	Layer
	box1.stl	Region	<input type="checkbox"/>	0	0	1	
	dexcs.stl	Surface	<input type="checkbox"/>	1	2	0	
	dexcs.eMesh	Feature	<input type="checkbox"/>	0	0	0	
	inlet	Patch	<input type="checkbox"/>	0	0	0	
	outlet	Patch	<input type="checkbox"/>	0	0	0	
	wall	Patch	<input type="checkbox"/>	0	0	0	
			<input type="checkbox"/>	0	0	0	

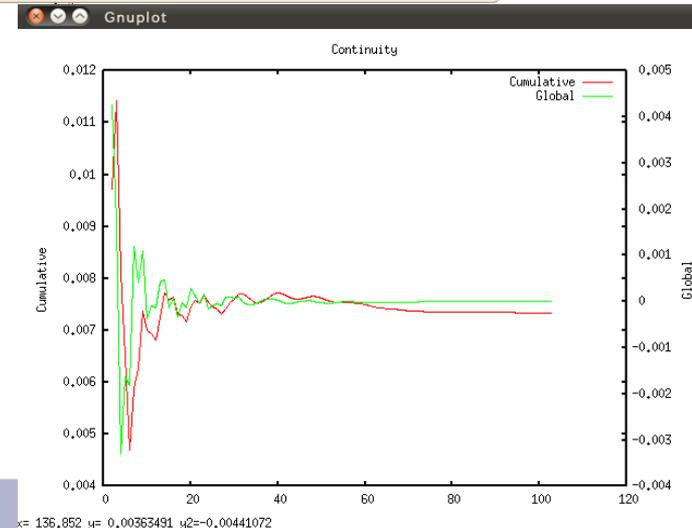
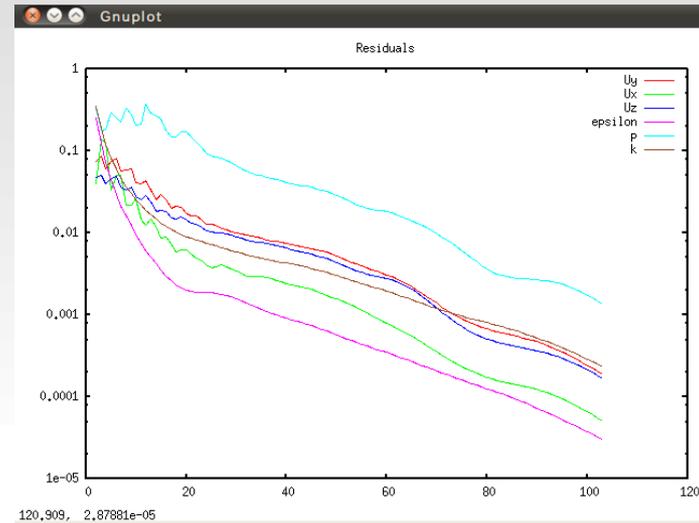
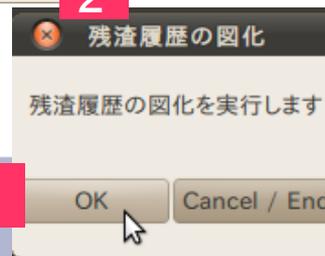
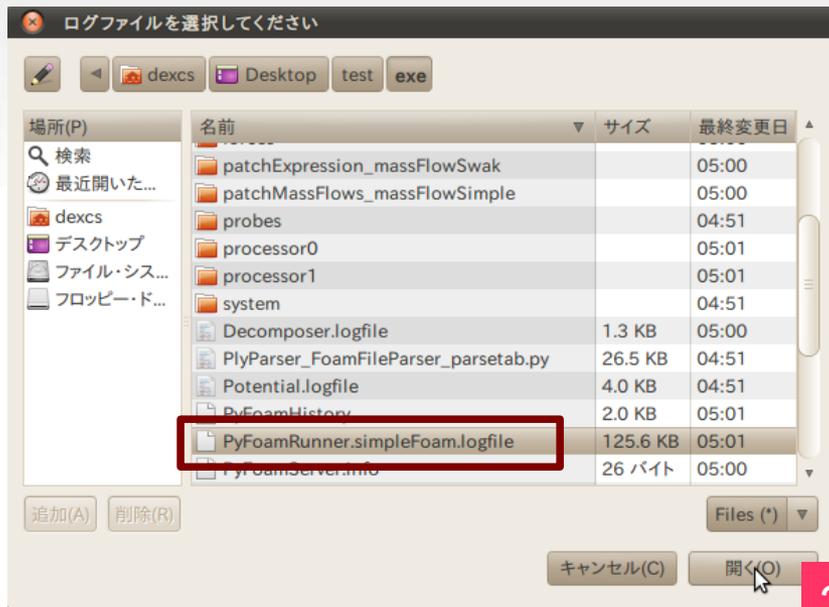
解析実行できま

Export... Exit



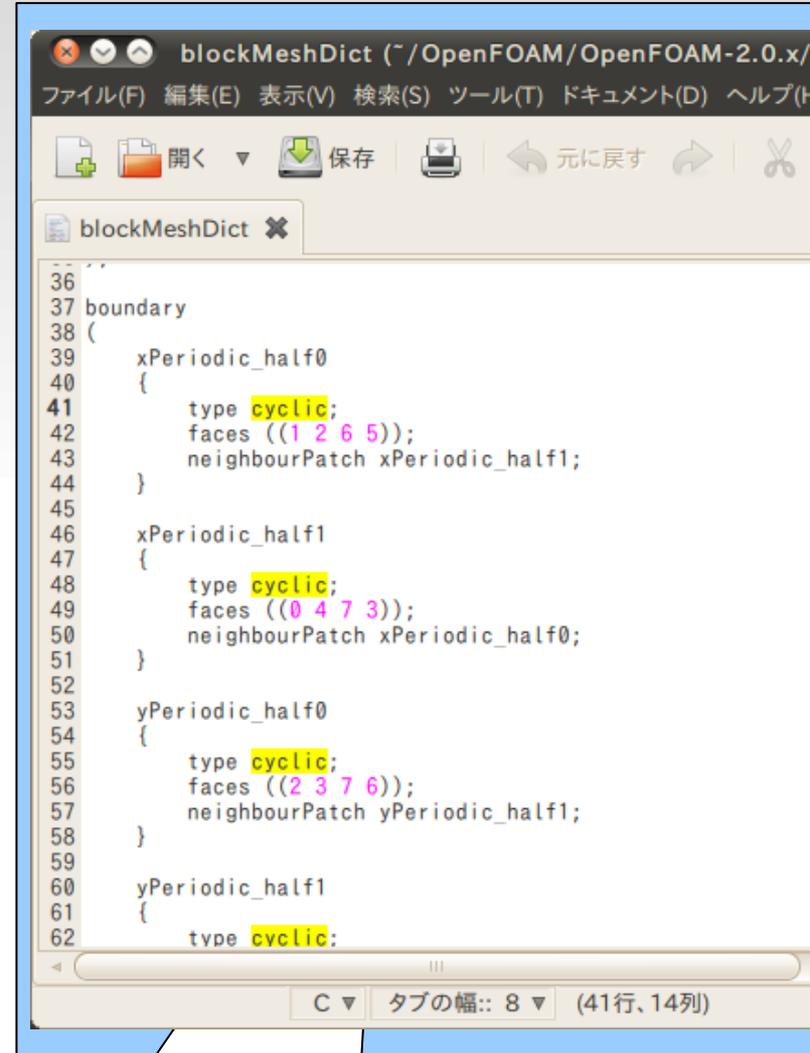
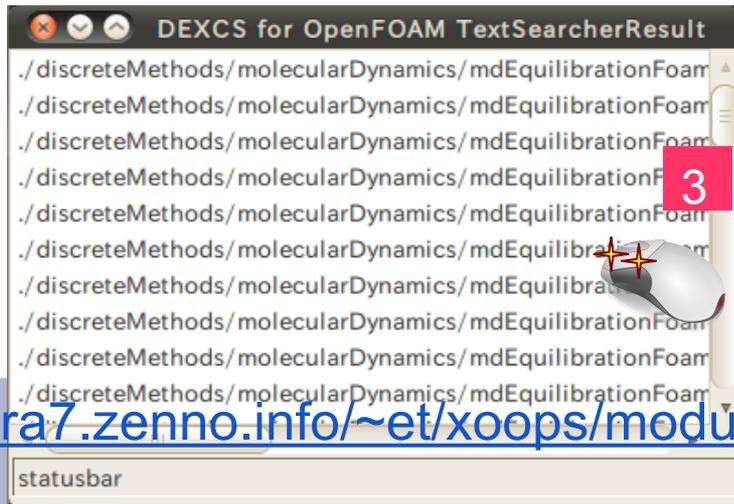
- ケースの初期化
- テンプレートケースの選択・変更
- blockMeshの表示・編集
- blockMeshの実行
- patch名の一括整合
- patch名の一括変更
- featureデータの作成
- snappyHexDictの簡易作成
- snappyHexMeshの実行
- 初期条件の一括参照・変更
- constデータの一括参照・変更
- systemデータの一括参照・変更
- ポテンシャル流れ場作成
- ソルバーの実行
- ソルバーの実行(プロット付)
- ソルバーの停止
- paraView可視化(paraFoam)
- paraView可視化(native)
- 時刻歴データの図化
- 残渣履歴の図化
- ソースコード検索

残差履歴の図化



ソースコード検索

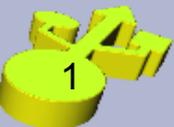
十徳ナイフのその他の機能については体験2-2にて説明します



OpenFOAM®の基礎知識概要

知っておきたいキーワードと、それらを調べる際の手掛かりについて説明

オープンCAEコンサルタント OCSE²
代表 野村 悦治



基礎知識目次

■ ファイル構成	3
■ マニュアル	5
■ linux環境	7
■ 標準ソルバー	12
■ ユーティリティ	14
■ ライブラリー	15
■ チュートリアルケース	16
■ ケースフォルダ	20
■ ケース構成ファイル	21
○ メッシュ、モデルパラメタ、計算パラメタなど	
■ 境界条件	29
■ メッシュ作成法	31
○ blockMesh / snappyHexMesh	
■ 並列計算	33
■ 可視化	35

ファイルの全体構成

custom - ファイル・ブラウザ

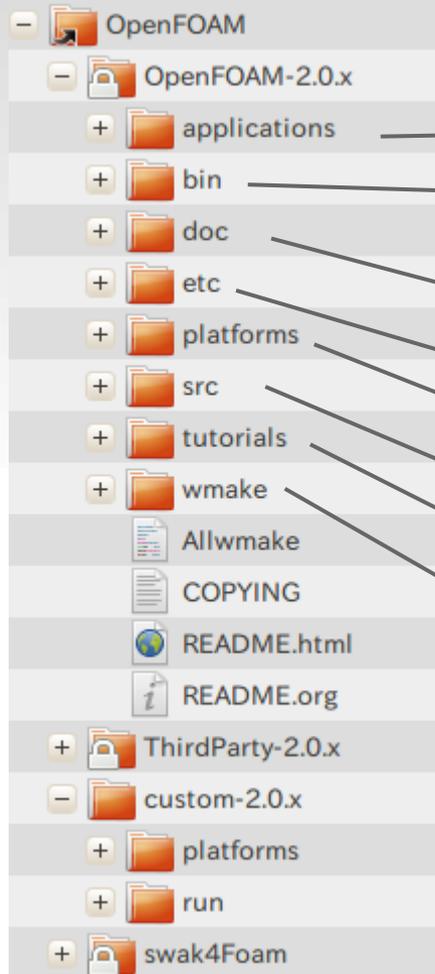
ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ヘルプ(H)

戻る 進む 50% 一覧表示

場所 custom

名前	サイズ	種類	更新日時
Desktop	3個のアイテム	フォルダ	2012年05月13日 2
OpenFOAM	4個のアイテム	フォルダへのリンク	2012年05月13日 2
OpenFOAM-2.0.x	12個のアイテム	フォルダ	2011年08月16日 1
ThirdParty-2.0.x	17個のアイテム	フォルダ	2011年08月16日 2
custom-2.0.x	2個のアイテム	フォルダ	2012年05月13日 2
swak4Foam	16個のアイテム	フォルダ	2011年08月17日 1

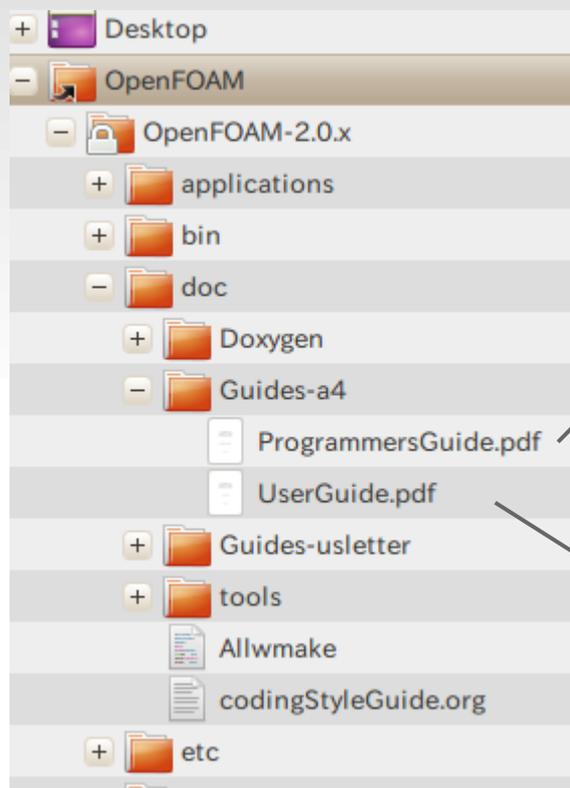
ファイルの全体構成



- 標準ソルバー、ユーティリティー
スクリプトなどコンパイル不要な
実行コマンド
- マニュアルなど
- 環境設定用
- コンパイル後のバイナリファイル
- ライブラリーのソースコード
- チュートリアルケース
- コンパイルシステム

OpenFOAMのマニュアル

(英文、原本)



+ GNU Free Documentation Licence	P-2
Trademarks	P-7
Contents	P-11
+ 1 Tensor mathematics	P-15
+ 2 Discretisation procedures	P-27
- 3 Examples of the use of OpenF...	P-45
+ 3.1 Flow around a cylinder	P-45
+ 3.2 Steady turbulent flow ove...	P-51
+ 3.3 Supersonic flow over a fo...	P-58
+ 3.4 Decompression of a tank ...	P-62
+ 3.5 Magneto hydrodynamic fl...	P-67
Index	P-70

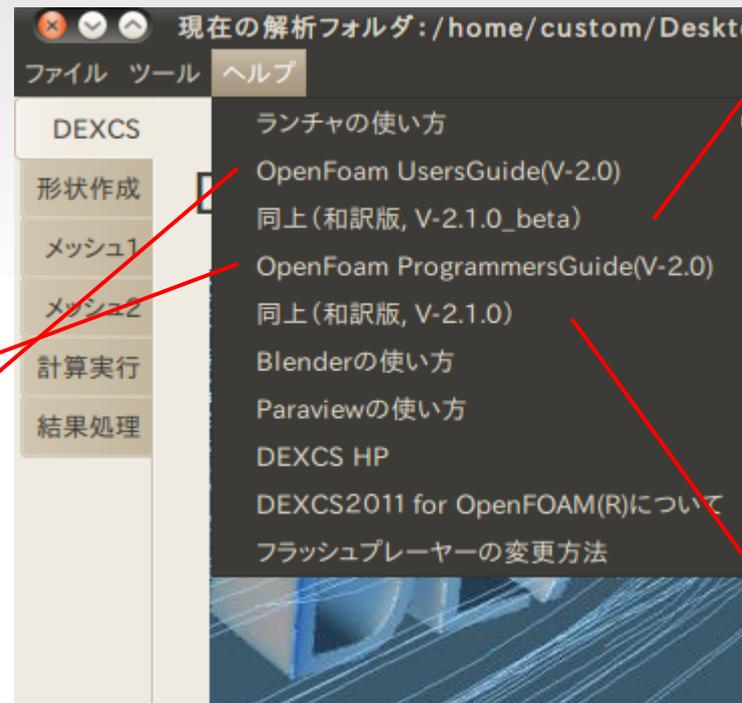
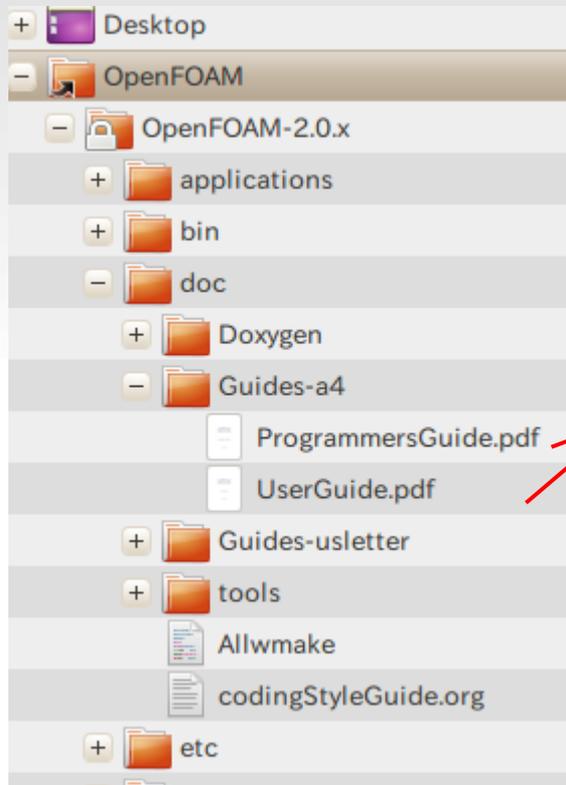
+ GNU Free Documentation Licence	U-2
Trademarks	U-7
Contents	U-11
1 Introduction	U-17
+ 2 Tutorials	U-19
+ 3 Applications and libraries	U-69
+ 4 OpenFOAM cases	U-103
+ 5 Mesh generation and conversion	U-127
+ 6 Post-processing	U-163
+ 7 Models and physical properties	U-179
Index	U-184

OpenFOAMのマニュアル

(和訳版、オープンCAE学会にて公開)

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 2.1.0
2012年2月18日

OpenFOAMユーザー会
一般社団法人 オープン CAE 学会

OpenFOAM

オープンソースCFDツールボックス

プログラマズ・ガイド 和訳

Linuxシェルの基礎

← → ↻ 🌐 www.opencae.jp/wiki/平成22年度OpenFOAM



wiki: 平成22年度OpenFOAM非圧縮性流体解析演習シリーズ

ツイート 0

平成22年度OpenFOAM非圧縮性流体解析演習

この演習シリーズは終了致しました。参加して頂いた皆様大変ありがとうございました。

概要

- 内容: OpenFOAMによる非圧縮性流体解析の演習シリーズ
- 時間: 1回90分間
- 回数: 全8回
- 形式: 基礎講義及び演習(講習者がノートPCを持ち込んで実際に)
- 講師: 今野 雅(東京大学)
- 定員: 24名
- 場所: 東京大学 工学部1号館1階14号講義室
- 地図: [本郷アクセスマップ](#), [工学部1号館](#)
- 備考: [OpenFOAM勉強会 for beginner](#)の前に併設して開催

準備

- 講習会をスムーズに受講して頂くために、講習会を申し込む際に必ず事前準備をしておいてくださるようお願い致します。

過去の演習

- 第1回(初級講習): 平成22年9月25日(土) 12:00-14:30 (受付開始 12:30)
 - 演習題目: 「Bashシェルの基礎とOpenFOAM概要および主要な非圧縮性ソルバー・チュートリアルの説明」
 - [配布資料](#) (PDF形式、13MB)
- 第2回(初級講習): 平成22年11月28日(日) 12:30-14:00 (受付開始 12:00)
 - 演習題目: 「キャビティ流れのチュートリアルによる格子生成・流体解析・可視化の基礎」

シェルで使う主なコマンド

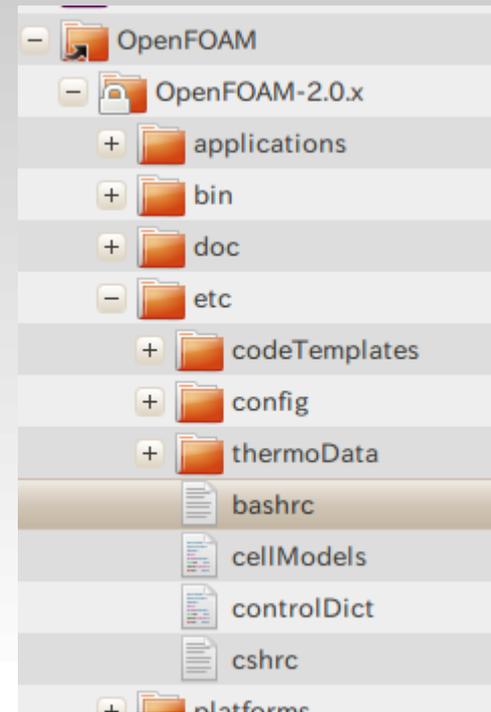
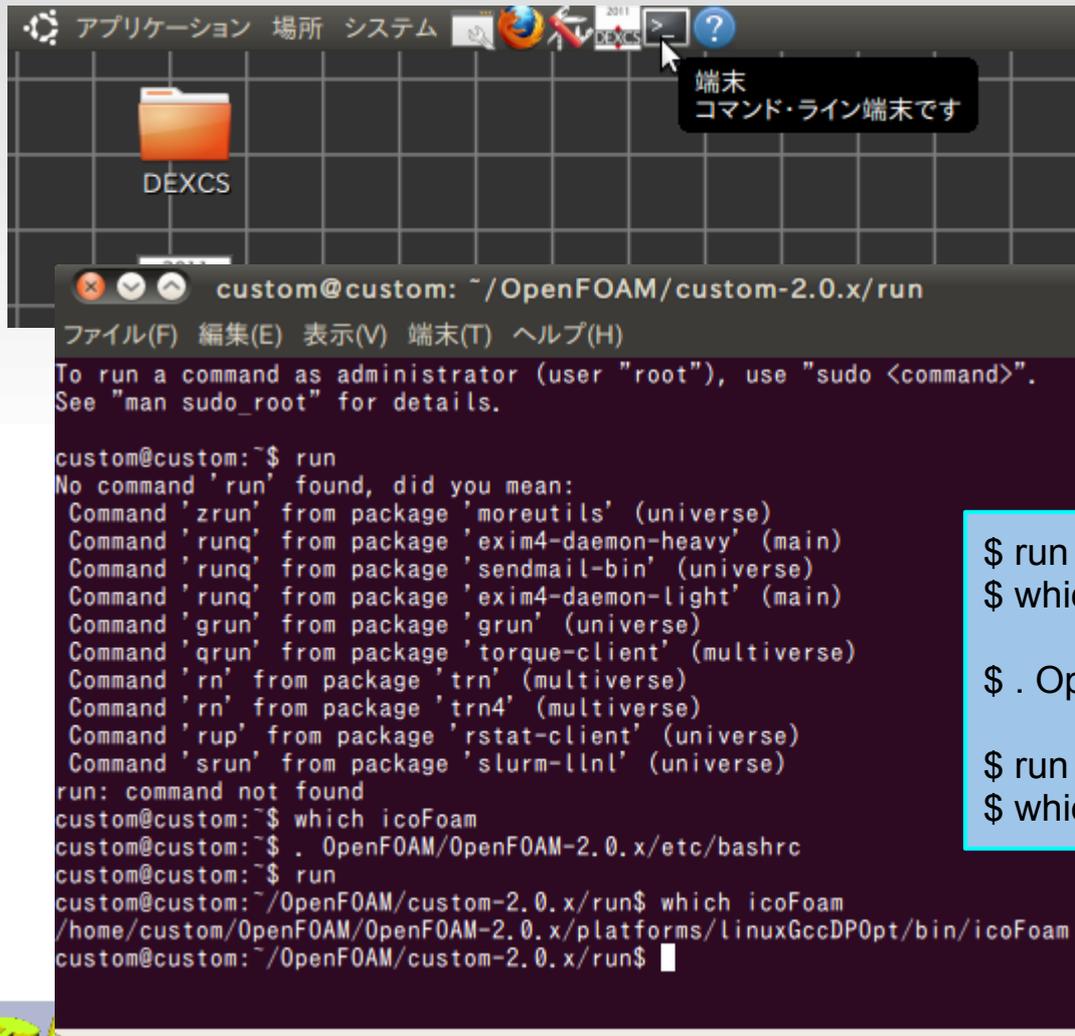
- ▶ **cd**: ディレクトリを移動する
- ▶ **cp**: ファイルをコピーする
- ▶ **mkdir**: ディレクトリを作る
- ▶ **pwd**: 現在のディレクトリの場所を確認する
- ▶ **ls**: ファイルやディレクトリの情報を表示する
- ▶ **more**: テキスト・ファイルの内容を見る
- ▶ **alias**: コマンドの別名(エイリアス)を表示・登録する



p.26 ~ p.37

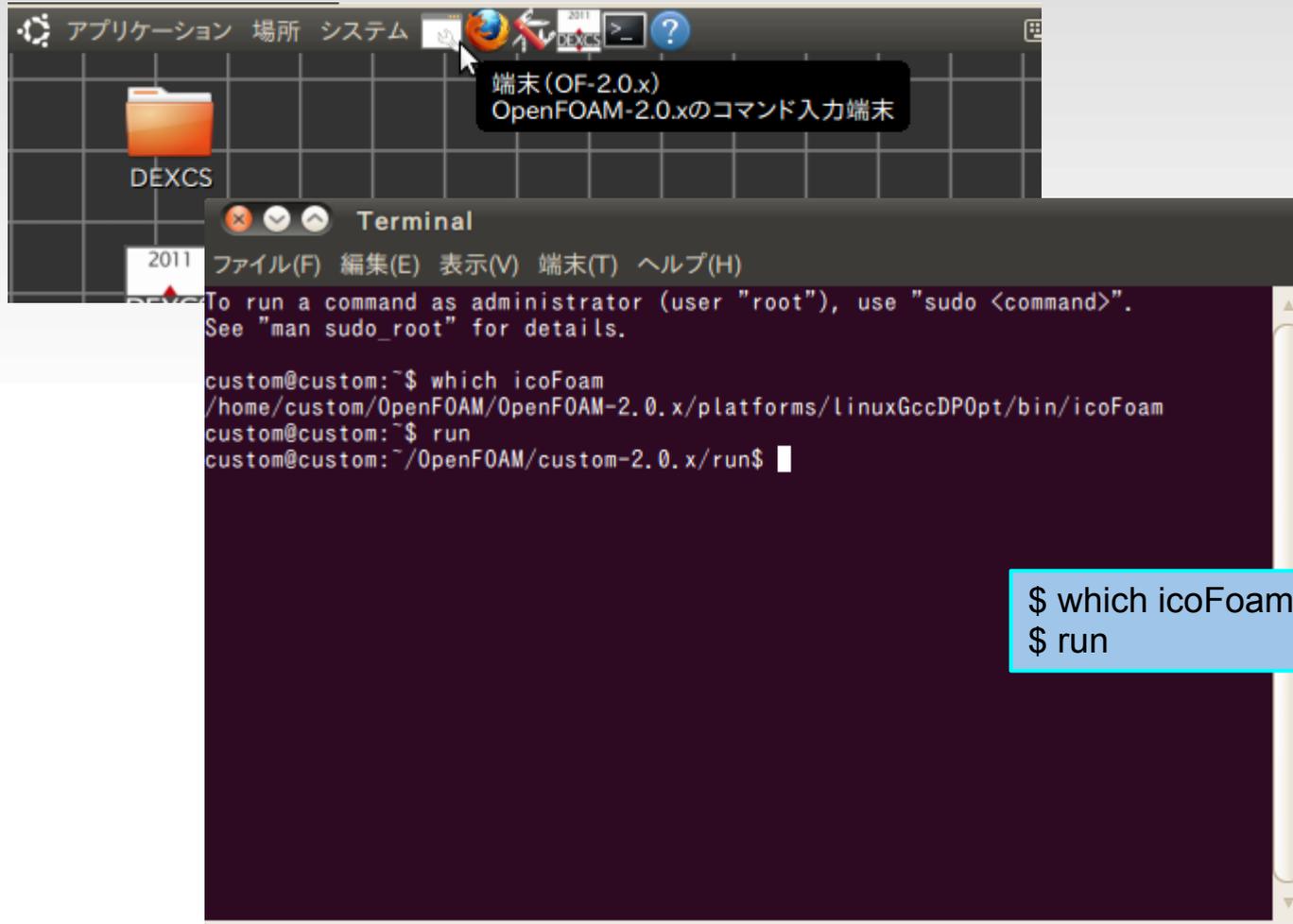
DEXCS2009
(ubuntu-8.04)
OpenFOAM-1.6.x

起動環境の組込



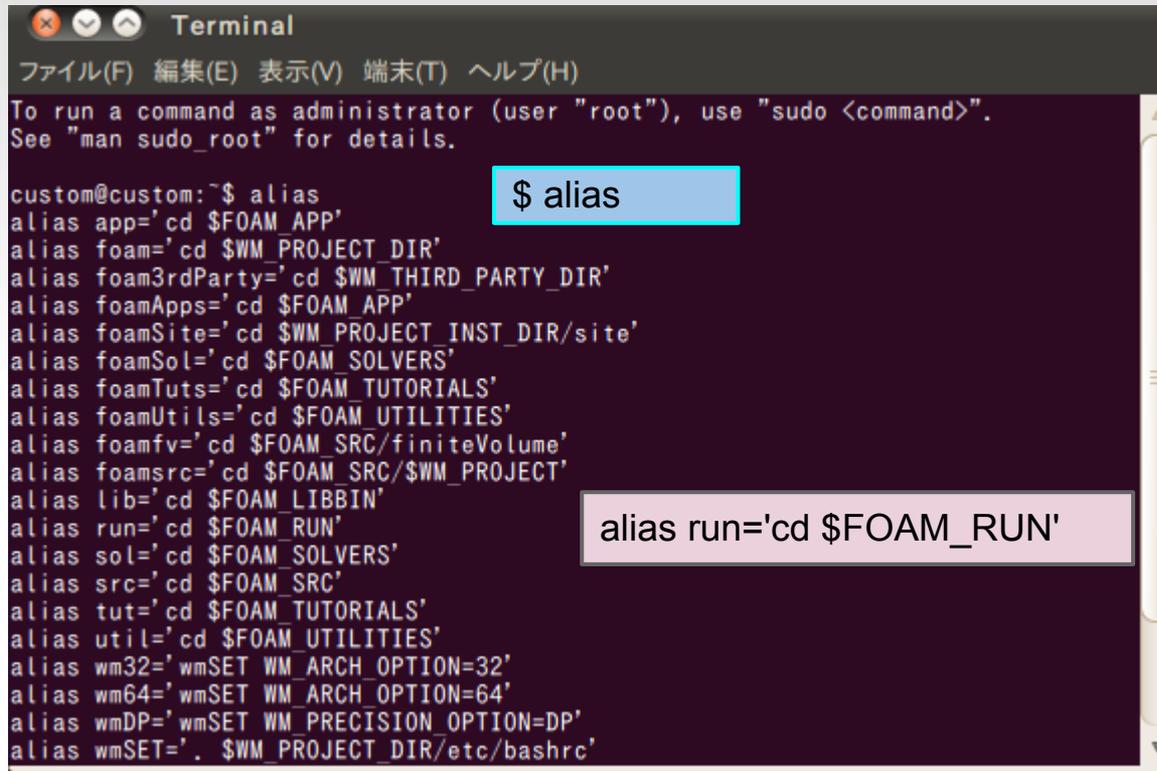
```
$ run
$ which icoFoam
$ . OpenFOAM/OpenFOAM-2.0.x/etc/bashrc
$ run
$ which icoFoam
```

起動用環境変数



\$ which icoFoam
\$ run

エイリアス



A terminal window titled "Terminal" with a menu bar containing "ファイル(F)", "編集(E)", "表示(V)", "端末(T)", and "ヘルプ(H)". The terminal content shows the command `alias` being executed, resulting in a list of aliases. Two callouts highlight specific parts: a blue box around the prompt `$ alias` and a pink box around the line `alias run='cd $FOAM_RUN'`.

```
custom@custom:~$ alias
alias app='cd $FOAM_APP'
alias foam='cd $WMM_PROJECT_DIR'
alias foam3rdParty='cd $WMM_THIRD_PARTY_DIR'
alias foamApps='cd $FOAM_APP'
alias foamSite='cd $WMM_PROJECT_INST_DIR/site'
alias foamSol='cd $FOAM_SOLVERS'
alias foamTuts='cd $FOAM_TUTORIALS'
alias foamUtils='cd $FOAM_UTILITIES'
alias foamfv='cd $FOAM_SRC/finiteVolume'
alias foamsrc='cd $FOAM_SRC/$WMM_PROJECT'
alias lib='cd $FOAM_LIBBIN'
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='wmSET WM_ARCH_OPTION=32'
alias wm64='wmSET WM_ARCH_OPTION=64'
alias wmDP='wmSET WM_PRECISION_OPTION=DP'
alias wmSET='. $WMM_PROJECT_DIR/etc/bashrc'
```

環境変数

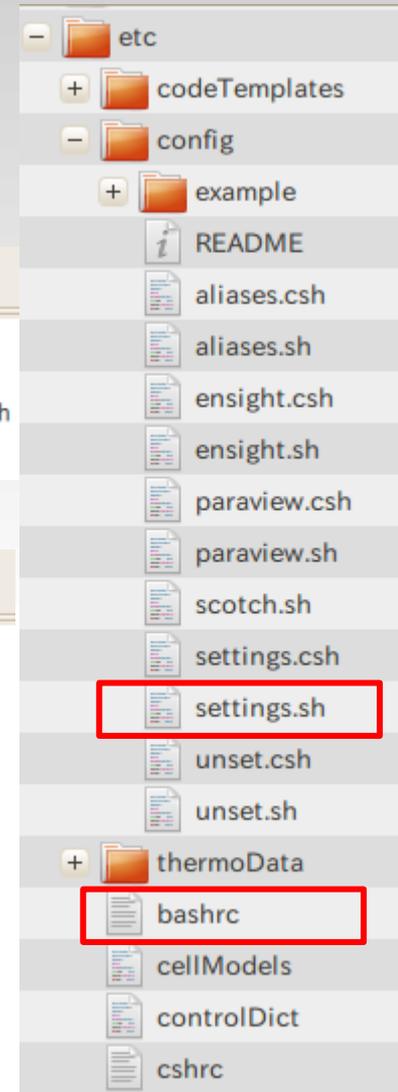
```
bashrc x
...
34 export WM_PROJECT=OpenFOAM
35 export WM_PROJECT_VERSION=2.0.x
...
```

```
bashrc x
205
206 # Source project setup files
207 # -----
208 _foamSource $WM_PROJECT_DIR/etc/config/settings.sh
209 _foamSource $WM_PROJECT_DIR/etc/config/aliases.sh
210
```

```
settings.sh x
...
195 export FOAM_UTILITIES=$FOAM_APP/utilities
196 export FOAM_SOLVERS=$FOAM_APP/solvers
197 export FOAM_RUN=$WM_PROJECT_USER_DIR/run
198
```

```
Terminal
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
alias foamUtils='cd $FOAM_UTILITIES'
alias foamfv='cd $FOAM_SRC/finiteVolume'
alias foamsrc='cd $FOAM_SRC/$WM_PROJECT'
alias lib='cd $FOAM_LIBBIN'
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='wmSET WM_ARCH_OPTION=32'
alias wm64='wmSET WM_ARCH_OPTION=64'
alias wmDP='wmSET WM_PRECISION_OPTION=DP'
alias wmSET='. $WM_PROJECT_DIR/etc/bashrc'
alias wmSP='wmSET WM_PRECISION_OPTION=SP'
alias wmSchedOFF='unset WM_SCHEDULER'
alias wmSchedON='export WM_SCHEDULER=$WM_PROJECT_DIR/wmake/wmakeScheduler'
alias wmUNSET='. $WM_PROJECT_DIR/etc/config/unset.sh'
custom@custom:~$
custom@custom:~$ echo $FOAM_RUN
/home/custom/OpenFOAM/custom-2.0.x/run
custom@custom:~$
custom@custom:~$ echo $WM_PROJECT
OpenFOAM
custom@custom:~$
```

```
$ echo $FOAM_RUN
$ echo $WM_PROJECT
```



標準ソルバー

OpenFOAM-2.0.x	
applications	
solvers	12個のアイテム
DNS	1個のアイテム
basic	3個のアイテム
combustion	10個
compressible	5個
discreteMethods	2個
electromagnetics	3個
financial	1個
heatTransfer	6個
incompressible	9個
lagrangian	9個
multiphase	10個
stressAnalysis	2個

+ GNU Free Documentation L...	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
- 第3章 アプリケーションとライブ...	U-73
+ 3.1 OpenFOAMのプログラ...	U-73
+ 3.2 アプリケーションやライブ...	U-75
3.3 アプリケーションの実行	U-85
+ 3.4 アプリケーションの並列...	U-86
3.5 標準のソルバ	U-90
3.6 標準のユーティリティ	U-94
3.7 標準のライブラリ	U-100

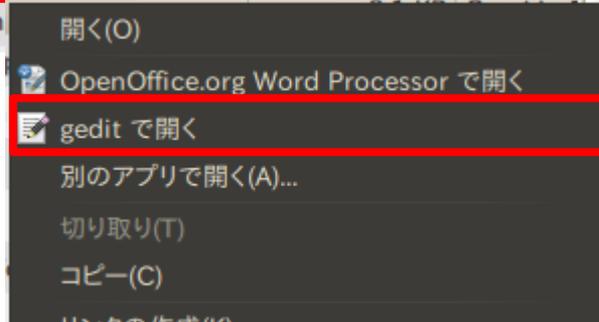
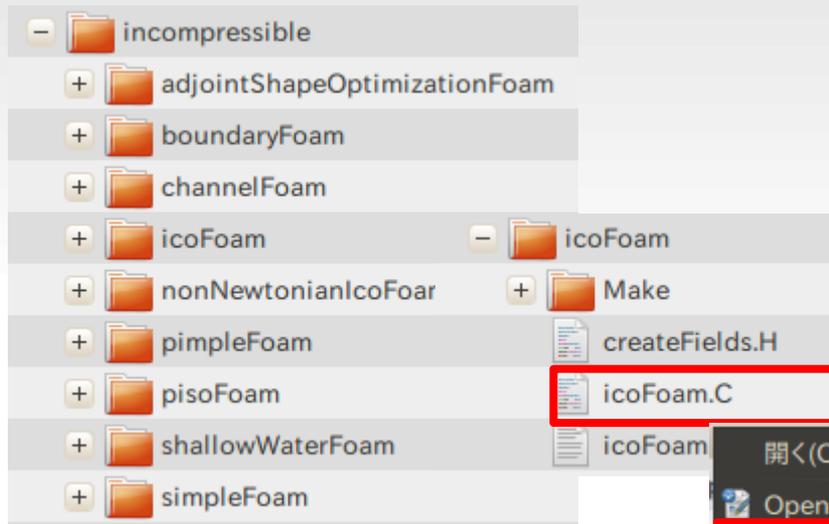
基礎的な CFD コード

laplacianFoam	固体の熱拡散のような単純なラプラス方程式を解く
potentialFoam	シンプルなポテンシャル流のコード. 完全ナビエ・ストークスコードを解く際の保存された初期値の生成にも使用できる
scalarTransportFoam	パッシブスカラの輸送方程式を解く

非圧縮性流れ

adjointShapeOptimizationFoam	圧力損失を生じる領域に, 随伴公式を使って推定された「ロッキング」を適用することで管路形状を最適化する, 非圧縮性・乱流の非ニュートン流体用定常ソルバ
boundaryFoam	1次元の非圧縮性・乱流用の定常状態ソルバで, 通常, 解析では流入口で境界層条件を発生させます.
channelFoam	チャンネル内流れ用の非圧縮 LES
icoFoam	非圧縮性, 層流の速度-圧力ソルバ. 非ニュートン流体も可
MRFSimpleFoam	MRF 領域のある非圧縮性・乱流・非ニュートン流体の定常ソルバ
nonNewtonianIcoFoam	非ニュートン流体の非圧縮性, 層流の非定常ソルバ
pimpleDyMFoam	ダイナミックメッシュをもつニュートン流体の非圧縮性乱流の PIMPLE (SIMPLE と PISO の融合) アルゴリズムによる非定常ソルバ
pimpleFoam	PIMPLE (SIMPLE と PISO の融合) アルゴリズムによる

ソルバーの構成ファイル



```
icoFoam.C x
31
32 #include "fvCFD.H"
33
34 // *****
35
36 int main(int argc, char *argv[])
37 {
38     #include "setRootCase.H"
39
40     #include "createTime.H"
41     #include "createMesh.H"
42     #include "createFields.H"
43     #include "initContinuityErrs.H"
44
45     // *****
46     Info<< "\nStarting time loop\n" << endl;
47
48     while (runTime.loop())
49     {
50         Info<< "Time = " << runTime.timeName() <<
51         #include "readPISOControls.H"
52         #include "CourantNo.H"
53         Matrix UEqn
54         ddt(U)
55         div(phi, U)
56         laplacian(nu, U)
57     }
```

ユーティリティ

OpenFOAM-2.0.x	
applications	
utilities	7個のアイテム
mesh	4個のアイテム
miscellaneous	5個のアイテム
parallelProcessing	4個のアイテム
postProcessing	12個のアイテム
preProcessing	14個のアイテム
surface	27個のアイテム
thermophysical	5個のアイテム
Allwmake	647

+ GNU Free Documentation L...	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
- 第3章 アプリケーションとライブ...	U-73
+ 3.1 OpenFOAMのプログラ...	U-73
+ 3.2 アプリケーションやライブ...	U-75
3.3 アプリケーションの実行	U-85
+ 3.4 アプリケーションの並列...	U-86
3.5 標準のソルバ	U-90
3.6 標準のユーティリティ	U-94
3.7 標準のライブラリ	U-100
第4章 OpenFOAMのケース...	U-105

メッシュ生成	マルチブロック・メッシュのジェネレータ 既存のパッチやファイルから読み込んだパッチを（デフォルトでは面の外側へ、オプションで反転して）押し出す。2Dメッシュ（すべての面が2点で、前後の面がない）のみ込み、与えられた厚さに押し出すことで3Dメッシュを作る。
blockMesh	faceZones を個別のメッシュに（別の領域として）押し出す。例えば、液体の膜領域を作るために
extrudeMesh	自動分割六面体メッシュ。細分化して面にスナップする
extrude2DMesh	
extrudeToRegionMesh	
snappyHexMesh	
メッシュの変換	
ansysToFoam	I-DEAS から出力した ANSYS インプットメッシュファイルを開いたメッシュを OpenFOAM 形式へ変換する
cfx4ToFoam	CFX 4 メッシュを OpenFOAM 形式へ変換する
datToFoam	datToFoam メッシュファイル内を読み、points ファイルに出力する。blockMesh との結合に使われる。
fluent3DMeshToFoam	Fluent のメッシュを OpenFOAM 形式に変換する
fluentMeshToFoam	Fluent のメッシュを OpenFOAM 形式に変換する。複数領域と、領域の境界の処理も扱える
foamMeshToFluent	OpenFOAM メッシュを Fluent メッシュ形式で出力する

ライブラリ

-	platforms	1個の
-	linuxGccDPOpt	2個の
+	bin	247個の
+	lib	102個の

3.7 標準のライブラリ

OpenFOAM 配布のライブラリは $\$FOAM_LIB/\$WM_OPTIONS$ ディレクトリ内にあり、コマンド欄に `lib` と入力すればすぐに見つかります。一方、名前は `lib` を前につけて、例えば `incompressibleTransportModels` が非圧縮性の輸送モデルのライブラリを含むというように合理的でかつ説明的です。表現を簡単にするためにライブラリは二つのタイプに分けられます。

一般的ライブラリ これらは一般的なクラスや表 3.7 に記載したような関連機能を備えています。
モデルライブラリ これらは表 3.8, 表 3.9, 表 3.10 に記載した計算連続体力学で使われるモデルを定めます。

+ GNU Free Documentation License	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
- 第3章 アプリケーションとライブラリ	U-73
+ 3.1 OpenFOAMのプログラミング	U-73
+ 3.2 アプリケーションやライブラリ	U-75
3.3 アプリケーションの実行	U-85
+ 3.4 アプリケーションの並列実行	U-86
3.5 標準のソルバ	U-90
3.6 標準のユーティリティ	U-94
3.7 標準のライブラリ	U-100
+ 第4章 OpenFOAMのケース	11-105

基本的な OpenFOAM ツールのライブラリ — OpenFOAM

algorithms	アルゴリズム
containers	コンテナクラス
db	データベースクラス
dimensionedTypes	dimensioned<Type> クラスと派生クラス
dimensionSet	dimensionSet クラス
fields	領域クラス
global	グローバルな設定
graph	graph クラス
interpolations	補間スキーム
matrices	行列クラス
memory	メモリ管理ツール
meshes	メッシュクラス
primitives	初期クラス

有限体積法ライブラリ — finiteVolume

cfdTools	CFD ツール
----------	---------

チュートリアルケース

-	tutorials	16個のアイテム
+	DNS	1個のアイテム
+	basic	3個のアイテム
+	combustion	7個のアイテム
+	compressible	8個のアイテム
+	discreteMethods	2個のアイテム
+	electromagnetics	2個のアイテム
+	financial	1個のアイテム
+	heatTransfer	7個のアイテム
+	incompressible	15個のアイテム
+	lagrangian	7個のアイテム
+	mesh	2個のアイテム
+	multiphase	14個のアイテム
+	stressAnalysis	2個のアイテム
	Allclean	490 バイト
	Allrun	2.8 KB
	Alltest	5.6 KB

(参考) 標準ソルバー

-	solvers	12個のアイテム
+	DNS	1個のアイテム
+	basic	3個のアイテム
+	combustion	10個のアイテム
+	compressible	5個のアイテム
+	discreteMethods	2個のアイテム
+	electromagnetics	3個のアイテム
+	financial	1個のアイテム
+	heatTransfer	6個のアイテム
+	incompressible	9個のアイテム
+	lagrangian	9個のアイテム
+	multiphase	10個のアイテム
+	stressAnalysis	2個のアイテム

チュートリアルケースの全自動実行方法

-	tutorials	16個のアイテム
+	DNS	1個のアイテム
+	basic	3個のアイテム
+	combustion	7個のアイテム
+	compressible	8個のアイテム
+	discreteMethods	2個のアイテム
+	electromagnetics	2個のアイテム
+	financial	1個のアイテム
+	heatTransfer	7個のアイテム
+	incompressible	15個のアイテム
+	lagrangian	7個のアイテム
+	mesh	2個のアイテム
+	multiphase	14個のアイテム
+	stressAnalysis	2個のアイテム
	Allclean	490 バイト
	Allrun	2.8 KB
	Alltest	5.6 KB

実行例、但し演習環境では実施しないこと

1

```
$ run
$ cp -r ../../OpenFOAM-2.0.x/tutorials/
./
$ cd tutorials/
$ ./Allrun
```

全ケースを自動実行
所要時間: 1週間以上、不明
必要ディスク容量: 100GB程度、不明

2

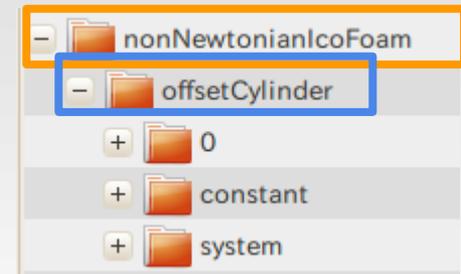
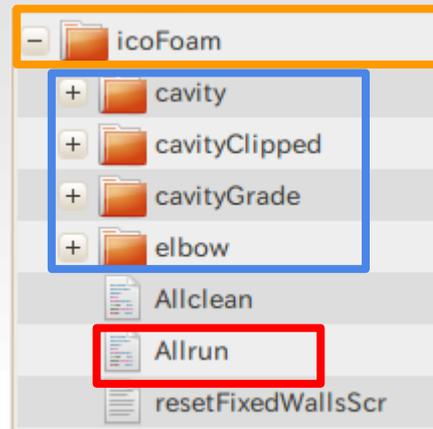
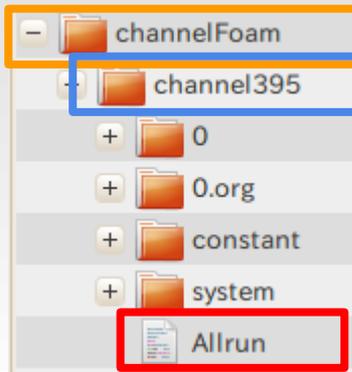
```
$ run
$ cp -r ../../OpenFOAM-2.0.x/tutorials/
./
$ cd tutorials/
$ ./Alltest
```

全ケースを自動実行、最初の1ステップのみ計算
所要時間: 1時間程度
必要ディスク容量: 数GB程度

チュートリアルケースの個別実行方法

***Foam:ソルバー名

ケース名



Allrun を実行

blockMesh を実行
↓
ソルバー(***Foam)を実行

```
Allrun x
1 #!/bin/sh
2 cd ${0%/*} || exit 1 # run from this directory
3
4 # Source tutorial run functions
5 . $WM_PROJECT_DIR/bin/tools/RunFunctions
6
7 # Get application directory
8 application=`getApplication`
9
10 runApplication blockMesh
11 runApplication $application
12 runApplication postChannel
13
14 #
```

チュートリアルケースの解説例

+ GNU Free Documentation Lice...	U-3
目次	U-11
第1章 はじめに	U-17
- 第2章 チュートリアル	U-19
+ 2.1 天井駆動のキャビティ流れ	U-19
+ 2.2 穴あき板の応力解析	U-48
+ 2.3 ダムの決壊	U-59
+ 第3章 アプリケーションとライブラリ	U-73
+ 第4章 OpenFOAMのケース	U-105
+ 第5章 メッシュの生成と変換	U-129
+ 第6章 後処理	U-165
+ 第7章 モデルと物性値	U-183
索引	U-187

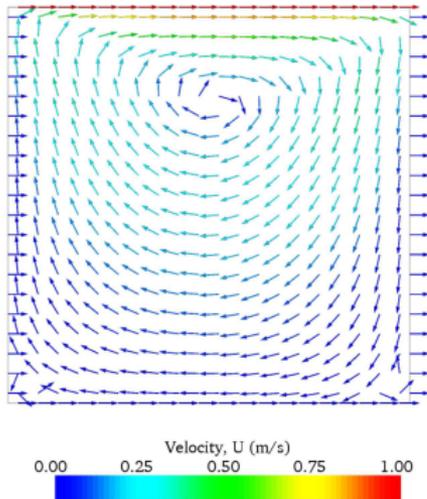


図 2.7 キャビティケースの速度

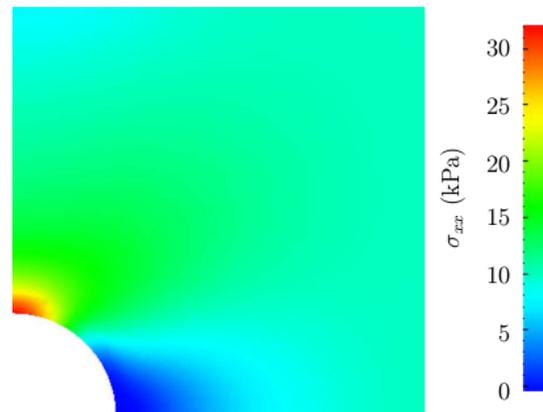
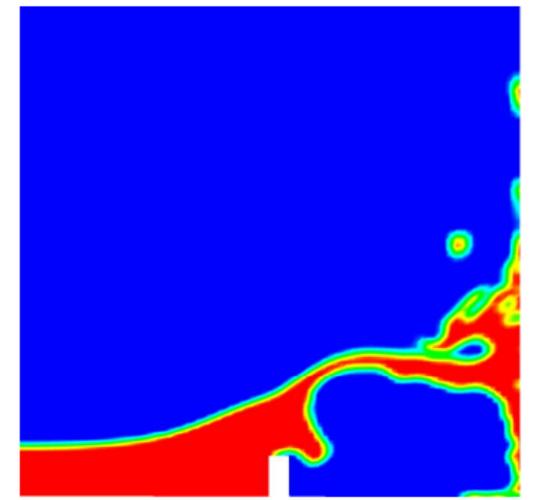
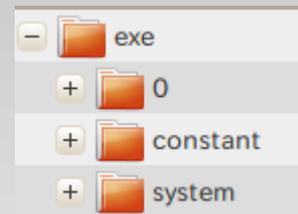


図 2.18 穴あき板における応力場



ケースフォルダ



U-106

第 4 章 OpenFOAM

+ GNU Free Documentation License	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
+ 第3章 アプリケーションとライブラリ	U-73
- 第4章 OpenFOAMのケース	U-105
4.1 OpenFOAMのケースのファイル構造	U-105
+ 4.2 基本的な入出力ファイルのフォーマット	U-106
4.3 時間とデータの入出力制御	U-113
+ 4.4 数値スキーム	U-115
+ 4.5 解法とアルゴリズム制御	U-122
+ 第5章 メッシュの生成と変換	U-129
+ 第6章 後処理	U-165
+ 第7章 モデルと物性値	U-183
索引	U-187

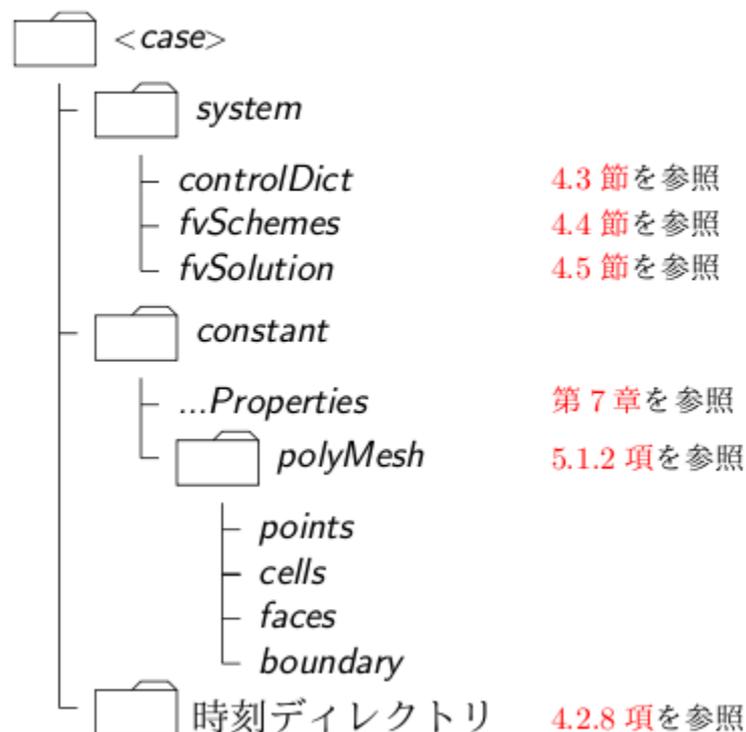
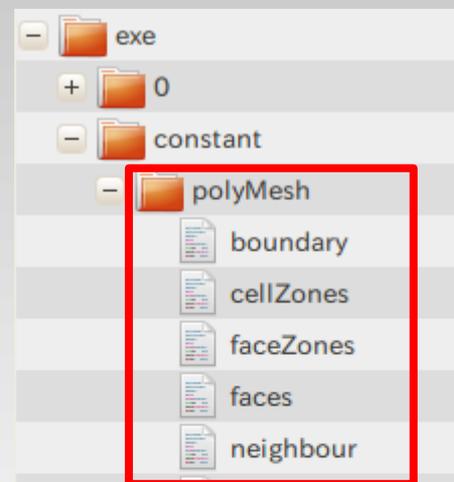


図 4.1 ケースディレクトリの構造

メッシュ(constant/polyMesh/)



+ GNU Free Documentation License	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
+ 第3章 アプリケーションとライブラリ	U-73
+ 第4章 OpenFOAMのケース	
- 第5章 メッシュの生成と変換	
- 5.1 メッシュの記法	
5.1.1 メッシュの仕様と妥当性の制約	
5.1.2 polyMeshの記述	
5.1.3 cellShapeツール	
5.1.4 1次元や2次元, 軸対称問題	
+ 5.2 境界	
+ 5.3 blockMeshユーティリティを使ったメッ...	
+ 5.4 snappyHexMeshユーティリティを使...	
+ 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...	
+ 第6章 後処理	
+ 第7章 モデルと物性値	
索引	

5.1.2 polyMesh の記述

constant ディレクトリのサブディレクトリである *polyMesh* には、そのケースの *polyMesh* データが全て収められています。この *polyMesh* の記述は面ベースであり、既に述べましたように、内部のセルは二つのセルと接続し、境界面はセルと境界のパッチを指定します。各面には「保有」セルと「隣接」セルが割り当てられ、面を通じた接続は、保有セルと隣接セルのラベルによって簡潔に記述することができます。境界の場合には、面に接続されたセルがその面の保有者であり、隣接セルには -1 のラベルが割り充てられます。以上を踏まえた上で、以下のファイルで構成される入出力の詳細をご覧ください。

points セルの頂点を記述するベクトルのリストです。ここで、リストにおける最初のベクトルは頂点0、次のベクトルの頂点1という風に番号付けします。

faces 面のリストです。各面は点中の頂点の番号のリストで成り立っています。ここで、先程と同様に、リスト中の最初の面の番号は0です。

owner 保有セルのラベルのリストです。面のリストと同じ順番に並んでますので、リストの最初のラベルは0番の面の保有セルのラベル、次のラベルは1番の面の保有セルのラベルということになります。

neighbour 隣接セルのラベルのリストです。

boundary パッチのリストです。以下のように、パッチ名の宣言で始まる各パッチに対するディ

constant/...Properties

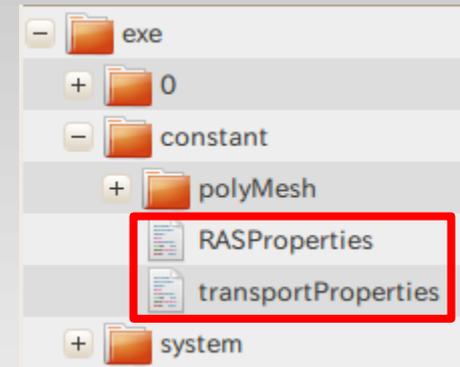
+ GNU Free Documentation License	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
+ 第3章 アプリケーションとライブラリ	U-73
+ 第4章 OpenFOAMのケース	U-105
+ 第5章 メッシュの生成と変換	U-129
+ 第6章 後処理	U-165
- 第7章 モデルと物性値	U-183
- 7.1 熱物理モデル	U-183
7.1.1 熱物性データ	U-184
- 7.2 乱流モデル	U-186
7.2.1 モデル係	
7.2.2 壁関数	
索引	

7.2 乱流モデル

乱流のモデリングを含むあらゆるソルバは *turbulenceProperties* デクシヨナリを読み込みます。このファイルの中では、*simulationType* キーワードで使用する乱流モデルとして次のいずれかを選択します。

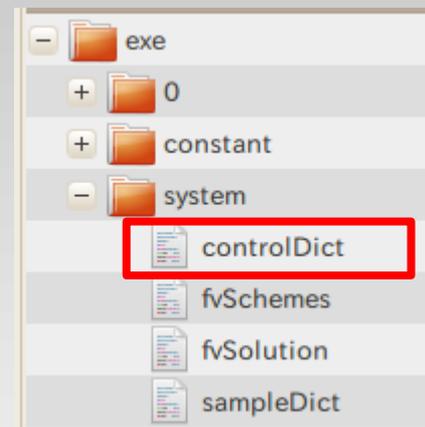
laminar 乱流モデルを使用しない

RASModel レイノルズ平均応力 (RAS) モデル



時間とデータの入出力制御

system/controlDict



- + GNU Free Documentation License U-3
- 目次 U-11
- 第1章 はじめに U-17
- + 第2章 チュートリアル U-19
- + 第3章 アプリケーションとライブラリ U-69
- 第4章 OpenFOAMのケース U-97
 - 4.1 OpenFOAMのケースのファイ... U-97
 - + 4.2 基本的な入出力ファイルの... U-98
 - 4.3 時間とデータの入出力制御 U-104
 - + 4.4 数値スキーム U-106
 - + 4.5 解法とアルゴリズム制御 U-112
- + 第5章 メッシュの生成と変換 U-119
- + 第6章 後処理 U-153
- + 第7章 モデルと物性値 U-171
- 索引 U-176

4.3 時間とデータの入出力制御

OpenFOAMのソルバは全て、データベースをセットアップすることによって、動き始めます。データベースは入出力を制御し、またデータの出力は通常実行中、時間ごとに要求されるので、時間はデータベースにとって不可避の要素です。controlDict ディクショナリはデータベースの作成に不可欠な入力パラメータを設定します。controlDictのキーワード入力項目は表 4.4 に記載されています。時間制御方式と writeInterval 入力だけは完全に強制的で、省略できる任意の項目には表 4.4 で示されたデフォルト値のデータベースが用いられます。

時間制御

startFrom	解析の開始時刻の制御
- firstTime	存在する時刻ディレクトリのうちで最初の時刻
- startTime	startTime の項目の入力により定める時刻
- latestTime	存在する時刻ディレクトリのうちで最近の時刻
startTime	startFrom の startTime を用いた解析の開始時刻
stopAt	解析の終了時刻の制御
- endTime	endTime の項目の入力により定める時刻
- writeNow	現在の時間ステップで解析を止めデータを書き出す
- noWriteNow	現在の時間ステップで解析を止めデータは書き出さない
- nextWrite	writeControl で指定した次のデータ書き出しの時間ステップで解析を止める
endTime	stopAt の endTime で指定した解析の終了時刻
deltaT	解析の時間ステップ

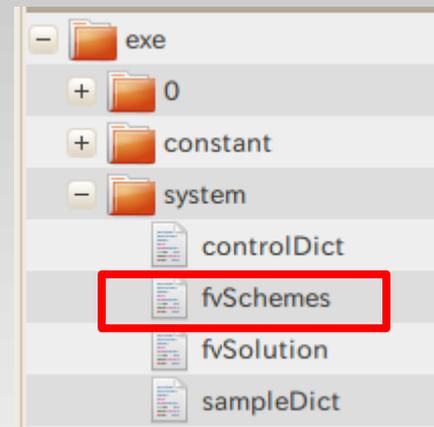
データの書き出し

writeControl	ファイルへのデータの書き出しのタイミングの制御
- timeStep†	タイムステップの writeInterval ごとにデータを書き出す

数値スキーム

system/fvSchemes

+ GNU Free Documentation License	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
+ 第3章 アプリケーションとライブラリ	U-69
- 第4章 OpenFOAMのケース	U-97
4.1 OpenFOAMのケースのファイ...	U-97
+ 4.2 基本的な入出力ファイルの...	U-98
4.3 時間とデータの入出力制御	U-104
- 4.4 数値スキーム	U-106
4.4.1 補間スキーム	U-108
4.4.2 表面法線方向勾配スキ...	U-109
4.4.3 勾配スキーム	U-110
4.4.4 ラプラシアンスキーム	U-110
4.4.5 発散スキーム	U-111
4.4.6 時間スキーム	U-112
4.4.7 流束の算出	U-112
+ 4.5 解法とアルゴリズム制御	U-112
+ 第5章 メッシュの生成と変換	U-119
+ 第6章 後処理	U-153
+ 第7章 モデルと物性値	U-171
索引	U-175



4.4.6 時間スキーム

一次の時間微分項 ($\partial/\partial t$) は, *ddtSchemes* サブディクショナリで指定します. 各項に対する離散化スキームは表 4.11 から選ぶことができます.

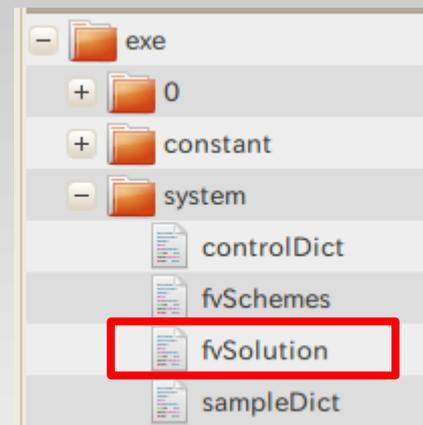
CrankNicholson スキームでは, Euler スキームと混合させる割合を決める係数 ψ を用います. $\psi = 1$ の場合には純粋な CrankNicholson, $\psi = 0$ の場合は純粋な Euler に対応します. 純粋な CrankNicholson では不安定なケースにおいては, 混合係数をいじることで計算を安定化させることができます.

スキーム	説明
Euler	1次, 制限, 陰的
CrankNicholson ψ	2次, 制限, 陰的
backward	2次, 陰的
steadyState	時間導関数について解かない

表 4.11 *ddtSchemes* において使用可能な離散化スキーム

解法とアルゴリズム制御

system/fvSolutions



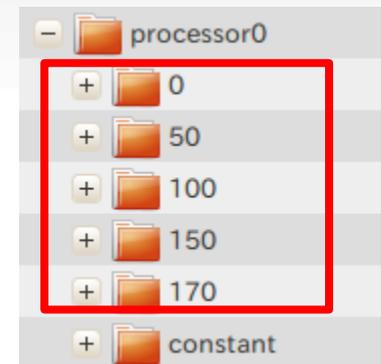
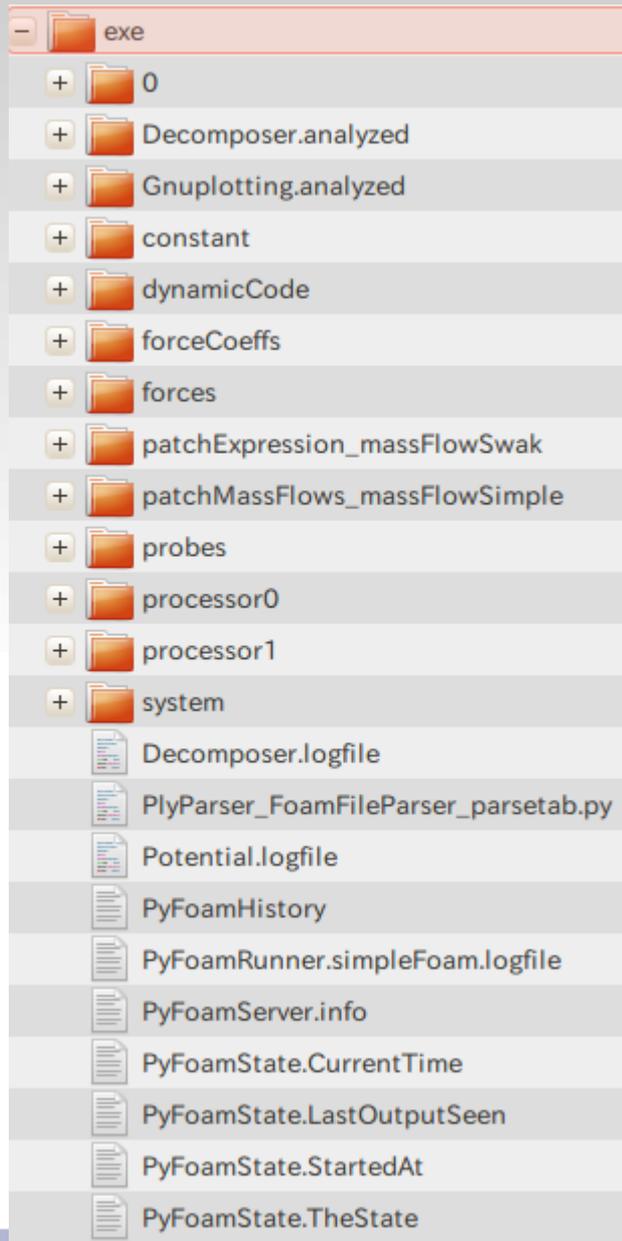
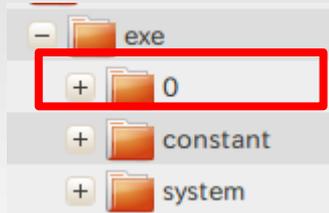
+ GNU Free Documentation License	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
+ 第3章 アプリケーションとライブラリ	U-69
- 第4章 OpenFOAMのケース	U-97
4.1 OpenFOAMのケースのファイ...	U-97
+ 4.2 基本的な入出力ファイルの...	U-98
4.3 時間とデータの入出力制御	U-104
+ 4.4 数値スキーム	U-106
- 4.5 解法とアルゴリズム制御	U-112
4.5.1 線形ソルバ制御	U-113
4.5.2 不足緩和解析	U-115
4.5.3 PISOとSIMPLEアルゴ...	U-117
4.5.4 その他のパラメタ	U-117
+ 第5章 メッシュの生成と変換	U-119
+ 第6章 後処理	U-153
+ 第7章 モデルと物性値	U-171
索引	U-175

5 解法とアルゴリズム制御

方程式のソルバ（求解機）、公差、およびアルゴリズムは `system` ディレクトリの `fvSolution` ディクショナリから制御されます。以下に示すのは、`icoFoam` ソルバに必要な `fvSolution` ディクショナリか

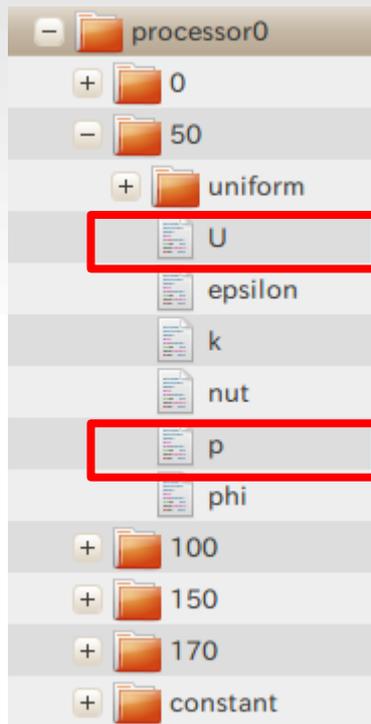
```
7
8 solvers
9 {
10     p
11     {
12         solver          PCG;
13         preconditioner  DIC;
14         tolerance       1e-06;
15         relTol          0;
16     }
17 }
```

時刻フォルダ



```
controlDict x
19
20 startFrom latestTime;
21
22 startTime 0;
23
24 stopAt endTime;
25
26 endTime 1000;
27
28 deltaT 1;
29
30 writeControl timeStep;
31
32 writeInterval 50;
33
34 purgeWrite 0;
35
36 writeFormat ascii;
```

Field変数



```
U x
16 // *****
17
18 dimensions      [0 1 -1 0 0 0];
19
20 internalField   nonuniform List<vector>
21 172560
22 (
23 (10.0011 0.000864482 0.000644335)
24 (10.0011 0.00253352 0.000710604)
25 (10.0011 0.00401844 0.000844031)
26 (10.0011 0.00518292 0.00104319)
27 (10.0011 0.00588646 0.00130079)
28 (10.0011 0.00599665 0.00159511)
29 (10.0011 0.00546662 0.00187491)
30 (10.0011 0.00439979 0.00208717)
31 (10.0011 0.00303521 0.00219468)
32 (10.0012 0.00167850 0.00218411)
```

```
p x
16 // *****
17
18 dimensions      [0 2 -2 0 0 0];
19
20 internalField   nonuniform List<scalar>
21 172560
22 (
23 0.775869
24 0.793849
25 0.828568
26 0.877439
27 0.936385
28 0.999903
29 1.06117
30 1.1137
31 1.15282
32 1.17701
33 1.18704
```

単位

- + GNU Free Documentation License U-3
- 目次 U-11
- 第1章 はじめに U-17
- + 第2章 チュートリアル U-19
- + 第3章 アプリケーションとライブラリ U-73
- 第4章 OpenFOAMのケース U-105
 - 4.1 OpenFOAMのケースのファイ... U-105
 - 4.2 基本的な入出力ファイルのフォ... U-106
 - 4.2.1 一般的な構文規則 U-106
 - 4.2.2 ディクショナリ U-107
 - 4.2.3 データファイルヘッダ U-107
 - 4.2.4 リスト U-108
 - 4.2.5 スカラとベクトル, テンソル U-109
 - 4.2.6 次元の単位 U-109
 - 4.2.7 次元付の型 U-110
 - 4.2.8 フィールド U-110
 - 4.2.9 ディレクティブとマクロ置換 U-111
 - 4.2.10 '043includeおよび'04... U-112
 - 4.2.11 '043codeStreamディレ... U-112
 - 4.3 時間とデータの入出力制御 U-113
 - + 4.4 数値スキーム U-115
 - + 4.5 解法とアルゴリズム制御 U-122
- + 第5章 メッシュの生成と変換 U-129
- + 第6章 後処理 U-165
- + 第7章 モデルと物性値 U-183
- 索引 U-187

```

16 // *****
17
18 dimensions      [0 1 -1 0 0 0 0];
19
20 internalField   nonuniform List<v
21 172560
22 (
23 (10.0011 0.000864482 0.000644335)
24 (10.0011 0.00253352 0.000710604)
25 (10.0011 0.00401844 0.000844031)
26 (10.0011 0.00518292 0.00104319)
27 (10.0011 0.00588646 0.00130079)
28 (10.0011 0.00599665 0.00159511)
29 (10.0011 0.00546662 0.00187491)
30 (10.0011 0.00439979 0.00208717)
31 (10.0011 0.00303521 0.00219468)
32 (10.0012 0.00167850 0.00218411)

```

m/s

```

16 // *****
17
18 dimensions      [0 2 -2 0 0 0 0];
19
20 internalField   nonuniform List<scalar>
21 172560
22 (
23 0.775869
24 0.793849
25 0.828568
26 0.877439
27 0.936385

```

m^2/s^2
(= Pa / ρ)

Pa=N/m²
N=kgm/s²
 ρ =kg/m³

U-110

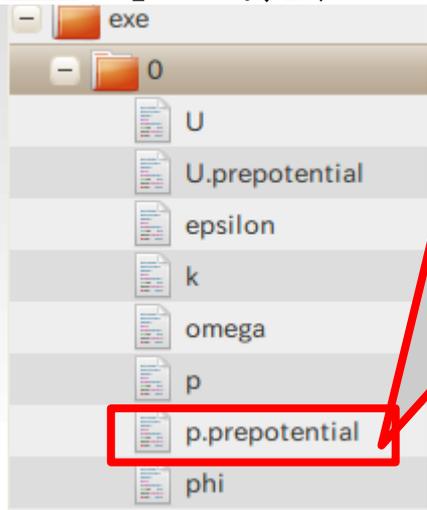
第4章 OpenFOAM

No.	物理量	SI 単位	USCS 単位
1	質量	キログラム (kg)	質量ポンド (lbm)
2	長さ	メートル (m)	フィート (ft)
3	時間		秒 (s)
4	温度	ケルビン (K)	ランキン温度 (°R)
5	物質量	モル (mol)*	ポンドモル (lbmol)
6	電流		アンペア (A)
7	光度		カンデラ (cd)

表 4.2 SI と USCS の基本単位

表 4.2 に記載するように各値は計測基準単位のそれぞれの物理量に対応しています。表 4.2 の単位系 (SI) と the United States Customary System (USCS) の基本単位ですが OpenFOAM の単位系も使えます。要求されることは入力データが選択した単位に合っているということ。特に重要なのは、OpenFOAM がいくつかの次元化された物理定数を必要とするということ。しておくことです。例えば熱力学のモデル化したある計算のための一般気体定数 R などが

境界条件 基本/派生型



```

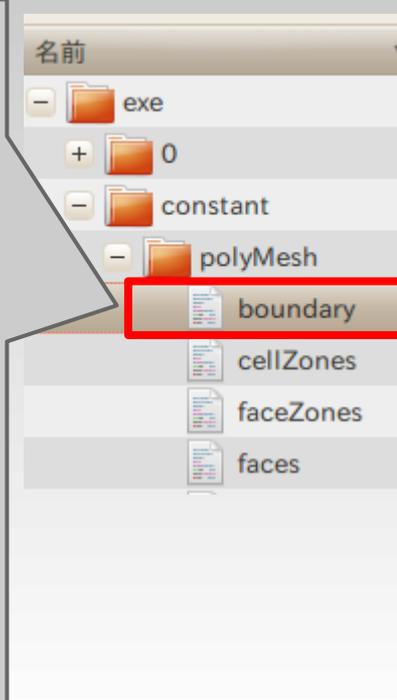
*****
...
22 {
23   type slip;
24 }
25 inlet
26 {
27   type zeroGradient;
28 }
29 Dexcs_Exportedfromblender
30 {
31   type zeroGradient;
32 }
33 outlet
34 {
35   type fixedValue;
36   value uniform 0.0;
37 }
38 }

```

```

boundary
{
  inlet
  {
    type patch;
    nFaces 196;
    startFace 1048095;
  }
  outlet
  {
    type patch;
    nFaces 196;
    startFace 1048291;
  }
  wall
  {
    type patch;
    nFaces 1624;
    startFace 1048487;
  }
  Dexcs_Exportedfromblender
  {
    type wall;
    nFaces 14086;
    startFace 1050111;
  }
}

```



- + GNU Free Documentation License U-3
- 目次 U-11
- 第1章 はじめに U-17
- + 第2章 チュートリアル U-19
- + 第3章 アプリケーションとライブラリ U-73
- + 第4章 OpenFOAMのケース U-105
- 第5章 メッシュの生成と変換 U-129
 - + 5.1 メッシュの記法 U-129
 - 5.2 境界 U-133
 - 5.2.1 パッチの形式の類型化 U-133
 - 5.2.2 基底型 U-136
 - 5.2.3 基本型 U-138
 - 5.2.4 派生型 U-138
 - + 5.3 blockMeshユーティリティを使... U-138
 - + 5.4 snappyHexMeshユーティリテ... U-147

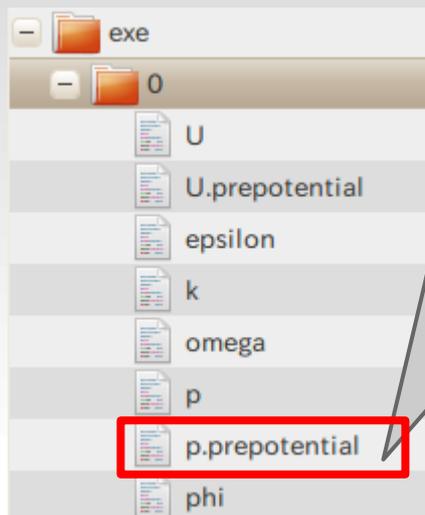
5.2.3 基本型

表 5.3 に基本型の種類を挙げます。

境界面の物理的特性

種類	物理量 ϕ に対して与える条件
fixedValue	ϕ の値が一定
fixedGradient	ϕ の勾配が一定
zeroGradient	ϕ の勾配が0
calculated	ϕ の境界条件が他の物理量から決まる
mixed	fixedValue と fixedGradient の組み合わせ、valueFraction に依存する条件
directionMixed	例えば法線方向と接線方向の異なるレベルでの組み合わせのような、テンソルの valueFraction に対しては mixed 条件

境界 基底型

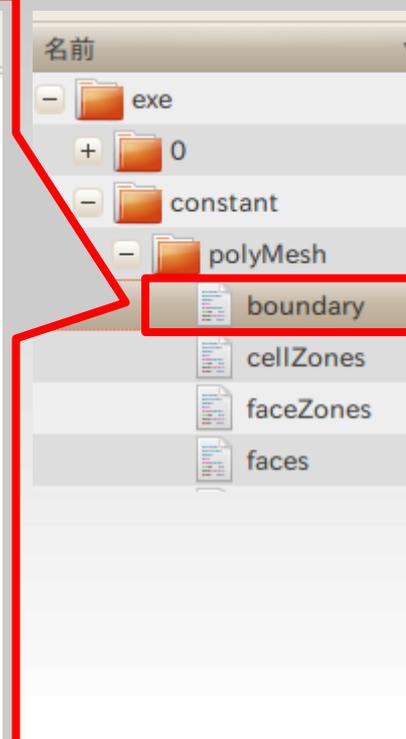


```

p.prepotential x
15 // * * * * *
16
17 dimensions [ 0 2 -2 0 0 0 ];
18 internalField uniform 0;
19 boundaryField
20 {
21   wall
22   {
23     type slip;
24   }
25   inlet
26   {
27     type zeroGradient;
28   }
29   Dexcs_Exportedfromblender
30   {
31     type zeroGradient;
32   }
33   outlet
34   {
35     type fixedValue;
36     value uniform 0.0;
37   }
38 }
    
```

```

boundary x
4
(
  inlet
  {
    type patch;
    nFaces 196;
    startFace 1048095;
  }
  outlet
  {
    type patch;
    nFaces 196;
    startFace 1048291;
  }
  wall
  {
    type patch;
    nFaces 1624;
    startFace 1048487;
  }
  Dexcs_Exportedfromblender
  {
    type wall;
    nFaces 14086;
    startFace 10501;
  }
)
    
```



メッシュ境界面の空間特性

- + GNU Free Documentation License U-3
 - 目次 U-11
 - 第1章 はじめに U-17
- + 第2章 チュートリアル U-19
- + 第3章 アプリケーションとライブラリ U-73
- + 第4章 OpenFOAMのケース U-105
- 第5章 メッシュの生成と変換 U-129
 - + 5.1 メッシュの記法 U-129
 - 5.2 境界 U-133
 - 5.2.1 パッチの形式の類型化 U-133
 - 5.2.2 基底型 U-136
 - 5.2.3 基本型 U-138
 - 5.2.4 派生型 U-138
 - + 5.3 blockMeshユーティリティを使... U-138
 - + 5.4 snappyHexMeshユーティリテ... U-147
 - + 5.5 メッシュの変換 U-155

種類	意味
patch	一般的なパッチ
symmetryPlane	対称面
empty	2次元形状の前後の面
wedge	軸対称形状のための、くさび型の前後
cyclic	周期境界面
wall	壁面（乱流の壁関数に使用）
processor	並列計算時のプロセッサ間の境界

表 5.2 基底型の境界の種類

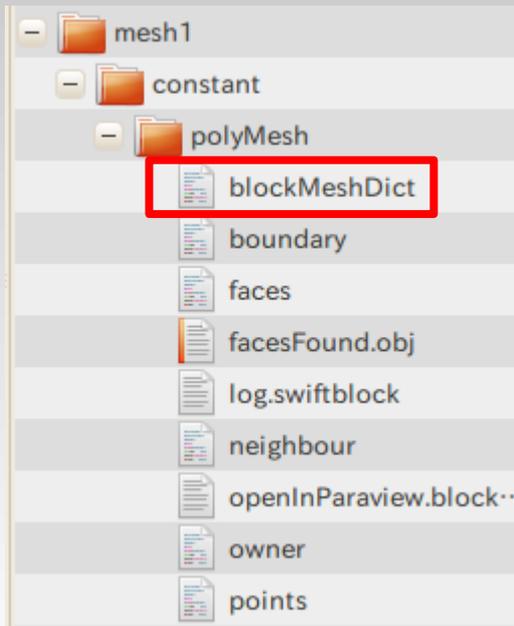
symmetryPlane 対称面

empty OpenFOAM が常に 3次元で形状を生成する一方で、2次元（1次元）可能です。そのためには、解が必要とされない3番目（2番目）の次元に法

blockMesh

- + GNU Free Documentation License U-3
- 目次 U-11
- 第1章 はじめに U-17
- + 第2章 チュートリアル U-19
- + 第3章 アプリケーションとライブラリ U-73
- + 第4章 OpenFOAMのケース U-105
- 第5章 メッシュの生成と変換 U-129
 - + 5.1 メッシュの記法 U-129
 - + 5.2 境界 U-133
 - 5.3 blockMeshユーティリティを使... U-138
 - 5.3.1 blockMeshDictファイル... U-141
 - 5.3.2 複数のブロック U-144
 - 5.3.3 8頂点までのブロックの... U-146
 - 5.3.4 bl...

- + 5.4 snappy
- + 5.5 メッシュ
- + 5.6 異なるシ
- + 第6章 後処理
- 第7章 モデルと
- + 7.1 熱物理
- + 7.2 乱流モ
- 索引



```
blockMeshDict
13
14 convertToMeters 1.0;
15
16 vertices
17 (
18     (1.0 -1.5 2.200000047683716)
19     (1.0 -1.5 -1.7999999523162842)
20     (-7.0 -1.5 -1.7999999523162842)
21     (-6.999999046325684 -1.5 2.2000010013580322)
22     (1.0000009536743164 -5.5 2.200000047683716)
23     (-7.0 -5.5 2.200000047683716)
24     (-7.000000953674316 -5.5 -1.7999999523162842)
25     (0.9999989867210388 -5.5 -1.8000010251998901)
26 );
27 blocks
28 (
29     hex (3 5 6 2 0 4 7 1) (14 14 29) simpleGrading (1
30 );
31
32 patches
33 (
34     patch inlet
35     (
36         (5 6 2 3)
```

5.3.1 blockMeshDict ファイルの記述

blockMeshDict ファイルは、表 5.5 で説明されているキーワードを使用す。convertToMeters キーワードは、メッシュ記述におけるすべての頂点の尺度因子を指定します。例えば、

```
convertToMeters 0.001;
```

は、すべての座標に 0.001 をかけることを意味します。すなわち、blockMeshDict で指定された値が mm になります。

5.3.1.1 頂点

メッシュのブロックの頂点は、vertices と名づけられた標準のリストとして以下のように与

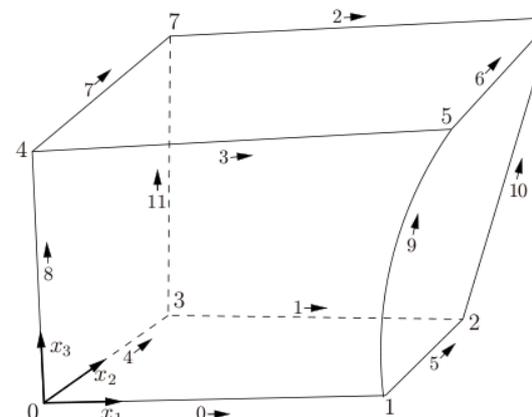
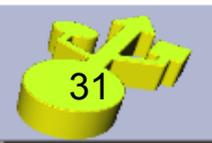
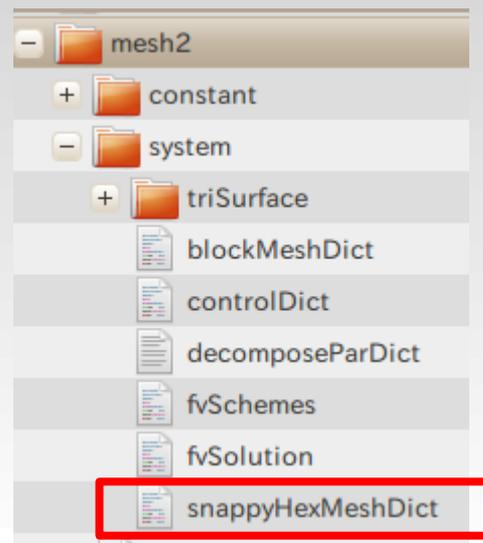


図 5.4 ひとつのブロック



snappyHexMesh

- + GNU Free Documentation License U-3
- 目次 U-11
- 第1章 はじめに U-17
- + 第2章 チュートリアル U-19
- + 第3章 アプリケーションとライブラリ U-73
- + 第4章 OpenFOAMのケース U-105
- 第5章 メッシュの生成と変換 U-129
 - + 5.1 メッシュの記法 U-129
 - + 5.2 境界 U-13~
 - + 5.3 blockMeshユーティリティを使... U-13
 - 5.4 snappyHexMeshユーティリテ... U-14
 - 5.4.1 snappyHexMeshによる... U-14
 - 5.4.2 六面体基礎メッシュの作成 U-14
 - 5.4.3 面と輪郭に合わせたセル... U-14
 - 5.4.4 セルの除去 U-15
 - 5.4.5 特定領域内のセルの分割 U-15
 - 5.4.6 面へのスナップ U-15
 - 5.4.7 メッシュレイヤ U-15
 - 5.4.8 メッシュの品質制御 U-15
 - + 5.5 メッシュの変換 U-15
 - + 5.6 異なるジオメトリ間のフィールド... U-16
- + 第6章 後処理 U-16
- 第7章 モデルと物性値 U-18
 - + 7.1 熱物理モデル U-18
 - 327.2 乱流モデル U-18
 - 索引 U-18



```

snappyHexMeshDict
14 castellatedMesh true;
15 snap true;
16 addLayers true;
17
18 geometry
19 {
20     Dexcs.stl
21     {
22         type triSurfaceMesh;
23         name Dexcs;
24     }
25     Box1.stl
26     {
27         type triSurfaceMesh;
28         name Box1;
29     }
30 };
31
32 castellatedMeshControls
33 {
34     features
35     (
36

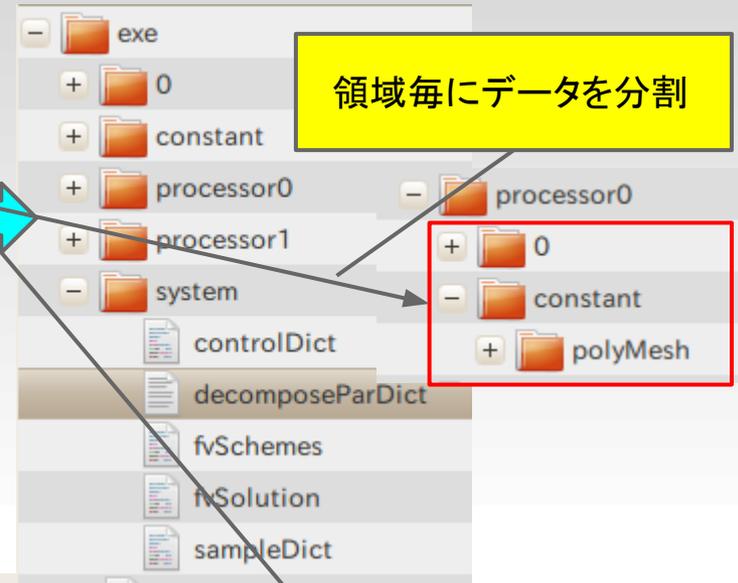
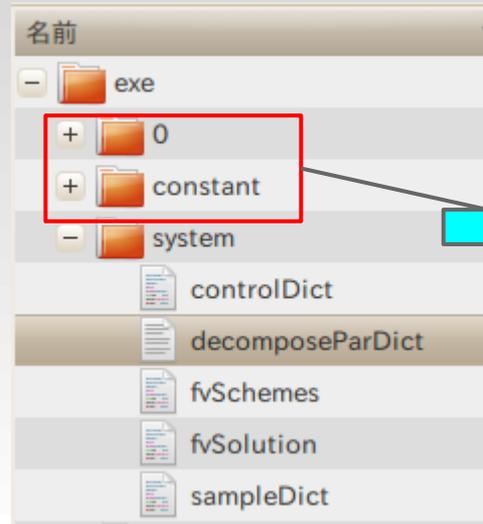
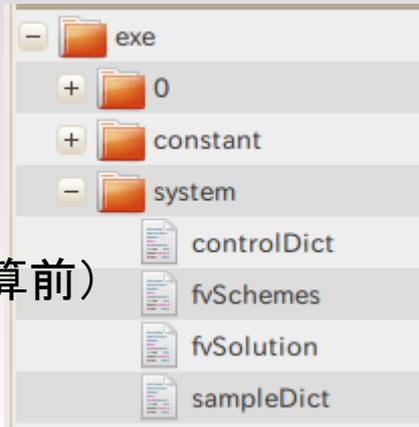
```

キーワード	意味
castellatedMesh	ギザギザのメッシュを作成するかどうか
snap	表面のスナップの有無
doLayers	レイヤの追加の有無
mergeTolerance	初期メッシュの有界ボックスの比として許容値をまとめる
debug	中間メッシュと画面プリントの記述の制御
	最終メッシュのみ記述
	中間メッシュの記述
	後処理のため cellLevel を付けた volScalarField を記述
geometry	.obj ファイルとして現在の交点を記述
castellatedMeshControls	表面に使用した全てのジオメトリのサブディクショナリ
snapControls	城壁メッシュ制御のサブディクショナリ
addLayersControls	表面スナップ制御のサブディクショナリ
meshQualityControls	レイヤ追加制御のサブディクショナリ
	メッシュ特性制御のサブディクショナリ

表 5.7 snappyHexMeshDict の最上位のキーワード

並列計算

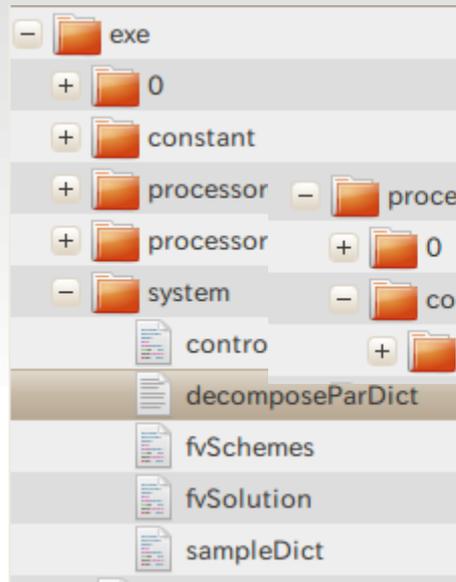
(計算前)



```
1 // * * * * * //
2 // * * * * * //
3 FoamFile
4 {
5     version 0.5;
6     format ascii;
7     root "ROOT";
8     case "CASE";
9     class dictionary;
10    object nix;
11 }
12 method scotch;
13 numberOfSubdomains 2;
14 scotchCoeffs
15 {
16 }
```

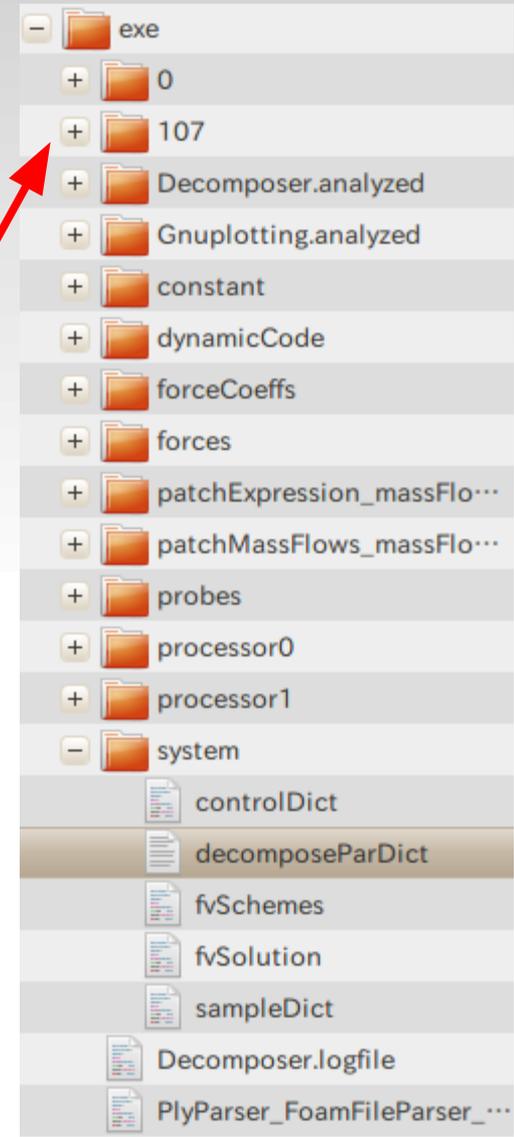
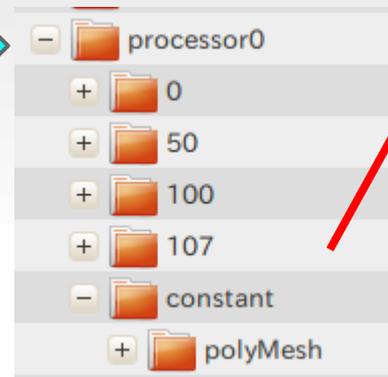
DEXCSランチャーでは、このファイルを自動作成しています

並列計算



```
$ reconstructPar -latestTime
```

引数(-latestTime)
なしで全ステップ



```
$ mpirun -np 2 simpleFoam -parallel
```

並列数

ソルバー名

後処理 (paraFoam)

+ GNU Free Documentation License	U-3
目次	U-11
第1章 はじめに	U-17
+ 第2章 チュートリアル	U-19
+ 第3章 アプリケーションとライブラリ	U-73
+ 第4章 OpenFOAMのケース	U-105
+ 第5章 メッシュの生成と変換	U-129
- 第6章 後処理	U-165
- 6.1 paraFoam	U-165
6.1.1 paraFoamの概要	U-166
6.1.2 Propertiesパネル	U-167
6.1.3 Displayパネル	U-167
6.1.4 ボタンツールバー	U-168
6.1.5 表示の操作	U-168
6.1.6 コンタのプロット	U-170
6.1.7 ベクトルのプロット	U-171
6.1.8 流線	U-171
6.1.9 画像の出力	U-171
6.1.10 アニメーション出力	U-172
6.2 Fluentによる後処理	U-172
6.3 Fieldviewによる後処理	U-174
+ 6.4 EnSightによる後処理	U-174
6.5 データのサンプリング	U-176
+ 6.6 ジョブのモニタと管理	U-179
- 第7章 モデルと物性値	U-183
+ 7.1 熱物理モデル	U-183
+ 7.2 乱流モデル	U-186
索引	U-187

Display パネル は、色など、選択されたモジュールの可視化の描画をコントロールします。
Information パネル はメッシュのジオメトリとサイズのようなケースの統計値を表示します。

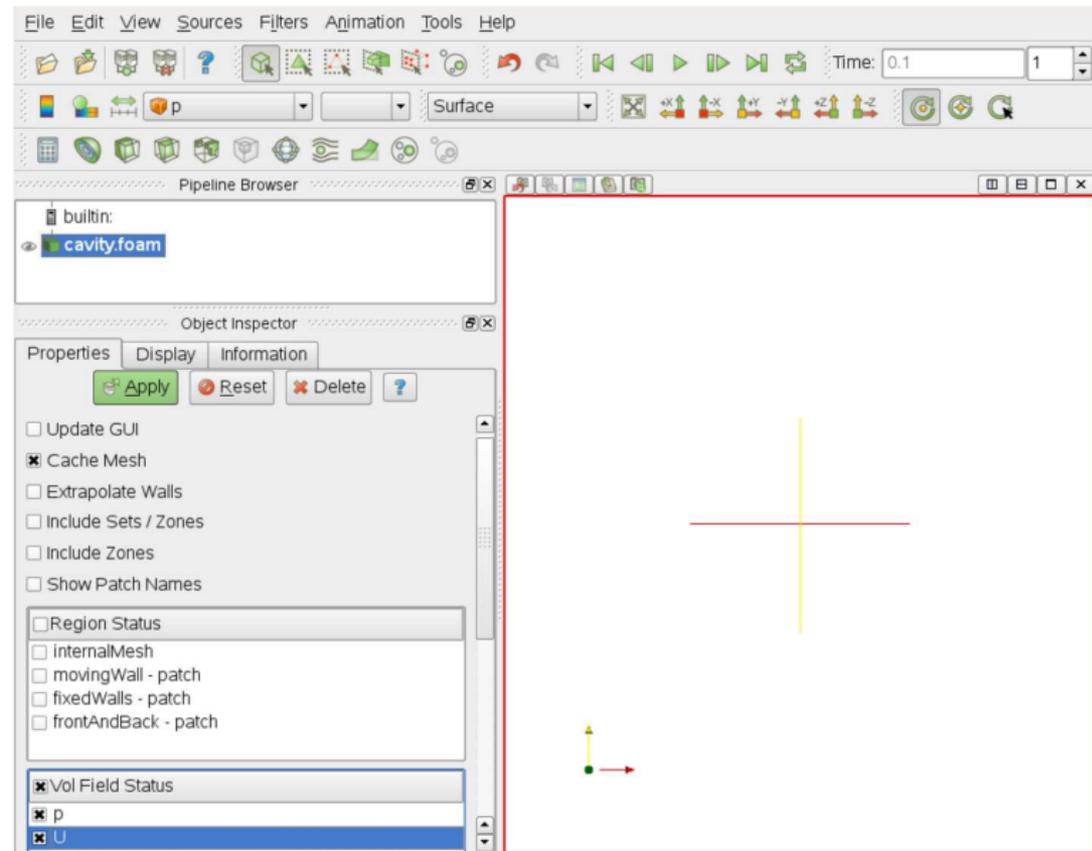
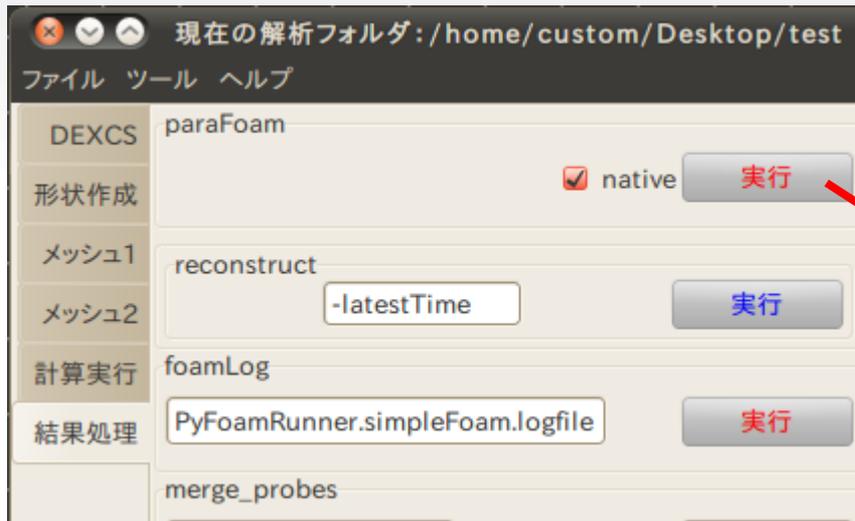


図 6.1 paraFoam の画面

ParaView / paraFoam

paraFoam は,OpenFOAM で提供されている読み込みモジュールを用いて,ParaView を立ち上げる厳密なスクリプトです

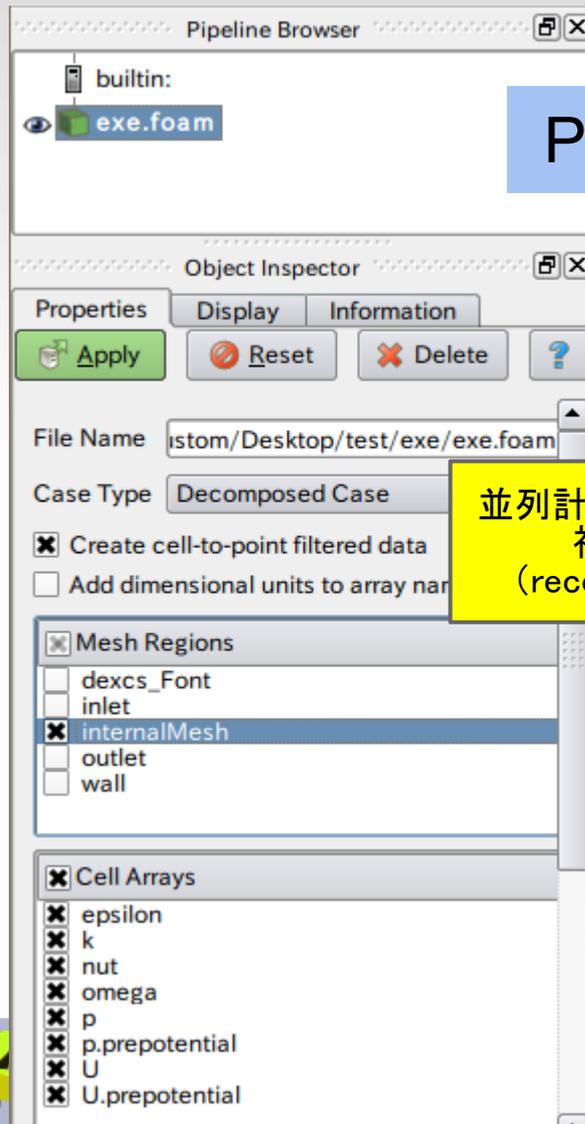


ParaView

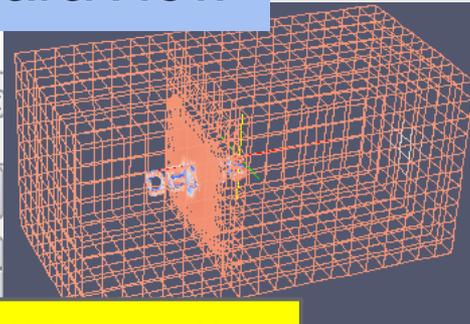


paraFoam

ParaView / paraFoam

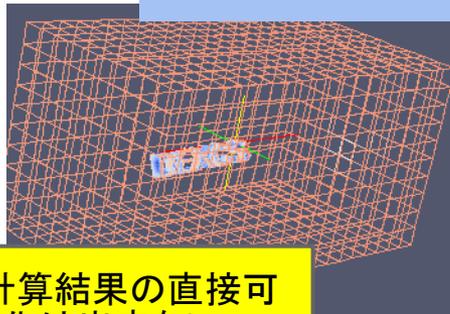


ParaView

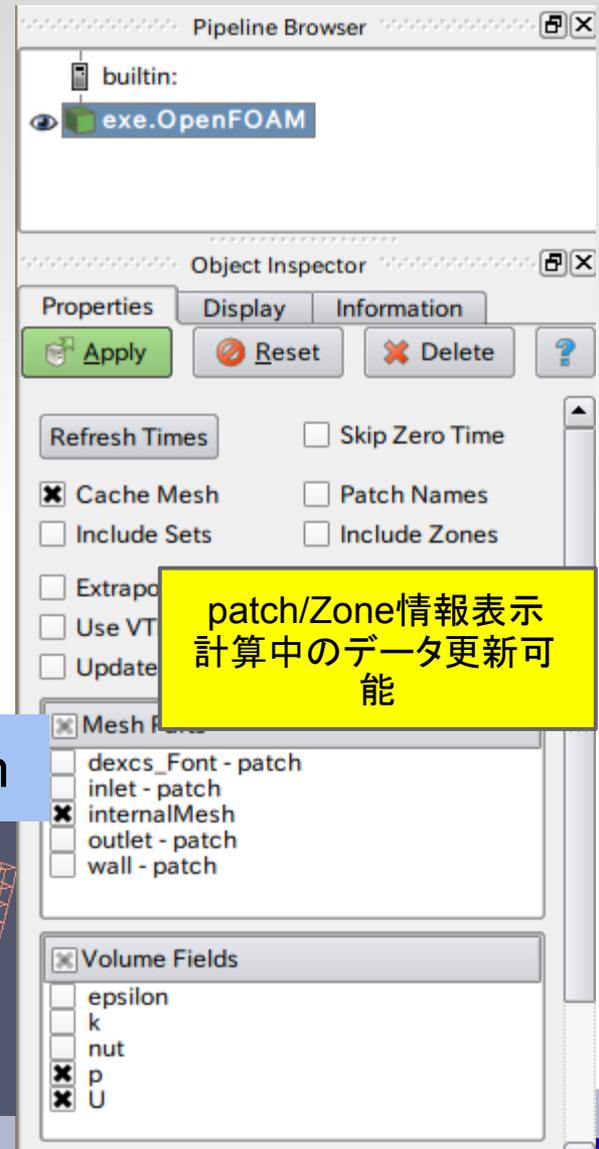


並列計算結果の直接可視化が可能
(reconstruct 不要)

paraFoam



並列計算結果の直接可視化は出来ない
(reconstruct 要)

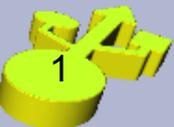


patch/Zone情報表示
計算中のデータ更新可能

DEXCS for OpenFOAM®の体験2-1

1. OpenFOAMの実践的活用法
2. blenderとSwiftツールの説明
3. 簡単なモデル変更・メッシュ作成演習
4. 標準チュートリアルケースの実行
5. 標準チュートリアルベースのメッシュ変更

オープンCAEコンサルタント OCSE^2
代表 野村 悦治



OpenFOAMの実践的活用法

(標準ソルバーが使える場合)

1. メッシュ作成

a. Swiftツール ⇒ snappyHexMesh

b. (他のソフトで作成) ⇒ ###ToFoam

2. 標準チュートリアルケースを精査し、自分が解きたい現象・モデルに近いものを探し出す。

3. 上記チュートリアルケースのメッシュを自前で作成したメッシュに置き換える。

4. モデルパラメタの細部詳細を整合

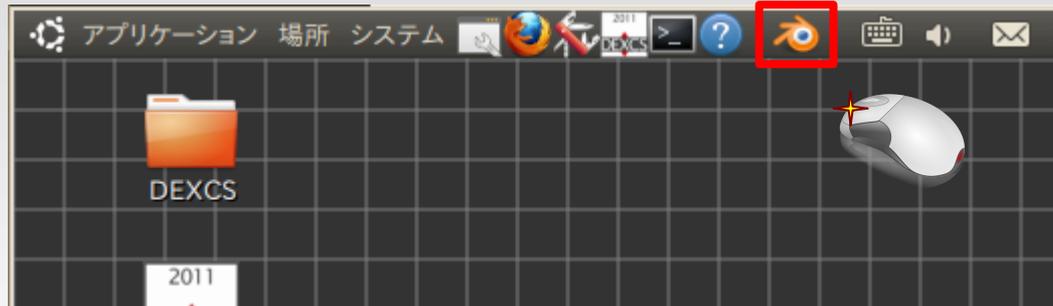
体験
2-1
で実践

体験
2-2
で実践

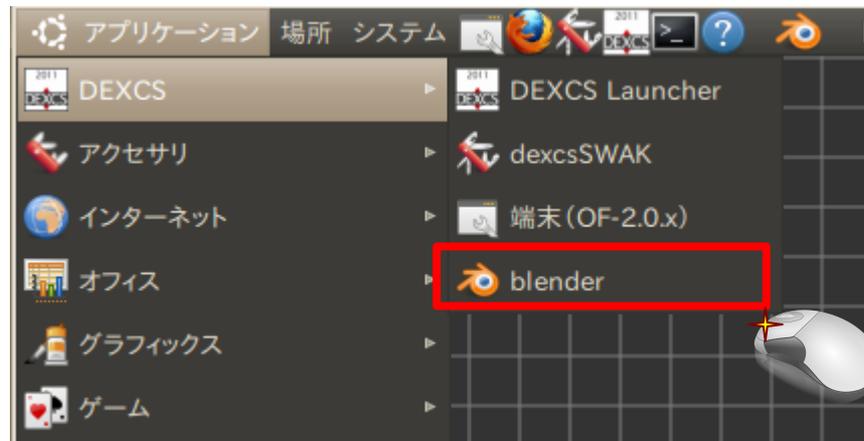
blenderとswiftツールの説明



blenderの起動方法

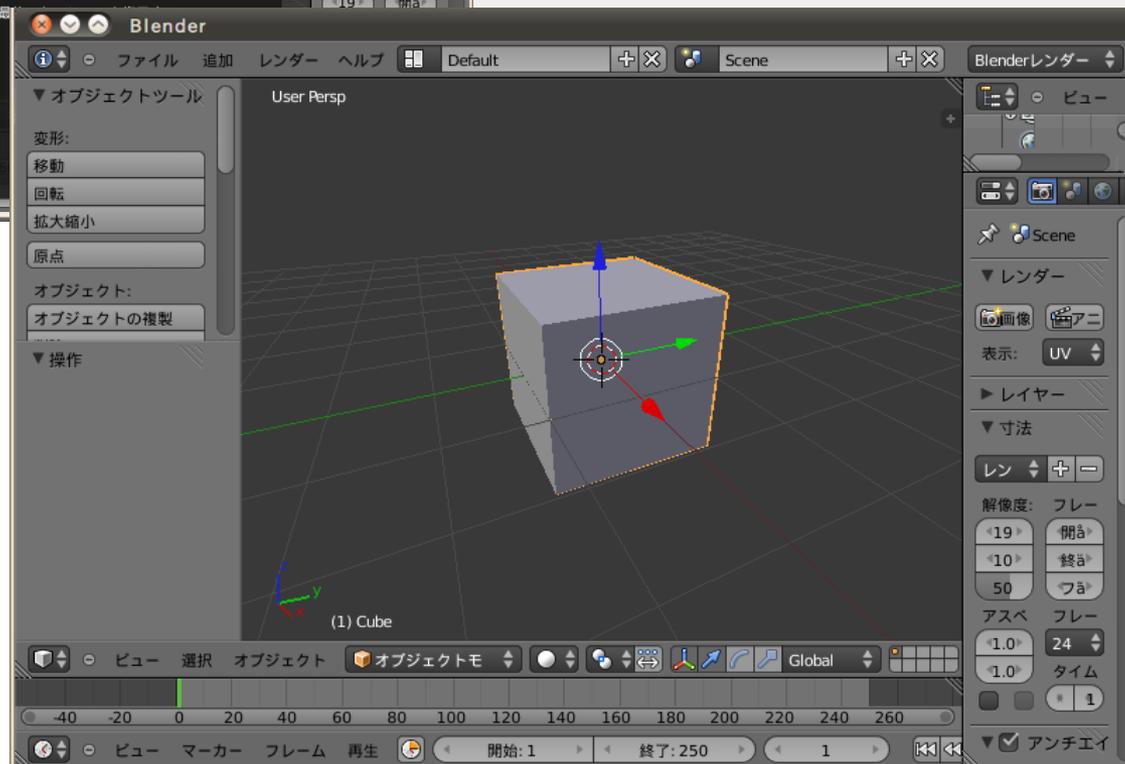
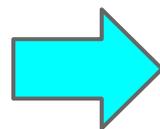


または





スプラッシュ画面
以外のどこかを
クリック



blenderの入門サイト

http://www.cgradproject.com/Blender_Tutorial/tutorial_Blender_tute_menu.html

CGrad Project. ▼ Open Menu

本当に初めての方のために作成したBlenderチュートリアル for 2.6

目次



- 1.はじめに
- 2.画面説明
- 3.マウス操作
- 4.ウィンドウ操作
- 5.3D View上の視点操作
- 6.モード
- 7.図形の追加
- 8.削除・Undo
- 9.ファイルの保存
- 10.マニピュレータ
- 11.メッシュ編集1
- 12.メッシュ編集2
- 13.メッシュ編集3
- 14.ライティング
- 15.カメラ設定
- 16.レンダリング

次の章へ

Blender 2.6 入門

- 初心者による初心者のためのBlenderチュートリアル 3DCGのモデリングからアニメーションまで -

トップ ご利用にあたって 個人情報保護に関する方針 お問い合わせ 特集(不定期連載)

Blender 2.4系はこちら

入門

- 01 Blenderのインストール
 - インストールの前に
 - Blenderのインストール(Windows)
 - Blenderのインストール(Linux)
- 02 基礎知識
 - Blenderでどんなことができるのか
 - 3DCG制作の流れ
 - 画面の見かた
 - 画面レイアウトとウィンドウ操作
 - コマンドの入力方法
 - レイヤおよびその操作
 - 様々な描画方法
 - 投影方法と視点
 - 様々な座標系
- 03 簡単な3DCGの制作
 - てんとう虫を描いてみよう
 - ライトとカメラを別レイヤに移動
 - モデリング
 - 体のモデリング(オブジェクトモード)
 - 体のモデリング(エディットモード)
 - 体のモデリング(顔部)
 - 体のモデリング(前脚と後脚)
 - 脚のモデリング(中脚)
 - 形状をなめらかにする
 - オブジェクトとメッシュの名称変更
 - マテリアルで質感を表現する
 - マテリアルの作成
 - マテリアルのメッシュへの割り当て
 - マテリアルへの質感の設定
 - テクスチャで模様を表現する
 - テクスチャとは
 - UVマップの作成
 - UV展開
 - イメージの作成
 - テクスチャの作成

Blender 2.6 入門

Blenderでどんな3DCG(静止画・アニメーション)を制作できるのか

トップ > 基礎知識 > Blenderでどんなことができるのか

Ads by Google [Blender](#) [Blender 3D](#) [3D Blender](#) [CGソフト](#)

Blenderでできること

Blenderは、3DCG(静止画・アニメーション)や3Dゲームを開発するための総合開発環境です。

Blenderは残念ながら『プロが利用する高額のソフトウェアに比べて、制作した3DCGが見劣りする』と評価されることがあります。それは事実かも知れませんが、それは、妥協の許されないとても高いレベルの3DCGを制作しようとした場合の話であって、Blenderでも高品質のコンテンツを制作することができます。

PAGE TOP

Blenderで制作した3DCGの紹介

Blenderは、無料で利用できるソフトウェアで制作したとは思えないような、高品質な3DCGを制作することができます。静止画は写真と見分けがつかない品質のものを制作することができますし、アニメーションについても劇場で上映してもおかしくない品質のものを制作することができます。

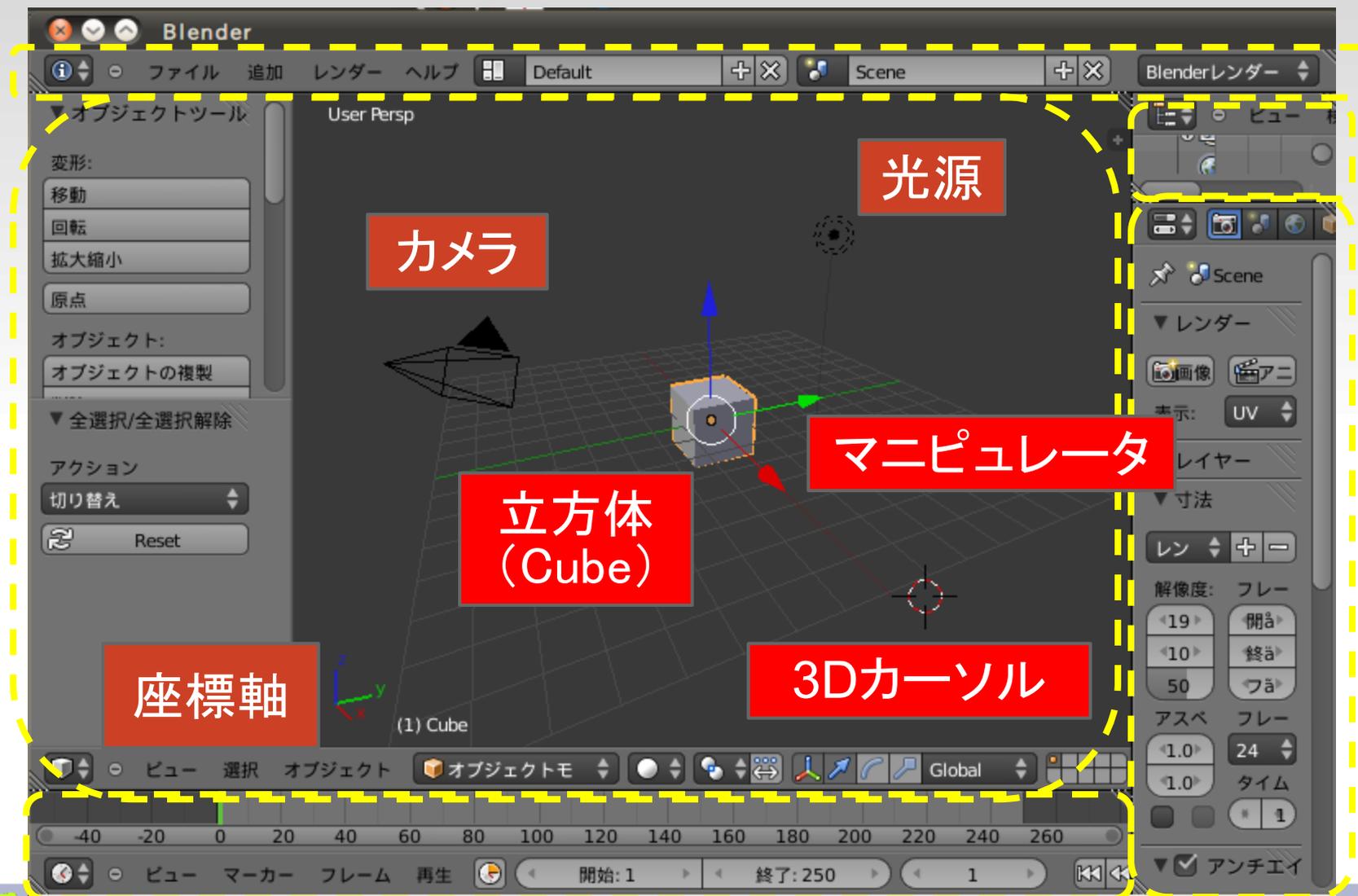
ここでは、Blenderで制作した静止画やアニメーションが掲載されているギャラリーを紹介いたします。Blenderでどんな3DCGが制作できるのか、品質はどの程度なのか、自分の目で確かめてください。

アートギャラリー

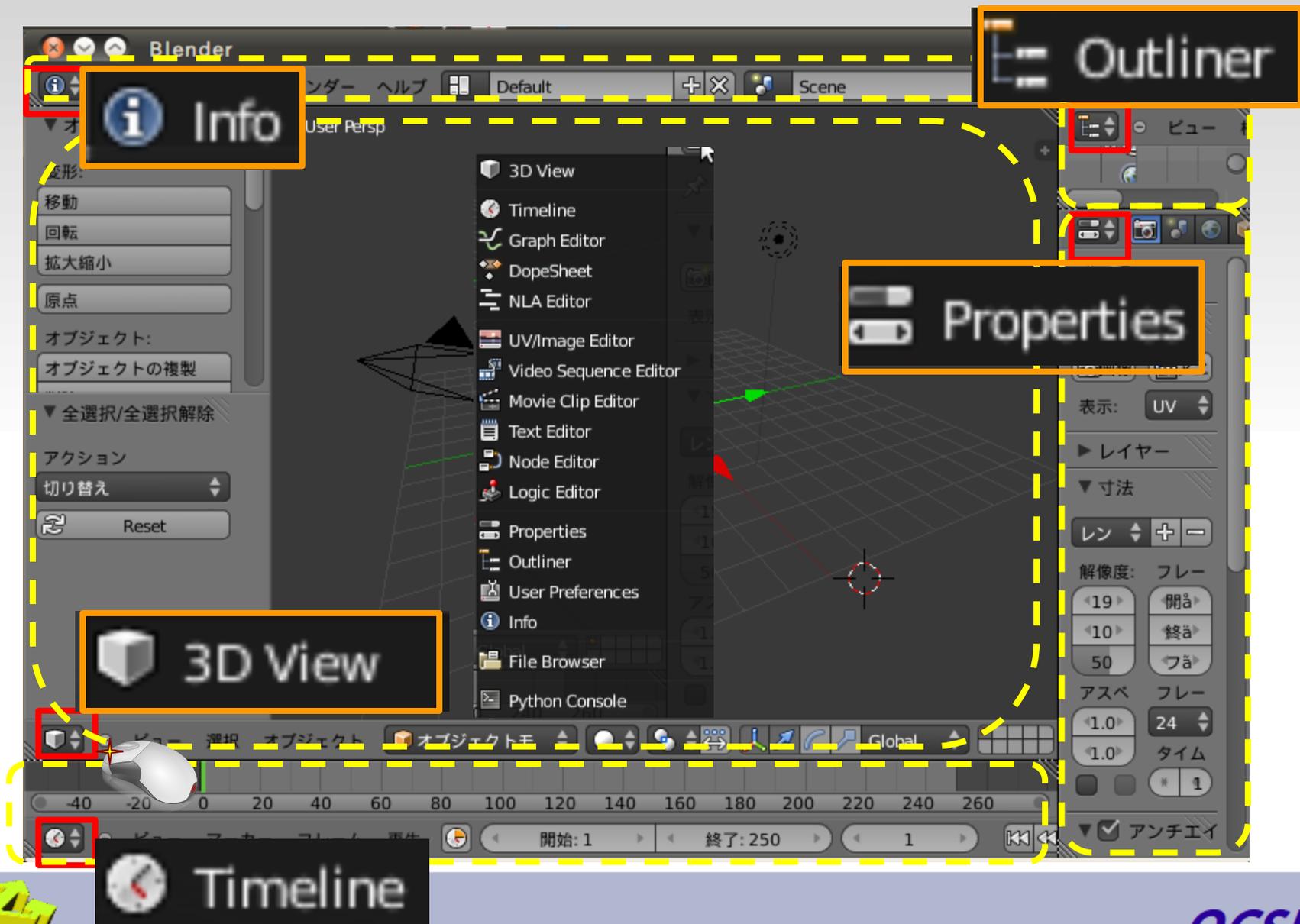
http://26.blender3d.biz/basis_whatisblender.html



画面の説明



画面配置と画面タイプの切り替え



マウス操作

 3D View

[テンキー]

■ 3D画面上

- ホイール回転⇒拡大・縮小 [+] [-]
 - + Shiftキー⇒上下移動 Ctrl + [8], [2]
 - + Ctrlキー⇒左右移動 Ctrl + [4], [6]
- ホイールボタンを押しながらドラッグ⇒回転
- **右ボタンクリック⇒オブジェクトの選択(選択解除はAキー)**
- 右ボタンを押しながらドラッグ⇒選択オブジェクトの移動
- 左ボタンクリック(何も無い場所)⇒3Dカーソル移動
- **左ボタンクリック(マニピュレータ上) & ドラッグ⇒オブジェクト移動**
Undo (Ctrl+Z) で元に戻す

■ その他の画面

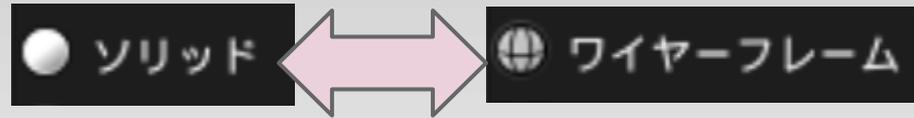
- 左ボタンクリック⇒選択・実行
- 右ボタンクリック⇒オプションメニュー表示

モード

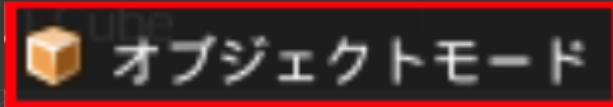


シェーディング

Zキーにて交互切換



Tabキーにて
交互切換



▼ オブジェクトツール
変形:
移動
回転
拡大縮小
原点
オブジェクト:
オブジェクトの複製
▼ 編集モードに切り替え

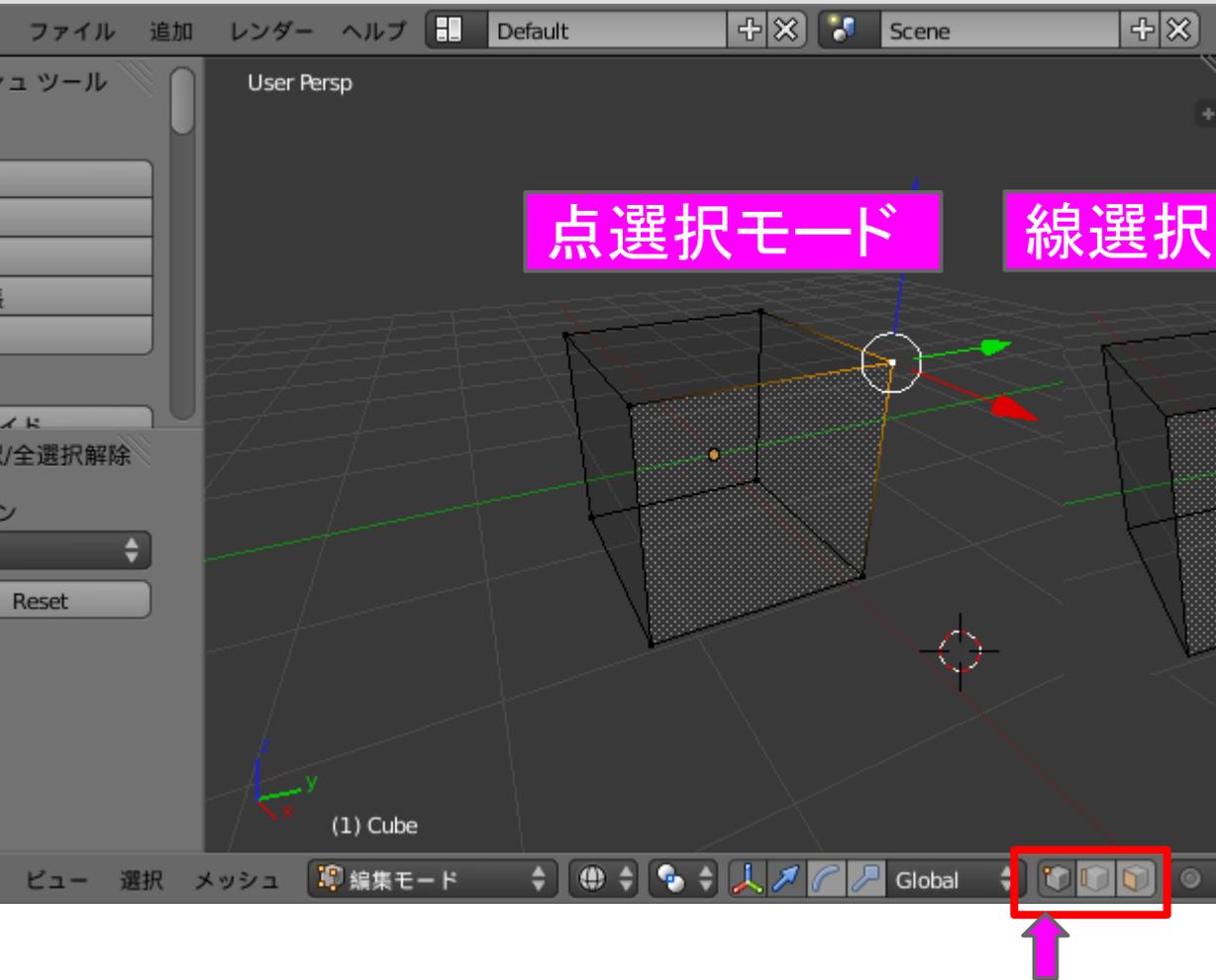
User Persp

3Dビューのシェーディング
● テクスチャ
● ソリッド
● ワイヤーフレーム
● バウンドボックス

(1) Cube

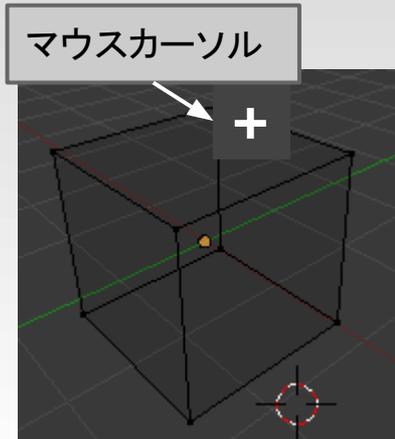
ビュー 選択 オブジェクト オブジェクトモ

編集モード / 選択モード

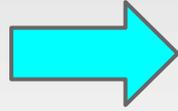


選択点の追加・解除(編集/点選択モード)

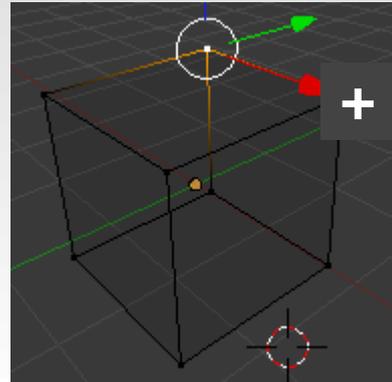
非選択



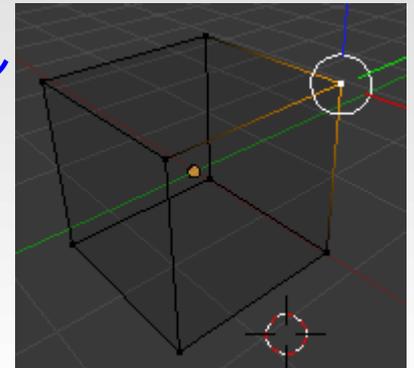
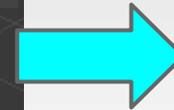
マウス
右ボタン



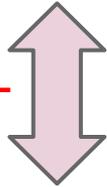
Aキー



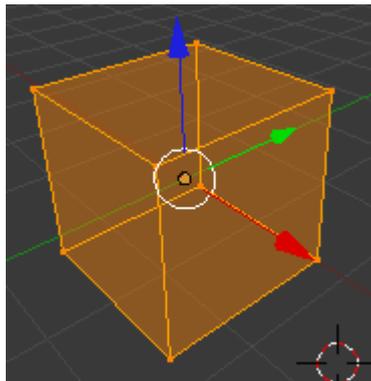
マウス
右ボタン



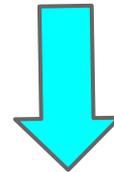
Aキー



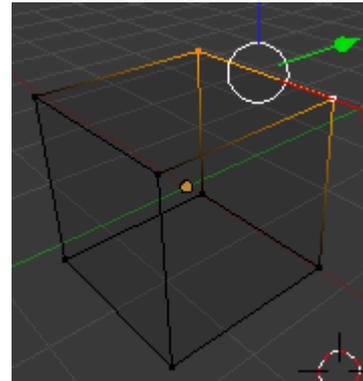
全選択



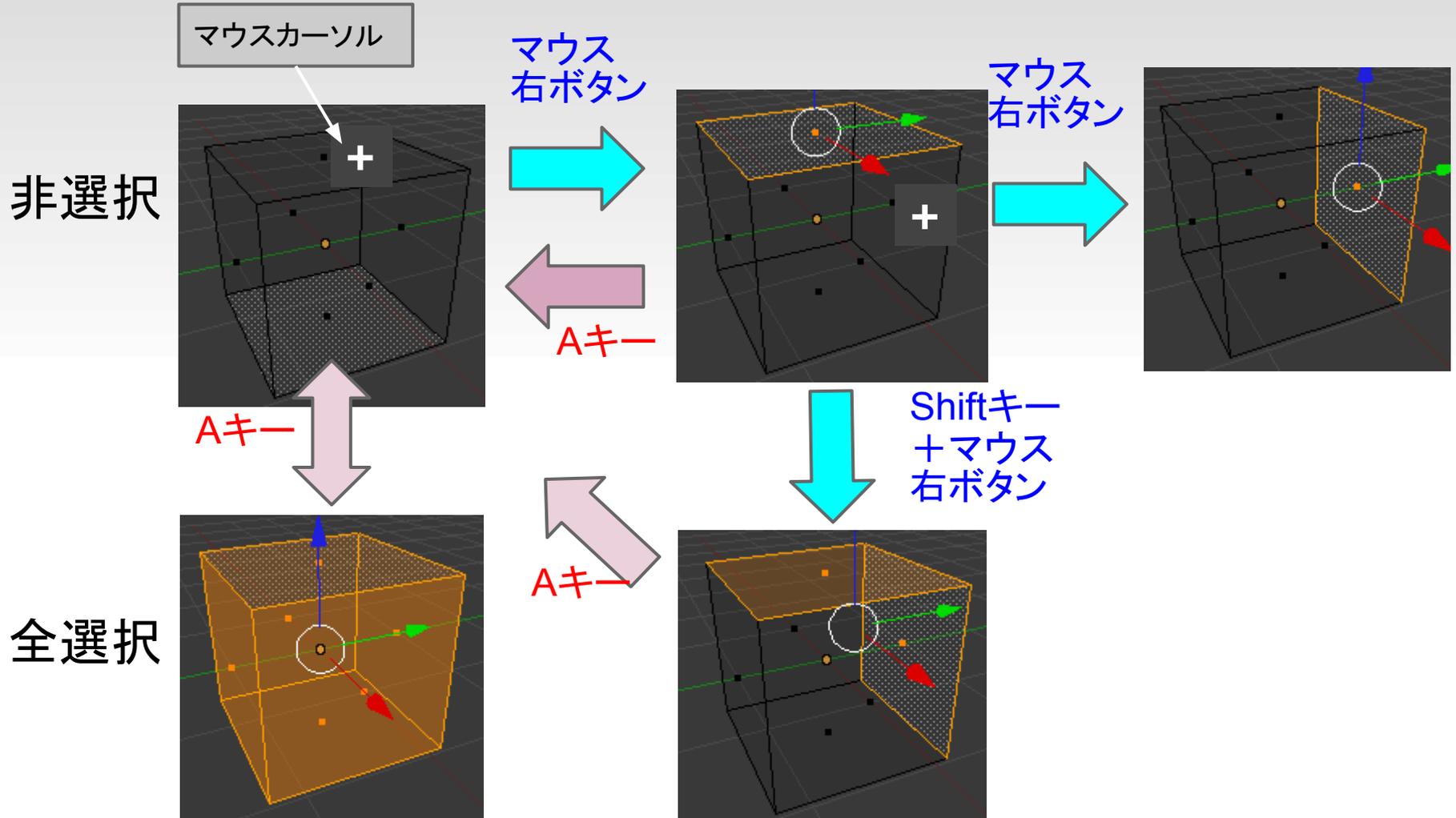
Aキー



Shiftキー
+マウス
右ボタン



選択点の追加・解除(編集/面選択モード)



ツールシェルフ

The image shows a Blender 3D Viewport with a cube in the center. On the left, the 'Object Tools' panel is visible with various manipulation tools like Move, Rotate, and Scale. A red arrow points from the 'View' button in the bottom toolbar to the 'Tools Shelf' in the context menu. The context menu is open, showing options like 'Toggle Layer Visibility', 'Zoom to Border', and 'Toggle Perspective/Orthographic'. The 'Tools Shelf' option is highlighted with a red box. A pink box with the text 'Tキー' (T key) is also present.

3D View

▼ オブジェクトツール

User Persp

変形:

移動
回転
拡大縮小

原点

オブジェクト:

オブジェクトの複製
削除
接続

シェーディング:

スムーズ フラット

キーフレーム:

キー挿入 削除

モーションパス:

パスを計算

▶ レイヤー移動

(0) Cube

グローバルビュー/ローカルビュー Numpad 7

全てのレイヤーを表示

ボーダーにズーム... Shift B

ボーダーでクリッピング... Alt B

視点を揃える

視点の操作

透視投影/平行投影 Numpad 5

カメラ設定

左 Ctrl Numpad 3

右 Numpad 3

後 Ctrl Numpad 1

前 Numpad 1

下 Ctrl Numpad 7

上 Numpad 7

カメラ Numpad 0

ツールシェルフ T

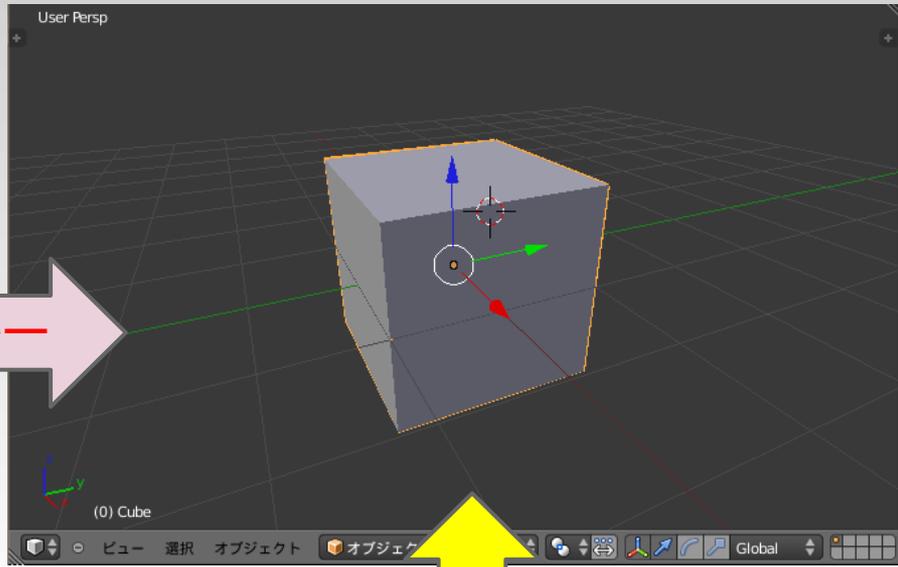
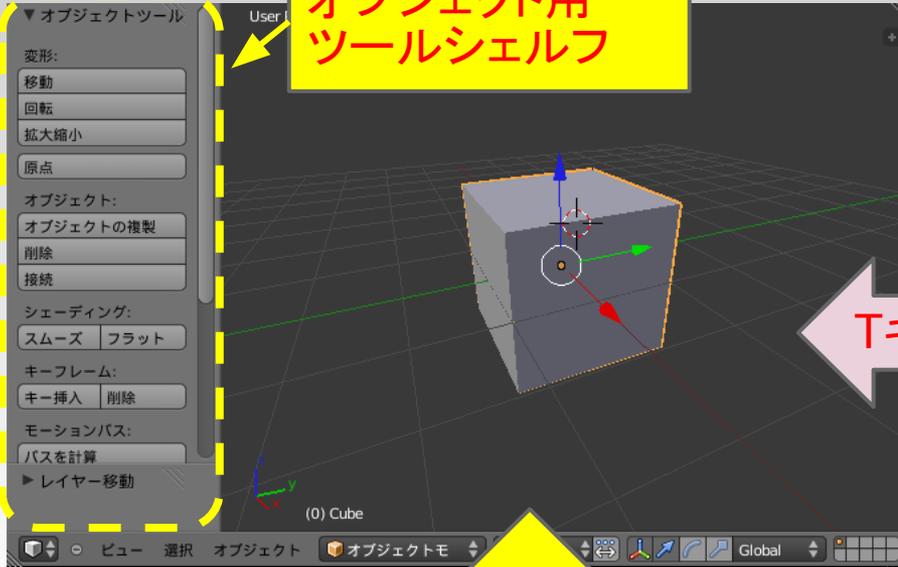
プロパティ N

ビュー 選択 オブジェクト オ

ビュー

Tキー

オブジェクト用
ツールシェルフ



Tキー

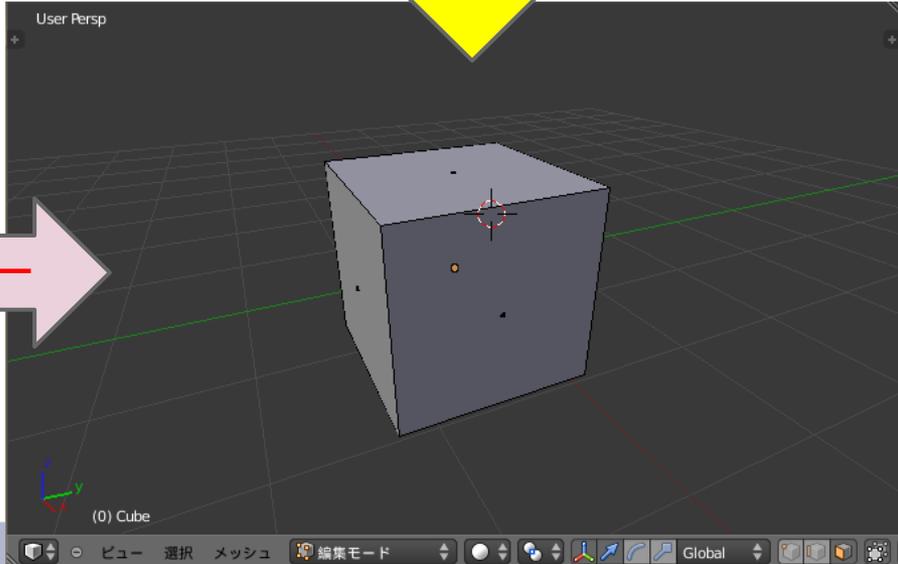
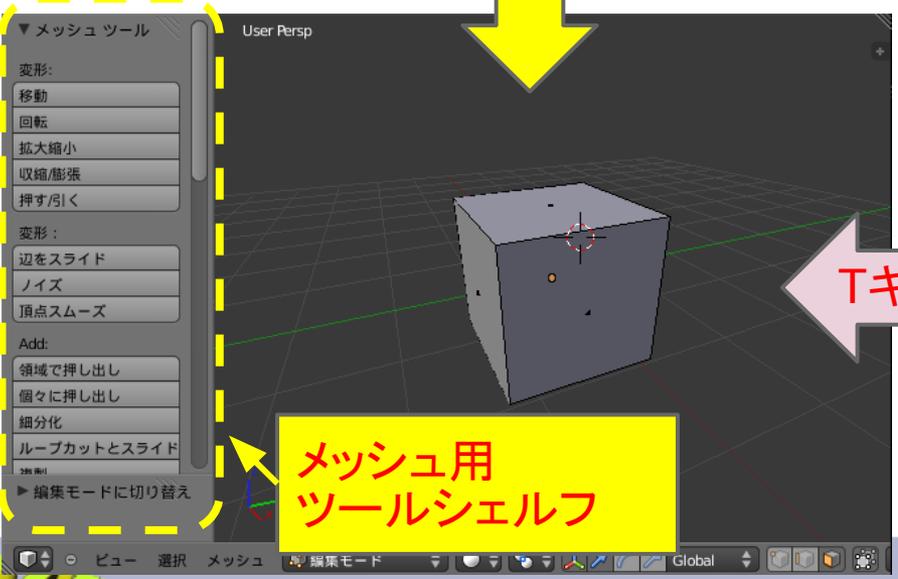


Tabキー



Tabキー

メッシュ用
ツールシェルフ



Tキー

プロパティ(数値パッド)

3D View

▼ メッシュ ツール

変形:

- 移動
- 回転
- 拡大縮小
- 収縮/膨張
- 押す/引く

変形:

- 辺をスライド
- ノイズ
- 頂点スムーズ

Add:

- 領域で押し出し
- 個々に押し出し
- 細分化
- ループカットとスライド

▶ 編集モードに切り替え

User Persp

(0) Cube

グローバルビュー/ローカルビュー Numpad 7

全てのレイヤーを表示

ボーダーにズーム... Shift B

ボーダーでクリッピング... Alt B

視点を揃える

視点の操作

透視投影/平行投影 Numpad 5

カメラ設定

左 Ctrl Numpad 3

右 Numpad 3

後 Ctrl Numpad 1

前 Numpad 1

下 Ctrl Numpad 7

上 Numpad 7

カメラ Numpad 0

ツールシェルフ T

プロパティ (0) Cube N

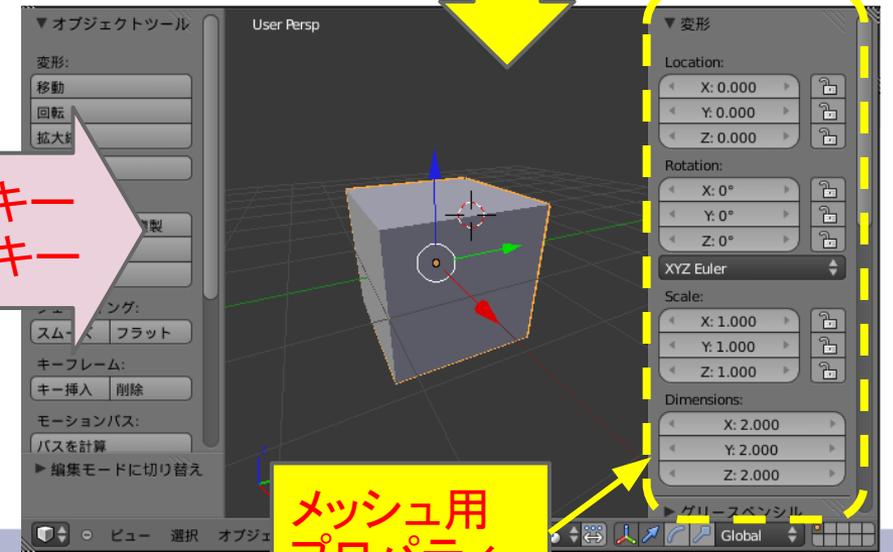
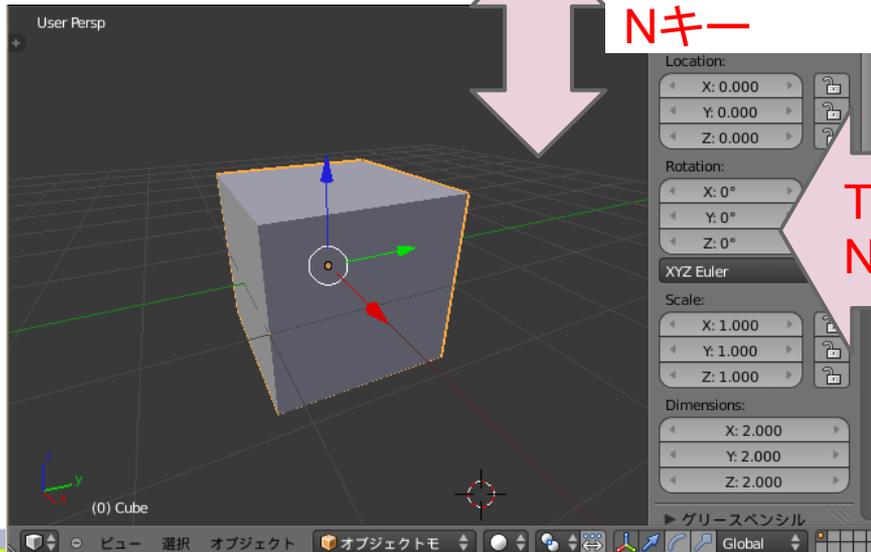
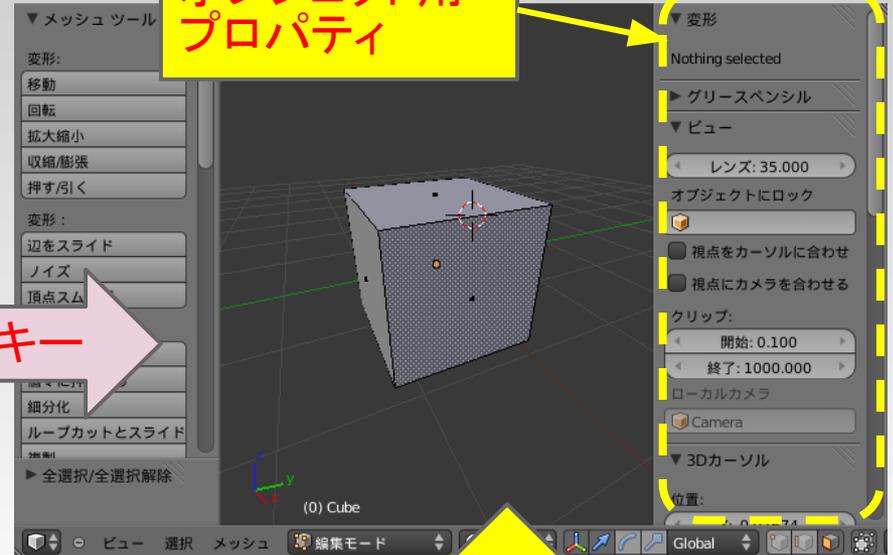
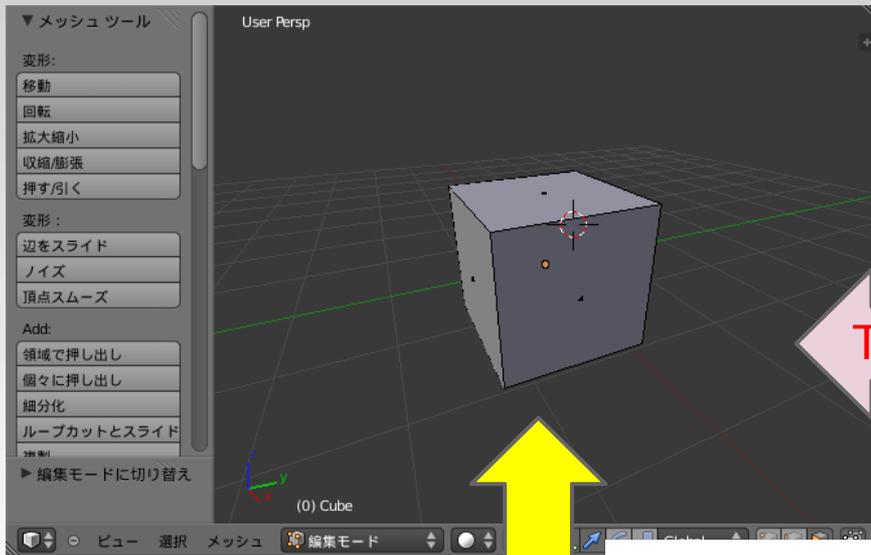
ビュー 選択 オブジェクト オブジェクト

ビュー 選択 メッシュ 編集モード Global

21

OCSE^2

Nキー



オブジェクト用
プロパティ

Tキー

Tabキー
Tキー
Nキー

Tabキー

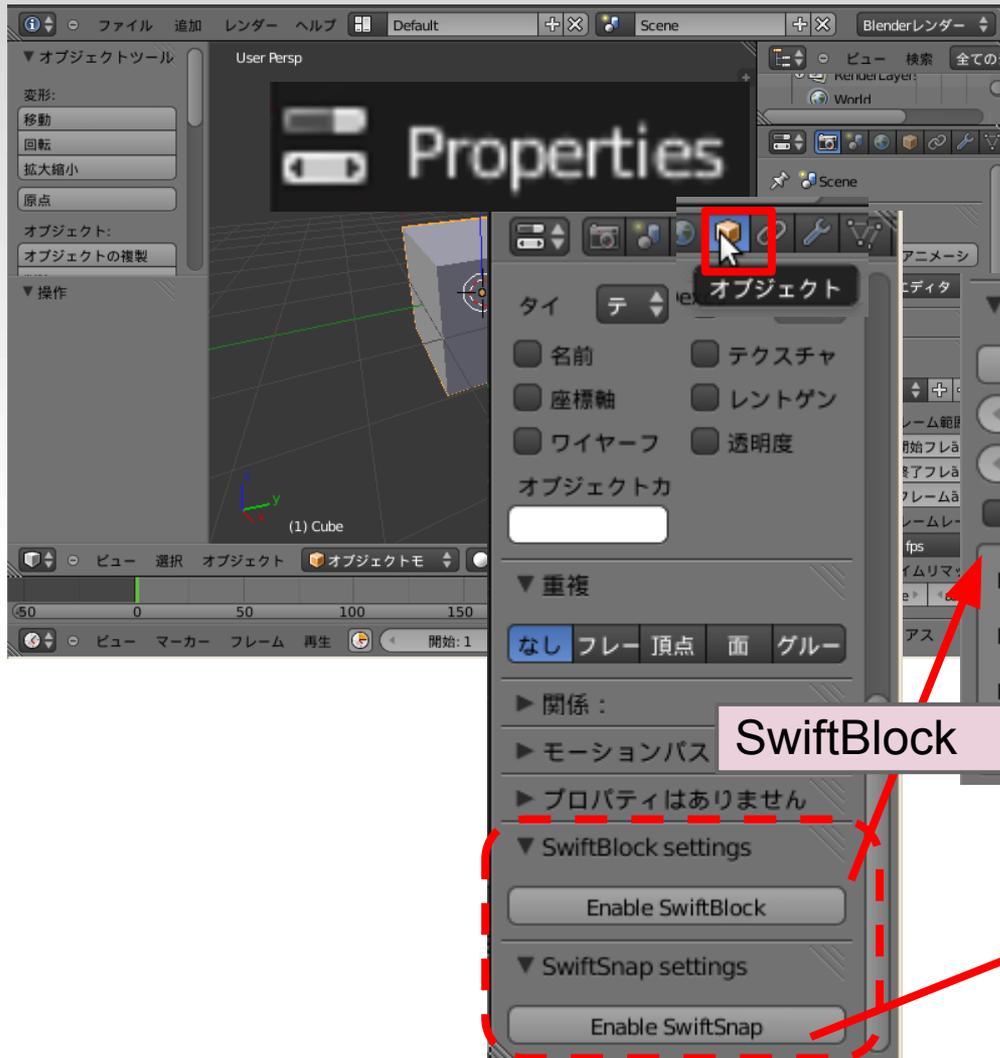
Tキー
Nキー

メッシュ用
プロパティ

swiftツール

- CFD Online, 2012/4/27 アナウンス
 - SwiftSnap and SwiftBlock, GUIs for OpenFOAM's meshers
 - <http://www.cfd-online.com/Forums/openfoam-meshing/100604-swiftnap-swiftblock-guis-openfoams-meshers.html>
- OpenFOAM Wiki にて、プログラム & 例題が公開
- プログラムはblender(ver.2.5?~)のアドオン
- blender, OS ヴァージョン問題、インストール方法
 - OCSE^2 HPにて公開

Swiftツールの起動方法

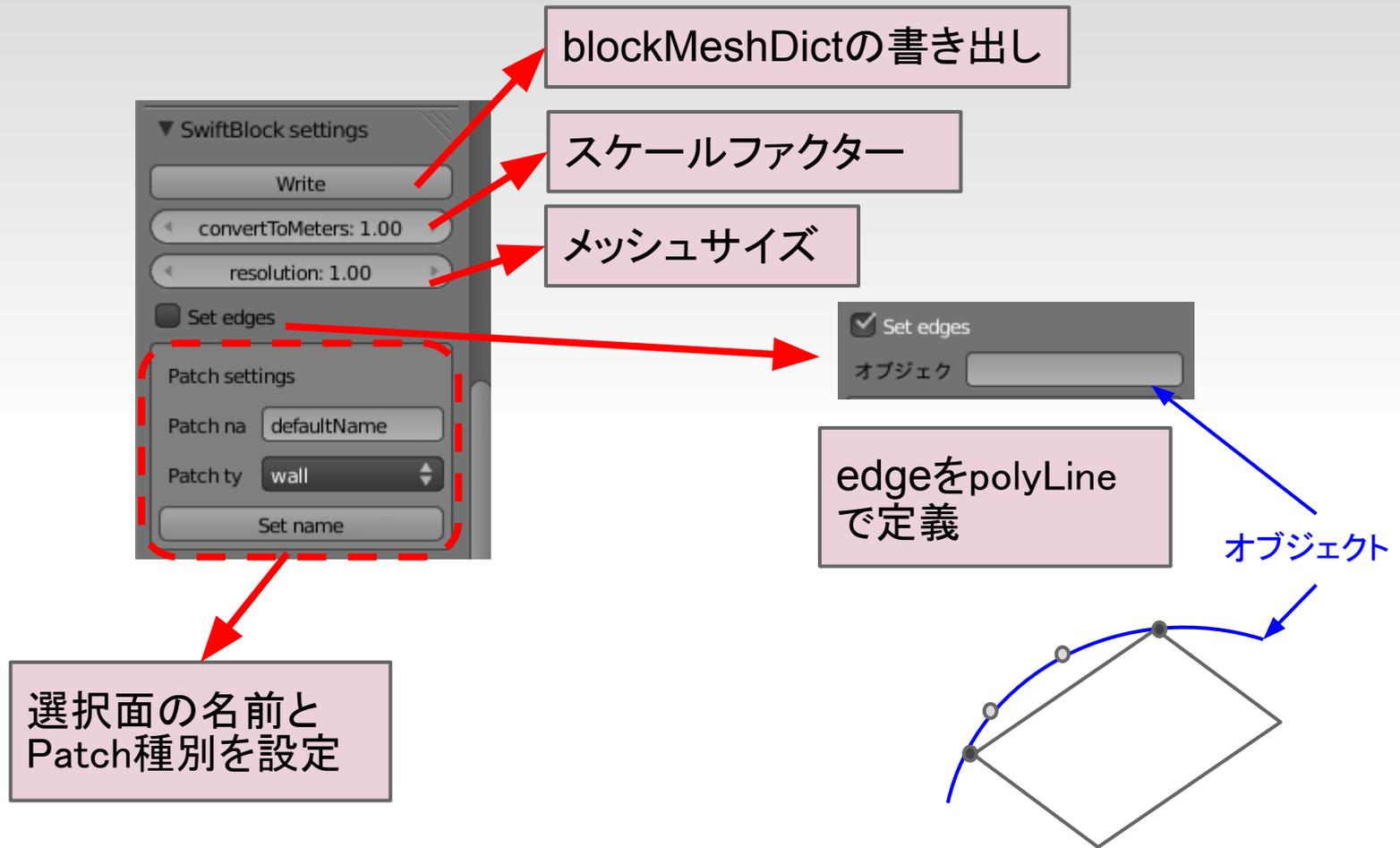


SwiftBlock

SwiftSnap



SwiftBlock



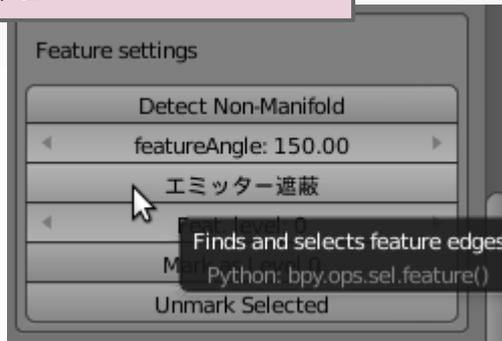
詳細説明は具体例にて後述
(p.36 ~ p.39)

SwiftSnap

スケールファクター

カーソル位置をメッシュ内部点として設定

FeatureEdgeの定義と設定



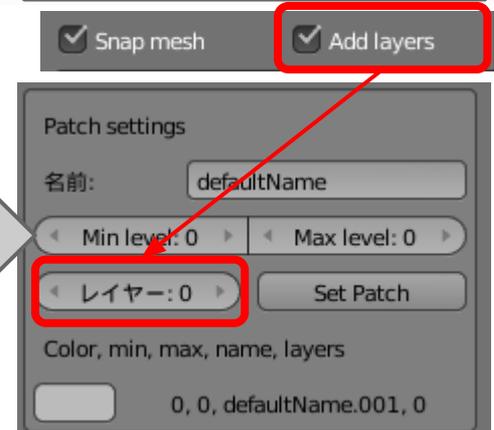
Region分割レベルの設定



blockMeshDictの書き出し

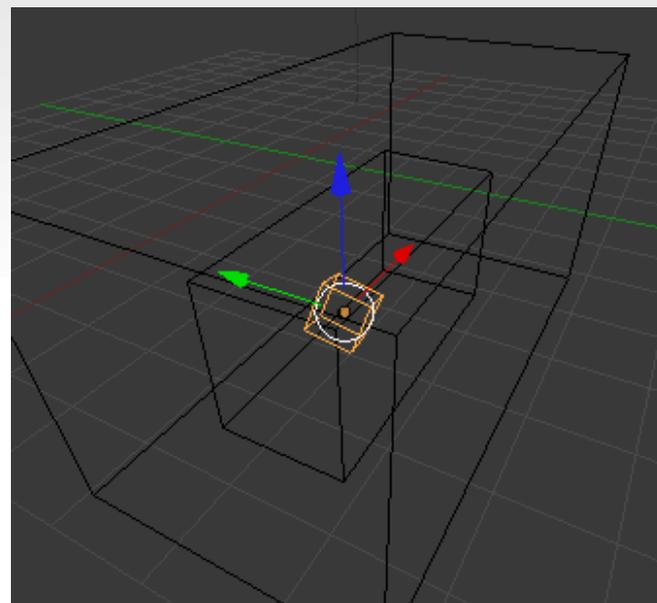
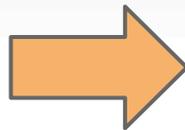
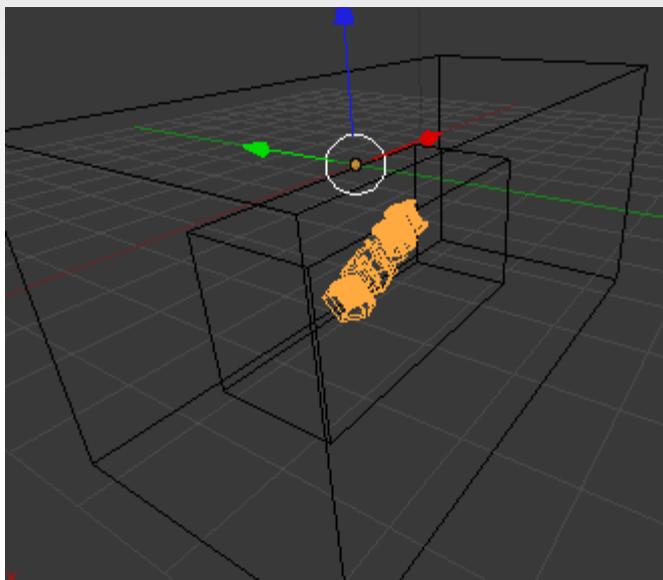
内部メッシュ作成には基礎メッシュ用のblockMeshDictを自動作成

Surface分割レベルとレイヤー設定



詳細説明は具体例にて後述
(p.41 ~ p.46)

簡単なモデル変更演習



解析フォルダ

1

解析フォルダ新規作成

解析フォルダを開く

終了

メッシュ1

メッシュ2

計算実行

結果処理

Change Directory

解析フォルダを /home/custom/Desktop/test1 に設定しました。

OK(O)

6

新規解析フォルダを選択してください

場所(P) 名前 サイズ 最終変更日

場所(P)	名前	サイズ	最終変更日
検索	Desktop		20:51
最近開いた...	OpenFOAM		昨日の 22:33

3

フォルダの作成(L)

場所(P) 名前 サイズ 最終変更日

場所(P)	名前	サイズ	最終変更日
検索	新しいフォルダの種類		
最近開いた...	DEXCS		09:17
custom	SwiftExamp		18:16
デスクトップ	test		
ファイル・シス...	DEXCS2		
フロッピー・ド...	test1		
	DEXCS		
	SwiftExamp		
	test		

4

test1 + Enterキーを押す

場所(P) 名前 サイズ 最終変更日

場所(P)	名前	サイズ	最終変更日
-------	----	-----	-------

5

6

形状作成 (blender)

現在の解析フォルダ: /home/custom/Desktop/test

ファイル ツール ヘルプ

DEXCS

形状作成 **1**

メッシュ1

メッシュ2

計算実行

結果処理

blender

解析実行できます

2 実行

使用するモデルを選択してください

custom Desktop test model

場所(P)	名前	サイズ	最終変更日
検索	dexcs.blend 3	533.4 KB	18:13
最近開いた...	dexcsSwitt.blend	511.9 KB	18:13

追加(A) 削除(R)

Files (*)

キャンセル(C) 開く(O) **4**

Blender [/home/custom/Desktop/test/model/dexcs.blend]

ファイル 追加 レンダー ヘルプ Default Scene Blenderレンダー

▼ オブジェクトツール

変形:
移動
回転
拡大縮小
原点

オブジェクト:
オブジェクトの複製

▼ 操作

ビュー 選択 オブジェクト オブジェクトモ Global

レンダー

レン

解像度: フレー

19 開

10 終

50 フ

アスペ フレー

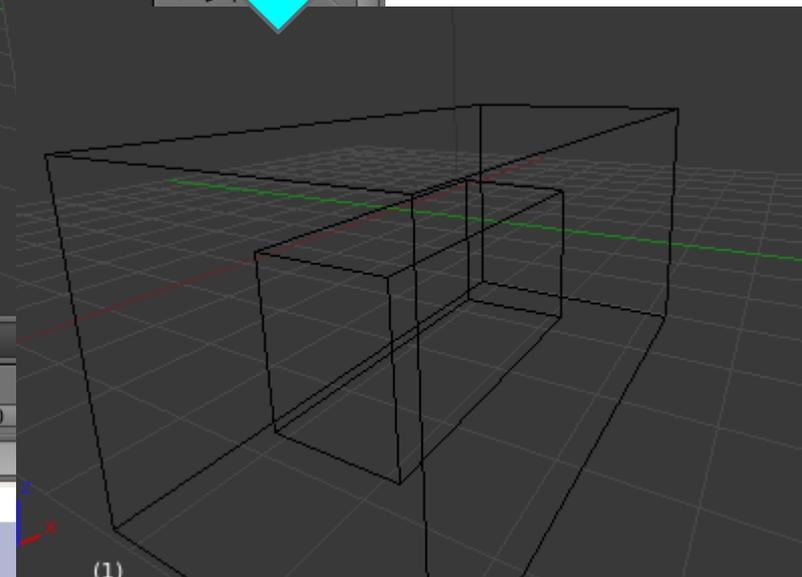
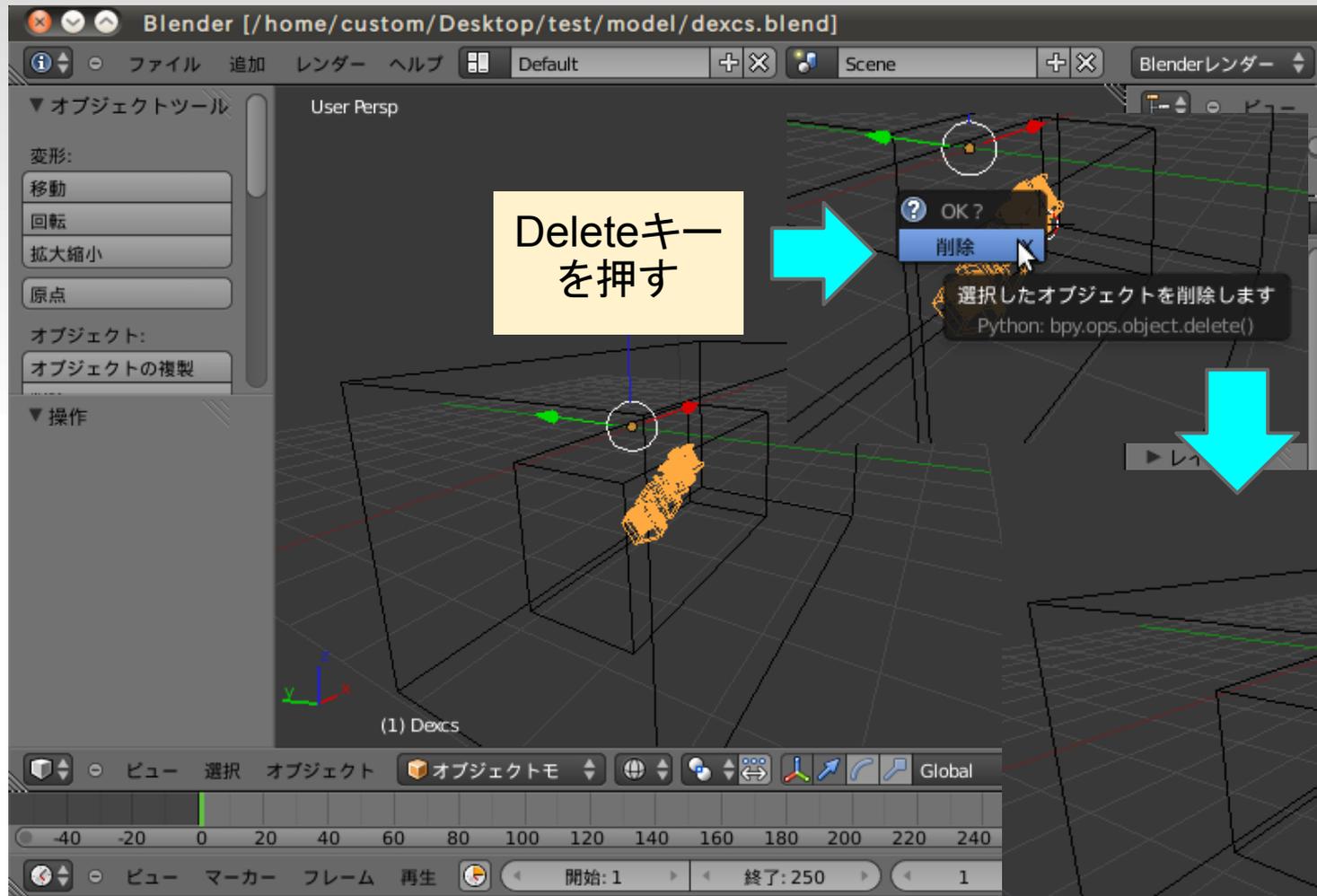
1.0 24

1.0 タイム

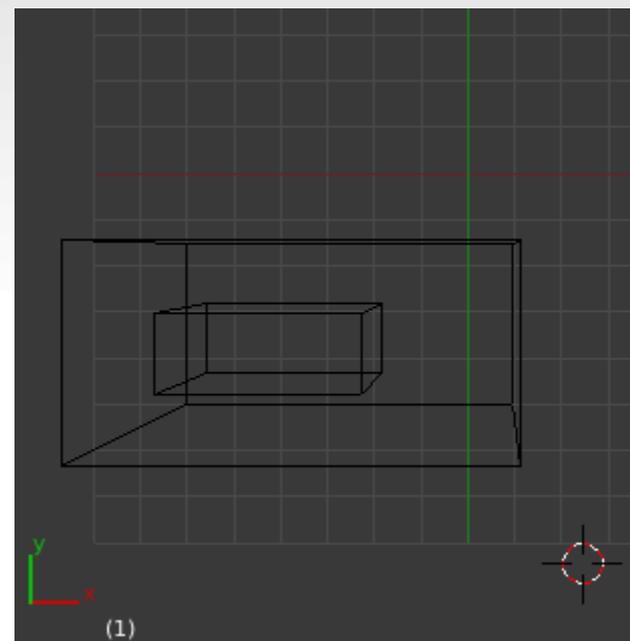
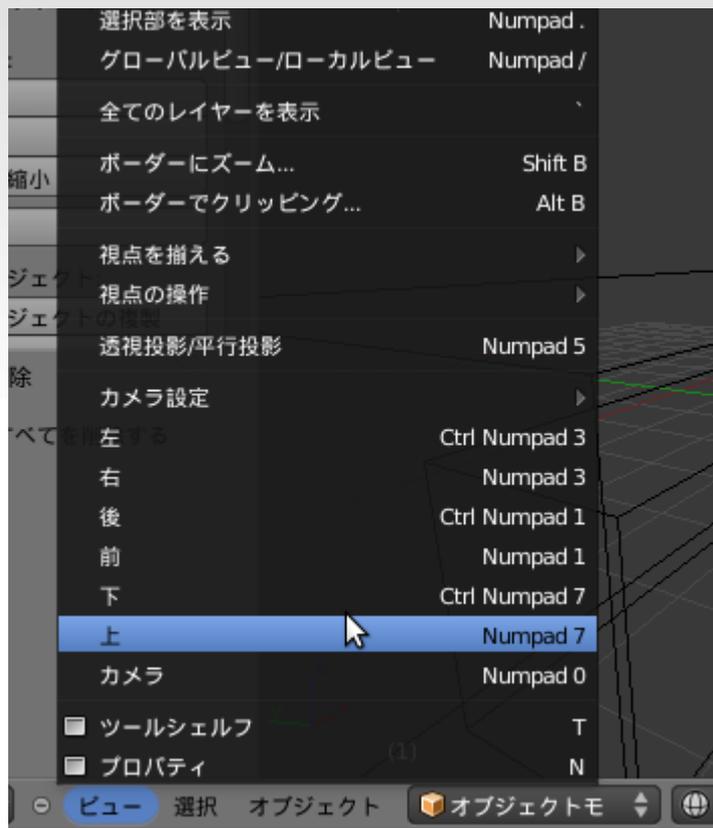
アンチエイ

(1) Dexcs

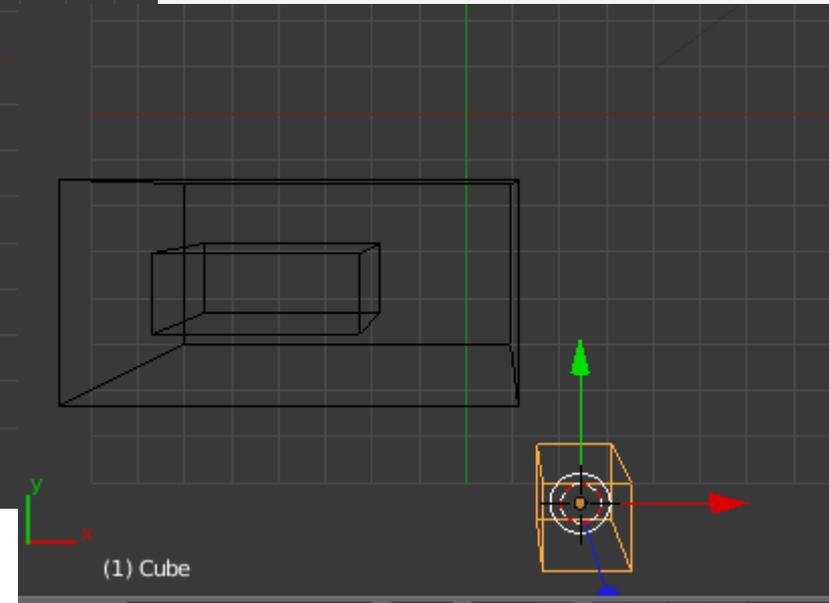
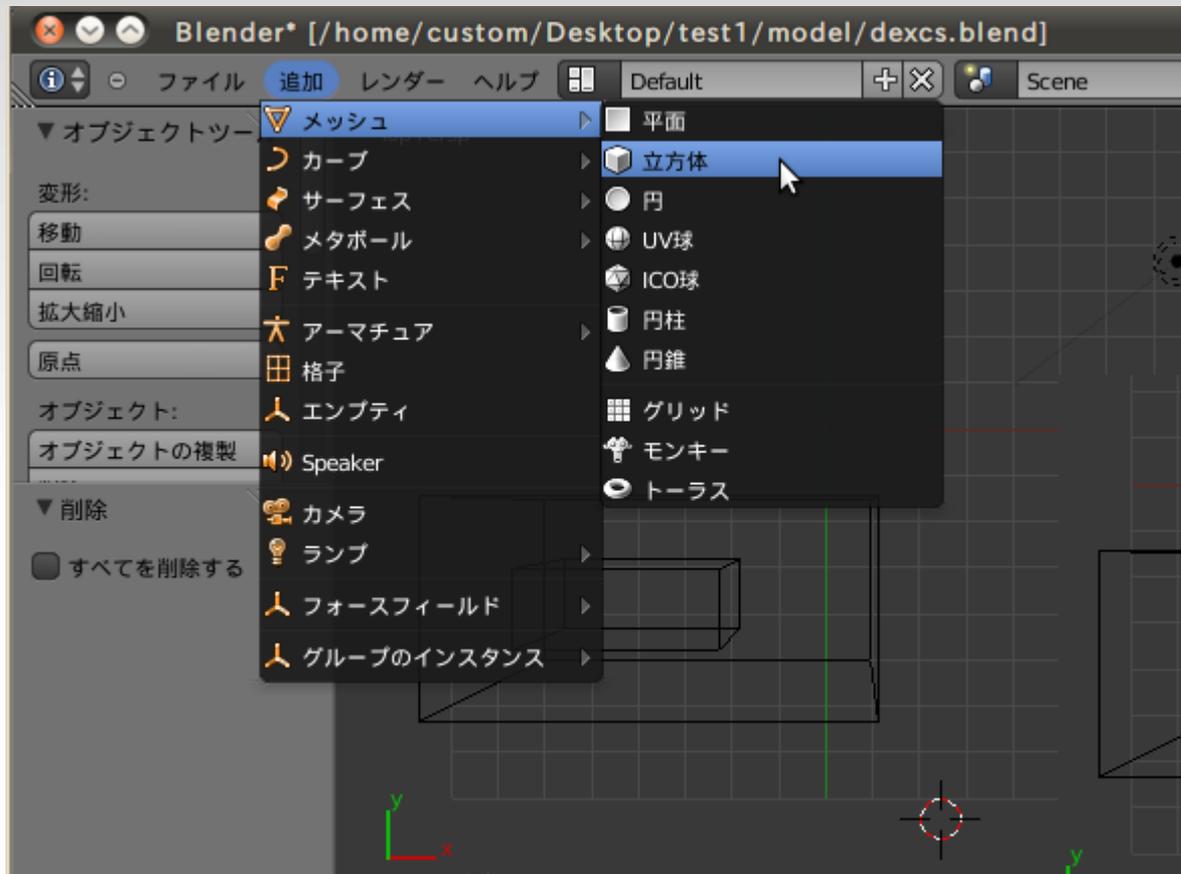
既存モデルの削除



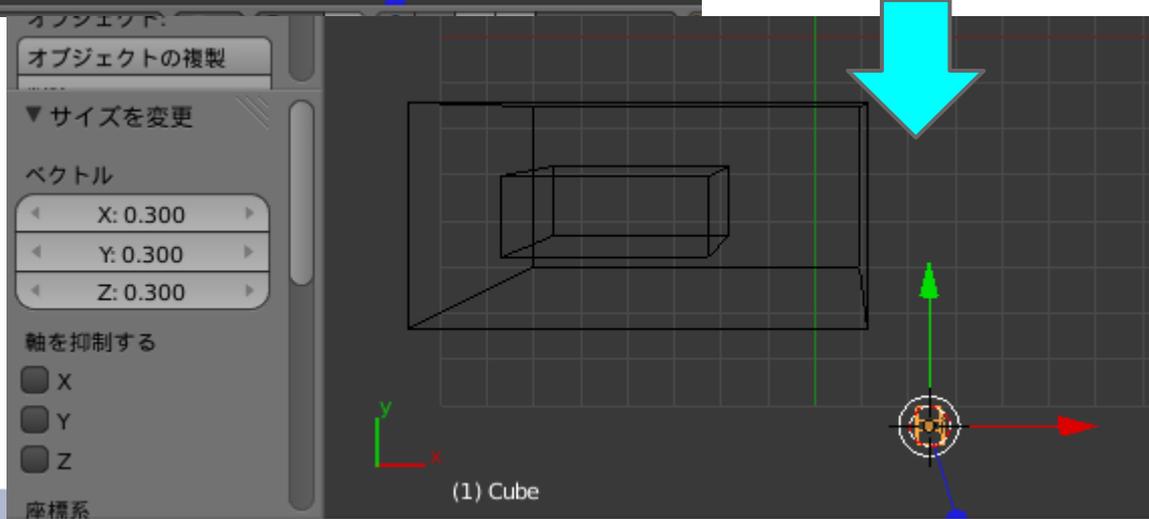
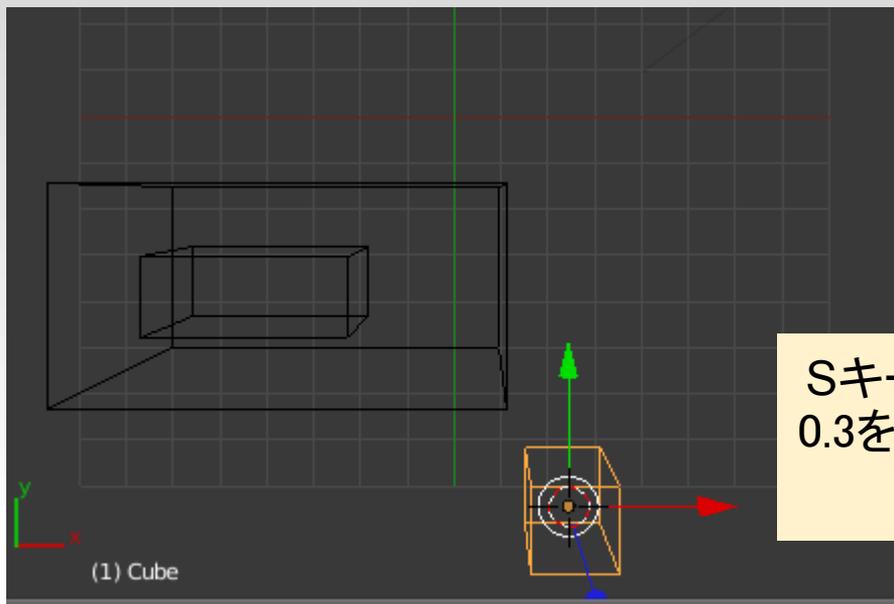
ビュー変更



簡単なモデル(立方体)の追加



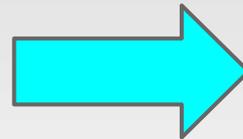
立方体のサイズ変更



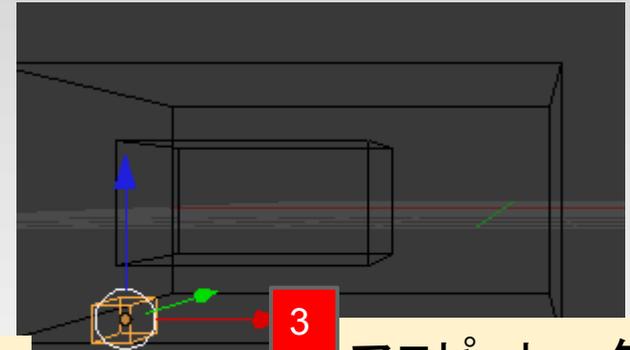
立方体の位置を調整



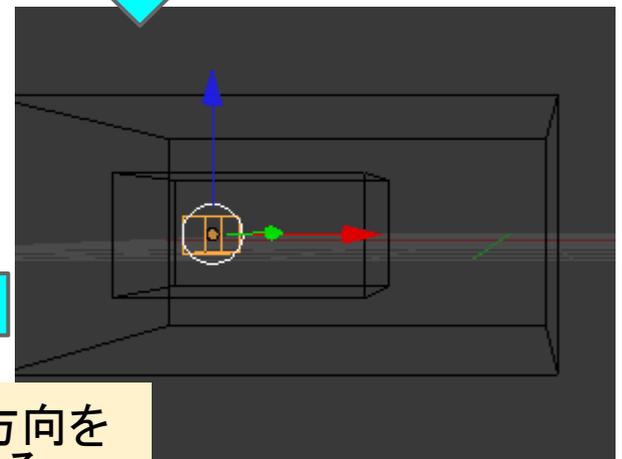
マニピュレータを
左ボタンクリック、
ドラッグで位置変更



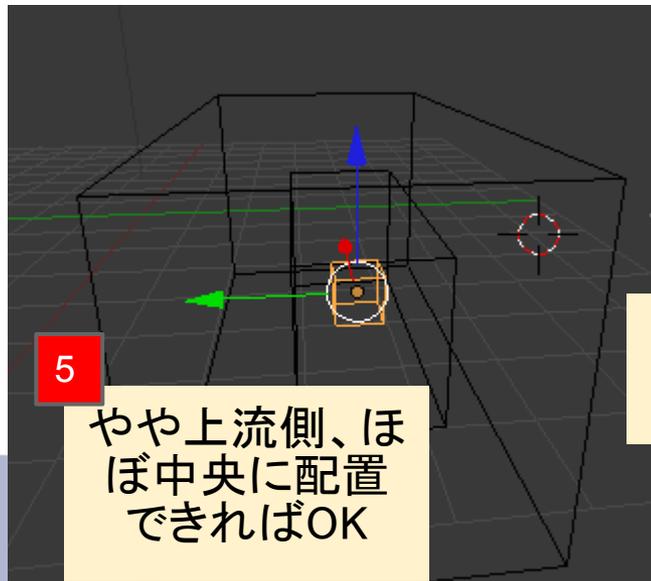
ビュー方向を
変える



マニピュレータを
左ボタンクリック、
ドラッグで位置変更



ビュー方向を
変える



やや上流側、ほ
ぼ中央に配置
できればOK

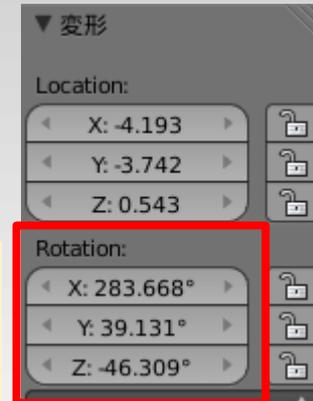


1'
数値パッド
で直接入力
も可能

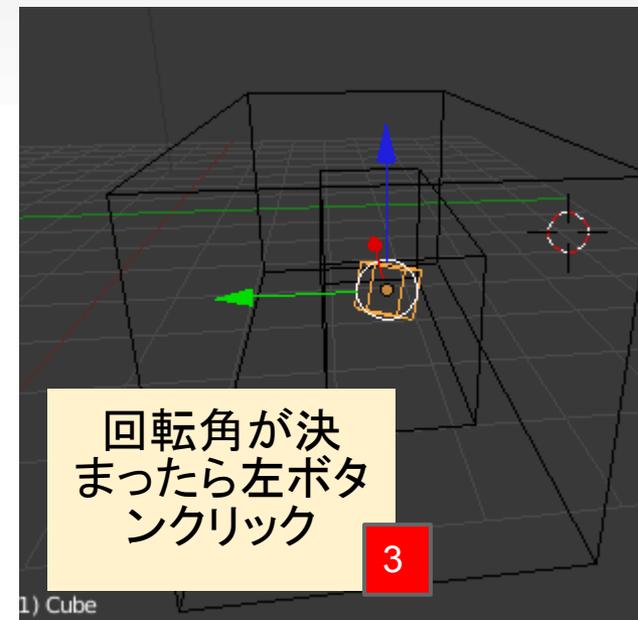
立方体の回転



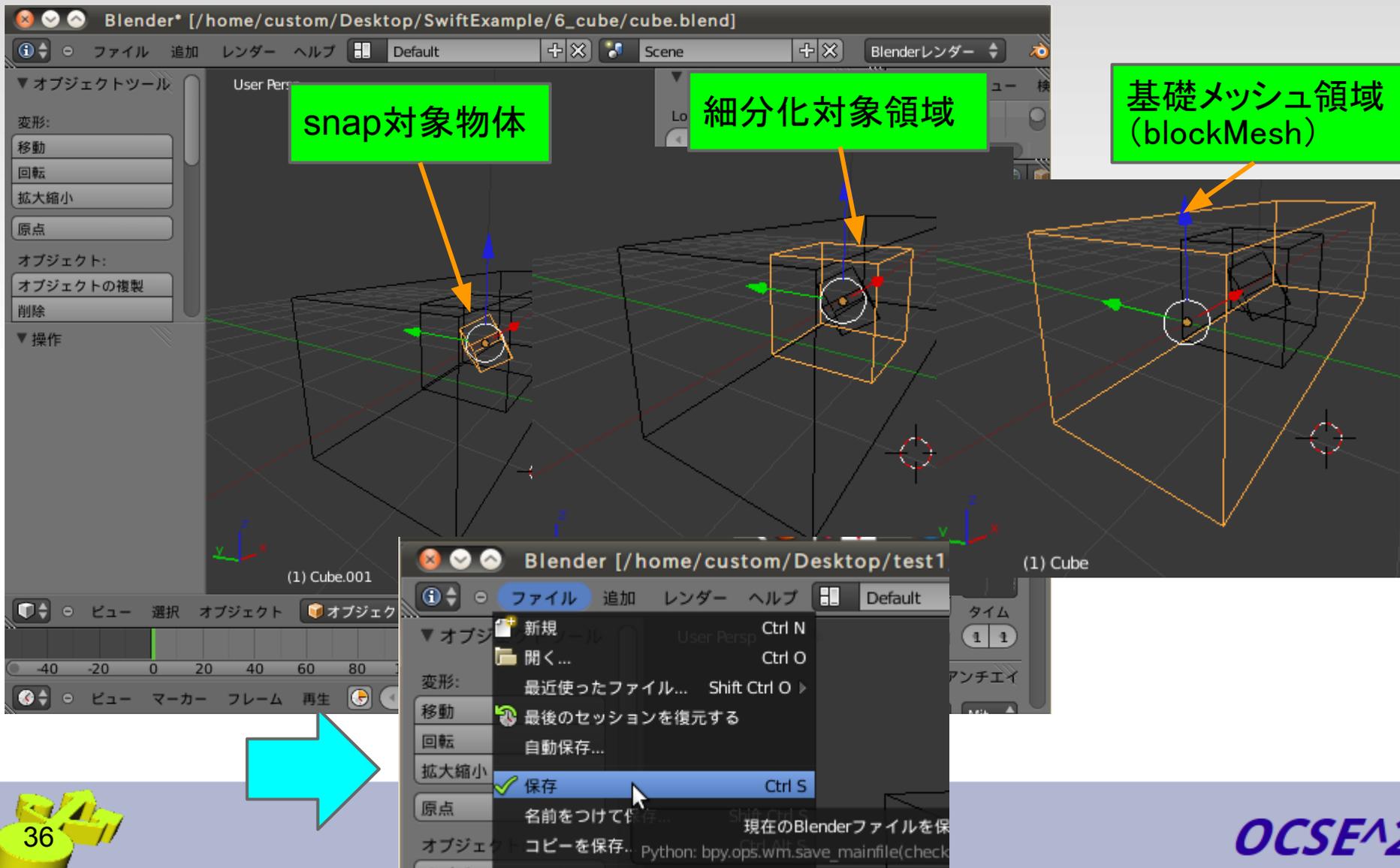
数値パッド
で直接入力
も可能



マウスの動き
に連れて回転



解析モデル確認



SwiftBlock起動

The image displays two overlapping screenshots of the Blender 2.79 interface. The top screenshot shows a 3D scene with a cube. A yellow callout box labeled '1' points to the cube, with the text 'Blockを選択' (Select Block). A yellow arrow labeled '右ボタンクリック' (Right-click) points to the mouse cursor on the cube. A red box labeled '2' highlights the 'SwiftBlock' icon in the top toolbar. The bottom screenshot shows the 'SwiftBlock settings' panel. A red box labeled '3' highlights the 'Enable SwiftBlock' button. A yellow arrow labeled '4' points to the 'SwiftBlock settings' panel, with the text '面編集モード' (Face Edit Mode) below it. A red box labeled '4' also highlights the 'SwiftBlock' icon in the bottom toolbar.

1 Blockを選択

右ボタンクリック

2

3

4

面編集モード

SwiftBlock設定

1 流入面を選択

2 Patch settings
Patch n inlet
Patch t patch
Set name

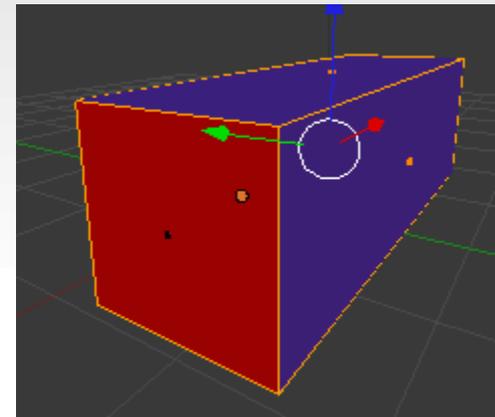
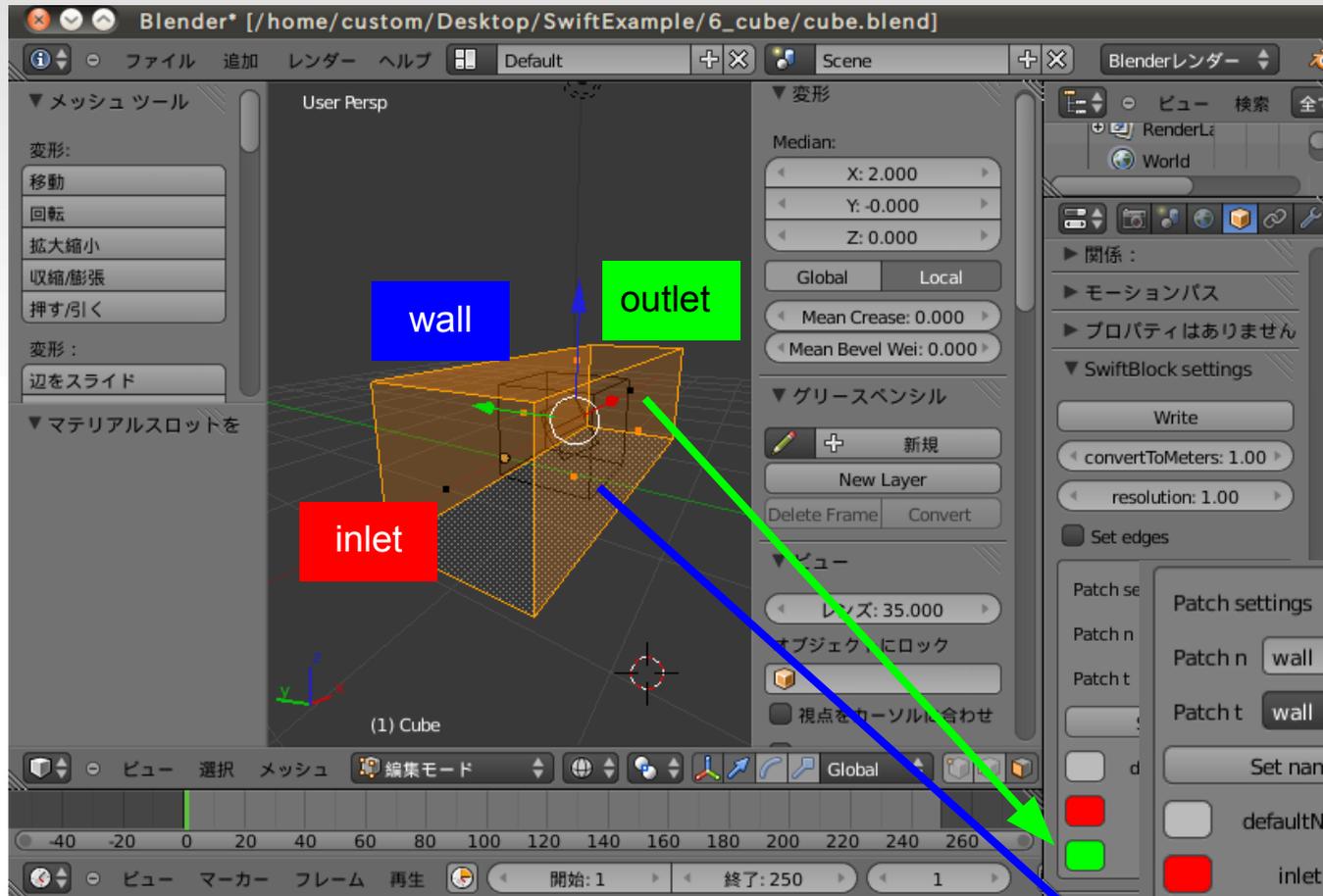
3 ボタンを押す

4 inlet patch

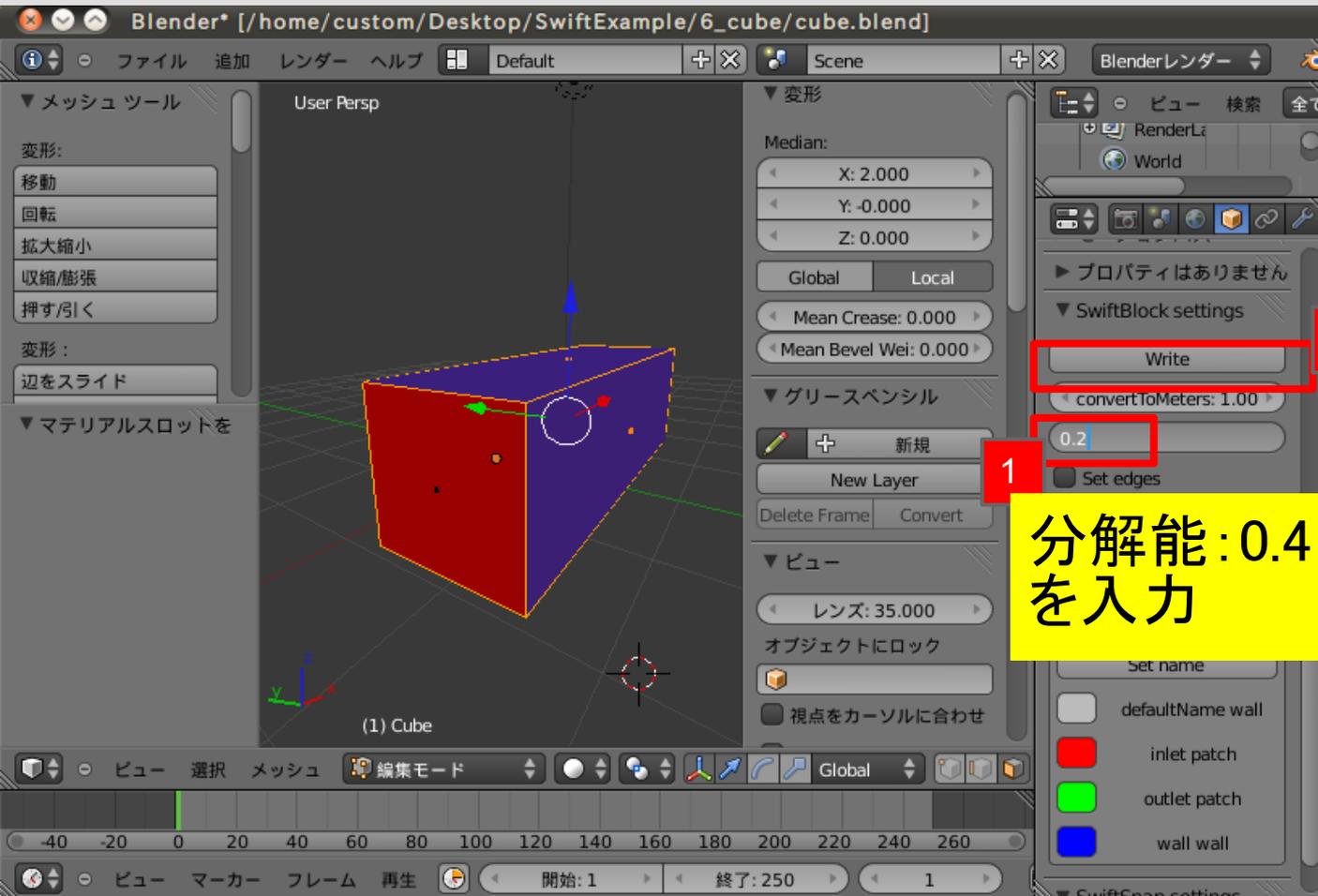
5 設定の確認

Zキー

SwiftBlock設定完了

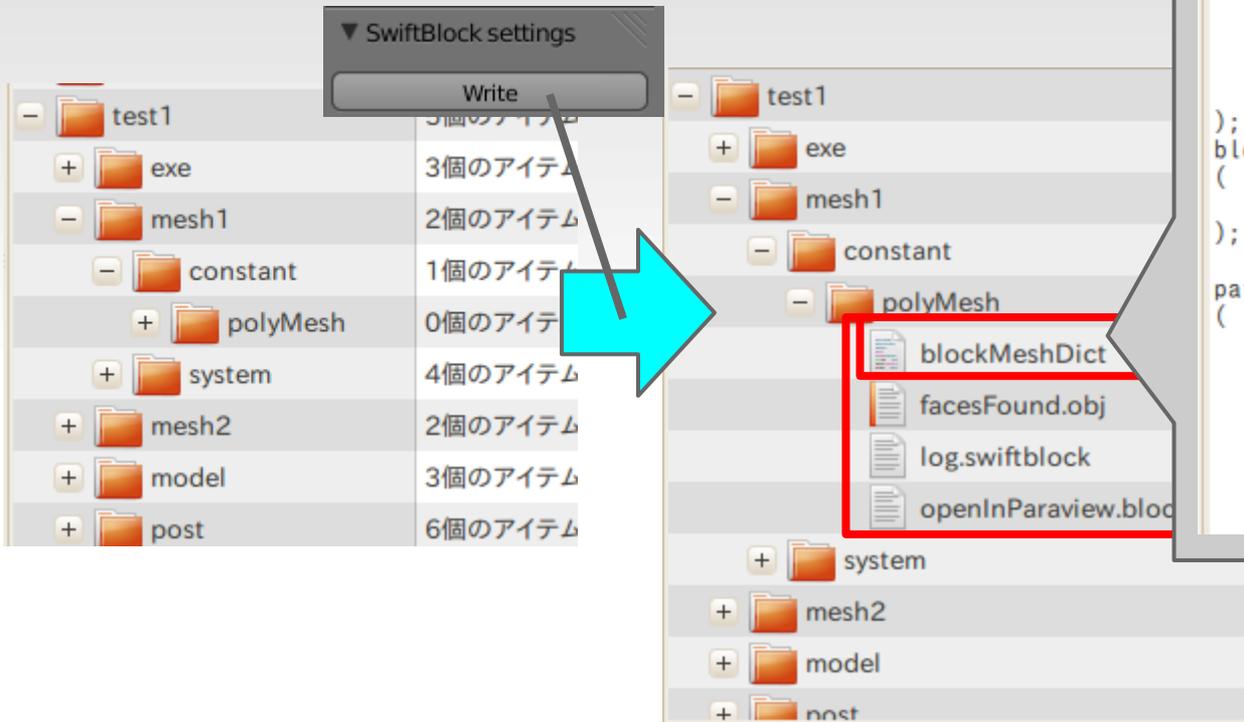


SwiftBlock出力



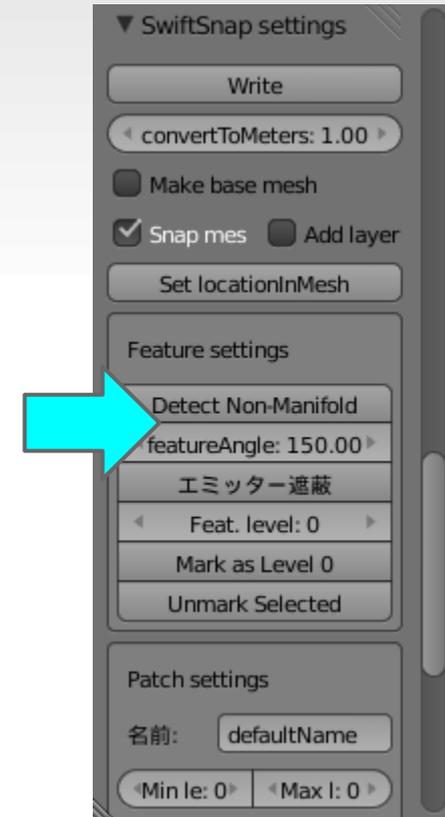
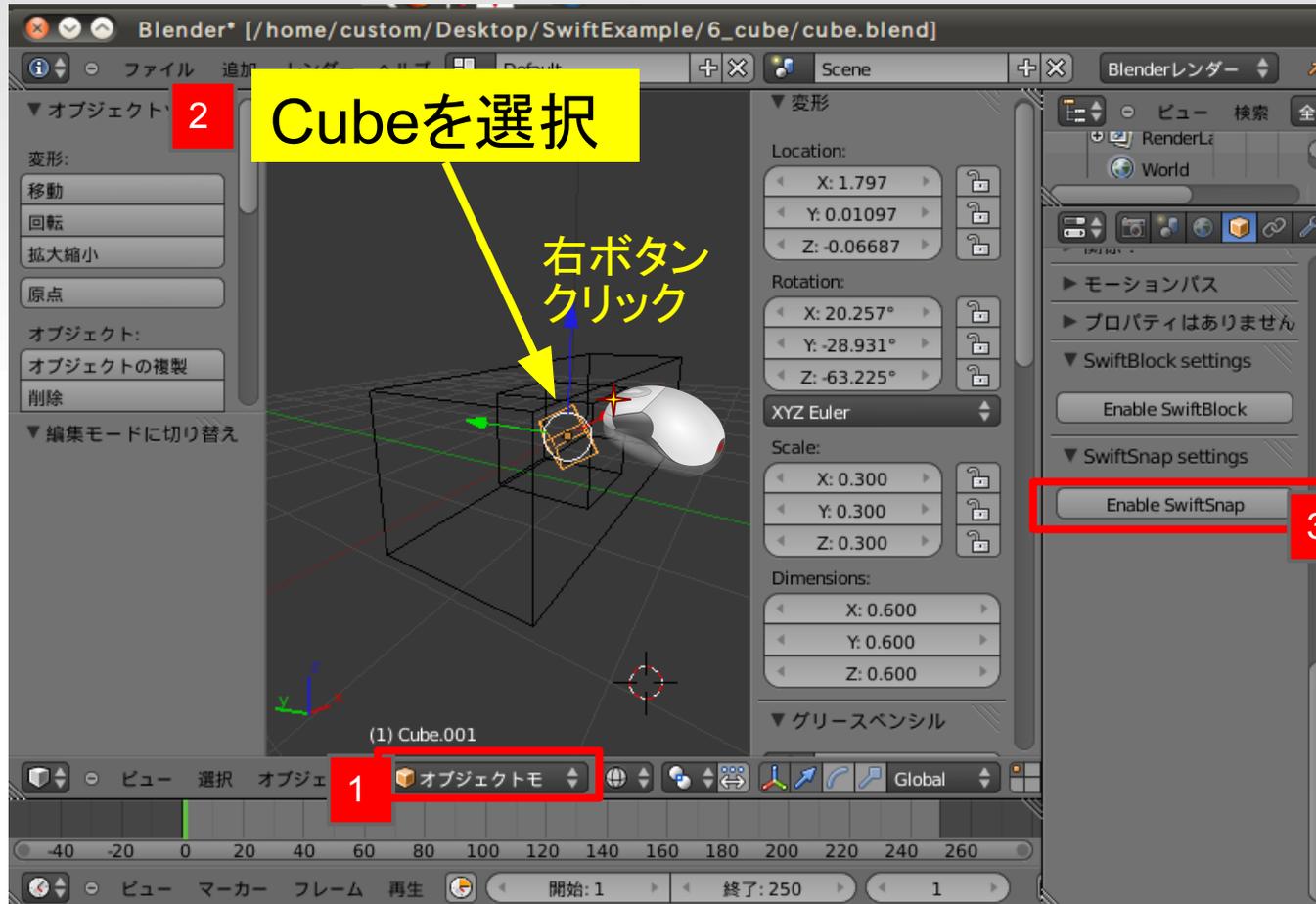
-	test1	5個
+	exe	3個
-	mesh1	2個
-	constant	1個
+	polyMesh	0個
+	system	4個
+	mesh2	2個
+	model	3個
+	post	6個

BlockMeshDict確認



```
blockMeshDict x
vertices
(
  (1.0 -1.5 2.200000047683716)
  (1.0 -1.5 -1.7999999523162842)
  (-7.0 -1.5 -1.7999999523162842)
  (-6.999999046325684 -1.5 2.2000010013580322)
  (1.0000009536743164 -5.5 2.200000047683716)
  (-7.0 -5.5 2.200000047683716)
  (-7.000000953674316 -5.5 -1.7999999523162842)
  (0.9999989867210388 -5.5 -1.8000010251998901)
);
blocks
(
  hex (3 5 6 2 0 4 7 1) (10 10 20) simpleGrading
);
patches
(
  patch inlet
  (
    (5 6 2 3)
  )
  patch outlet
  (
    (4 0 1 7)
  )
);
```

SwiftSnap起動



フィーチャーエッジの設定

1

1

2

2

2

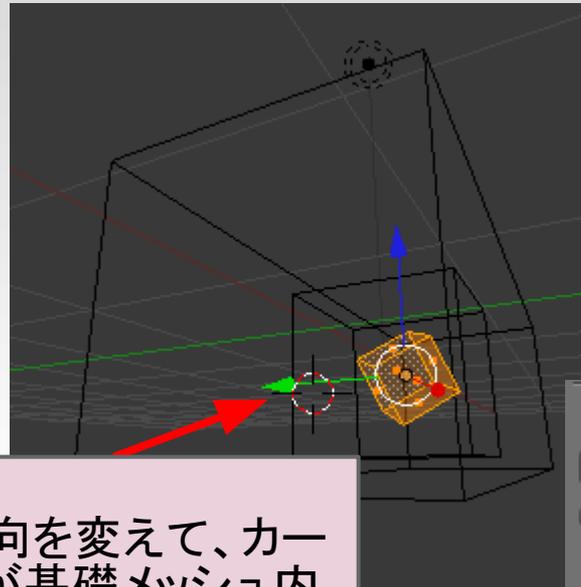
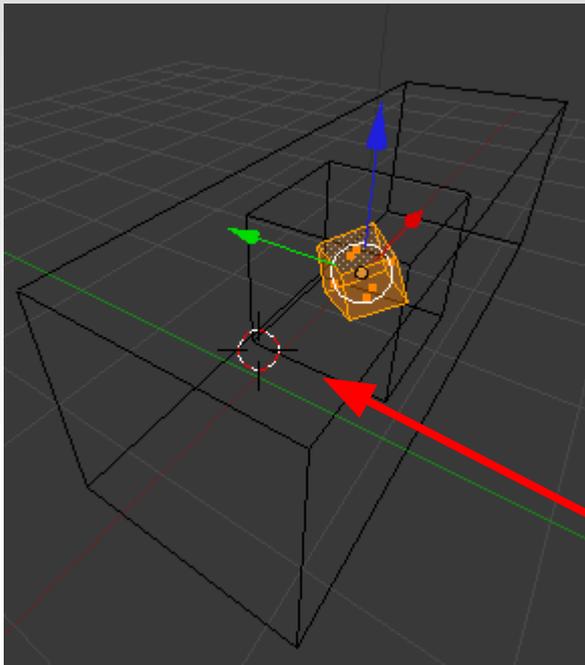
パラメタ 設定された

フィーチャー(輪郭) が抽出された

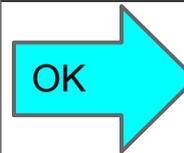
抽出されない場合はfeatureAngleを変えてやり直す

43

メッシュ内部点の設定



視方向を変えて、カーソルが基礎メッシュ内部、snap対象物体の外側にあることを確認



SwiftSnap settings

Write

convertToMeters: 1.00

Make base mesh

Snap mesh Add layers

Set locationInMesh

Feature settings

Detect Non-Manifold

featureAngle: 150.00

エミッター遮蔽

Feat. level: 0

Mark as Level 0

Unmark Selected

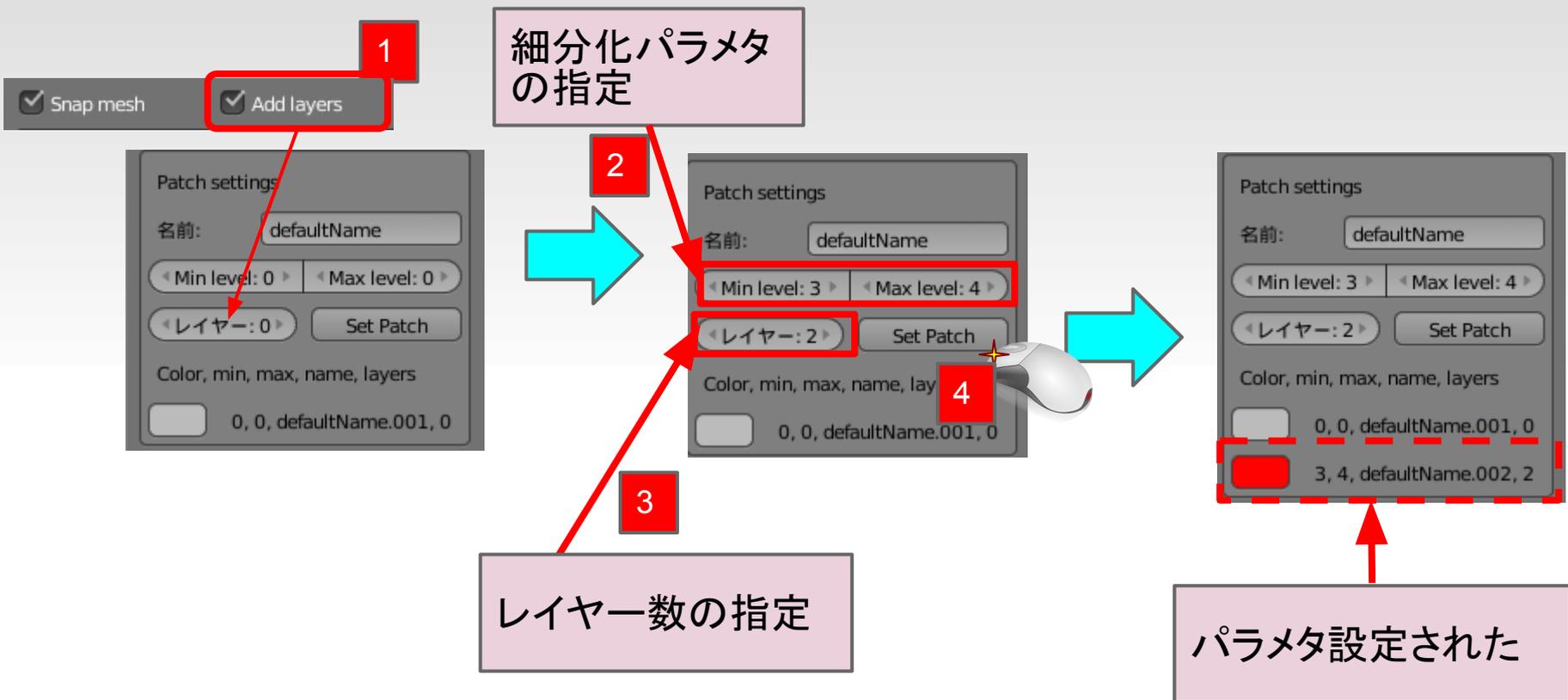
Patch settings

名前: defaultName

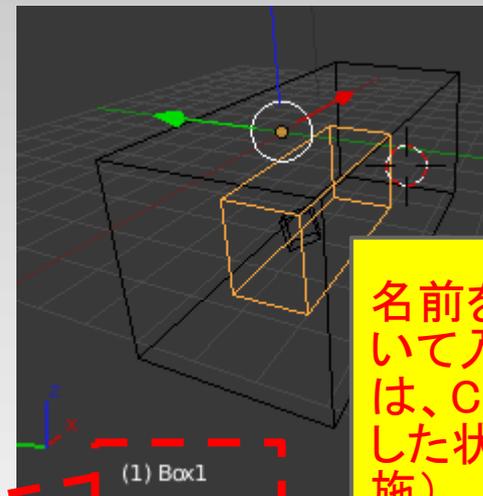
Min lev: 0 Max lev: 0



表面細分化とレイヤー指定



領域細分化の指定

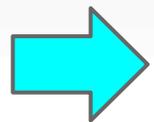


名前を覚えておいて入力(作業は、Cubeを選択した状態にて実施)

領域オブジェクトの名前を入力してEnter

1

細分化パラメタの指定

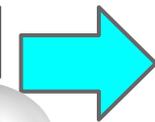


領域オブジェクトが存在すればボタンが有効になる

2



3

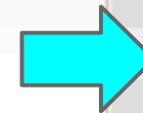
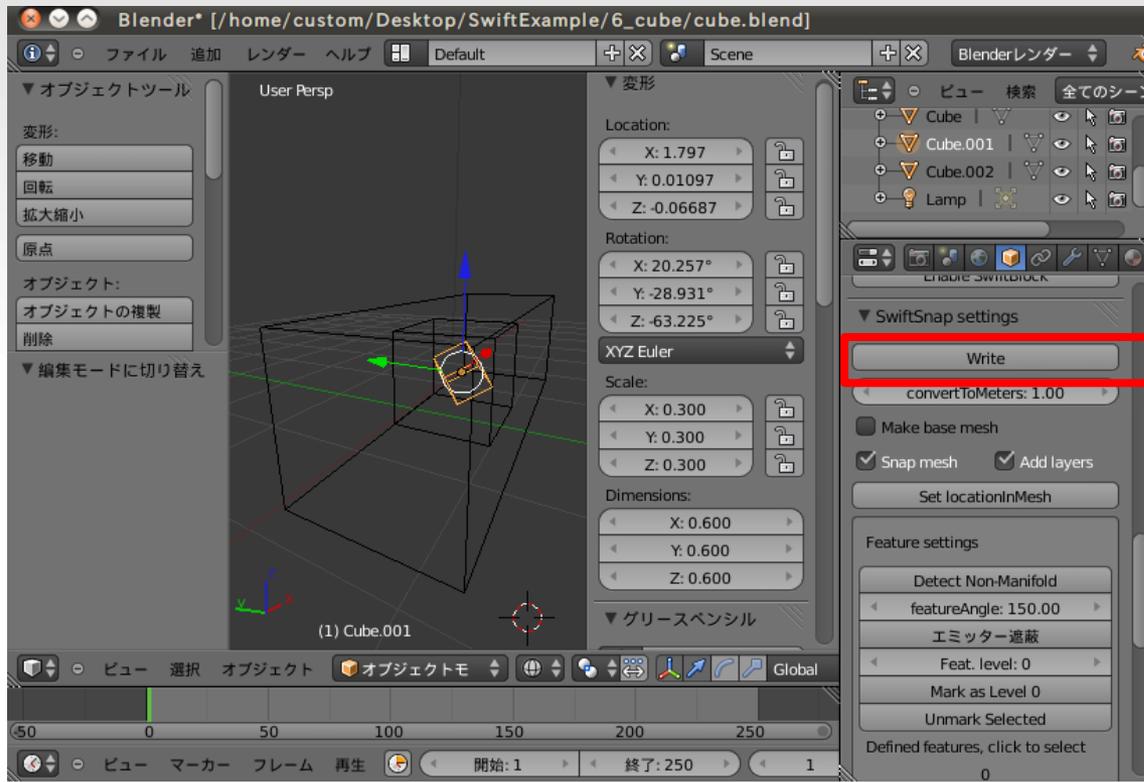


パラメタ設定された

領域が複数存在すれば、複数個の設定が可能。



SwiftSnap書き出し



-	test1	5個のアイテム
+	exe	3個のアイテム
+	mesh1	2個のアイテム
-	mesh2	2個のアイテム
+	constant	1個のアイテム
-	system	5個のアイテム
+	triSurface	0個のアイテム
	controlDict	1.3 KB
	decomposeParDict	1.4 KB
	fvSchemes	1.6 KB
	fvSolution	1.2 KB
+	model	3個のアイテム
+	post	6個のアイテム

snappyHexMeshDict確認

The image shows a file explorer window on the left and a code editor window on the right. The file explorer shows a directory structure with folders like 'test1', 'exe', 'mesh1', 'mesh2', 'constant', 'system', and 'triSurface'. A 'SwiftSnap settings' dialog box is open over the 'system' folder, with a 'Write' button highlighted by a cyan arrow. The 'system' folder is expanded to show files: 'Box1.stl', 'defaultName.002.stl', 'level0.eMesh', 'level0.obj', 'controlDict', 'decomposeParDict', 'fvSchemes', 'fvSolution', and 'snappyHexMeshDict'. A red box highlights the first four files, and another red box highlights 'snappyHexMeshDict'. A red box with the text '生成されたファイル' (Generated files) points to the first four files. The code editor shows the 'snappyHexMeshDict' configuration file with the following content:

```
castellatedMesh true;
snap true;
addLayers true;

geometry
{
    defaultName.002.stl
    {
        type triSurfaceMesh;
        name defaultName.002;
    }
    Box1.stl
    {
        type triSurfaceMesh;
        name Box1;
    }
};

castellatedMeshControls
{
    features
    (
        {
            file "level0.eMesh";
            level 0;
        }
    )
};
```

生成されたファイル

Swiftモデルの保存

The image illustrates the steps to save a Swift model in Blender. The top window shows the File menu with '名前をつけて保存...' (Save As...) selected, highlighted with a red box and the number 1. The bottom window shows the file browser with 'dexcsSwift.blend' selected, highlighted with a red box and the number 2. The 'Save As Blend' button is highlighted with a red box and the number 3.

Blender* [/home/custom/Desktop/test1/model/dexcs.blend]

ファイル 追加 レンダー ヘルプ Default Scene Blenderレンダー

新規 Ctrl N
開く... Ctrl O
最近使ったファイル... Shift Ctrl O
最後のセッションを復元する
自動保存...
保存 Ctrl S
名前をつけて保存... Shift Ctrl S
コピーを保存... Ctrl Alt S
ユーザー設定... Ctrl Alt U
ユーザー設定の保存 Ctrl U
初期設定を読み込む
リンク Ctrl Alt O
アペンド Shift F1
インポート

変形
Location:
X: -4.760
Y: -3.665
Z: 0.08529
Rotation:
X: 198.277°
Y: 14.421°
Z: 123.775°
XYZ Euler
Scale:
X: 0.300
Y: 0.300
Z: 0.300

ビュー 検索 全てのシーン
Box1
Camera
Cube
Lamp
Enable SwiftBlock
SwiftSnap settings
Write
convertToMeters: 1.00
Make base mesh
Snap mesh Add layers

Blender* [/home/custom/Desktop/test1/model/dexcs.blend]

新しいディレクトリを作成
隠したものを表示

システム
cdrom
rofs

Bookmarks
追加
custom

/home/custom/Desktop/test1/model/
Save As Blend
dexcsSwift.blend
キャンセル
dexcs.blend 348 KB
dexcsSwift.blend 511 KB

メッシュ作成

現在の解析フォルダ: /home/custom/Desktop/test

ファイル ツール ヘルプ

DEXCS blockMesh

形状作成 blockMeshDict編集 表示

メッシュ1

メッシュ2 **メッシュ確認** 初期化(やり直し) **実行 1**

計算実行 autoPatch

結果処理 二面挟角(deg) 60 実行

createPatch

createPatchDict編集 実行

メッシュ確定-> メッシュ2 (snappyHexMeshing)

実行 2

メッシュ確定-> exe

実行

解析実行できます

現在の解析フォルダ: /home/custom/Desktop/test

ファイル ツール ヘルプ

DEXCS surfaceFeatureExtract

形状作成 includedAngle 150 選択実行

メッシュ1

メッシュ2 形状選択 Dict編集 初期化(やり直し)

計算実行 並列分割

結果処理 並列分割数 2

snappyHexMesh

停止 **実行 3**

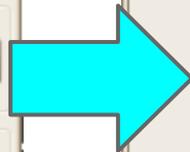
メッシュ確認

native 実行

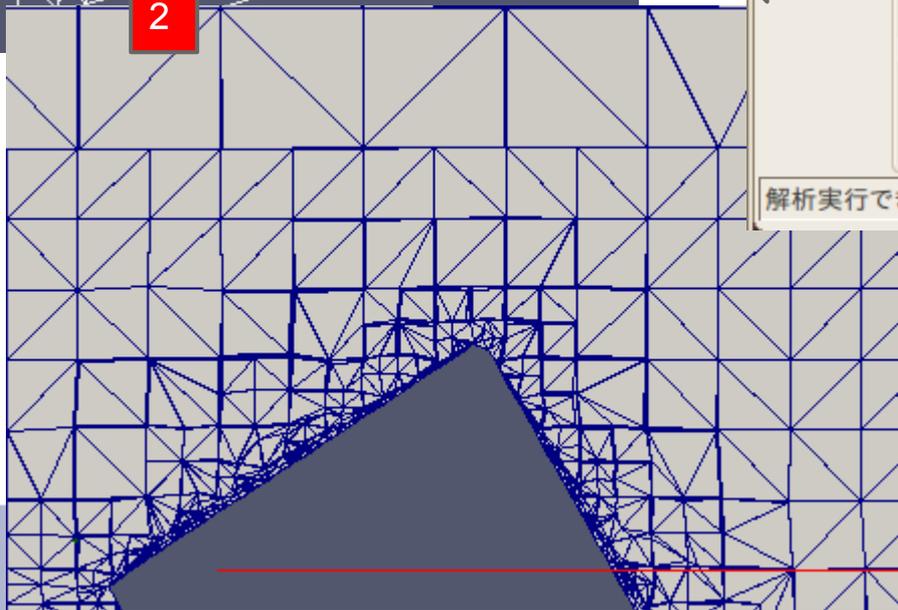
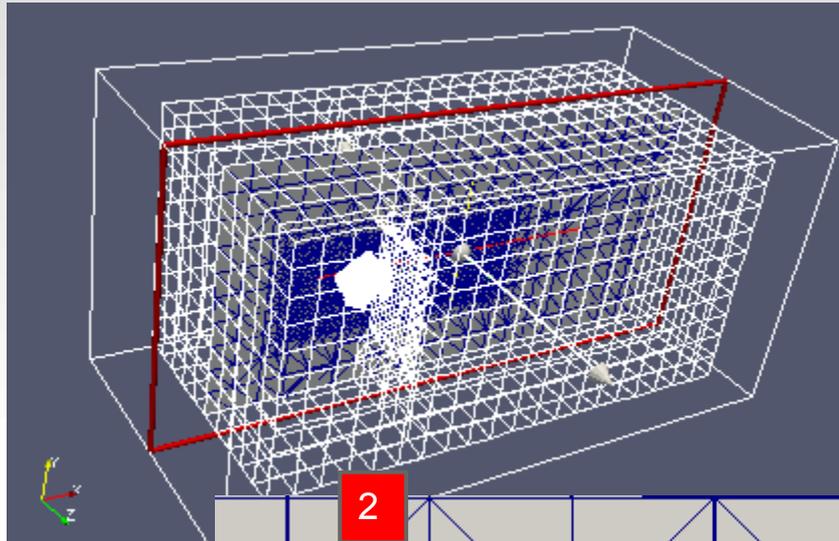
メッシュ確定-> exe

time 3 実行

解析実行できます



メッシュ確認⇒確定



現在の解析フォルダ: /home/custom/Desktop/test
ファイル ツール ヘルプ

DEXCS surfaceFeatureExtract
includedAngle 150 選択実行

形状作成

メッシュ1 snappyHexMeshDict
形状選択 Dict編集 初期化(やり直し)

メッシュ2

計算実行 並列分割
並列分割数 2

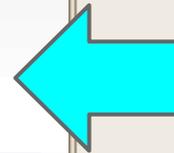
結果処理

snappyHexMesh
停止 実行

メッシュ確認
 native 実行 1

メッシュ確定-> exe
time 3 実行 3

解析実行できます

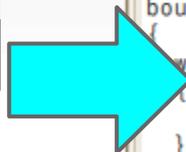


境界名の確認

ファイル ツール ヘルプ

DEXCS	計算初期化(clearCase)	実行
形状作成		
メッシュ1	system/controlDict,...	編集
メッシュ2	constant	編集
計算実行		編集
結果処理	Field変数	編集
	初期化 patch名取得	
	初期流れ場作成 (potentialFoam)	実行
	並列分割 (decomposePar)	
	並列分割数 <input type="text" value="2"/>	
	計算実行(simpleFoam)	停止 実行

解析実行できます



```
U (~/Desktop/test1/exe/0) - 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
U x epsilon x k x omega x p x phi
class volVectorField;
object U;
}
// *****
dimensions [ 0 1 -1 0 0 0 0 ];
internalField uniform (0 0 0);
boundaryField
{
    wall
    {
        type slip;
    }
    inlet
    {
        type          fixedValue;
        value         uniform (10 0 0);
    }
    Dexcs_Exportedfromblender
    {
        type          fixedValue;
        value         uniform (0 0 0);
    }
    outlet
    {
        type zeroGradient;
    }
}
```

境界名の修正

The image shows a gedit window with the following menu: 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 toolbar includes icons for opening, saving, and navigating. The window title is "U (~ / Desktop / test1 / exe / 0) - gedit".

The main editor displays the contents of a file named "boundary". The code is as follows:

```
class volVectorField;
object U;
}
// *****
dimensions [ 0 1 -1 0 0 0 ];
internalField uniform (0 0 0);
boundaryField
{
  wall
  {
    type slip;
  }
  inlet
  {
    type          fixedValue;
    value         uniform (10 0 0);
  }
  defaultName.002_Exportedfromblender
  {
    type          fixedValue;
    value         uniform (0 0 0);
  }
  outlet
  {
    type zeroGradient;
  }
}
```

Annotations and actions:

- 1**: A red box highlights the "boundary" tab in the window's tab bar.
- 2**: A red box highlights the text "Dexcs Exportedfromblender" in the code. A yellow box labeled "コピー" (Copy) is positioned over it.
- 3**: A red box highlights the "boundary" tab in the window's tab bar.
- 4**: A red box highlights the text "Dexcs Exportedfromblender" in the code. A yellow box labeled "ペースト" (Paste) is positioned below it.

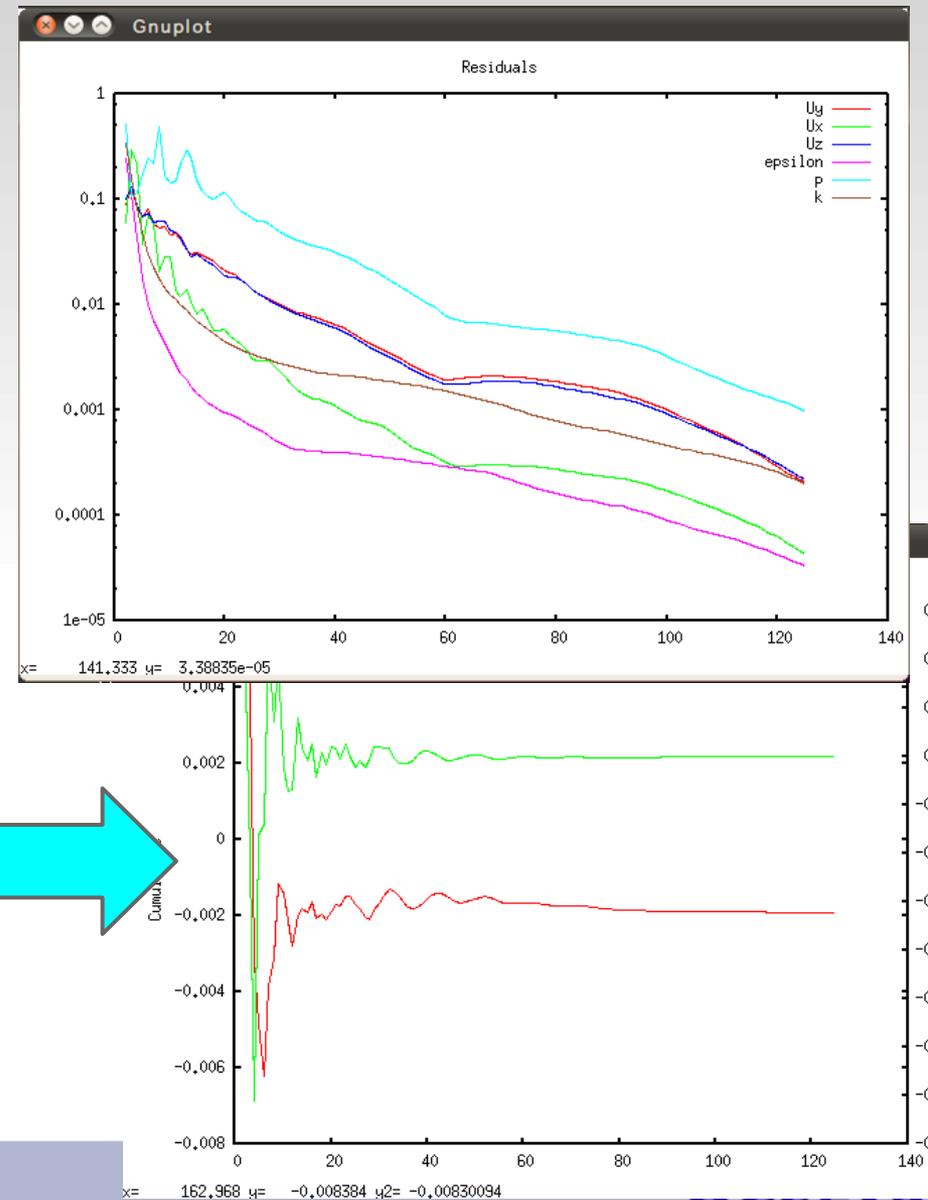
A large blue arrow points from the "ペースト" box to the "boundary" tab in the second window. A dashed red box labeled "defaultName.002_Exportedfromblender" is shown in the code, with a red arrow pointing from the "ペースト" box to it.

計算実行

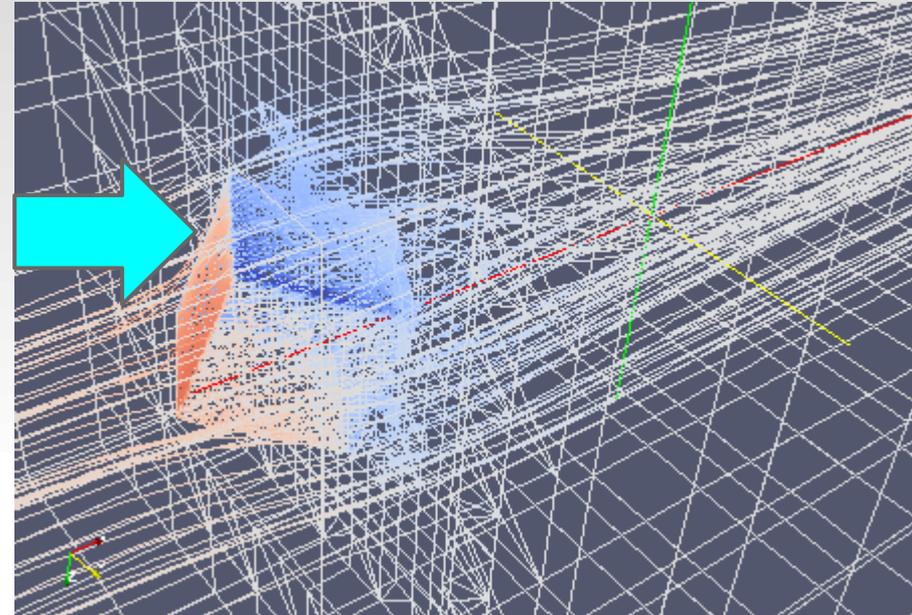
ファイル ツール ヘルプ

DEXCS	計算初期化(clearCase)	実行
形状作成		
メッシュ1	system/controlDict,...	編集
メッシュ2	constant	編集
計算実行		編集
結果処理	Field変数	
	初期化 patch名取得	編集
	初期流れ場作成 (potentialFoam)	実行 1
	並列分割 (decomposePar)	
	並列分割数 2	
	計算実行(simpleFoam)	停止 実行 2

解析実行できます



計算結果確認



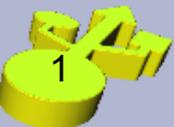
memo



DEXCS for OpenFOAM®の体験2-2

1. ランチャー演習の振り返り
2. blenderとSwiftツールの説明
3. 簡単なモデル変更演習
4. 標準チュートリアルケースの実行
5. 標準チュートリアルベースのメッシュ変更

オープンCAEコンサルタント OCSE^2
代表 野村 悦治



標準チュートリアルケースの実行演習

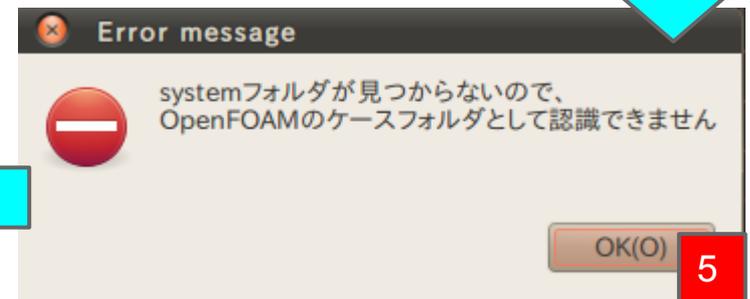
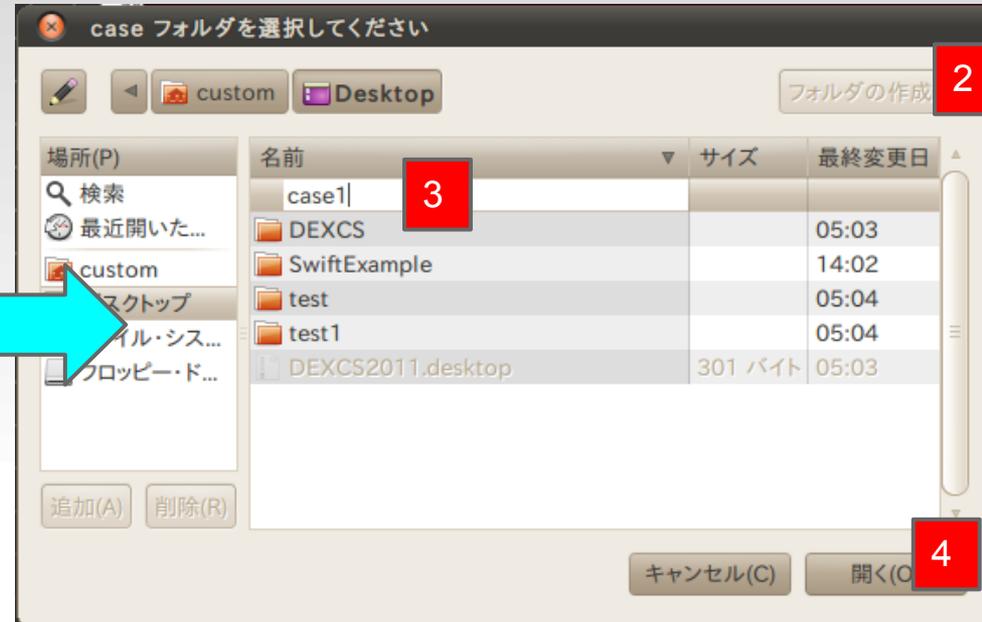
1. incompressible/pisoFoam/ras/cavity
2. multiphase/interFoam/laminar/damBreak
3. heatTransfer/buoyantBoussinesqSimpleFoam
/iglooWithFridges



DEXCS十徳ナイフの起動

The screenshot illustrates the process of launching the DEXCS Ten-Blade Knife application on a Linux desktop. The desktop background features a grid pattern and several icons, including a folder named 'DEXCS' and a file named 'DEXCS2011'. The system tray at the top right shows the date and time as '6月22日 (金) 午前 8:45' and the user 'custom'. The application menu is open, showing a search bar and a list of applications. The 'dexcsSWAK' application is highlighted, and a red box with the number '1' is placed over it. A tooltip for 'DEXCS 十徳ナイフ' is visible. In the bottom right corner, the system tray icon for 'dexcsSWAK' is also highlighted with a red box and the number '1'. A terminal window titled 'stdout/stderr' is open in the foreground, showing a blank prompt. A red box with the number '2' is placed over the terminal window. The 'DEXCS SWAK' application window is open, showing a settings dialog for 'DEXCS 十徳ナイフ'. The dialog has checkboxes for 'FileManager' and 'Terminal', both of which are checked. There are input fields for 'case', 'Ncore' (set to 1), and 'solver/application' (set to 'simpleFoam'). A 'ケースの初期化' dropdown menu and an '実行' button are also visible. A red box with the number '3' is placed over the application window. In the bottom left corner, there is a yellow ribbon icon with the number '3' on it.

caseフォルダの作成



DEXCS十徳ナイフの起動画面

The screenshot displays a Linux desktop environment with the following elements:

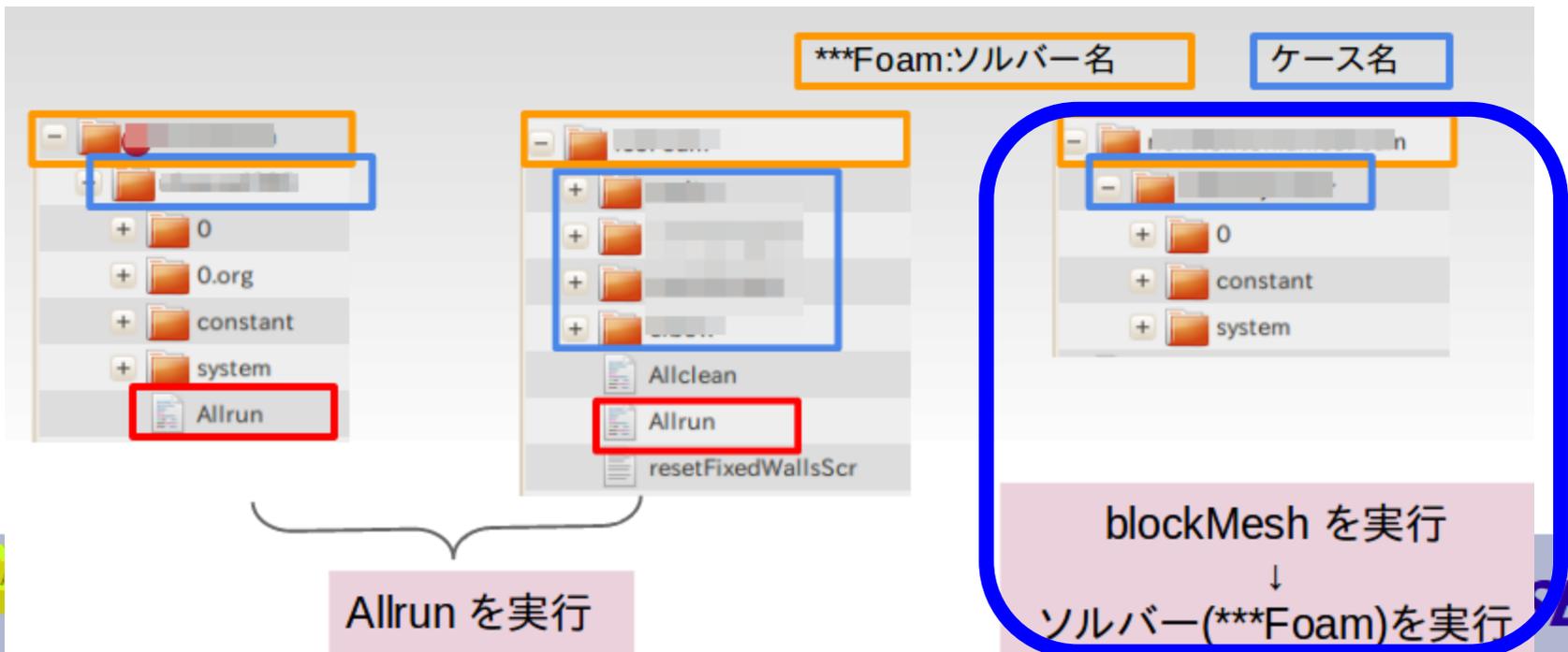
- DEXCS SWAK Application:** A window titled "DEXCS SWAK" with a menu bar "設定 ツール 終了". The main area is titled "DEXCS 十徳ナイフ" and contains:
 - Checkboxes for "FileManager" and "Terminal", both of which are checked.
 - A text input field containing the path "/home/custom/Desktop/case1" and a button labeled "case".
 - A field for "Ncore" set to "1", a field for "solver/application" containing "simpleFoam", and a dropdown menu for "ケースの初期化".
 - An "実行" (Execute) button.
- File Browser:** A window titled "case1 - ファイルブラウザ" with a menu bar "ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ヘルプ(H)". The address bar shows the path "custom / デスクトップ / case1".
- Terminal:** A window titled "Terminal" with a menu bar "ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)". The terminal output shows:

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
custom@custom: ~/Desktop/case1$
```

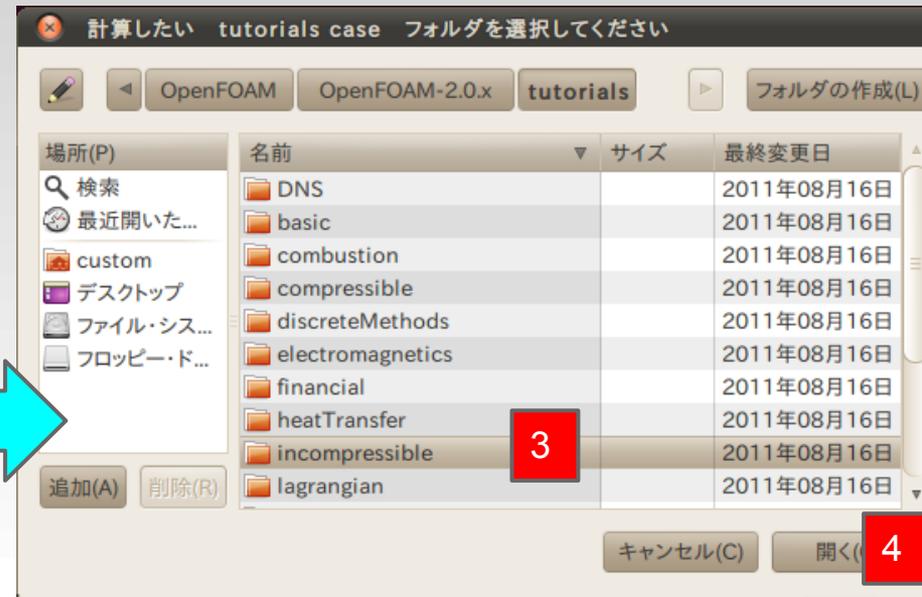
Red arrows indicate the workflow: one arrow points from the "case" button in the DEXCS SWAK window to the "case1" folder in the file browser, and another arrow points from the "Terminal" checkbox in the DEXCS SWAK window to the terminal window.

標準チュートリアルケース1の実行演習

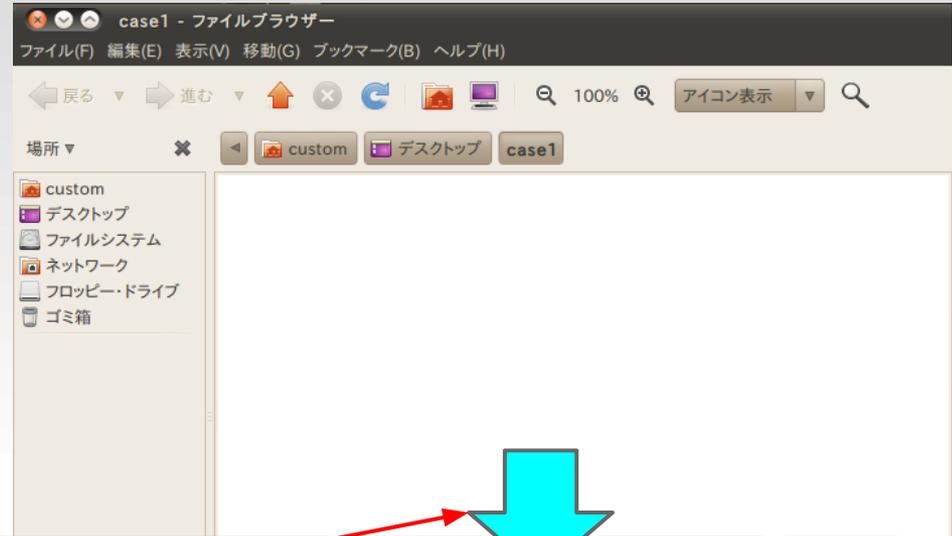
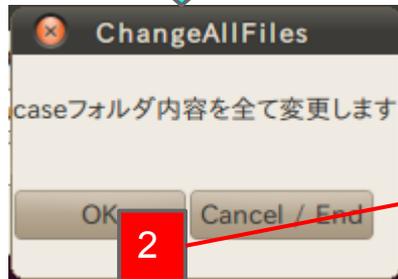
1. incompressible/pisoFoam/ras/cavity
2. multiphase/interFoam/laminar/damBreak
3. heatTransfer/buoyantBoussinesqSimpleFoam/iglooW



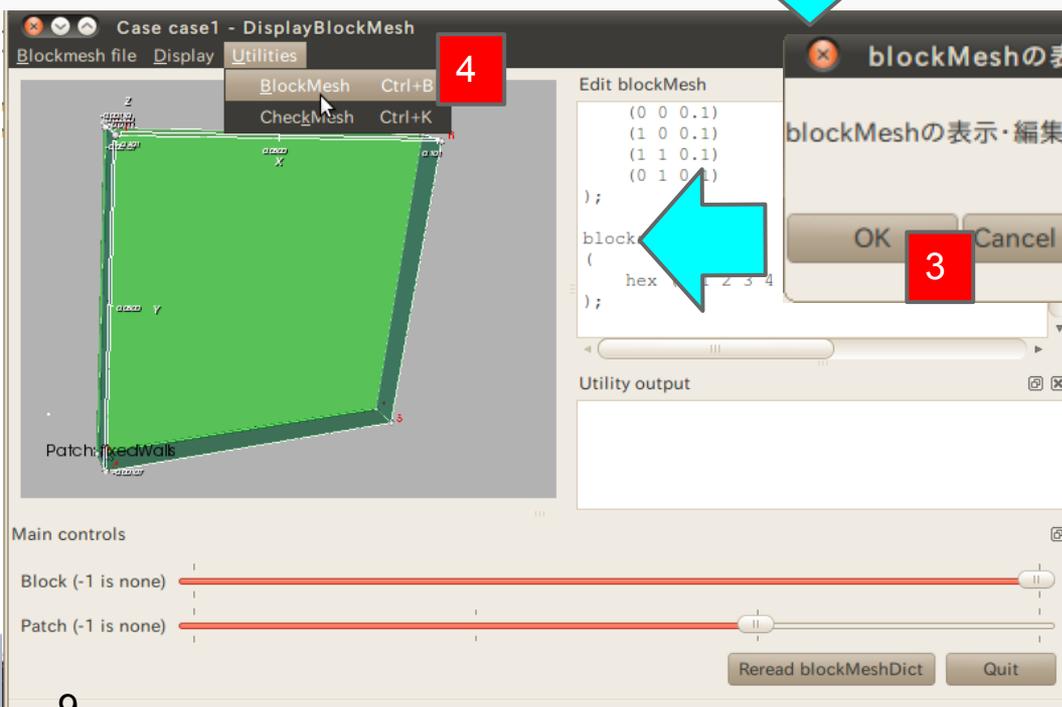
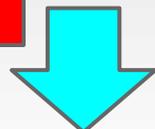
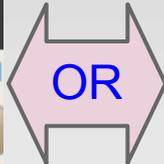
テンプレートケースの選択・変更



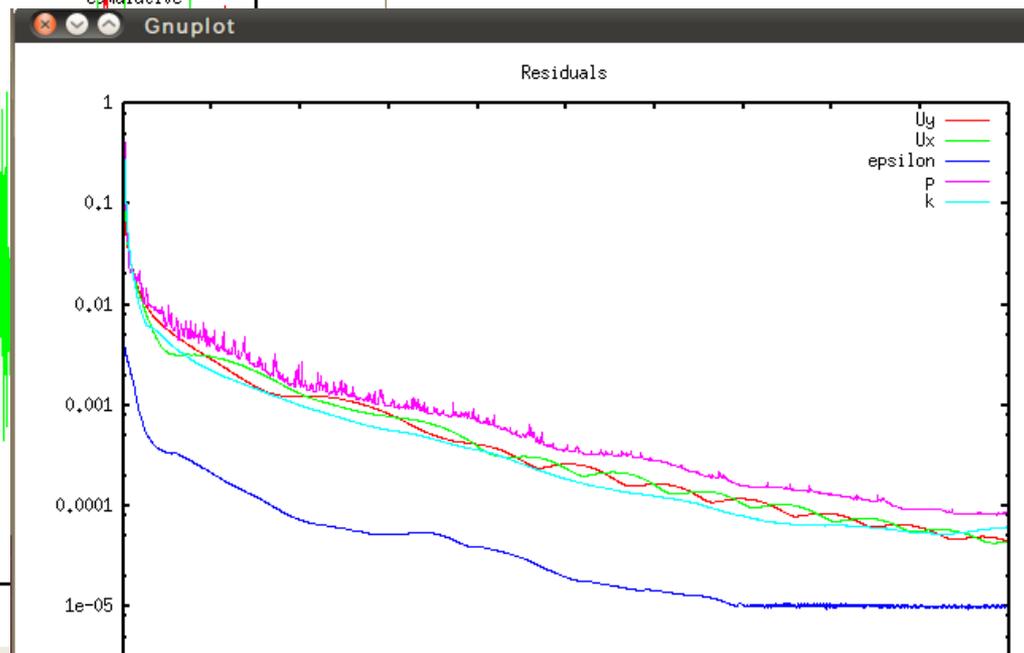
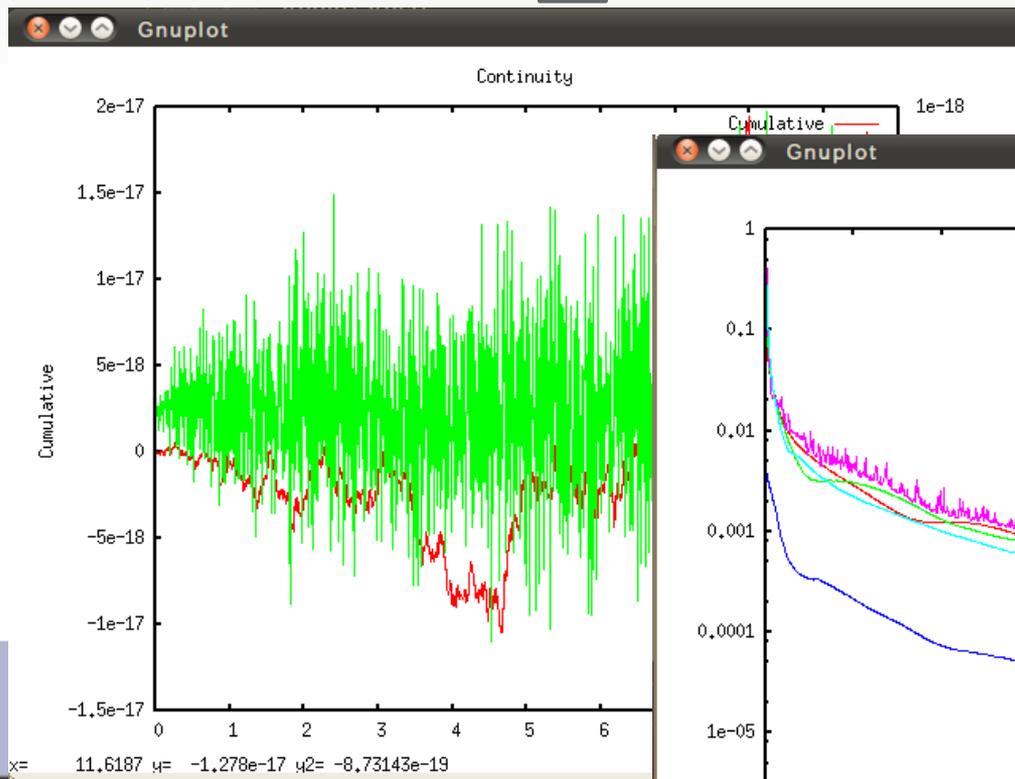
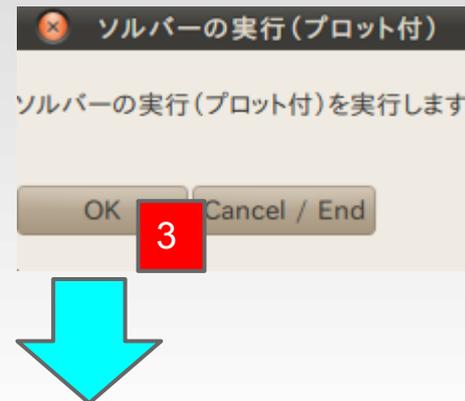
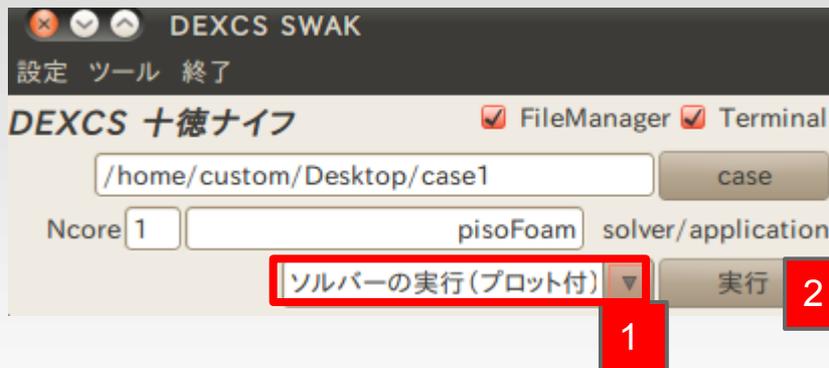
テンプレートケースの選択・変更



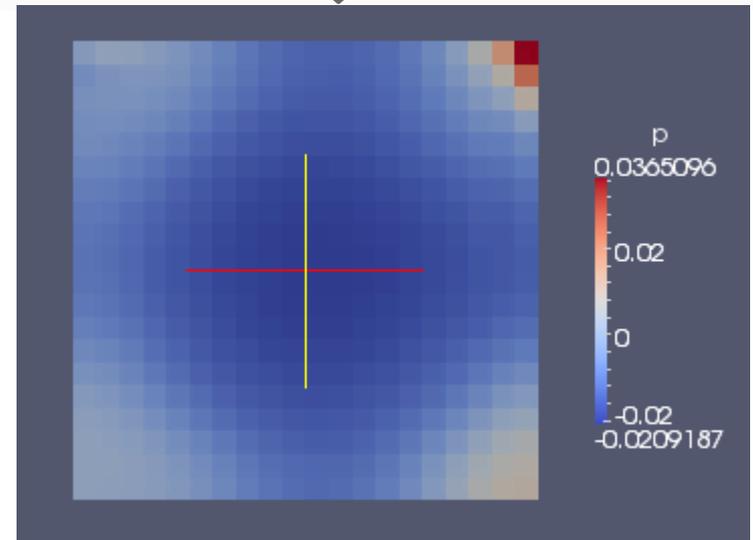
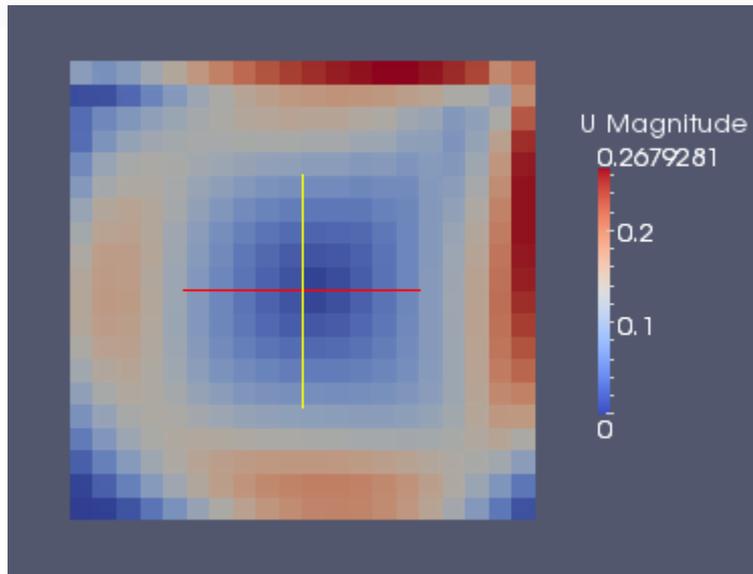
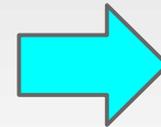
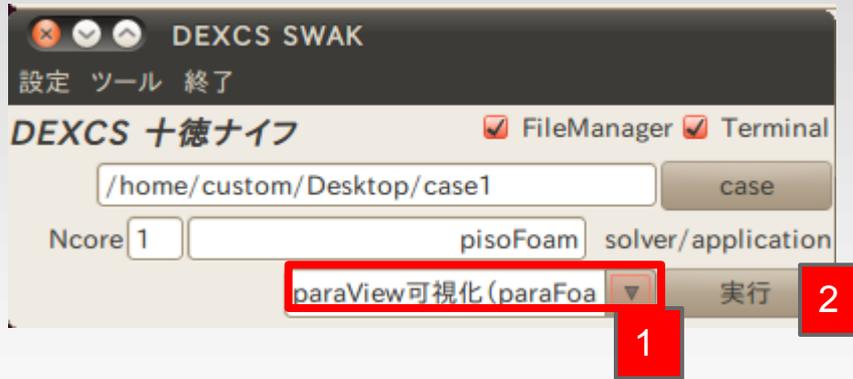
blockMesh



ソルバーの実行



標準チュートリアルケース1の可視化



標準チュートリアルケース2の実行演習

1. incompressible/pisoFoam/ras/cavity
2. multiphase/interFoam/laminar/damBreak
3. heatTransfer/buoyantBoussinesqSimpleFoam/iglooW

The screenshot displays the OpenFOAM GUI interface with three panels. The central panel is highlighted with a blue border and contains a red box around the 'Allrun' file. A bracket below this panel points to a pink box labeled 'Allrun を実行'. The top of the central panel has an orange box labeled '***Foam:ソルバー名'. The right panel has an orange box labeled 'ケース名' and a red box around the 'Allrun' file. A bracket below this panel points to a pink box labeled 'blockMesh を実行' and 'ソルバー(***Foam)を実行'. The left panel shows a file tree with folders '0', '0.org', 'constant', and 'system', and a file 'Allrun' highlighted with a red box.

***Foam:ソルバー名

ケース名

Allrun を実行

blockMesh を実行
↓
ソルバー(***Foam)を実行

テンプレートケースの選択・変更

case フォルダを選択してください

custom Desktop case2 フォルダの作成(L)

2

DEXCS SWAK
設定 ツール 終了

DEXCS 十徳ナイフ

FileManager Terminal

case 1

Ncore 1 pisoFoam solver/application

テンプレートケースの選択... 実行

3 4

計算したい tutorials case フォルダを選択してください

multiphase interFoam laminar damBreak フォルダの作成(L)

5

場所(P)

名前 サイズ 最終変更日

capillaryRise		2011年08月16日
damBreak		2011年08月16日
Allclean	421 バイト	2011年08月16日
Allrun	1.4 KB	2011年08月16日

6

追加(A) 削除(R)

キャンセル(C) 開く(C)

7

ChangeAllFiles

caseフォルダ内容を全て変更します

OK Cancel / End

10

Close Window

メッシュデータも変更しますか?

いいえ(N) はい(Y)

9

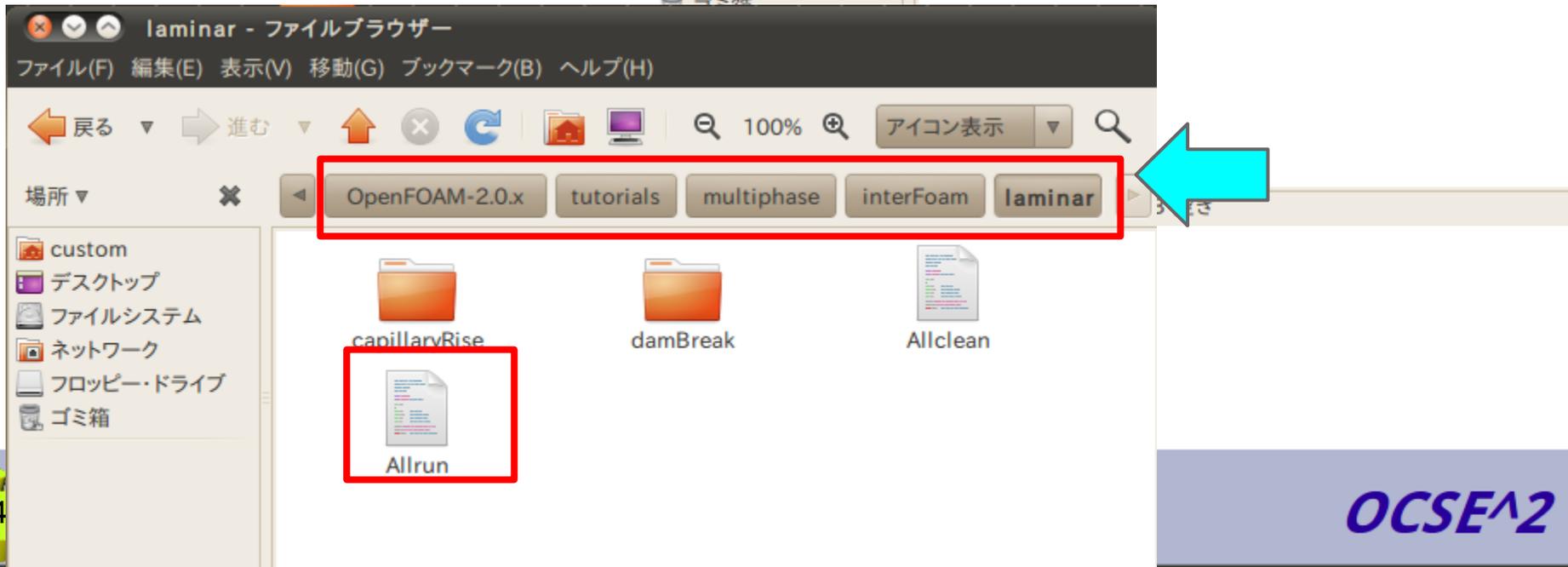
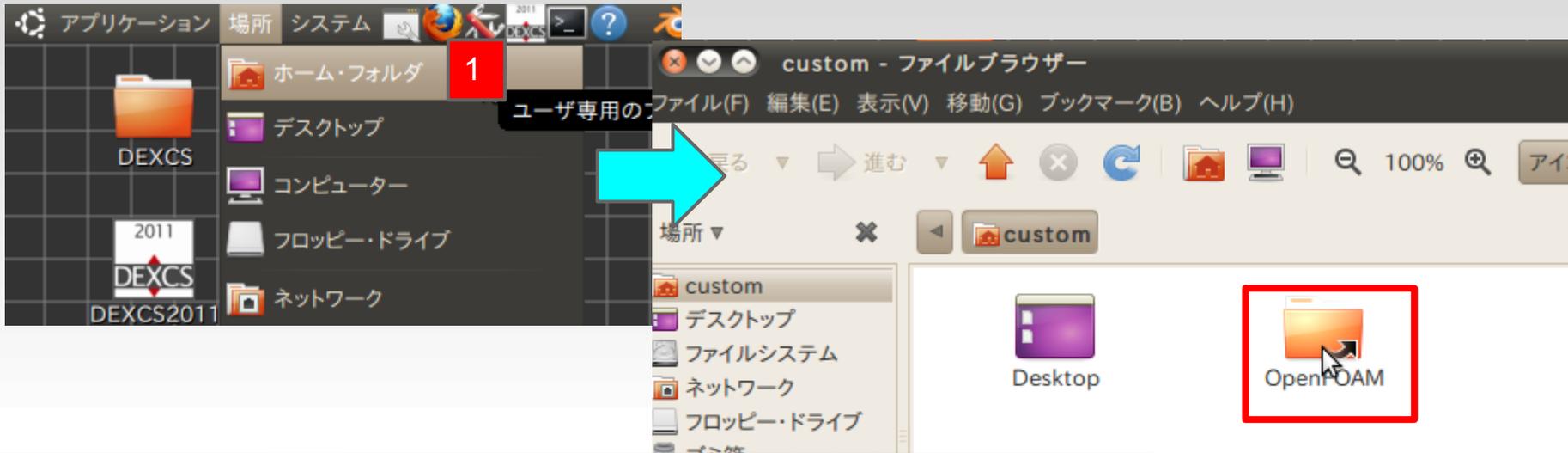
Set Solver

interFoam を
計算solverとして設定しました。

OK(O)

8

Allrun の確認



Allrun の内容確認

1 右ボタン
クリック

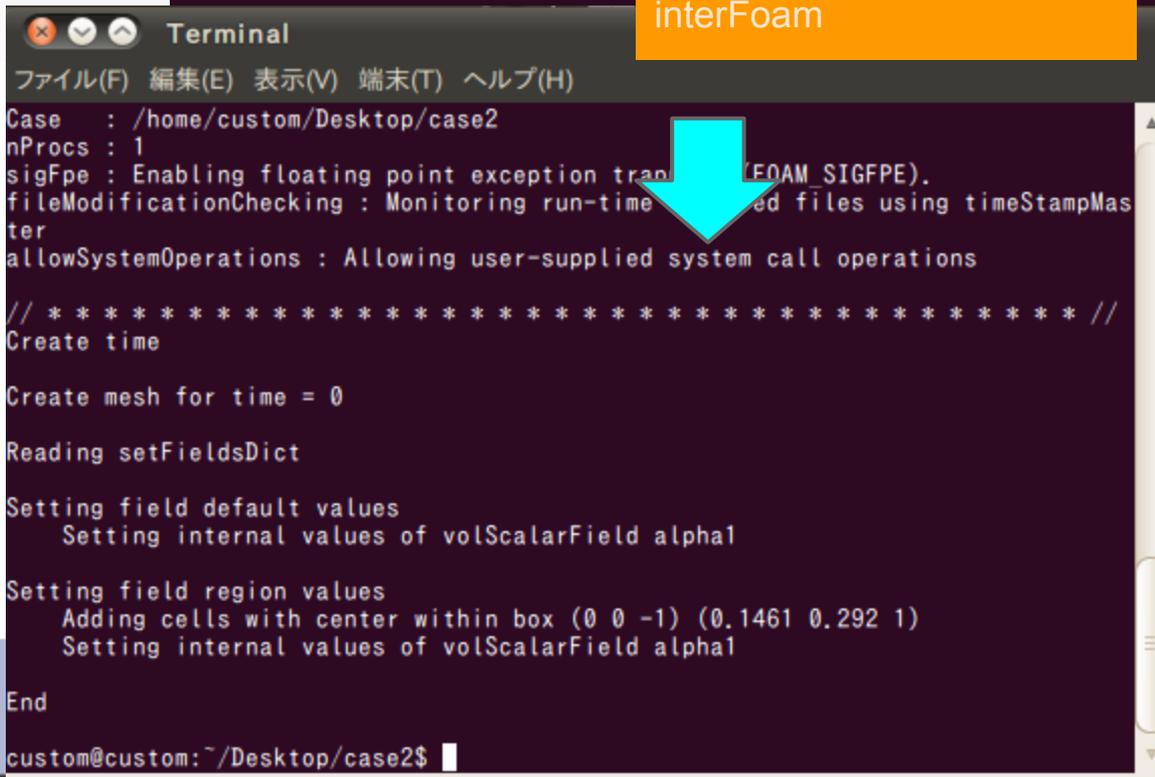
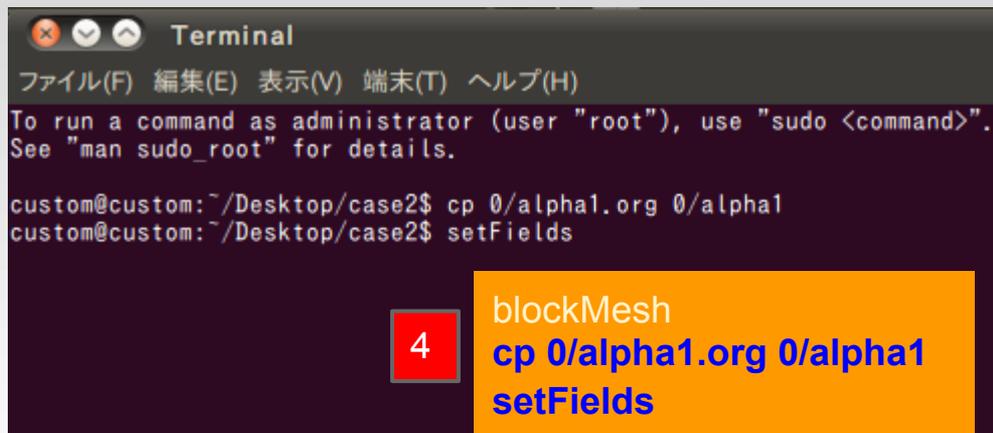
2

3

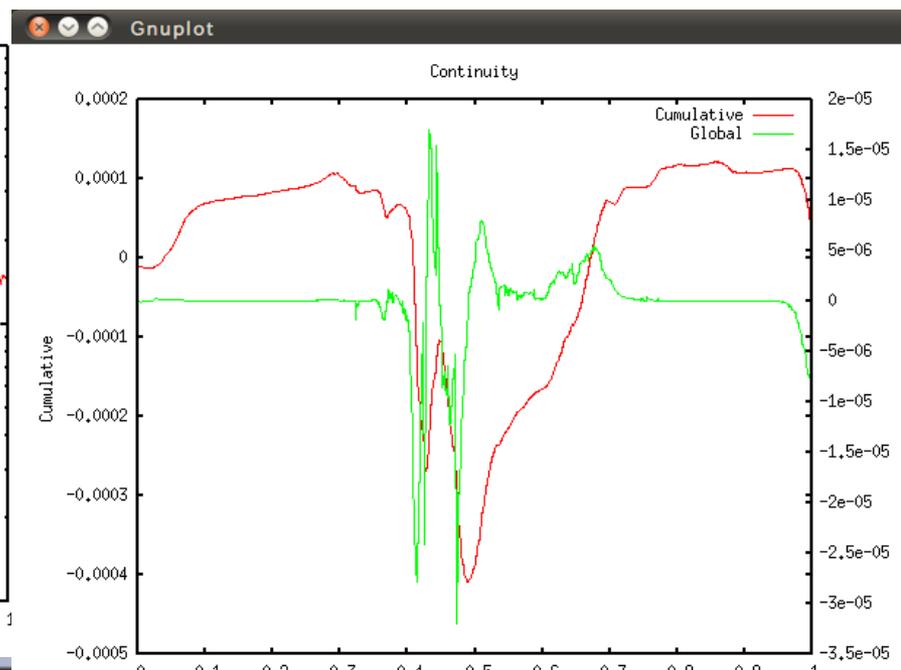
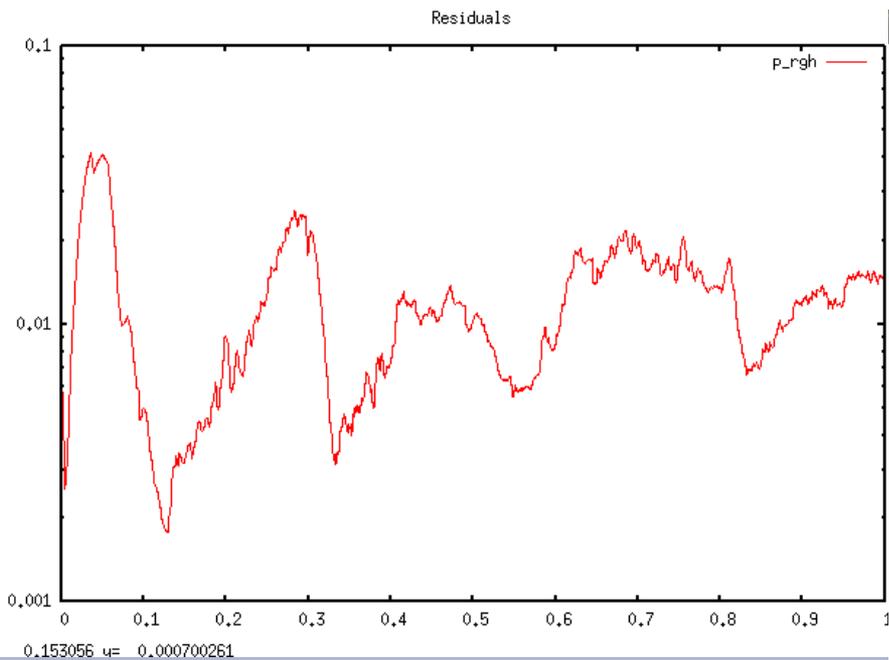
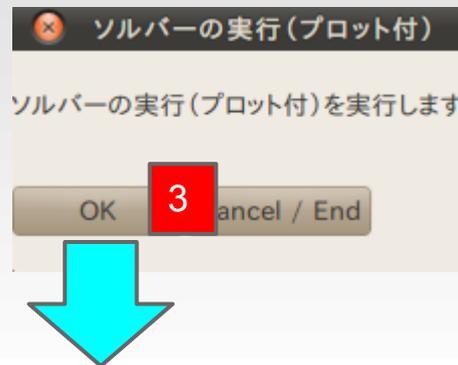
```
23 mv temp.$$ $controlDict
24 }
25
26 # Do damBreak
27 (
28   cd damBreak || exit
29
30   runApplication blockMesh
31   cp 0/alpha1.org 0/alpha1
32   runApplication setFields
33   runApplication `getApplication`
34 )
35
36 # Clone case
37 cloneCase damBreak damBreakFine
38 (
39
```

blockMesh
cp 0/alpha1.org 0/alpha1
setFields
interFoam

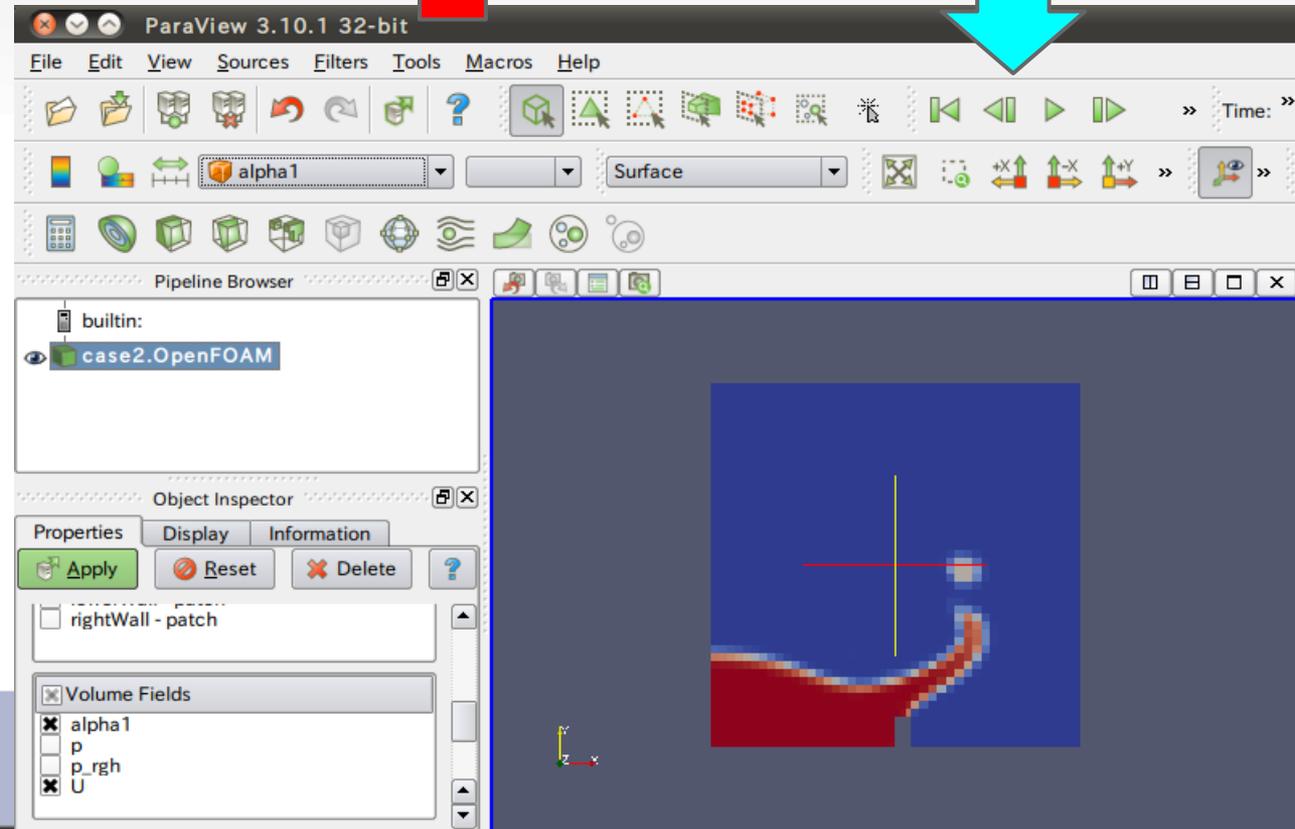
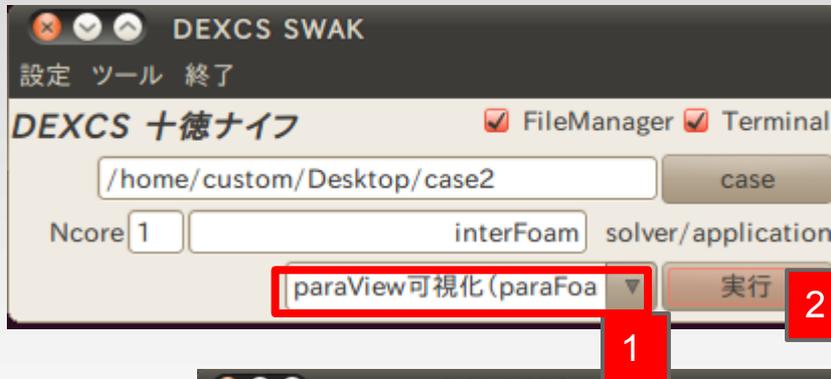
Allrun 内容の個別実行



ソルバーの実行



標準チュートリアルケース2の可視化



標準チュートリアルケース3の実行演習

1. incompressible/pisoFoam/ras/cavity
2. multiphase/interFoam/laminar/damBreak
3. heatTransfer/buoyantBoussinesqSimpleFoam

/iglooWithFridges



Allrun を実行

blockMesh を実行
↓
ソルバー(***Foam)を実行

テンプレートケースの選択・変更

case フォルダを選択してください

custom Desktop case3 フォルダの作成(L)

2

DEXCS SWAK
設定 ツール 終了
DEXCS 十徳ナイフ
FileManager Terminal
/home/custom/Desktop/case3 case 1
Ncore 1 interFoam solver/application
テンプレートケースの選択... 実行 4

5
heatTransfer buoyantBoussinesqSimpleFoam
iglooWithFridges 6

場所(P)	名前	サイズ	最終変更日
	0		2011年08月16日
	constant		2011年08月16日
	system		2011年08月16日
	Allrun	313 バイト	2011年08月16日

追加(A) 削除(R)

キャンセル(C) 開く(O)

ChangeAllFiles
caseフォルダ内容を全て変更します
OK Cancel / End 10

Close Window
メッシュデータも変更しますか?
いいえ(N) はい(Y) 9

Set Solver
buoyantBoussinesqSimpleFoam を
計算solverとして設定しました。
OK(O) 8

Allrun の内容確認

The screenshot shows a Mac desktop environment. A file browser window titled 'case3 - ファイルブラウザー' is open, displaying a folder named 'custom' containing a file named 'Allrun'. The file 'Allrun' is highlighted with a red rectangular box. To the right, a terminal window titled 'Allrun (~ / Desktop / case3) - gedit' is open, showing the contents of the 'Allrun' file. The terminal text is as follows:

```
1 #!/bin/sh
2 cd ${0%/*} || exit 1 # run from this directory
3
4 # Source tutorial run functions
5 . $WM_PROJECT_DIR/bin/tools/RunFunctions
6
7 runApplication blockMesh
8 runApplication snappyHexMesh -overwrite
9 runApplication `getApplication`
10
11 #
```

Lines 7, 8, and 9 of the terminal output are highlighted with an orange rectangular box. A blue arrow points from this box to a summary box at the bottom right of the image.

blockMesh
snappyHexMesh -overwrite
buoyantBoussinesqSimpleFoam

Allrun の実行

アプリケーション 場所 システム 6月22日 (金) 午前11:25 custom

stdout/stderr
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
EXEC ... rm -rf *: cp -r /home/custom/OpenFOAM/OpenFOAM-2.0.x/tutorials/heatTransfer/buoyantBoussinesqSimpleFoam/iglooWithFridges/* ./ ... Completed.
Allrun ファイルが見つかりました
chmod: /home/custom/OpenFOAM/OpenFOAM-2.0.x/tutorials/heatTransfer/buoyantBoussinesqSimpleFoam/iglooWithFridges/Allrun のパーミッションの変更: 許可されていない操作です
blockMeshDict ファイルが見つかりました

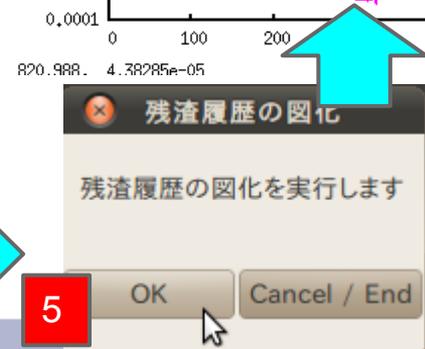
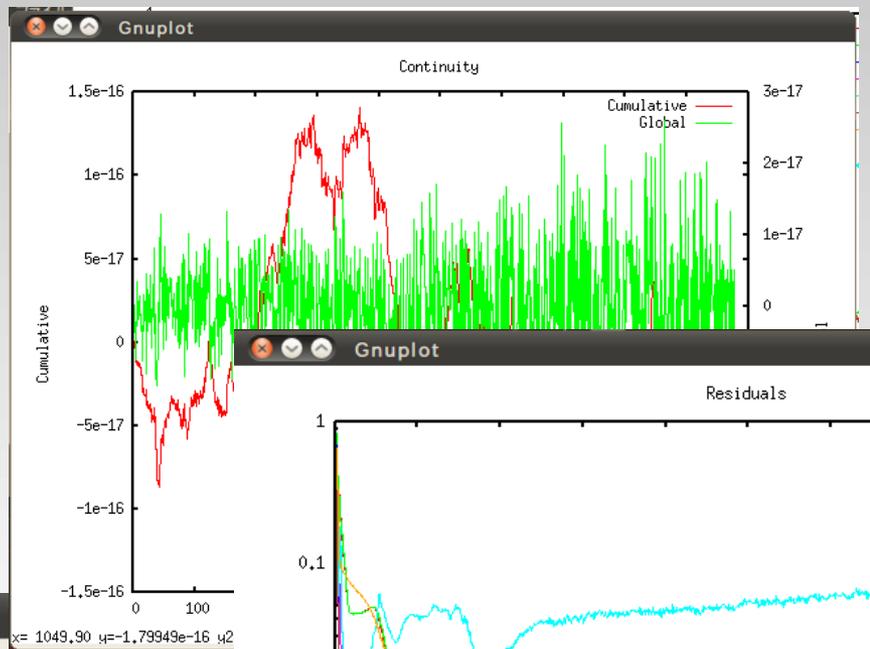
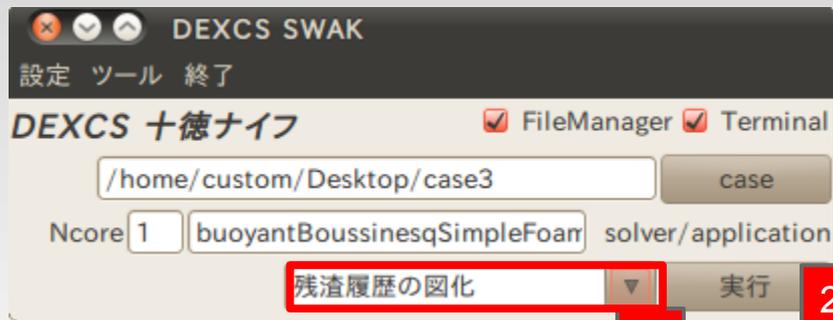
Terminal
ファイル(F) 編集(E) 表示(V) 端末(T) ヘルプ(H)
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
custom@custom: ~/Desktop/case3\$ chmod +x Allrun
custom@custom: ~/Desktop/case3\$./Allrun

case3 - ファイルブラウザ
ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ヘルプ(H)
場所 ▼ custom デスクトップ case3
custom
デスクトップ
ファイルシステム
ネットワーク
フロッピー・ドライブ
ゴミ箱
0 100
constant system

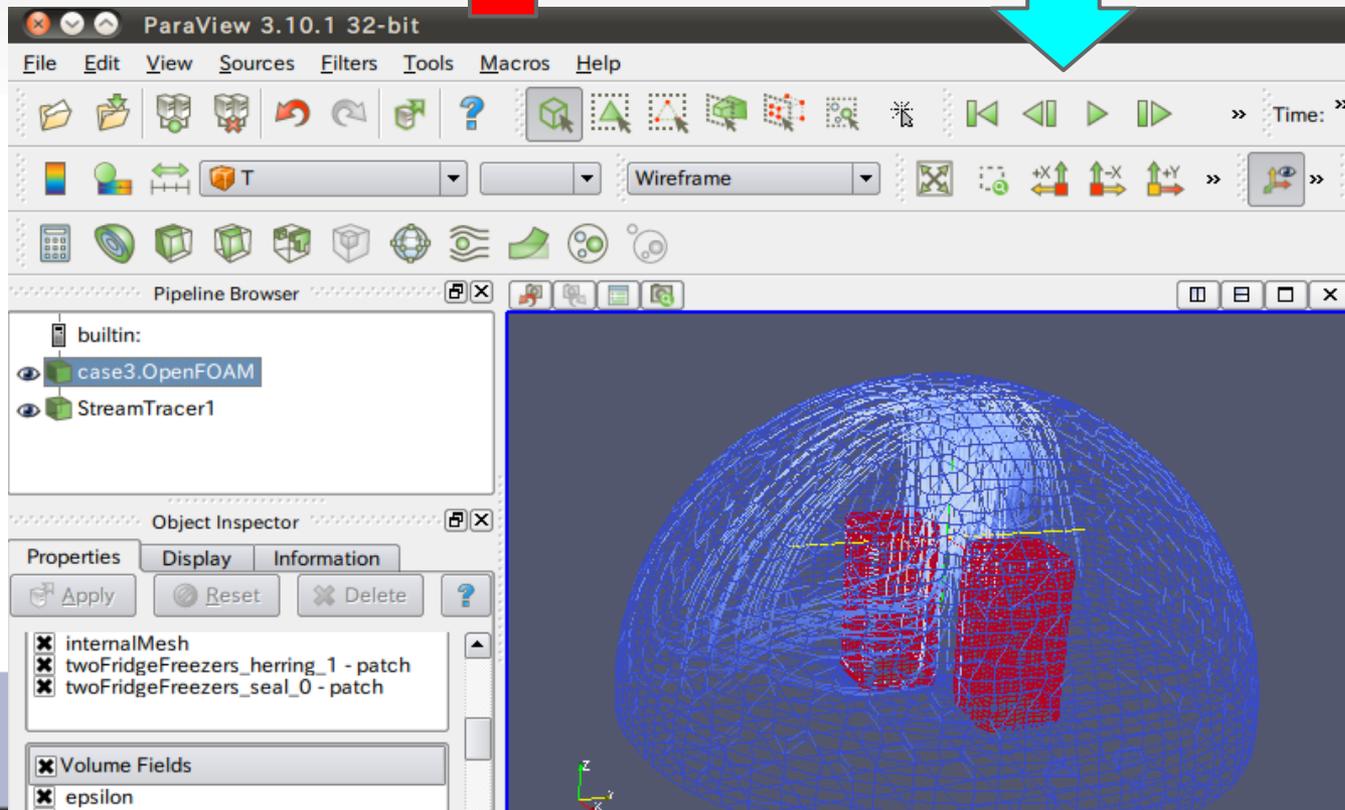
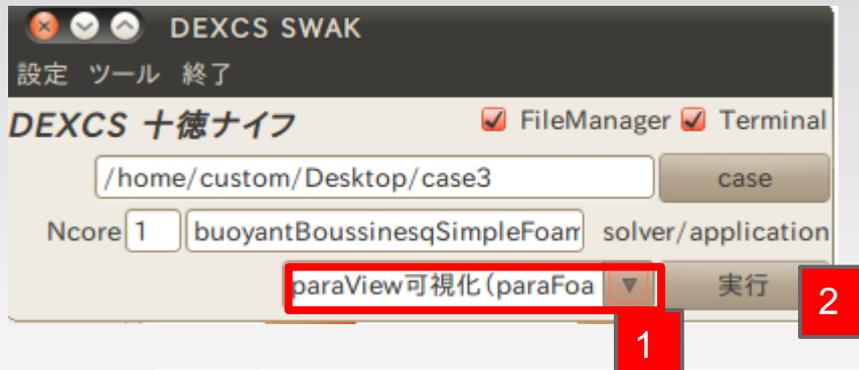
chmod +x Allrun
./Allrun

22

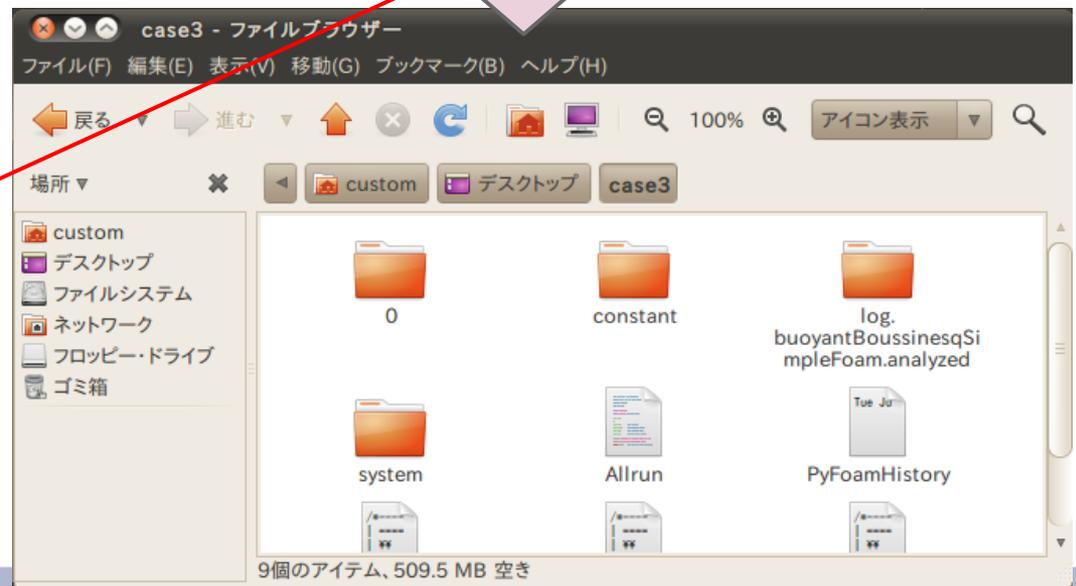
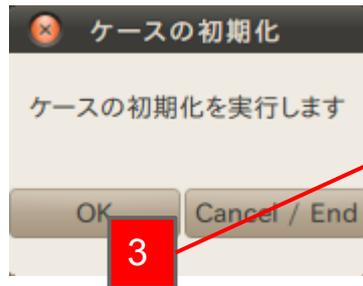
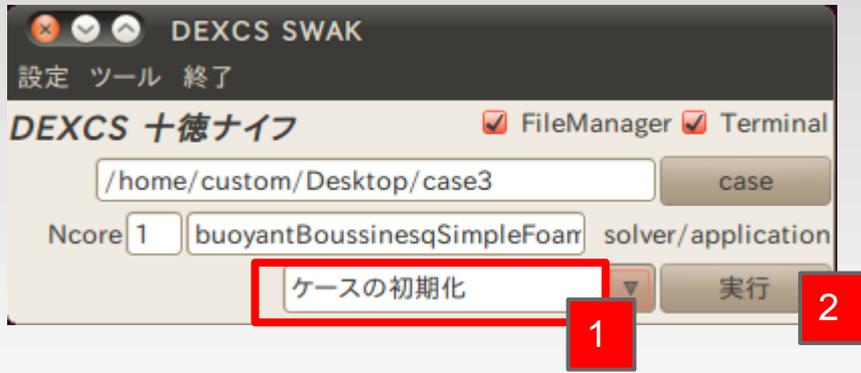
残差履歴の図化



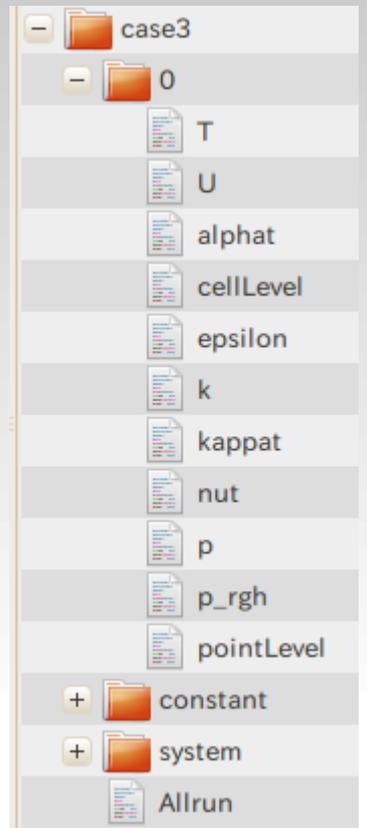
標準チュートリアルケース3の可視化



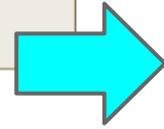
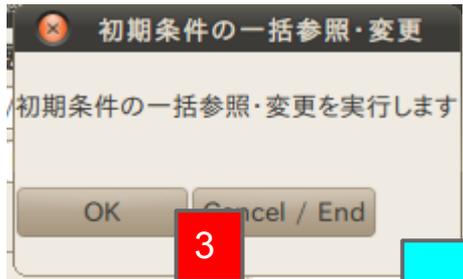
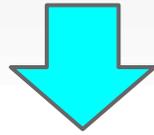
ケースの初期化



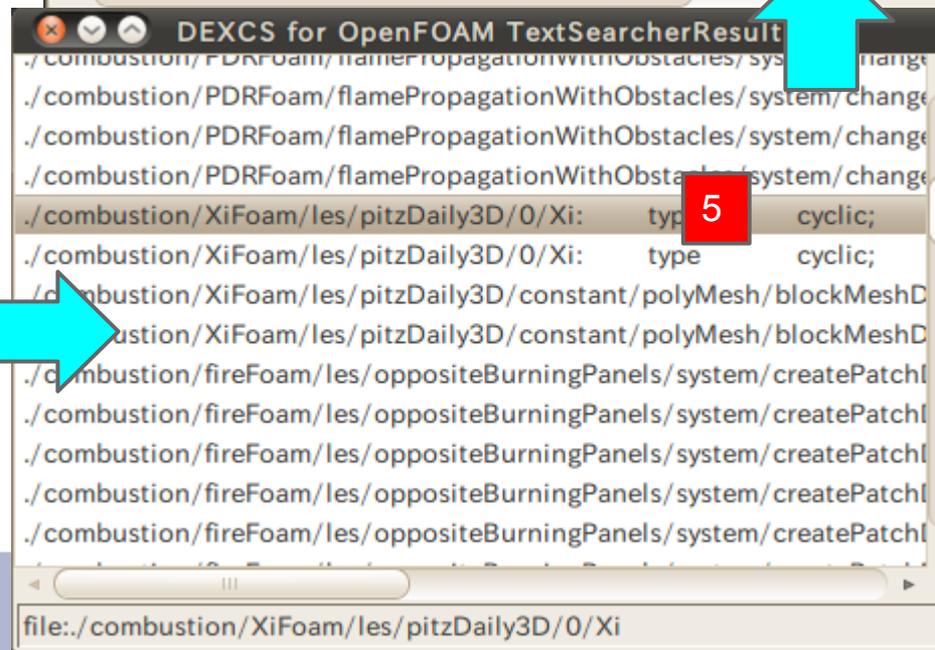
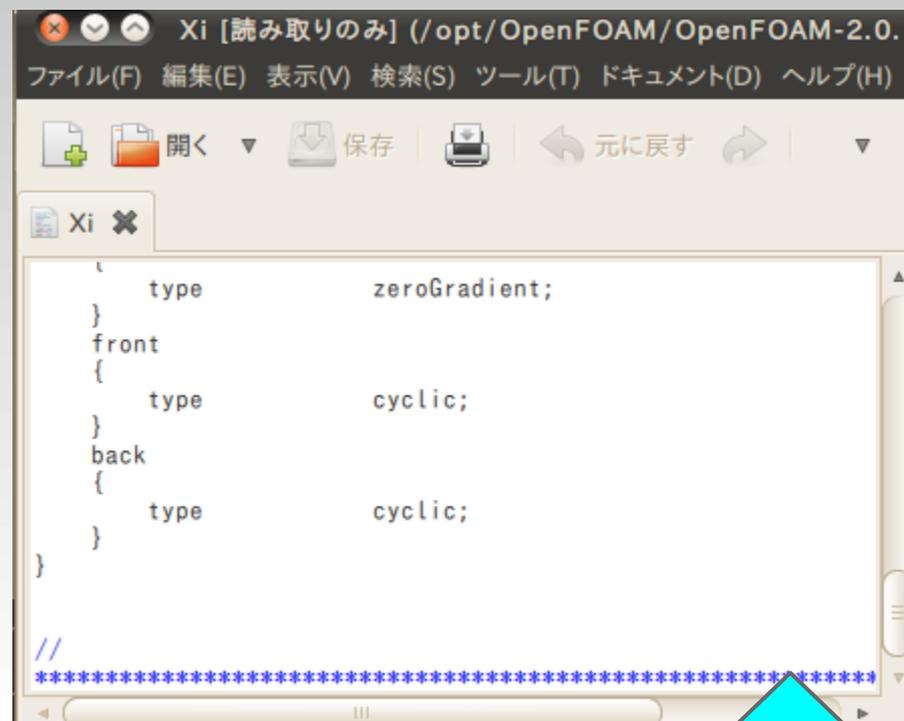
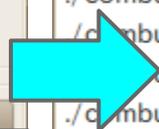
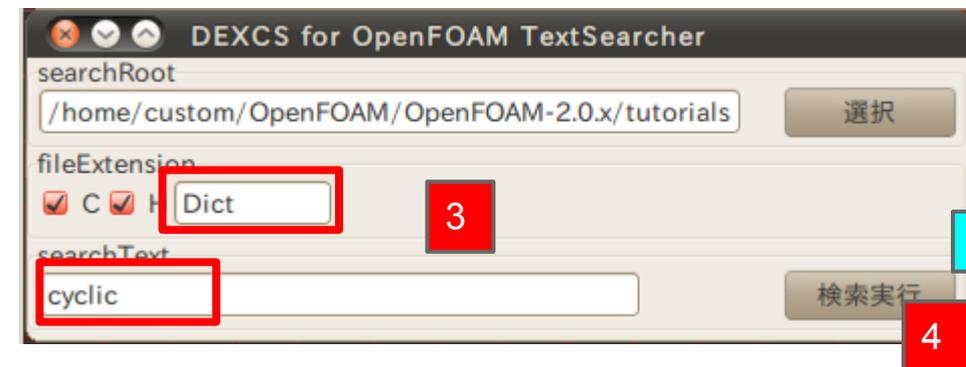
初期条件の一括参照・変更



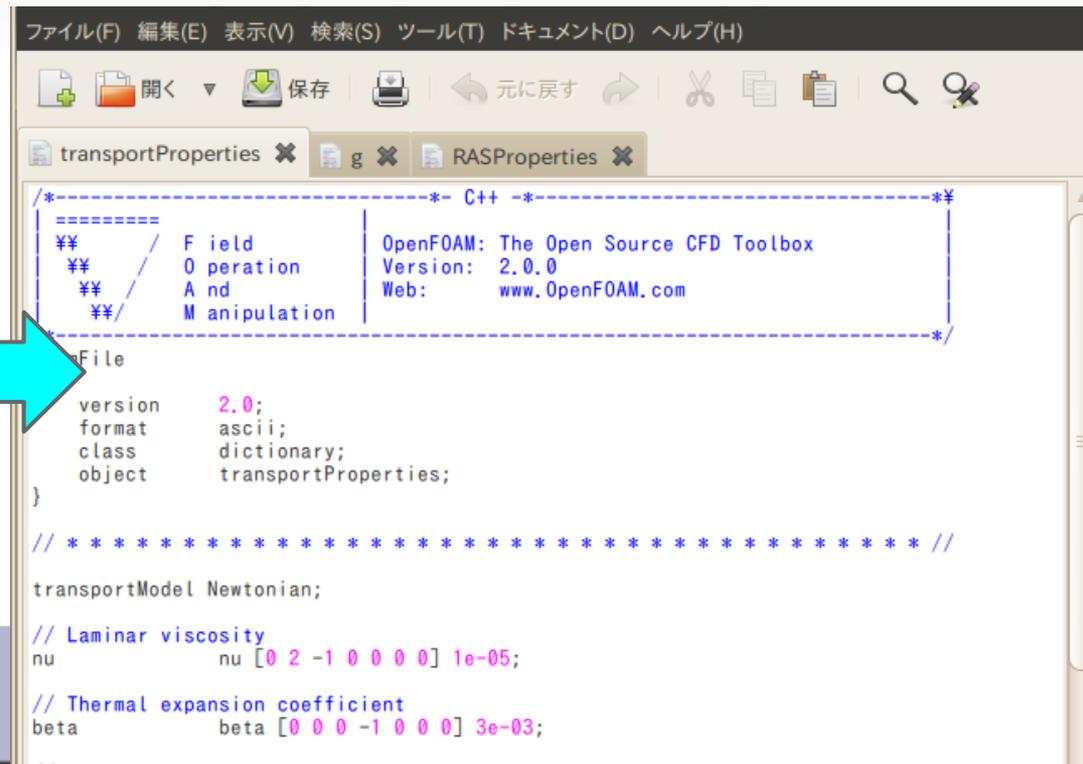
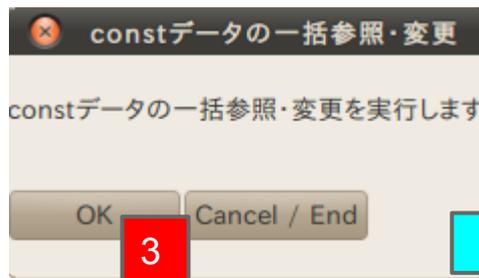
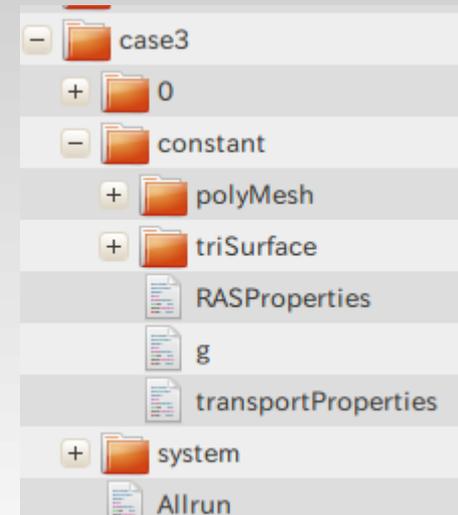
boundaryファイルは初期条件Fileldファイルではありませんが、参照用として同時オープンされます。



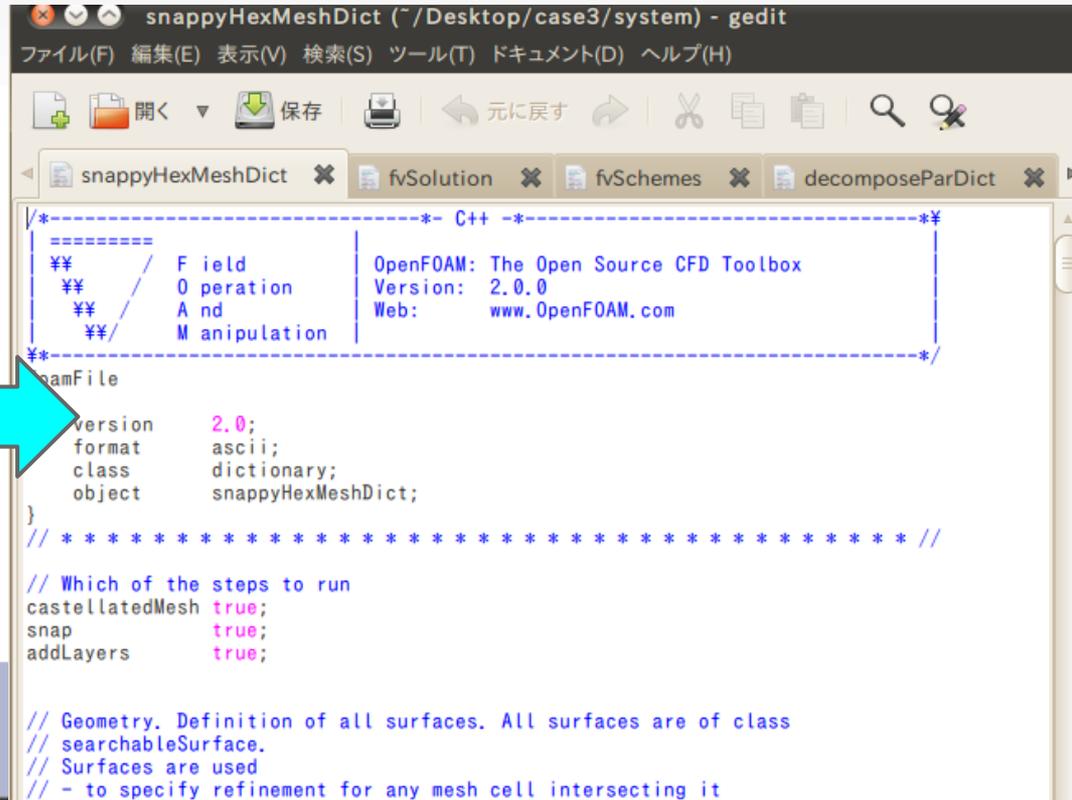
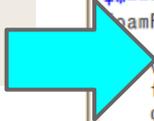
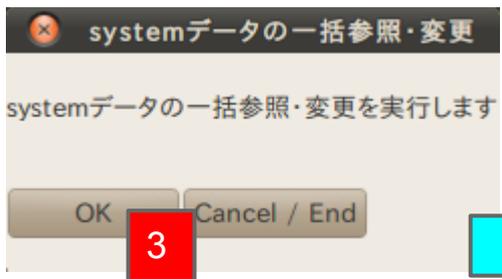
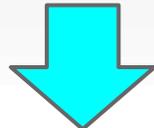
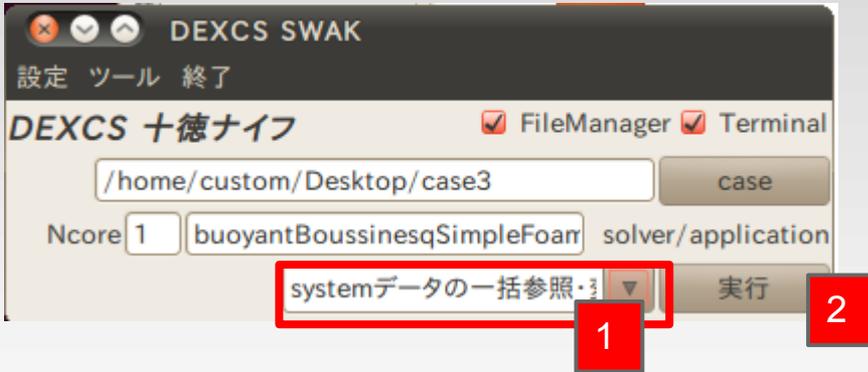
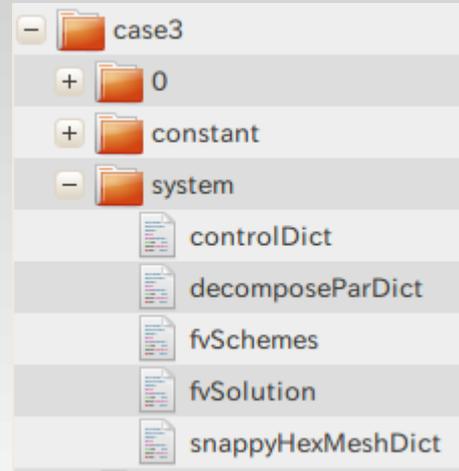
ソースコード検索



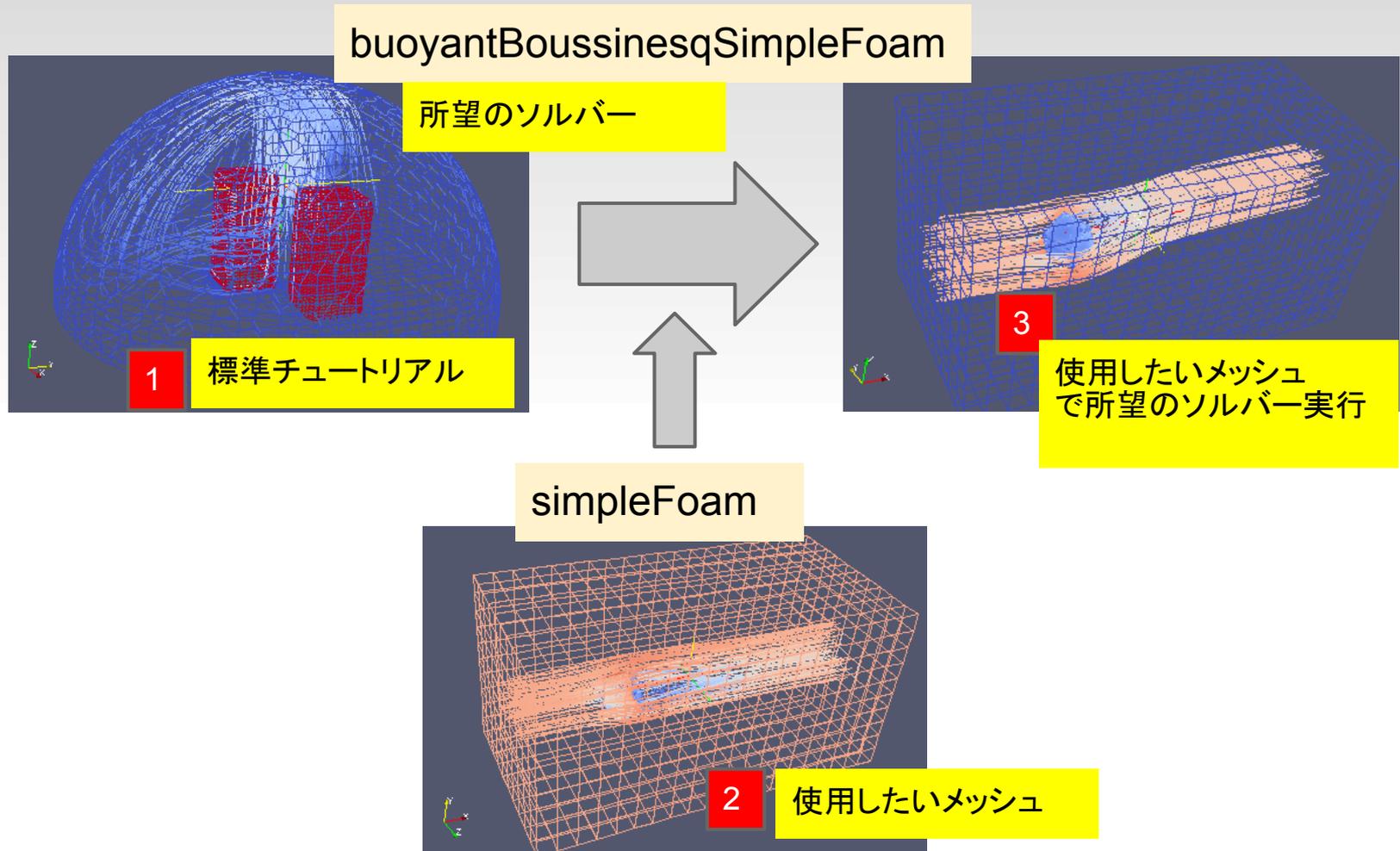
constデータの一括参照・変更



systemデータの一括参照・変更

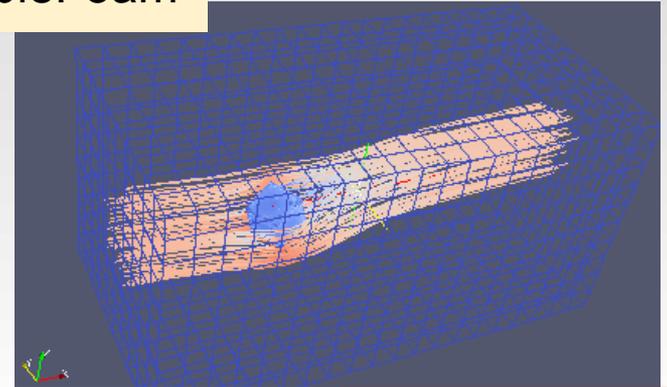
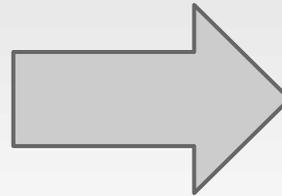
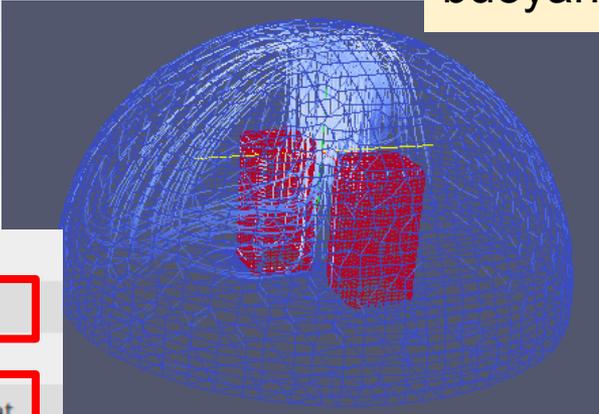


標準チュートリアルベースのメッシュ変更



標準チュートリアルベースのメッシュ変更

buoyantBoussinesqSimpleFoam

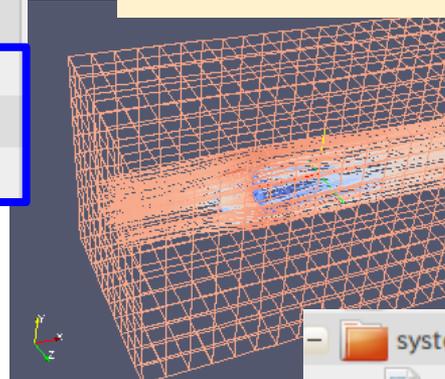


- 0
- T
- U
- alphat
- epsilon
- k
- kappat
- nut
- p
- p_rgh

- constant
- polyMesh
- triSurface
- RASProperties
- g
- transportProperties

- system
- controlDict
- decomposeParDict
- fvSchemes
- fvSolution
- snappyHexMeshDict

simpleFoam



- 0
- U
- U.prepotential
- epsilon
- k
- omega
- p
- p.prepotenti
- phi

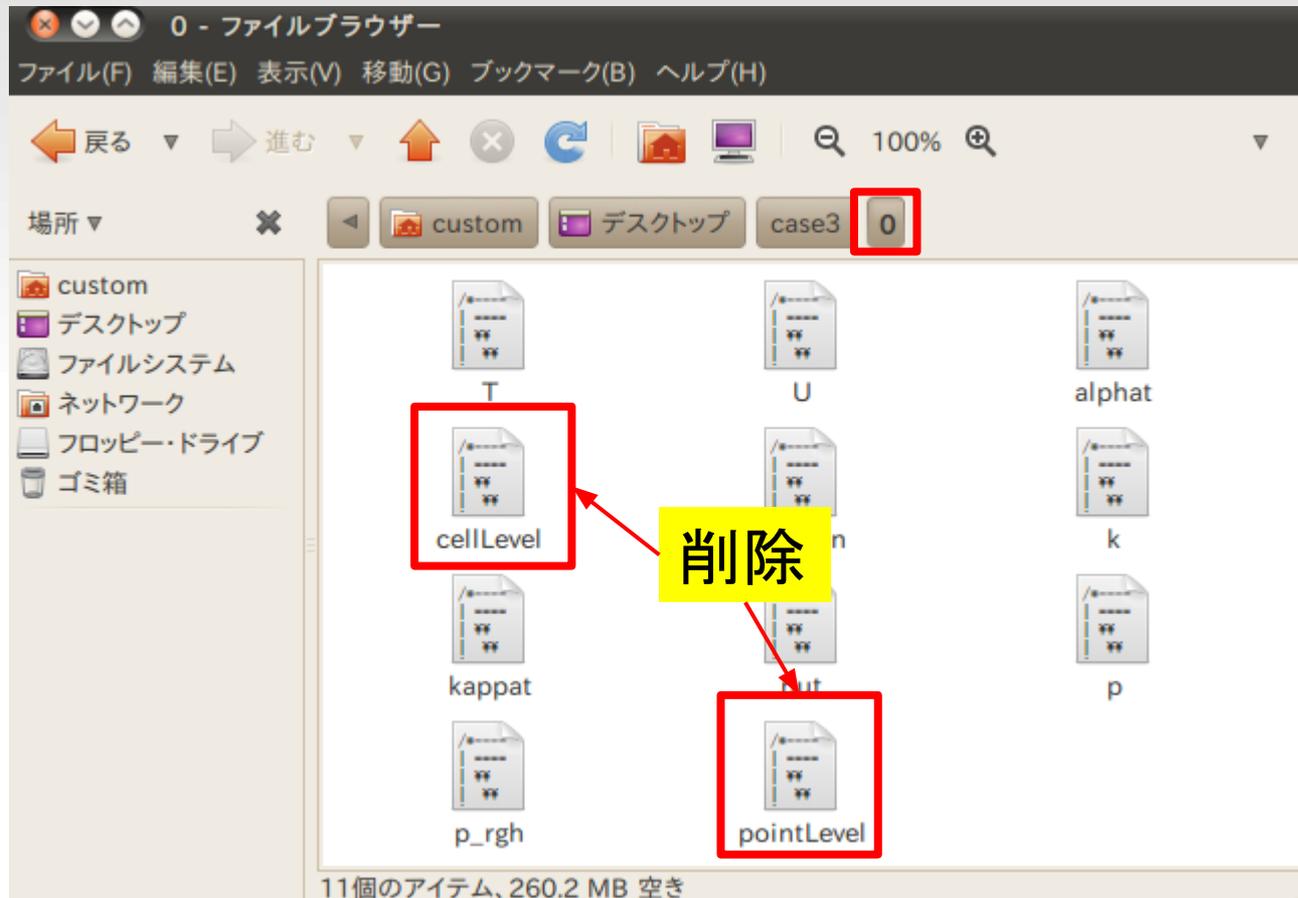
T, p_rgh, ... ??

流用可能

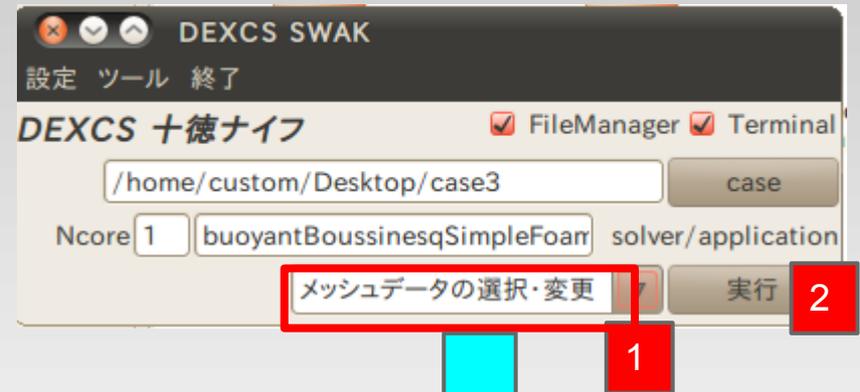
- constant
- polyMesh
- RASProperties
- transportProperties

- system
- controlDict
- decomposeParDict
- fvSchemes
- fvSolution

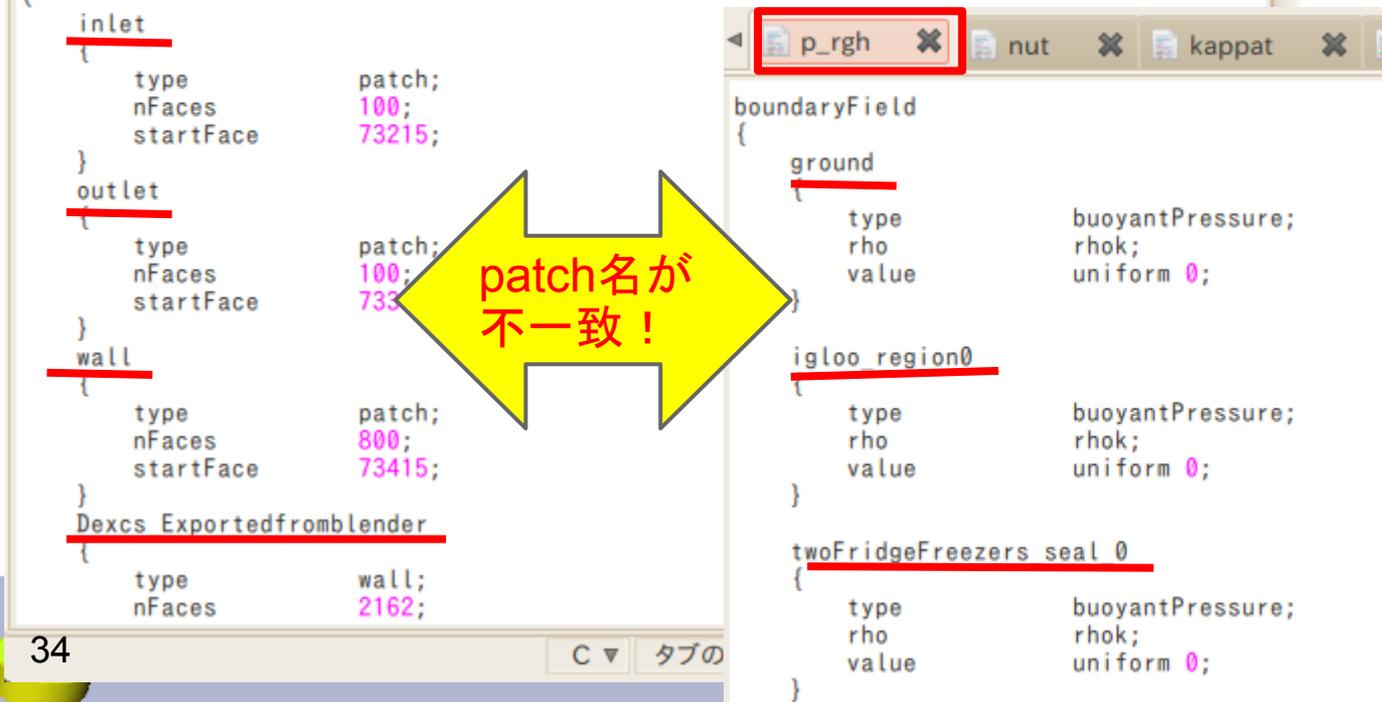
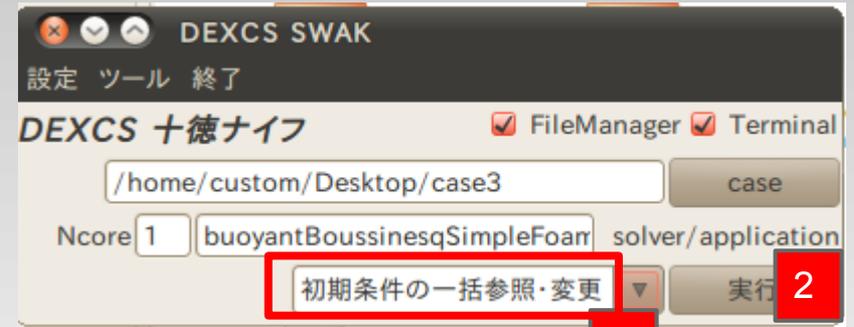
Field変数の確認と不要データの削除



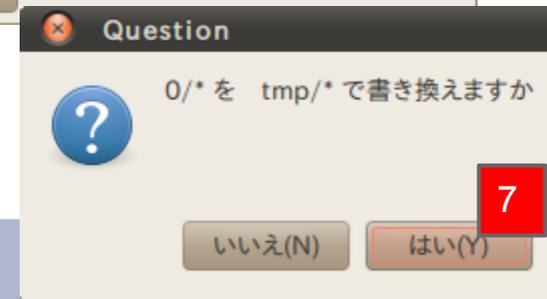
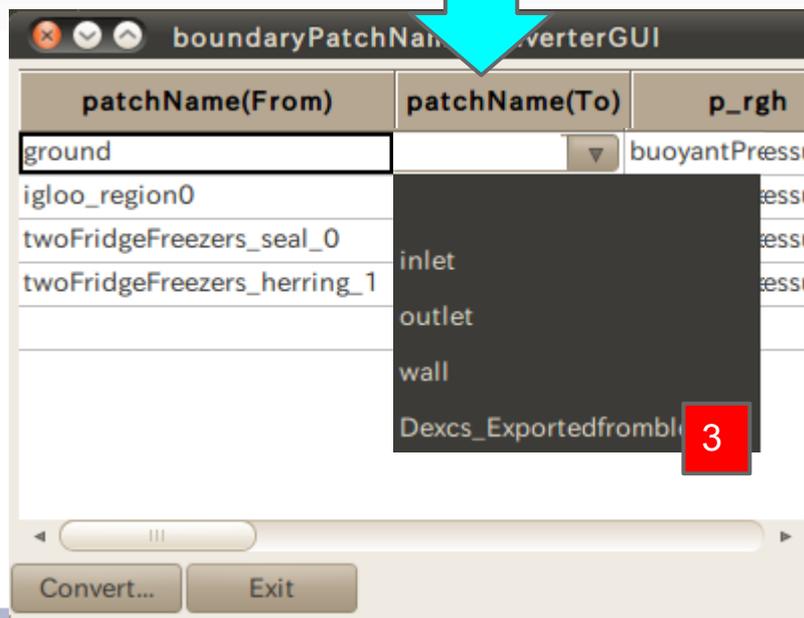
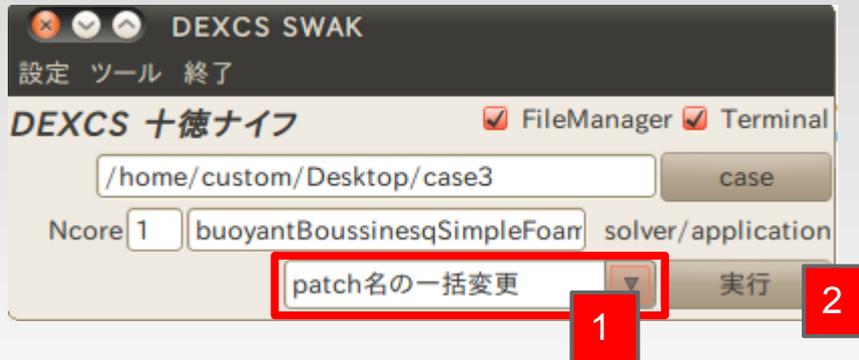
メッシュ変更



初期・境界条件 データの確認



patch名の一括変更



流用可能な初期条件ファイルをコピー

The image illustrates a three-step process for copying and renaming initial condition files in a Windows File Explorer window:

- 削除 (Delete):** A red box highlights the file 'epsilon' in the 'case3' folder. A red arrow points to a yellow box labeled '削除' (Delete).
- コピー (Copy):** A red box highlights the file 'U' in the 'case3' folder. A red arrow points to a yellow box labeled 'コピー' (Copy).
- 名前変更 (Rename):** A red box highlights the file 'U' in the 'test1' folder. A red arrow points to a yellow box labeled '名前変更' (Rename).

Additional annotations include:

- A red box around the file 'k' in the 'case3' folder, with a blue arrow pointing to the file 'U' in the 'test1' folder.
- A red box around the file 'U.prepotential' in the 'test1' folder, with a blue arrow pointing to the file 'U' in the 'case3' folder.
- A red box around the file 'epsilon' in the 'test1' folder, with a blue arrow pointing to the file 'U' in the 'case3' folder.

The File Explorer window shows the following details:

- Window title: 0 - ファイルブラウザ
- Menu: ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ヘルプ(H)
- Navigation: 戻る (Back), 進む (Forward), 上 (Up), 下 (Down), 刷新 (Refresh), ホーム (Home), コンピュータ (Computer), 検索 (Search), 100% (Zoom), アイコン表示 (Icon View)
- Location: 場所 (Location) custom デスクトップ case3 0
- Files in 'case3': T, U, alphas, k, kappat, nut, epsilon
- Location: 場所 (Location) custom デスクトップ test1 exe 0
- Files in 'test1': U, U.prepotential, epsilon, k, omega, p

初期・境界条件データの再確認

boundary (~ / Desktop / case3 / constant / polyMesh) - 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

p_rgh nut kappat alphas T boundary U p k epsilon

```
inlet
{
  type          patch;
  nFaces        100;
  startFace     73215;
}
outlet
{
  type          patch;
  nFaces        100;
  startFace     73315;
}
wall
{
  type          patch;
  nFaces        889;
  startFace     73415;
}
Dexcs_Exportedfromblender
{
  type          wall;
  nFaces        2162;
}
```

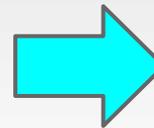
patch名はすべて一致

```
dimensions [ 0 1 -1 0 0 0 0 ];
internalField uniform (0 0 0);
boundaryField
{
  wall
  {
    type slip;
  }
  inlet
  {
    type          fixedValue;
    value         uniform (10 0 0);
  }
  Dexcs_Exportedfromblender
  {
    type          fixedValue;
    value         uniform (0 0 0);
  }
  outlet
  {
    type zeroGradient;
  }
}
```

```
inlet
{
  type          buoyantPressure;
  rho           rhok;
  value         uniform 0;
}
Dexcs_Exportedfromblender
{
  type          buoyantPressure;
  rho           rhok;
  value         uniform 0;
}
twoFridgeFreezers_seal_0
{
  type          buoyantPressure;
  rho           rhok;
  value         uniform 0;
}
twoFridgeFreezers_herring_1
```

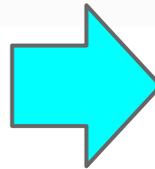
patch名は一部不一致

patch名の一括整合



細部手修正 (p_rgh)

```
p_rgh
14 internalField uniform 0;
15
16 boundaryField
17 {
18   inlet
19   {
20     type buoyantPressure;
21     rho rhok;
22     value uniform 0;
23   }
24   Dexcs_Exportedfromblender
25   {
26     type buoyantPressure;
27     rho rhok;
28     value uniform 0;
29   }
30   wall
31   {
32     type zeroGradient;
33   }
34   outlet
35   {
36     type zeroGradient;
37   }
38 }
```



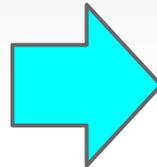
```
p_rgh p nut
12 dimensions [ 0 2 -2 0 0 0 ];
13
14 internalField uniform 0;
15
16 boundaryField
17 {
18   inlet
19   {
20     type buoyantPressure;
21     rho rhok;
22     value uniform 0;
23   }
24   Dexcs_Exportedfromblender
25   {
26     type buoyantPressure;
27     rho rhok;
28     value uniform 0;
29   }
30   wall
31   {
32     type slip;
33   }
34   outlet
35   {
36     type fixedValue;
37     value uniform 0;
38   }
39 }
```

変更箇所

全Field共通
変更箇所

細部手修正 (p)

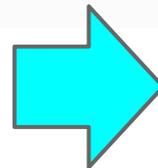
```
p x
..
12 dimensions [ 0 2 -2 0 0 0 ];
13
14 internalField uniform 0;
15
16 boundaryField
17 {
18   inlet
19   {
20     type calculated;
21     value $internalField;
22   }
23   Dexcs_Exportedfromblender
24   {
25     type calculated;
26     value $internalField;
27   }
28   wall
29   {
30     type zeroGradient;
31   }
32   outlet
33   {
34     type zeroGradient;
35   }
36 }
37
```



```
p x nut x kappat
..
12 dimensions [ 0 2 -2 0 0 0 ];
13
14 internalField uniform 0;
15
16 boundaryField
17 {
18   inlet
19   {
20     type calculated;
21     value $internalField;
22   }
23   Dexcs_Exportedfromblender
24   {
25     type calculated;
26     value $internalField;
27   }
28   wall
29   {
30     type slip;
31   }
32   outlet
33   {
34     type zeroGradient;
35   }
36 }
..
```

細部手修正 (T)

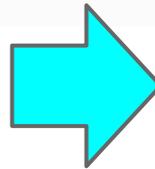
```
T x boundary x
..
12 dimensions [ 0 0 0 1 0 0 0 ];
13
14 internalField uniform 265;
15
16 boundaryField
17 {
18   inlet
19   {
20     type fixedValue;
21     value uniform 265;
22   }
23   Dexcs_Exportedfromblender
24   {
25     type fixedValue;
26     value uniform 265;
27   }
28   wall
29   {
30     type zeroGradient;
31   }
32   outlet
33   {
34     type zeroGradient;
35   }
36 }
37
```



```
T x boundary x
..
12 dimensions [ 0 0 0 1 0 0 0 ];
13
14 internalField uniform 265;
15
16 boundaryField
17 {
18   inlet
19   {
20     type fixedValue;
21     value uniform 265;
22   }
23   Dexcs_Exportedfromblender
24   {
25     type fixedValue;
26     value uniform 365;
27   }
28   wall
29   {
30     type slip;
31   }
32   outlet
33   {
34     type zeroGradient;
35   }
36 }
37
```

細部手修正 (U)

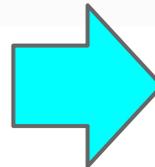
```
U x T x boundary x
..
12 dimensions [ 0 1 -1 0 0 0 ];
13
14 internalField uniform (0 0 0);
15
16 boundaryField
17 {
18   inlet
19   {
20     type fixedValue;
21     value uniform (0 0 0);
22   }
23   Dexcs_Exportedfromblender
24   {
25     type fixedValue;
26     value uniform (0 0 0);
27   }
28   wall
29   {
30     type zeroGradient;
31   }
32   outlet
33   {
34     type zeroGradient;
35   }
36 }
```



```
U x p_rgh x p x
..
12 dimensions [ 0 1 -1 0 0 0 ];
13
14 internalField uniform (0 0 0);
15
16 boundaryField
17 {
18   inlet
19   {
20     type fixedValue;
21     value uniform (0.2 0 0);
22   }
23   Dexcs_Exportedfromblender
24   {
25     type fixedValue;
26     value uniform (0 0 0);
27   }
28   wall
29   {
30     type slip;
31   }
32   outlet
33   {
34     type zeroGradient;
35   }
36 }
37 }
```

細部手修正 (nut)

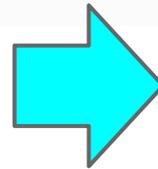
```
nut x kappat x k x
12
13 dimensions [ 0 2 -1 0 0 0 ];
14
15 internalField uniform 0;
16
17 boundaryField
18 {
19   inlet
20   {
21     type nutkWallFunction;
22     value uniform 0;
23   }
24   Dexcs_Exportedfromblender
25   {
26     type nutkWallFunction;
27     value uniform 0;
28   }
29   wall
30   {
31     type zeroGradient;
32   }
33   outlet
34   {
35     type zeroGradient;
36   }
37 }
```



```
nut x kappat x k x
12
13 dimensions [ 0 2 -1 0 0 0 ];
14
15 internalField uniform 0;
16
17 boundaryField
18 {
19   inlet
20   {
21     type fixedValue;
22     value uniform 0;
23   }
24   Dexcs_Exportedfromblender
25   {
26     type nutkWallFunction;
27     value uniform 0;
28   }
29   wall
30   {
31     type slip;
32   }
33   outlet
34   {
35     type zeroGradient;
36   }
37 }
```

細部手修正 (kappat)

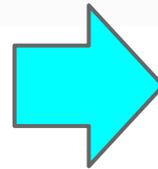
```
kappat x k x epsilon x alphas
12
13 dimensions [ 0 2 -1 0 0 0 ];
14
15 internalField uniform 0;
16
17 boundaryField
18 {
19   inlet
20   {
21     type kappatJayatillekeWallFunction;
22     Prt 0.85;
23     value uniform 0;
24   }
25   Dexcs_Exportedfromblender
26   {
27     type kappatJayatillekeWallFunction;
28     Prt 0.85;
29     value uniform 0;
30   }
31   wall
32   {
33     type zeroGradient;
34   }
35   outlet
36   {
37     type zeroGradient;
38   }
39 }
```



```
kappat x k x epsilon x alphas
12
13 dimensions [ 0 2 -1 0 0 0 ];
14
15 internalField uniform 0;
16
17 boundaryField
18 {
19   inlet
20   {
21     type fixedValue;
22     Prt 0.85;
23     value uniform 0;
24   }
25   Dexcs_Exportedfromblender
26   {
27     type kappatJayatillekeWallFunction;
28     Prt 0.85;
29     value uniform 0;
30   }
31   wall
32   {
33     type slip;
34   }
35   outlet
36   {
37     type zeroGradient;
38   }
39 }
```

細部手修正 (alphat)

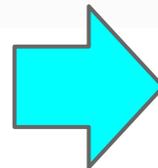
```
12
13 dimensions [ 1 -1 -1 0 0 0 ];
14
15 internalField uniform 0;
16
17 boundaryField
18 {
19     inlet
20     {
21         type alphasWallFunction;
22         value uniform 0;
23     }
24     Dexcs_Exportedfromblender
25     {
26         type alphasWallFunction;
27         value uniform 0;
28     }
29     wall
30     {
31         type zeroGradient;
32     }
33     outlet
34     {
35         type zeroGradient;
36     }
37 }
```



```
12
13 dimensions [ 1 -1 -1 0 0 0 ];
14
15 internalField uniform 0;
16
17 boundaryField
18 {
19     inlet
20     {
21         type fixedValue;
22         value uniform 0;
23     }
24     Dexcs_Exportedfromblender
25     {
26         type alphasWallFunction;
27         value uniform 0;
28     }
29     wall
30     {
31         type slip;
32     }
33     outlet
34     {
35         type zeroGradient;
36     }
37 }
```

細部手修正 (k)

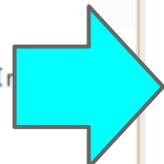
```
k x  epsilon x  *p_rgh x  *nut x
dimensions [ 0 2 -2 0 0 0 ];
internalField uniform 0.35;
boundaryField
{
    wall
    {
        type slip;
    }
    inlet
    {
        type turbulentIntensityKineticEnergyInlet;
        intensity 0.05;
        value uniform 0.35;
    }
    Dexcs_Exportedfromblender
    {
        type kqRWallFunction;
        value uniform 0.35;
    }
    outlet
    {
        type zeroGradient;
    }
}
```



```
k x  epsilon x  boundary x
{
    dimensions [ 0 2 -2 0 0 0 ];
    internalField uniform 0.00015;
    boundaryField
    {
        wall
        {
            type slip;
        }
        inlet
        {
            type turbulentIntensityKineticEnergyInlet;
            intensity 0.05;
            value uniform 0.00015;
        }
        Dexcs_Exportedfromblender
        {
            type kqRWallFunction;
            value uniform 0.00015;
        }
        outlet
        {
            type zeroGradient;
        }
    }
}
```

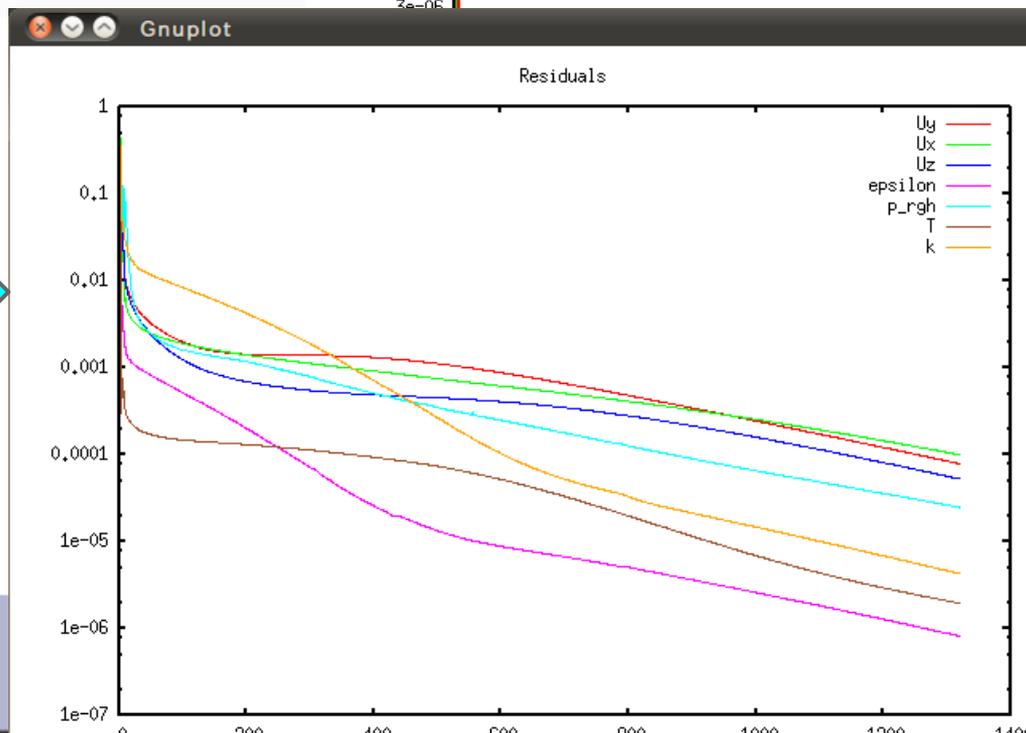
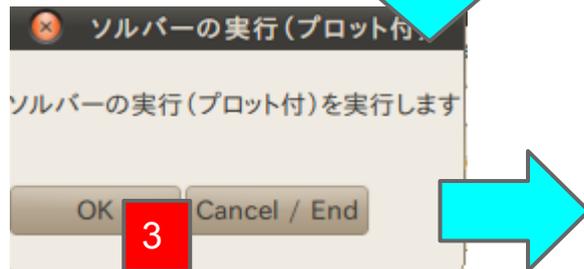
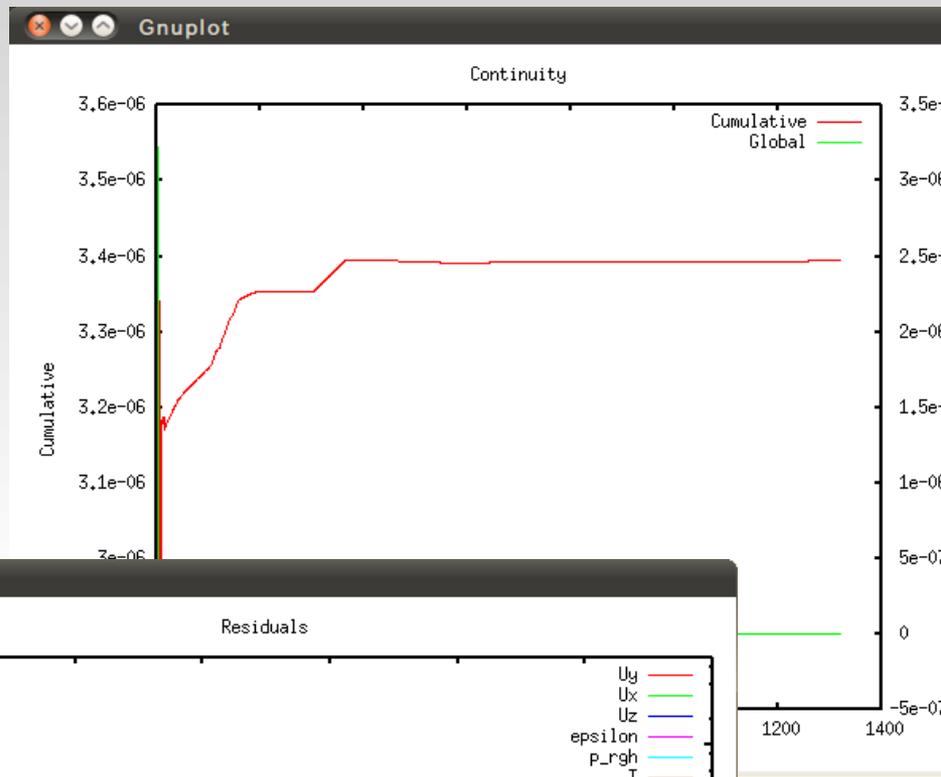
細部手修正 (epsilon)

```
epsilon x *p_rgh x *nut x *kappat x
dimensions [ 0 2 -3 0 0 0 0 ];
internalField uniform 14.855;
boundaryField
{
  wall
  {
    type slip;
  }
  inlet
  {
    value uniform 14.855;
    type turbulentMixingLengthDissipationRateInlet;
    mixingLength 0.01;
  }
  Dexcs_Exportedfromblender
  {
    type epsilonWallFunction;
    value uniform 14.855;
  }
  outlet
  {
    type zeroGradient;
  }
}
```



```
epsilon x boundary x
dimensions [ 0 2 -3 0 0 0 0 ];
internalField uniform 0.00003;
boundaryField
{
  wall
  {
    type slip;
  }
  inlet
  {
    value uniform 0.00003;
    type turbulentMixingLengthDissipationRateInlet;
    mixingLength 0.01;
  }
  Dexcs_Exportedfromblender
  {
    type epsilonWallFunction;
    value uniform 0.00003;
  }
  outlet
  {
    type zeroGradient;
  }
}
```

ソルバーの実行



流れ場の可視化

DEXCS SWAK
設定 ツール 終了

DEXCS 十徳ナイフ FileManager Terminal

/home/custom/Desktop/case3 case

Ncore 1 buoyantBoussinesqSimpleFoam solver/application

paraView可視化 (paraFoam) 実行

1 2

paraView可視化 (paraFoam) を実行します

OK Cancel / End

3

Pipeline Browser

- builtin:
- case3.OpenFOAM
- StreamTracer1

Object Inspector

Properties Display Information

Apply Reset Delete

- epsilon
- k
- kappat
- nut
- p
- p_rgh
- T
- U

U Magnitude

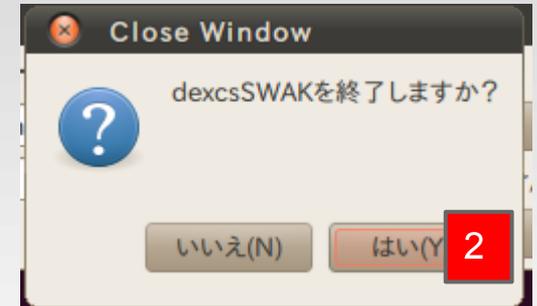
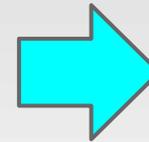
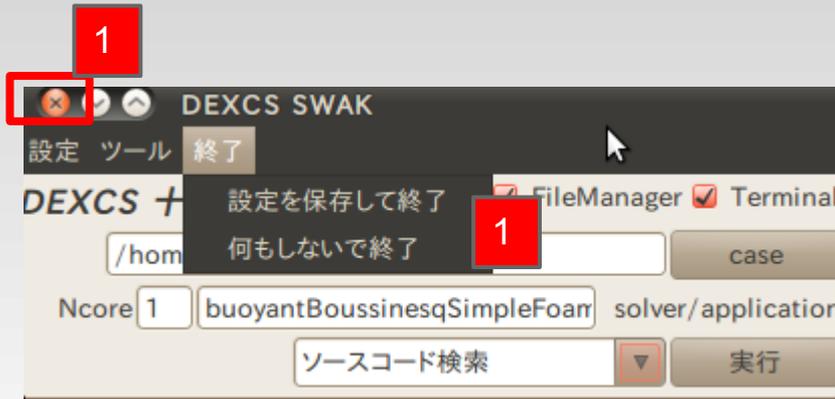
0.0416836 0.1 0.2 0.281320

265.00009 334.85049

280 300 320

50

終了



以上、本当にお疲れ様でした



今後の活用に向けて

オープンCAEコンサルタント OCSE²
代表 野村 悦治



isoイメージファイル/DVDの利用方法

BIOS設定(起動順をDVD優先)
パソコンをDVDから起動

DVD



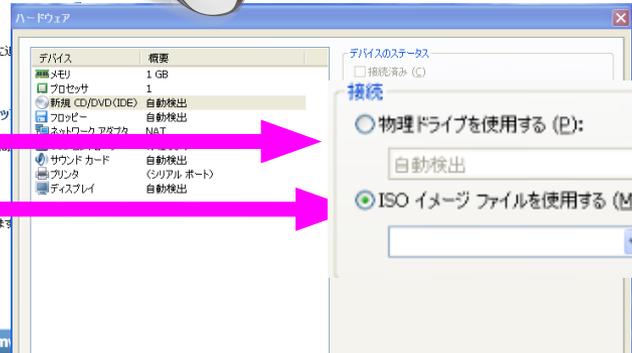
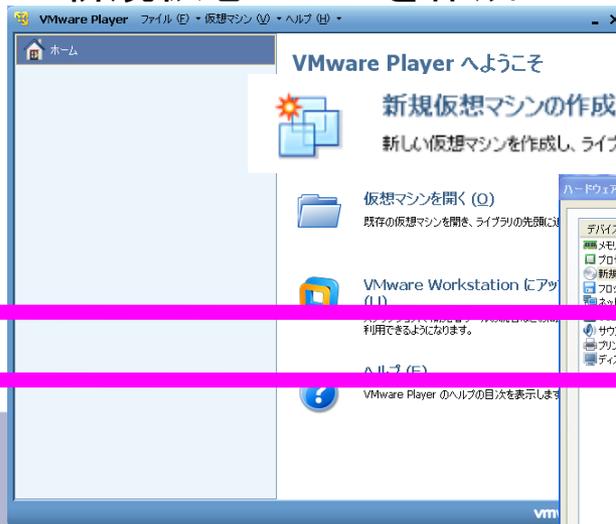
イメージ書き込み
/抽出



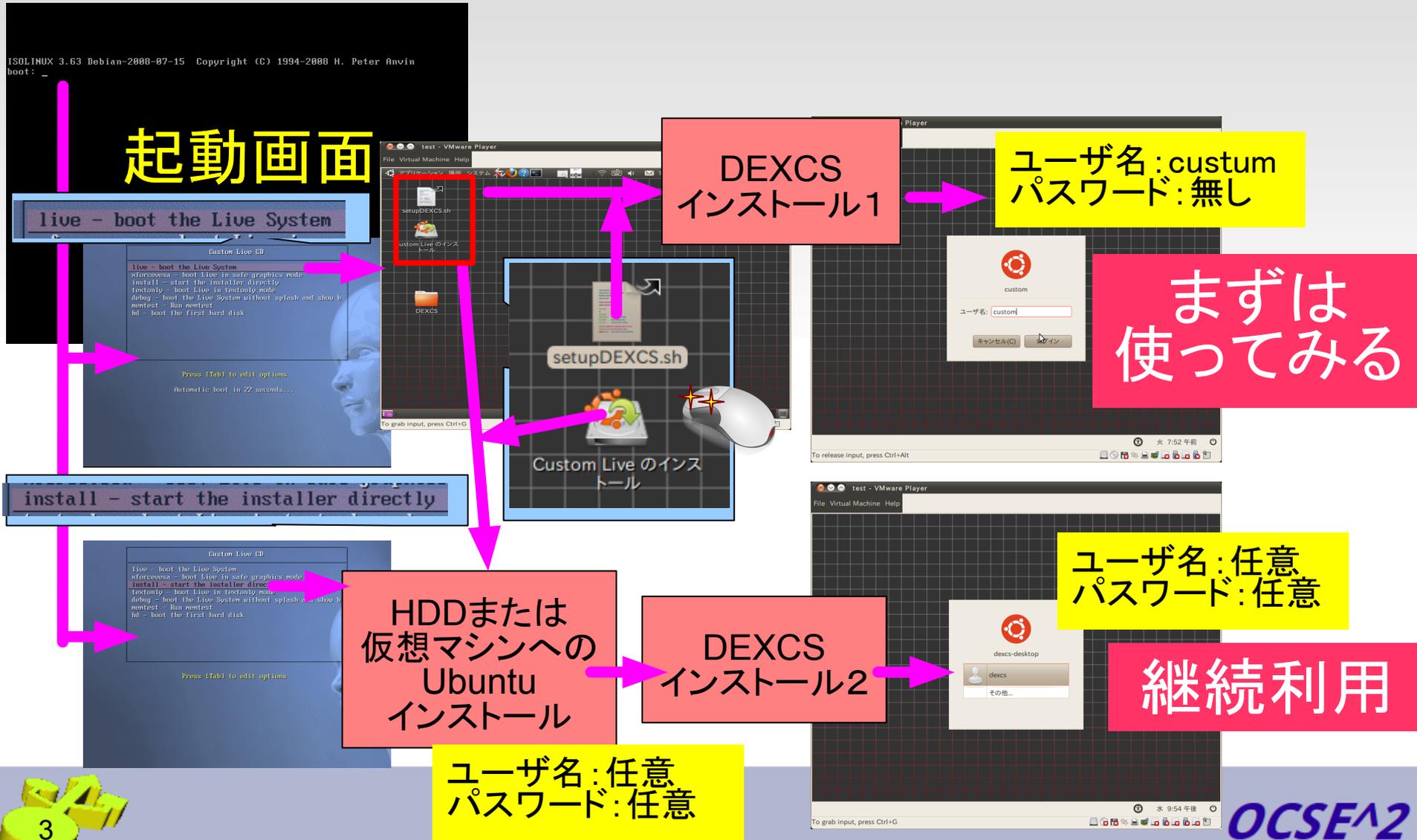
isoイメージ
ファイル

仮想マシンプレーヤー
(vmware, virtualBox,...)

新規仮想マシンを作成

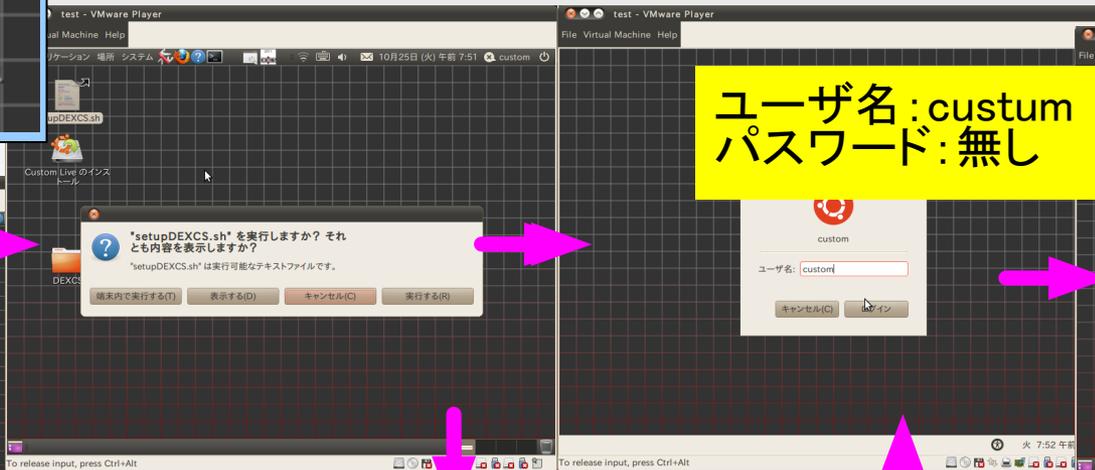
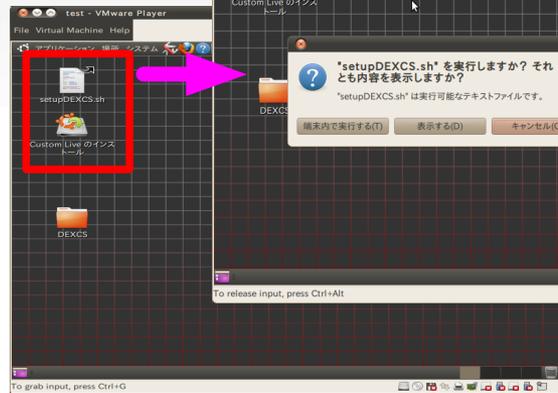
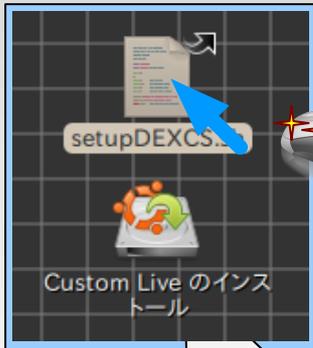


インストールの手順と利用形態

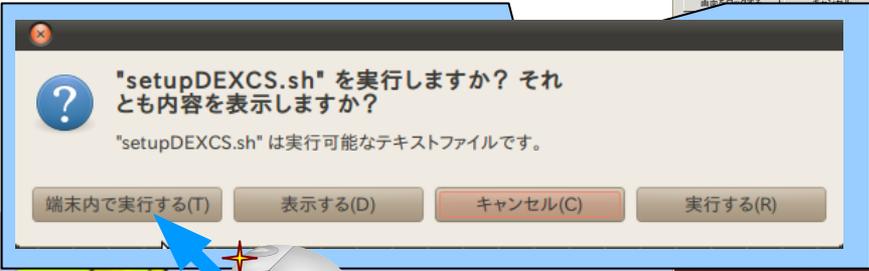
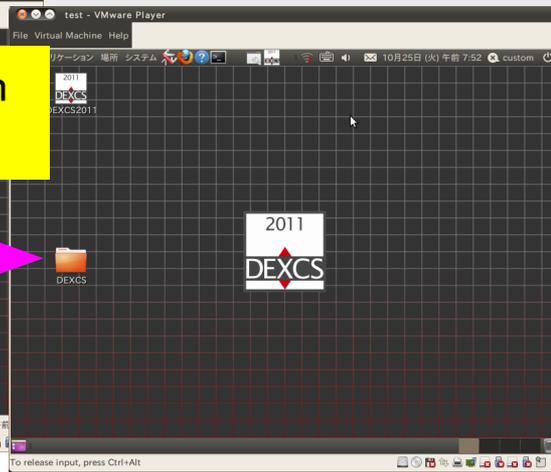


DEXCSインストール1の手順

ライブDVDで使用する場合

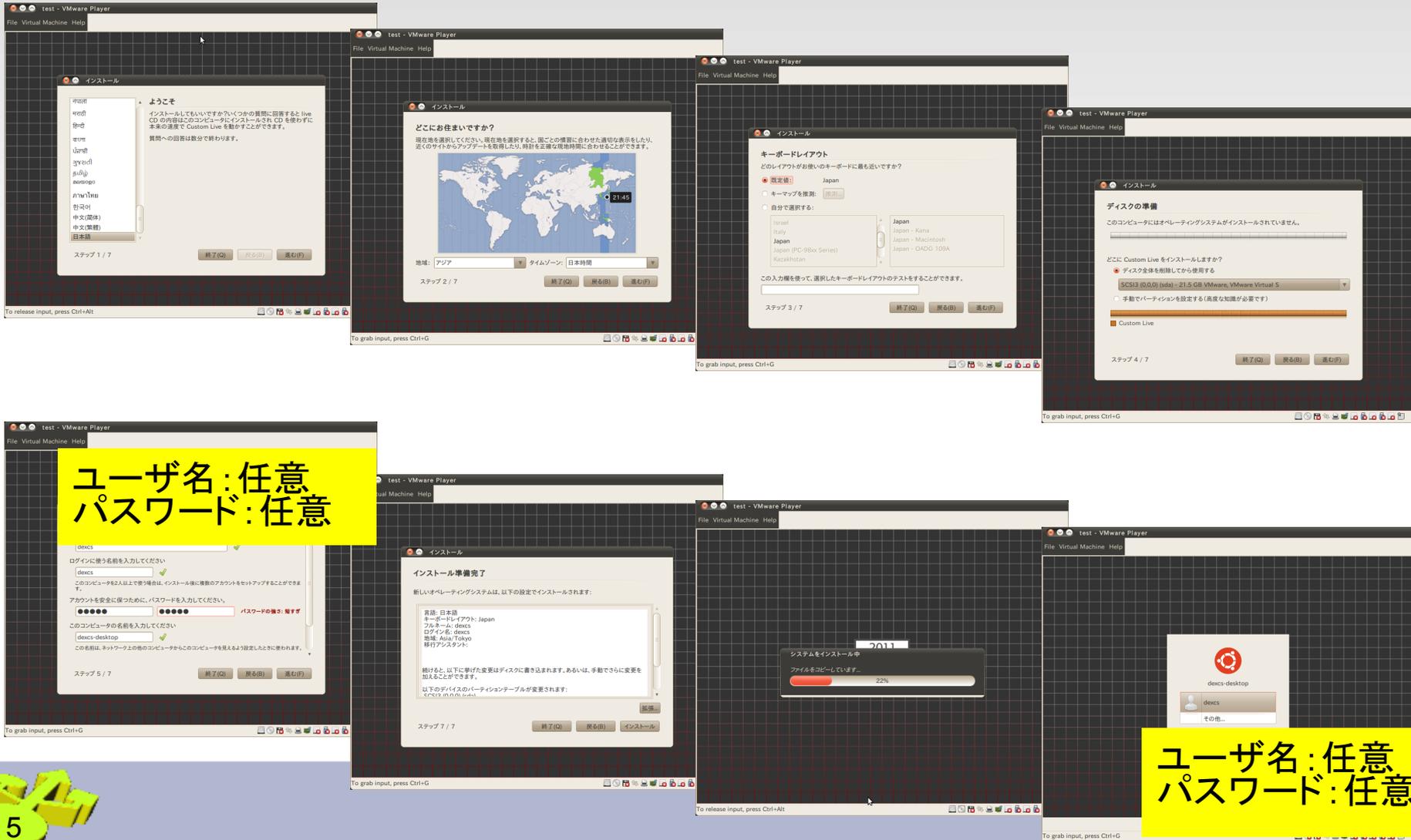


ユーザ名: custom
パスワード: 無し



このメニューは出てこない場合もあります

ubuntuインストールの手順



DEXCSインストール2の手順

HDDまたは仮想マシンへubuntuをインストールした後で実施

setupDEXCS.sh

test - VMware Player

ユーザー名:任意
パスワード:任意

2011
DEXCS

いくつかのプログラムが未だ実行中です:

- 不明 応答なし
- gnome-keyring-daemon 応答なし

プログラムが終了するのを待っています。実行中のプログラムを途中で強制終了させてしまうと、これまでの処理結果を失ってしまう可能性があります。

画面をロックする キャンセル **とにかくログアウトする**

このメニューは出てこない場合もあります

6

推獎利用方法



推奨利用方法1

OpenFOAM初学者向け

Linux(ubuntu)のコマンド入力が不得手な人

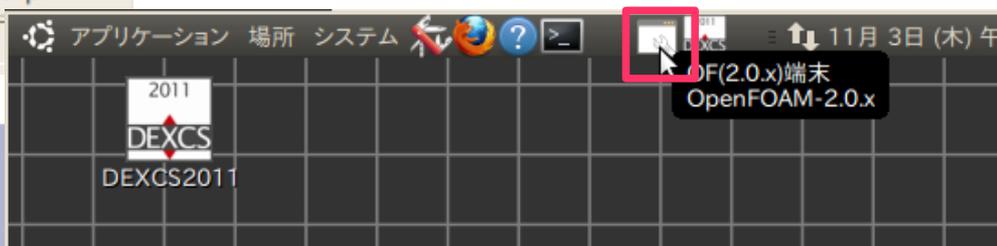


- 1 ランチャーの動作確認と全体概要の理解
 - 3 OpenFOAMの動作とファイル構成の実体理解
- 公開版情報

2 フラッシュプレーヤーの変更
(表示品質が改善されます)

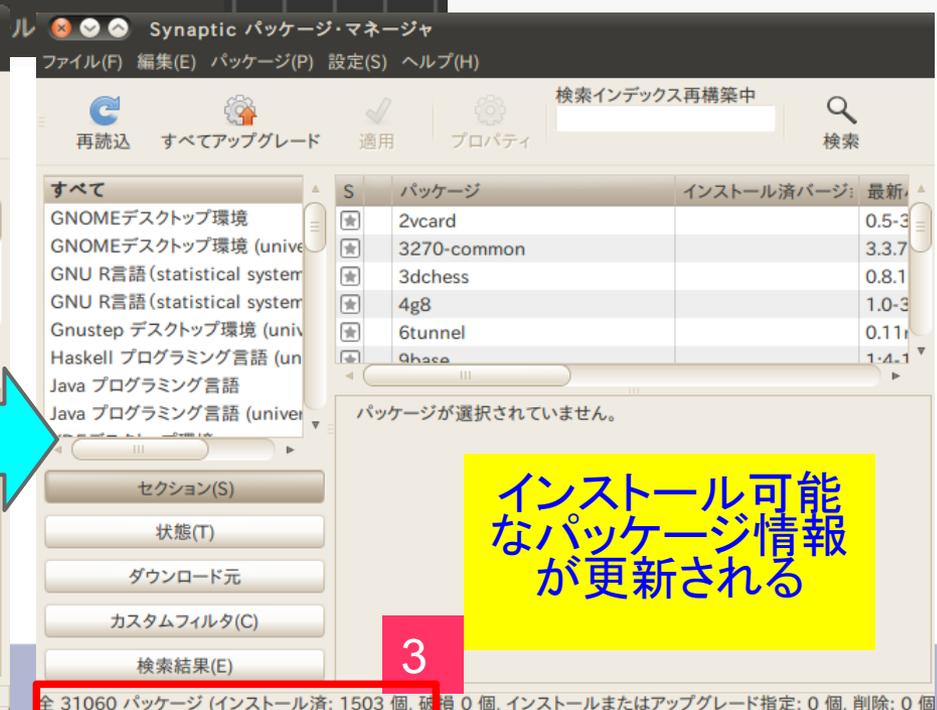
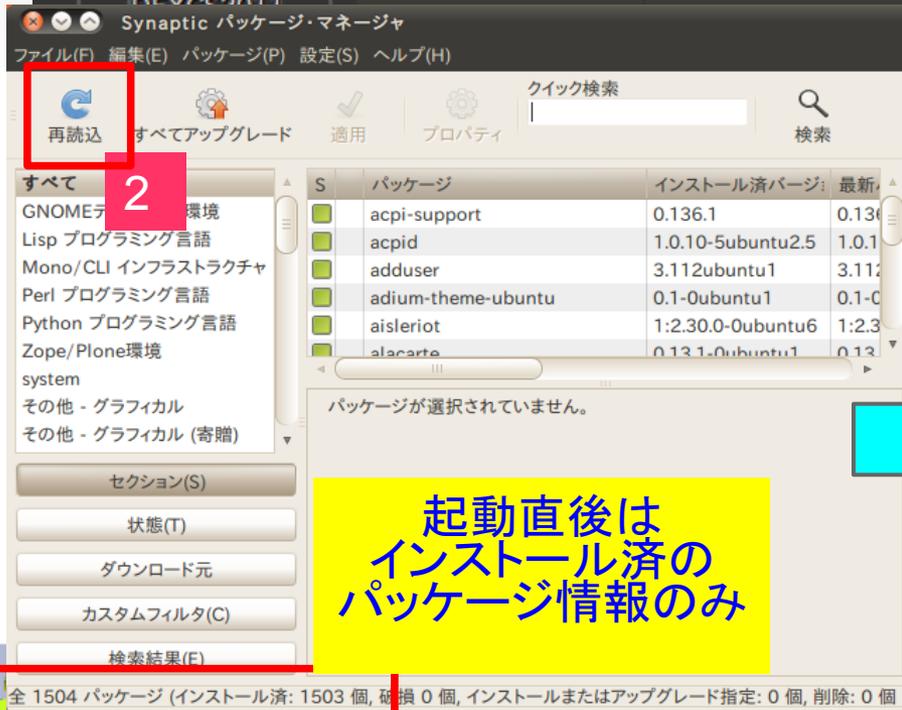
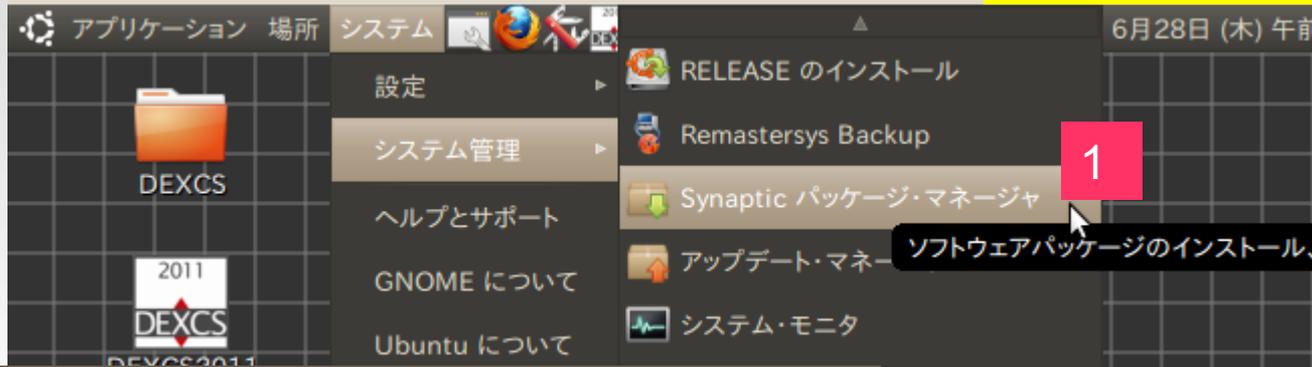
4 OpenFOAMマニュアル

コマンド入力が不得手な人



フラッシュプレーヤの変更方法(1)

要インターネット接続環境



フラッシュプレーヤの変更方法(2)

gnashを削除

Synaptic パッケージ・マネージャ

ファイル(F) 編集(E) パッケージ(P) 設定(S) ヘルプ(H)

再読み込み すべてアップグレード 適用 プロパティ

クイック検索

検索

すべて

- GNOMEデスクトップ環境
- GNOMEデスクトップ環境 (univer
- GNU R言語 (statistical sy
- GNU R言語 (statistical sy
- Gnustep デスクトップ環境 (univ
- Haskell プログラミング言語 (un
- Java プログラミング言語
- Java プログラミング言語 (univer

1

S	パッケージ	インストール済バージョン	最新バージョン
★	gmysqlcc		
★	gnarwl		
■	gnash	0.8.7	
■	gnash		
★	gnash		

free SW

スクリーン

Gnash is a standalon

plugin for alpha stat

The plugin

Gnash supports the majority of Flash opco version 7, and

2

マークを外す
インストール指定
再インストール指定
アップグレード指定
削除指定
完全削除指定
プロパティ
推奨パッケージもインストール指定
提案パッケージをインストール指定

依存により要求された変更を追加しますか?

選択された変更はほかのパッケージにも影響があります。処理の実行には以下の変更も必要になります。

削除されるパッケージ

- mozilla-plugin-gnash

キャンセル(C) マーク(M)

3

フラッシュプレーヤの変更方法(3)

flashplugin-installer をインストール

Synaptic パッケージ・マネージャ

ファイル(F) 編集(E) パッケージ(P) 設定(S) ヘルプ(H)

再読み込み すべてアップグレード 適用 プロパティ

クイック検索

すべて

- GNOMEデスクトップ環境
- GNOMEデスクトップ環境 (univer
- GNU R言語 (statistical system
- GNU R言語 (statistical system
- Gnustep デスクトップ環境 (un
- Haskell プログラミング言語 (un
- Java プログラミング言語
- Java プログラミング言語 (univer

1

パッケージ

- flamerobin
- flamethrower
- flashgot
- flashplugin-installer
- マークを外す
- 2 インストール指定
- 再インストール指定
- アップグレード指定
- 削除指定
- 完全削除指定
- プロパティ
- 推奨パッケージもインストール指定
- 提案パッケージをインストール指定

インストール指定

2

以下の変更を適用しますか?

指定された変更リストを適用前に参照することができる最後のチャンスです。

- + 削除されるパッケージ
- + インストールされるパッケージ
- + 未変更のままにする

サマリ

63 個のパッケージが保留されアップグレードされません
1 個の新規パッケージがインストールされます
2 個のパッケージが削除されます
459 kB のディスク容量が開放されます
20.8 kB のファイルをダウンロードする必要があります

パッケージのダウンロードのみ行う(D)

3

キャンセル(C) 適用(A)

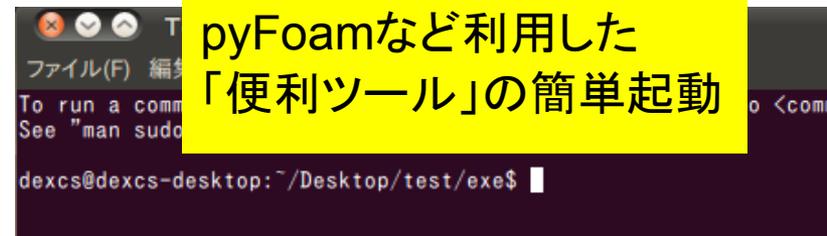
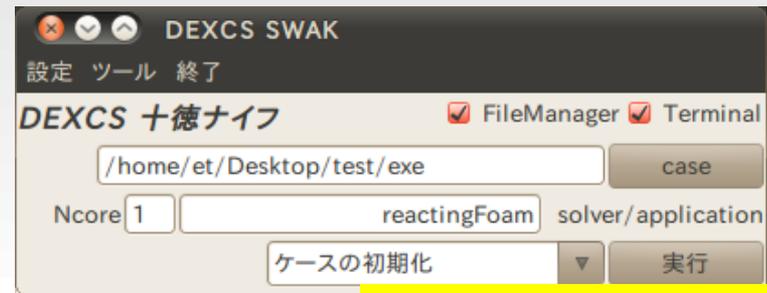
全 31060 パッケージ (インストール済: 1503 個, 破損 0 個, インストールまたはアップグレード指定: 0 個, 削除指定:

推奨利用方法2

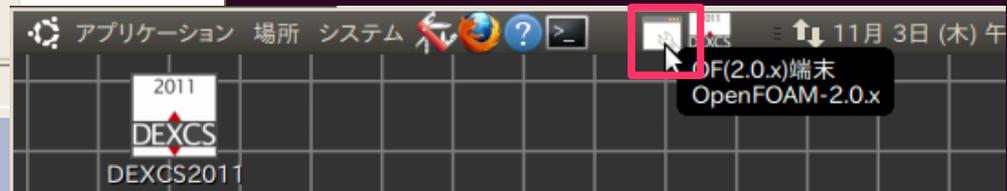
OpenFOAM利用経験者

OpenFOAMの動作と
ファイル構成を理解している

Linux(ubuntu)のコマンド入力が不得手な人



コマンド入力に不便を感じない人
pyFoam などのコマンドも熟知している人



推奨利用方法3-1

DEXCSをカスタマイズしたい人向け

ランチャーGUI(ボタン配置など)の変更

Desktop/DEXCS/dexcxs.wgx

blockMeshDict編集

Label blockMeshDict編集

Handler EditBlockMeshDict

推奨利用方法3-2

DEXCSをカスタマイズしたい人向け

ランチャースクリプトの変更

Desktop/DEXCS/launcherOpen/dexcs.py

EditBlockMeshDict(self, event)

```
def EditBlockMeshDict(self, event): # wxGlade: MyFrame

if not os.path.exists(self.mesh1Dir):
    self.ErrorDialog(u"解析フォルダを設定してください")
    return False
else:
    os.chdir(self.mesh1Dir)
    title = "EditBlockMeshDict"
    msg = "blockMeshDict を編集します"
    cmd = "gedit ./constant/polyMesh/blockMeshDict"
    self.runCommand(msg, title, cmd)
    self.setStatusBar()

def RunDisplayBlockMesh(self, event): # wxGlade:
if not os.path.exists(self.mesh1Dir):
    self.ErrorDialog(u"解析フォルダを設定してください")
    return False
else:
    os.chdir(self.mesh1Dir)
    title = "RunDisplayBlockMesh"
    cmd = "python ./constant/polyMesh/blockMeshDict"
    self.runCommand(msg, title, cmd)
    self.setStatusBar()
```

注意事項

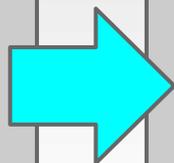
- ソースコード更新 (git pull ⇒ Allwmake) した場合
 - ~/OpenFOAM/OpenFOAM-2.0.x/etc/controlDict
AllowSystemOperation 1
 - swak4Foam ⇒ Allwmake
- 十徳ナイフのバグフィックス
 - <http://mogura7.zenno.info/~et/wordpress/ocse/?p=481> の記事を参考にプログラムをダウンロードし、入れ替えて使用の事。

公開版との相違点と64bit版について

公開版

特別版

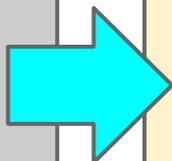
32bit版



1. blender変更 + Swiftツール
2. ランチャー、十徳ナイフ改良
3. (Swiftサンプル)

体験コース

64bit版



1. <http://mogura7.zenno.info/~et/wordpress/ocse/?p=337>の記事を参考にインストール可能
2. デスクトップ上のDEXCSフォルダを入れ替えば流用可能(但し、ユーザー名は同一にしてあること)
3. デスクトップ上のフォルダ(SwiftExample)をそのまま流用可能

継続利用にお薦め
近日製作予定(有料?)

オンラインでの情報元

■ Web

- SGI社、OpenFOAM Foundation(英語)
- オープンCAE学会(日本語)

■ Discussion Board

- OpenFOAM Forum (英語)
- OpenFOAMディスカッションボード (日本語)

■ Wiki

- OpenFOAM Wiki (英語)
- OFWikiJa [OpenFOAM Wiki日本語版] (日本語)

SGI, OpenFOAM Foundation

<http://www.openfoam.com/>



The open source CFD toolbox



[Home](#) | [Features](#) | [Support](#) | [Training](#) | [Resources](#) | [News](#) | [OpenFOAM Foundation](#)

[About us](#) | [Contact](#) | [Jobs](#) | [Legal](#)

Get OpenFOAM

- [OpenFOAM v2.1.0](#)
- [Download OpenFOAM](#)

Our Services

- [OpenFOAM Training](#)
- [Software Support](#)
- [Development Support](#)
- [Run on SGI® Cyclone](#)

Further Help

- [Official Documentation](#)
- [Report a Bug](#)
- [Run on Amazon EC2®](#)
- [Run on Windows](#)

About OpenFOAM

OpenFOAM is a free, open source CFD software package developed by [the OpenFOAM Team](#) at [SGI Corp](#) and distributed by the [OpenFOAM Foundation](#). It has a large user base across most areas of engineering and science, from both commercial and academic organisations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to solid dynamics and electromagnetics. [More...](#)

Our commitment to the users

OpenFOAM comes with full commercial support from SGI, including [software support](#), [contracted developments](#) and a programme of [training courses](#). These activities fund the development, maintenance and release of OpenFOAM to make it an extremely viable commercial open source product.

Our commitment to open source

SGI and the OpenFOAM Team are committed to open source software, establishing the [OpenFOAM Foundation](#) to manage and promote OpenFOAM. Through its bylaws, it formalises the commitment, begun by OpenCFD in 2004, to ensure OpenFOAM will **always be free open source only**. SGI contributes the code developments and maintenance work of the OpenFOAM Team to the Foundation.

FOLLOW US ON [twitter](#)

News on Twitter

"#OpenFOAM Training: last places available for #San #Francisco 25-28 June, 2012. Venue: <http://t.co/ijZ4Cq8B> Booking: <http://t.co/xHCPcTZC>"
May 2nd 2012

"#OpenFOAM #Foundation and #Advanced courses scheduled 01-04 Oct 2012 in #Chicago <http://t.co/X1uiQXkN>. Register at <http://t.co/xHCPcTZC>"
Apr 25th 2012

[More news on Twitter...](#)

Main News

OpenFOAM bug reporting upgraded
The OpenFOAM bug reporting has been upgraded to Mantis 1.2.8 and relocated to <http://www.openfoam.org/bugs>
Feb 25th 2012

OpenFOAM® version 2.1.0 released
The OpenFOAM Foundation is pleased to announce the release of version 2.1.0 of the OpenFOAM open source CFD toolbox.
Dec 19th 2011

SGI acquires OpenCFD Ltd.
SGI, a trusted leader in technical



18



2

オープンCAE学会(日本語)

<http://www.opencae.jp/>



Search bar and navigation menu: ログイン | 個人設定 | ヘルプガイド | Trac について | 登録 | Wiki | Blog | タイムライン | リポジトリブラウザ | 検索 | フォーラム | スタートページ | ページ一覧 | このページの履歴 | 最終更新 10時間前

一般社団法人オープンCAE学会 —The Open CAE Society of Japan—

- [オープンCAEとは](#)
- [オープンCAE学会の概要](#)
- [入会のご案内と会費](#)

ニュース

- (2012/05/06) [ソフトウェアマニュアル翻訳](#)にて公開している OpenFOAM 2.1.0 プログラムズガイド和訳に、索引を追加しました。
- (2012/04/10) 2012年6月8日～9日に国立オリンピック記念青少年総合センターでオープンCAEワークショップ2012を行います。詳細は未定ですが、9日午前中に行う[講習会のアンケート](#)を行なっておりますので、是非ご協力ください。
- (2012/02/18) [ソフトウェアマニュアル翻訳](#)に OpenFOAM 2.1.0 ユーザガイド和訳(β版)およびプログラムズガイド和訳(β版)を追加しました。
- (2012/02/05) [ソフトウェアマニュアル翻訳](#)に OpenFOAM 2.0.0 ユーザガイド和訳(β版)を追加しました。
- (2011/11/03) 2011年12月1日～2日に国立オリンピック記念青少年総合センターで行う[オープンCAEシンポジウム2011](#)の[申し込み](#)を開始しました。(終了)
- (2011/10/01) 2011年12月1日～2日に国立オリンピック記念青少年総合センターで[オープンCAEシンポジウム2011](#)を行います。(終了)
- (2011/08/28) [ソフトウェアマニュアル翻訳](#)に OpenFOAM 2.0.0 プログラムズガイド和訳(β版)を追加しました。
- (2011/08/12) [平成22年度OpenFOAM非圧縮性流体解析演習シリーズ](#)に第8回(中級講習)「buoyantBoussinesqSimpleFoamソルバーや乱流モデルのカスタマイズ」の配布資料を掲載しました。

[ニュース・アーカイブ](#)

オープンCAE学会について

- [オープンCAE学会の概要](#)
- [オープンCAE学会定款](#)
- [入会のご案内と会費](#)
- [ディスクロージャ](#)
- [プライバシー・ポリシー](#)
- [ロゴ](#)

事業のご案内

- [オープンCAEシンポジウム](#)
- [オープンCAE講習会](#)
- [分科会](#)
- [オープンCAE勉強会](#)
- [ディスカッションボードの運営](#)
- [ソフトウェアマニュアル翻訳](#)

サポート、トレーニング、受託
OpenFOAM

■爆発・災害コンサルティング
爆発研究所

株式会社 **森村設計**
P.TAKAHASHI & ASSOCIATES, LTD.

sgi.

OpenCAE ロゴTシャツ
KeityS office

OpenCAE **opencae**

OpenCAE 今週土曜にOpenFOAM非圧縮性流体解析演習シリーズの第8回(中級講習、最終回)「buoyantBoussinesqSimpleFoam

OpenFOAM Forum (英語)

<http://www.cfd-online.com/Forums/openfoam/>



[Sponsors]

- Home
- News
- Forums**
- Wiki
- Links
- Jobs
- Books
- Events
- Tools
- Feeds
- About
- Search

Home > Forums > OpenFOAM

OpenFOAM

Welcome, E.Mogura.

You last visited: May 1, 2012 at 05:15

Private Messages: Unread 0, Total 1.

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

Sub-Forums : OpenFOAM

SEARCH THIS FORUM ▾

Forum	Last Post	Threads	Posts
 OpenFOAM News & Announcements	 enGrid training?? by atg Today 04:20 >	106	485
 OpenFOAM Installation (1 Viewing)	 gcc by wyldckat Yesterday 23:35 >	1,157	9,352
 OpenFOAM Meshing & Mesh Conversion (2 Viewing)	 SwiftSnap and SwiftBlock,... by kalle Yesterday 21:42 >	1,392	7,781
 OpenFOAM Pre-Processing (1 Viewing)	 Modelling wave impacts on... by tfuwa May 4, 2012 23:47 >	583	2,876
 OpenFOAM Running, Solving & CFD (1 Viewing)	 TwoPhaseEulerFoam and... by alberto	5,300	30,749

OpenFOAM Forum (英語)

Software User Forums 				
	ANSYS (20 Viewing) Topics related to the software packages sold by ANSYS Inc.	 Can CFX run 2D problems by happy Today 11:19 	48,896	156,516
	CD-adapco (2 Viewing) Topics related to the software packages sold by CD-adapco.	 Oscillating coefficient values by calim_cfd Today 00:56 	6,122	20,885
	CFdesign Topics related to the software packages sold by Autodesk Inc. (Blue Ridge Numerics).	 How to best handle quilts... by Mithrandir April 5, 2012 03:44 	11	44
	Edge Topics related to the Edge software from FOI.	 Welcome to Edge Forum! by kid April 12, 2012 19:37 	15	59
	FloEFD, FloWorks & FloTHERM (1 Viewing) Topics related to the software packages sold by Mentor Graphics Corporation.	 Slow calculations with... by Rpaplowl Yesterday 05:52 	220	912
	FLOW-3D (1 Viewing) Topics related to the software packages sold by Flow Science Inc.	 Flow3D Mesh Assessment by JBurnham April 26, 2012 07:36 	593	2,341
	NUMECA Topics related to the software packages sold by NUMECA International.	 air under hull by laurent98 May 3, 2012 07:38 	324	878
	OpenFOAM (10 Viewing) Topics related to the OpenFOAM software.	 Find maximum value in the... by nandiganavishal Today 08:11 	16,224	89,897
	Phoenics (2 Viewing) Topics related to the software packages sold by CHAM Ltd.	 run ear in background by monty87 April 30, 2012 21:22 	1,382	4,052

OpenFOAMディスカッションボード(日本語)

<https://groups.google.com/forum/?fromgroups&hl=ja#!forum/openfoam>

The screenshot shows the Google Groups interface for the OpenFOAM Japanese discussion board. The page title is "OpenFOAM" with a star icon, a "+1" icon, and a "0" icon. The search bar at the top contains the text "グループまたはメッセージを検索する". The main content area displays a list of topics, each with a star icon, a title, and a brief description. The topics listed are:

- ★ 5月の東京初心者向け勉強会のご案内 (1)
作成者: siba - 1 件の投稿 - 4 ビュー - 最終更新: 5月3日 (3 日前)
- ★ 第14回オープンCAE初心者勉強会@東海のご案内
作成者: ryos - 1 件の投稿 - 5 ビュー - 最終更新: 5月1日 (4 日前)
- ★ 並列計算技術講習会についてのアンケート (1)
作成者: Masataka - 1 件の投稿 - 3 ビュー - 最終更新: 5月1日 (5 日前)
- ★ 第16回オープンCAE初心者勉強会のご案内
作成者: ryos
- ★ 【勉強会 for...】
作成者: tomil
- ★ オープンCAE初心者勉強会 (1)
作成者: ryos - 1 件の投稿 - 2
- ★ MRFInterFoamでのワーニング
作成者: Sakuma - 9 件の投稿 - 43 ビュー - 最終更新: 4月23日 (12 日前)

On the left side, there is a navigation menu with the following items:

- グループ
- マイグループ
- ホーム
- スター付き
- お知らせ
- Google Groups Anno...
- 最近表示したフォ... 629
- 最近の検索
- git pull (openfoam 内)
- 最近投稿したフォ... 515
- OpenFOAM 515
- お気に入り

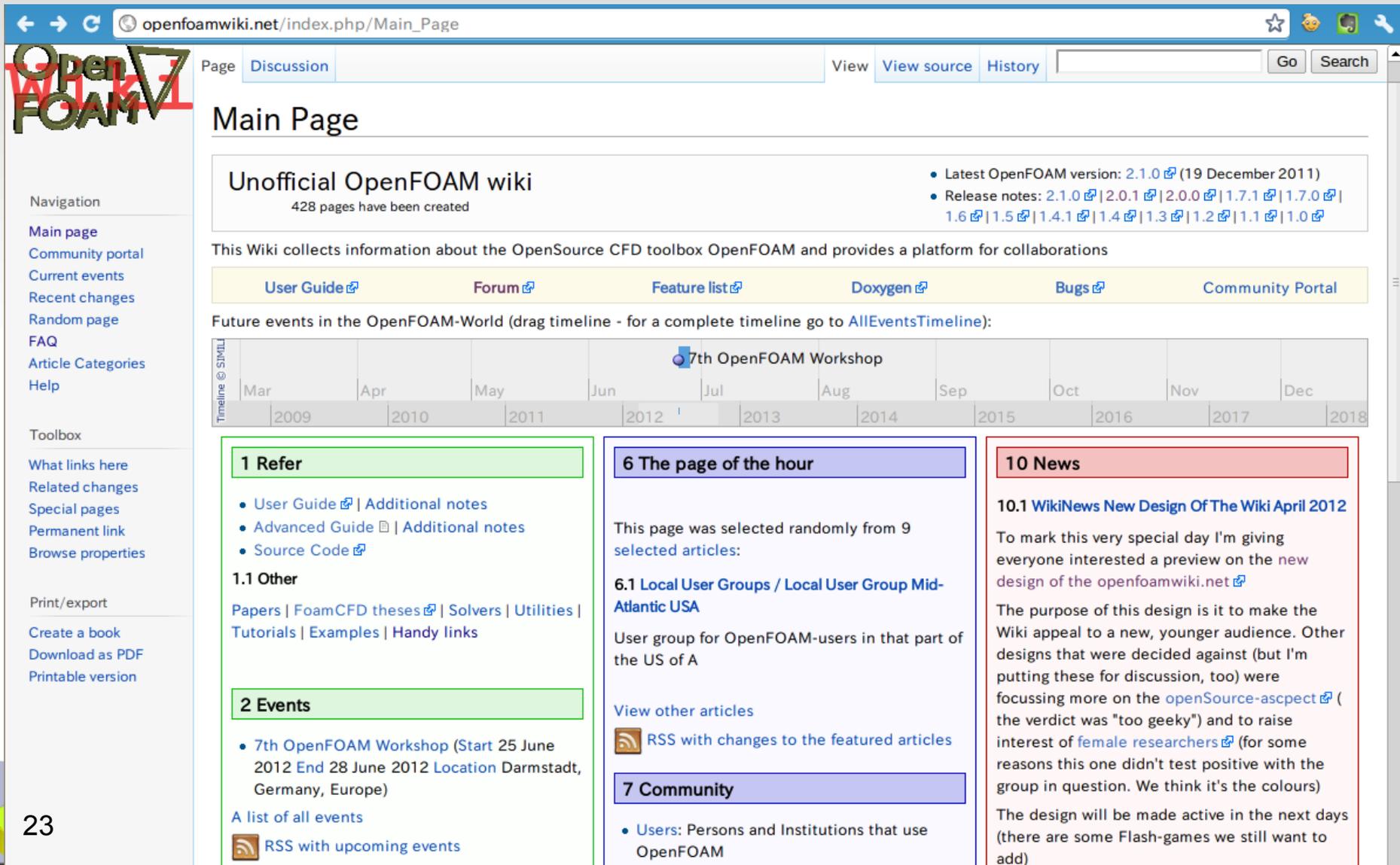
At the bottom left, there is a yellow box with the text: "グループのスターアイコンをクリックして、お気に入りに追加します".

わからない事があったらここで聞くのが一番
(匿名可)

各地(東京・岐阜・大阪・富山・広島)で
勉強会も開催中(1/月程度)

OpenFOAM Wiki (英語)

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



The screenshot shows the OpenFOAM Wiki Main Page in a browser. The page title is "Main Page" and the URL is "http://openfoamwiki.net/index.php/Main_Page". The page content includes a navigation sidebar on the left, a main content area with a "Unofficial OpenFOAM wiki" section, a "Future events in the OpenFOAM-World" timeline, and three featured article boxes: "1 Refer", "6 The page of the hour", and "10 News".

Navigation

- Main page
- Community portal
- Current events
- Recent changes
- Random page
- FAQ
- Article Categories
- Help

Toolbox

- What links here
- Related changes
- Special pages
- Permanent link
- Browse properties

Print/export

- Create a book
- Download as PDF
- Printable version

Main Page

Unofficial OpenFOAM wiki
428 pages have been created

- Latest OpenFOAM version: [2.1.0](#) (19 December 2011)
- Release notes: [2.1.0](#) | [2.0.1](#) | [2.0.0](#) | [1.7.1](#) | [1.7.0](#) | [1.6](#) | [1.5](#) | [1.4.1](#) | [1.4](#) | [1.3](#) | [1.2](#) | [1.1](#) | [1.0](#)

This Wiki collects information about the OpenSource CFD toolbox OpenFOAM and provides a platform for collaborations

[User Guide](#) | [Forum](#) | [Feature list](#) | [Doxygen](#) | [Bugs](#) | [Community Portal](#)

Future events in the OpenFOAM-World (drag timeline - for a complete timeline go to [AllEventsTimeline](#)):

Timeline © SIMILI

Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
2009	2010	2011	2012	2013	2014	2015	2016	2017	2018

7th OpenFOAM Workshop

1 Refer

- [User Guide](#) | [Additional notes](#)
- [Advanced Guide](#) | [Additional notes](#)
- [Source Code](#)

1.1 Other

[Papers](#) | [FoamCFD theses](#) | [Solvers](#) | [Utilities](#) | [Tutorials](#) | [Examples](#) | [Handy links](#)

2 Events

- [7th OpenFOAM Workshop \(Start 25 June 2012 End 28 June 2012 Location Darmstadt, Germany, Europe\)](#)

[A list of all events](#)

[RSS with upcoming events](#)

6 The page of the hour

This page was selected randomly from 9 selected articles:

6.1 Local User Groups / Local User Group Mid-Atlantic USA

User group for OpenFOAM-users in that part of the US of A

[View other articles](#)

[RSS with changes to the featured articles](#)

7 Community

- Users:** Persons and Institutions that use OpenFOAM

10 News

10.1 WikiNews New Design Of The Wiki April 2012

To mark this very special day I'm giving everyone interested a preview on the new design of the [openfoamwiki.net](#)

The purpose of this design is it to make the Wiki appeal to a new, younger audience. Other designs that were decided against (but I'm putting these for discussion, too) were focussing more on the [openSource-ascpect](#) (the verdict was "too geeky") and to raise interest of [female researchers](#) (for some reasons this one didn't test positive with the group in question. We think it's the colours)

The design will be made active in the next days (there are some Flash-games we still want to add)

OFWikiJa [OpenFOAM Wiki日本語版] (日本語)

<http://www.ofwikija.org/index.php>

[本文](#) [ノート](#) [ソースを表示](#) [履歴](#)

OpenFOAM

ナビゲーション

- [メインページ](#)
- [リリースノート](#)
- [ダウンロードとインストール](#)
- [ユーザーガイドの和訳](#)
- [FAQ](#)
- [チュートリアル](#)
- [ユーティリティ](#)
- [ライブラリ](#)
- [勉強会](#)
- [リンク](#)
- [最近更新したページ](#)
- [おまかせ表示](#)
- [ヘルプ](#)

検索

ツールボックス

- 24 [リンク元](#)
■ [リンク先の更新状況](#)

メインページ

OFWikiJa - OpenFOAM Wiki日本語版

OpenFOAMはオープンソースのCFDツール集です. 初めての方は[OpenFOAMとは](#)をご覧ください.

このサイトに関する話題は、Googleグループ[OpenFOAMユーザー会](#)において議論したいと思います。ご興味のある方は、是非OpenFOAMユーザー会にご入会下さい。

ニュース

- スпам投稿が多くなってきたので、[ConfirmEdit](#)のSimple Captchaを導入しました。Masa 2012年2月5日 (日) 05:33 (UTC)
- 東京大学HA8000クラスタシステムでのOpenFOAM-2.0.xの並列化効率のページに小規模ケースを追加しました。サンプルケースも更新しました。Masa 2011年9月26日 (月) 12:23 (UTC)
- 東京大学HA8000クラスタシステムへのOpenFOAM-2.0.xのインストールと実行のページにcavityのサンプルケースを追加しました。Masa 2011年9月21日 (水) 11:12 (UTC)
- 文献を追加しました。皆様もご自由に追加ください。Masa 2011年9月19日 (月) 08:45 (UTC)
- 東京大学HA8000クラスタシステムでのOpenFOAM-2.0.xの並列化効率のページを作成しました。Masa 2011年9月17日 (土) 10:44 (UTC)
- 東京大学HA8000クラスタシステムへのOpenFOAM-2.0.xのインストールと実行のページを作成しました。Masa 2011年9月10日 (土) 13:20 (UTC)
- OpenFOAM-2.0.xのダウンロードとインストールのページを作成しました。Masa
- リリースノート OpenFOAM®_v2.0.0のページを作成しました。Masa 2011年6月18日
- OpenFOAM勉強会 for beginnerのページを作成しました。Masa 2010年7月3日 (土) 01:47 (UTC)
- [ユーザーガイド 1.6系の和訳](#)のBeta版を掲載しました。Masa 2010年5月27日 (木) 10:06 (UTC)
- サーバーの不具合により、2008年12月8日以降の更新が再度全て無くなってしまいました。Googleのキャッシュに等により順次コンテンツを復元致します。大変申し訳ありませんが、暫くお待ちください。Masa 2010年5月27日 (木) 09:54 (UTC)