

# Swiftツールによる OpenFOAM®用メッシュ作成

オープンCAE学会 理事  
オープンCAEコンサルタント  
OCSE<sup>2</sup> 代表  
野村 悦治

OPENFOAM and OpenCFD are registered trademarks of SGI Corp.



# 講習手順

1. 実習環境の説明
2. 実習環境の起動
3. blenderとswiftツールの説明
4. 実習例題の概要説明
5. 実習
6. まとめ

OpenFOAM®の標準的なメッシュ作成ツールとしてblockMeshとsnappyHexMeshが存在しますが、3次元形状データを基にこれらに必要なデータ(\* \* Dict)を作成できるフリーで実用的なGUIツールが存在しませんでした。  
本講習では、この課題を解決すべく公開されたツール(SwiftBlock,SwiftSnap)を組み込んだDEXCS for OpenFOAM®(特別版)を使って、ツールの使用法を具体例にて説明し、その可能性を探ります。

狙い

# 実習環境の説明

## 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](#))を組み込みました。

- 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  $\beta$  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

# 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](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> )

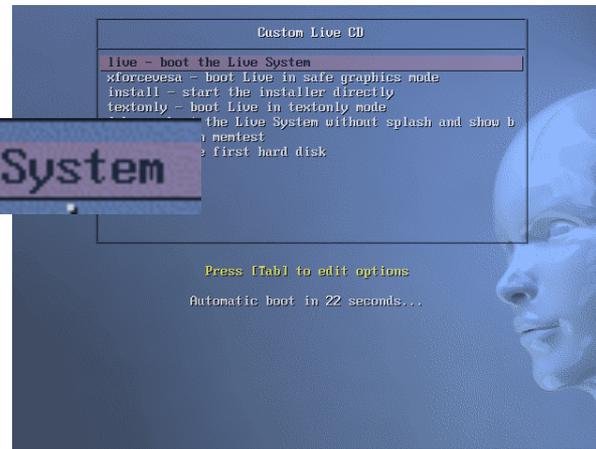
# 実習環境の起動



# 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

1 ダブルクリック

Custom Live のインストール

SwiftExample

2 クリック

"setupDEXCS.sh" を実行しますか? それとも内容を表示しますか?  
"setupDEXCS.sh" は実行可能なテキストファイルです。

端末内で実行する(T) 表示する(D) キャンセル(C) 実行する(R)

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

3

このダイアログは出ない場合もある





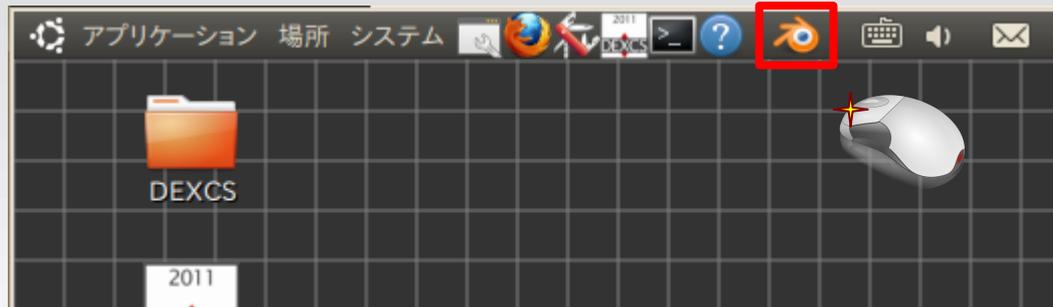


実習環境の構築完了

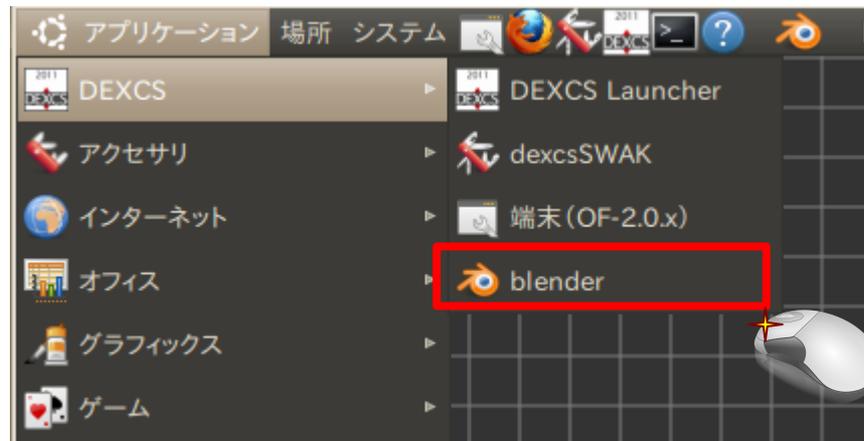
# blenderとswiftツールの説明

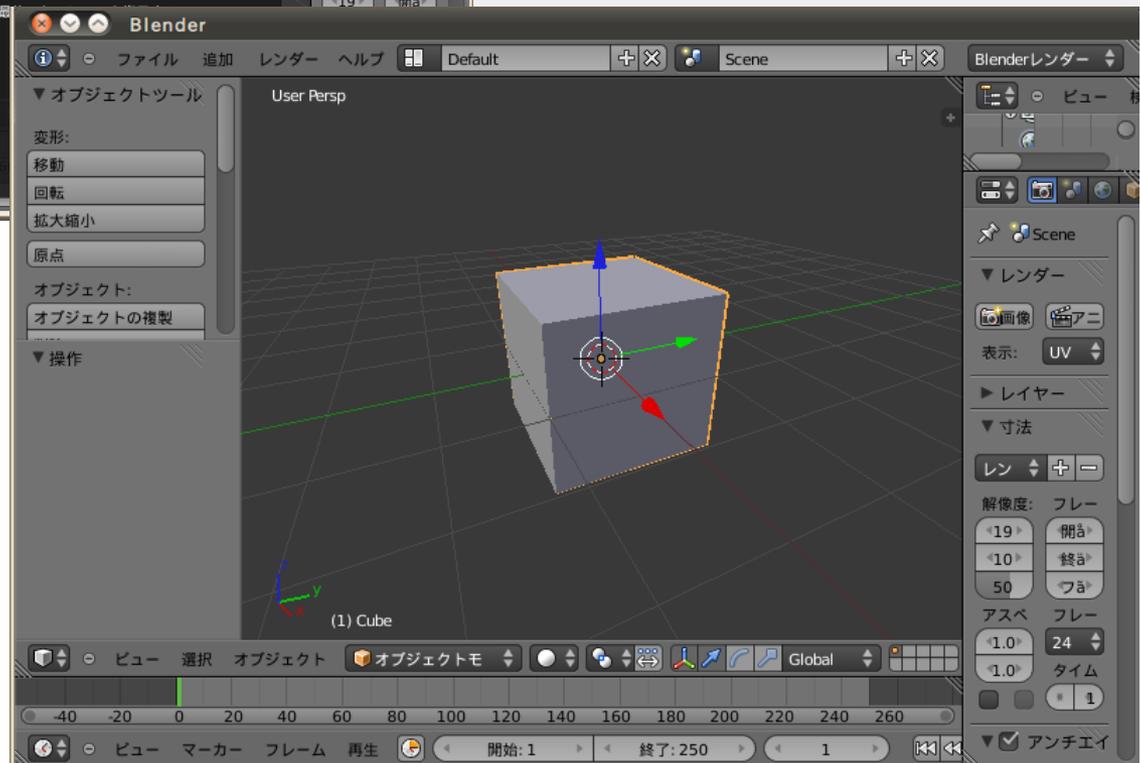


# blenderの起動方法

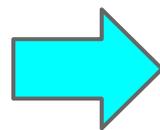


または





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](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

## 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な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。

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

以下は、Blenderを利用して制作した静止画の公開ページです。Copyright © 2013

目次

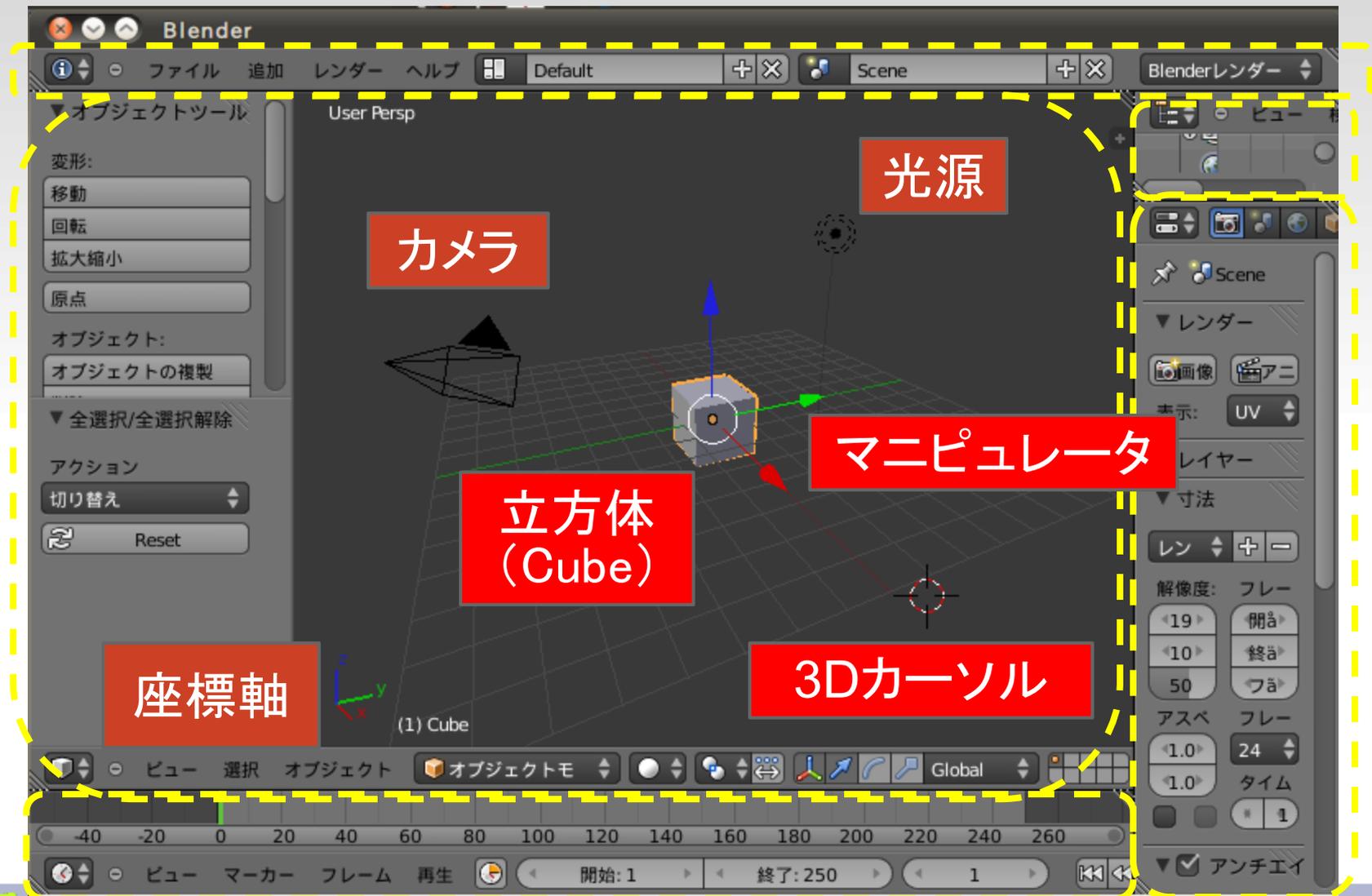


- 1.はじめに
  - 2.画面説明
  - 3.マウス操作
  - 4.ウィンドウ操作
  - 5.3D View上の視点操作
  - 6.モード
  - 7.図形の追加
  - 8.削除・Undo
  - 9.ファイルの保存
  - 10.マニピュレータ
  - 11.メッシュ編集1
  - 12.メッシュ編集2
  - 13.メッシュ編集3
  - 14.ライティング
  - 15.カメラ設定
  - 16.レンダリング
  - 17.マテリアル設定
  - 18.テクスチャ設定1
  - 19.テクスチャ設定2
  - 20.ワールド設定
  - 21.アニメーション1
  - 22.アニメーション2
  - 23.アニメーション3
  - 24.アニメーション4
  - 25.アニメーション5
  - 26.ノード機能
  - 27.最後に&アンケート
- Appendix 1. インタフェースの日本語化

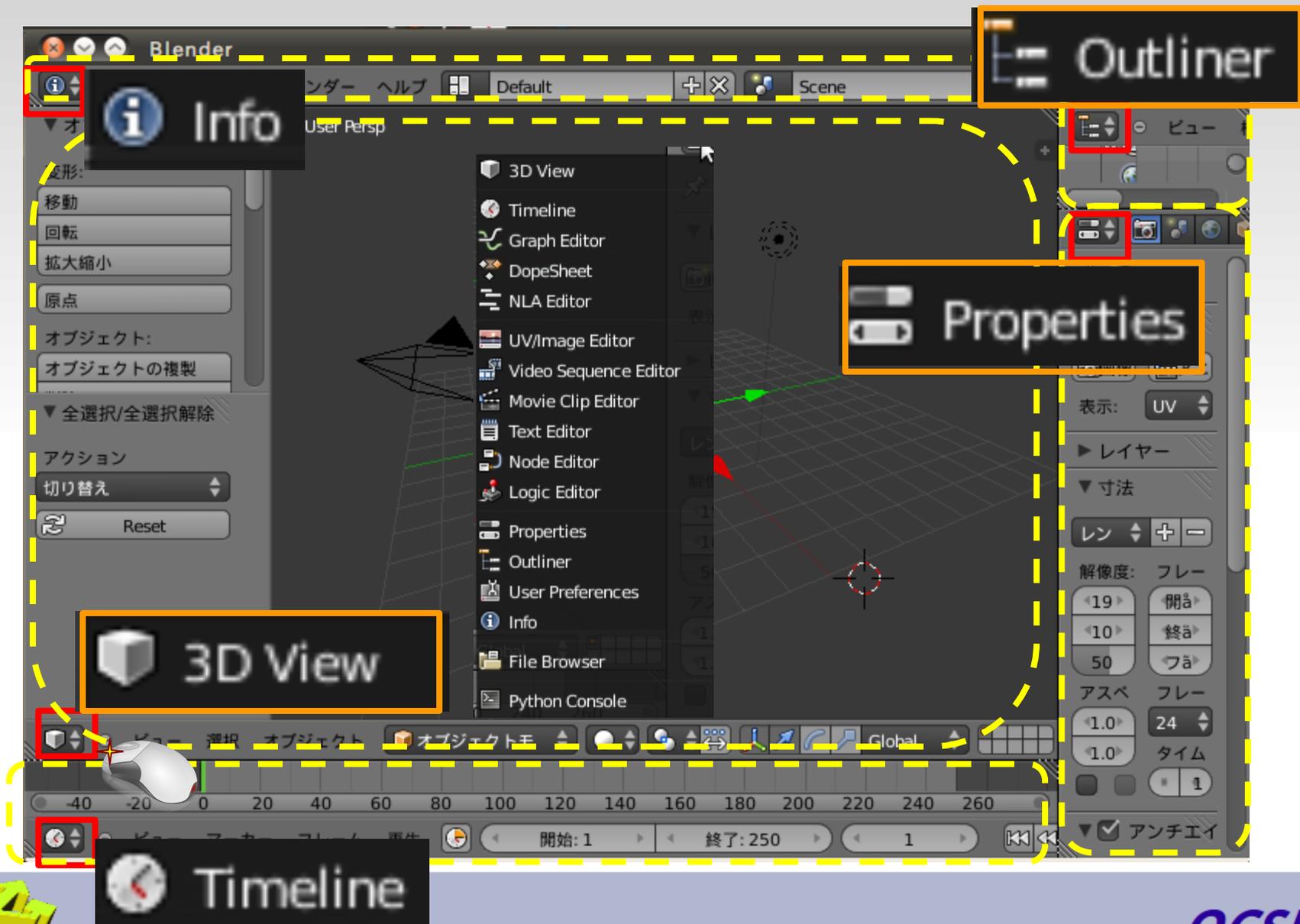
[次の章へ](#)

[http://26.blender3d.biz/basis\\_whatisblender.html](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画面上

 3D View

[テンキー]

- 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**
- 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**

## ■ その他の画面

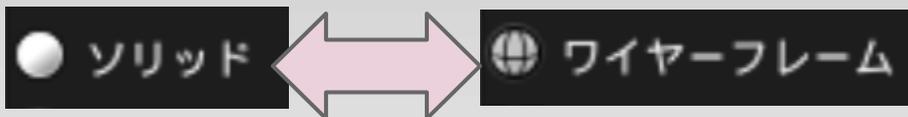
- 左ボタンクリック⇒選択・実行
- 右ボタンクリック⇒オプションメニュー表示

# モード

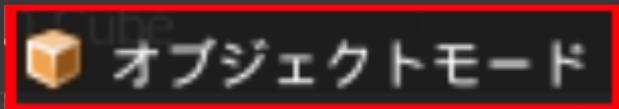


# シェーディング

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

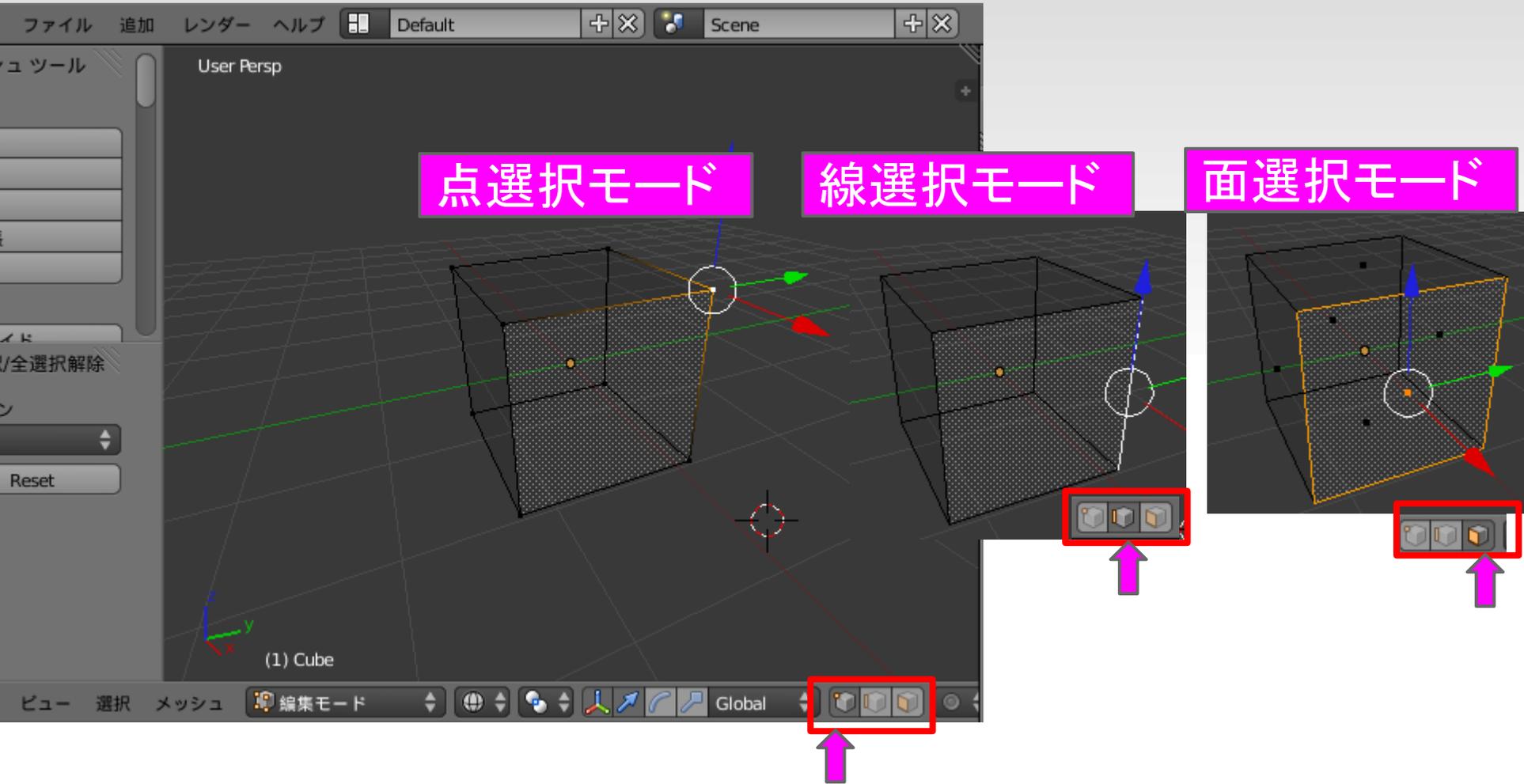
3Dビューのシェーディング

- テクスチャ
- ソリッド
- ワイヤーフレーム
- バウンズボックス

(1) Cube

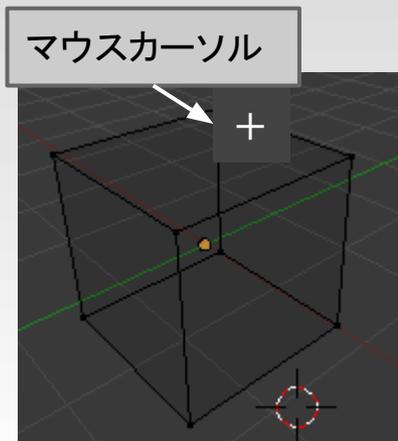
ビュー 選択 オブジェクト オブジェクトモ

# 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)

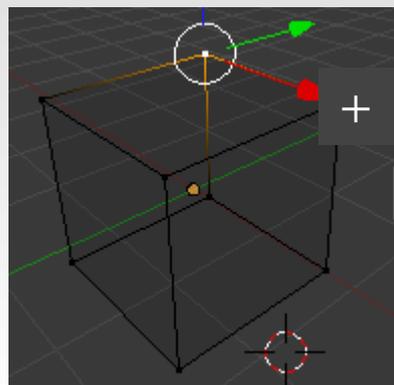
非選択



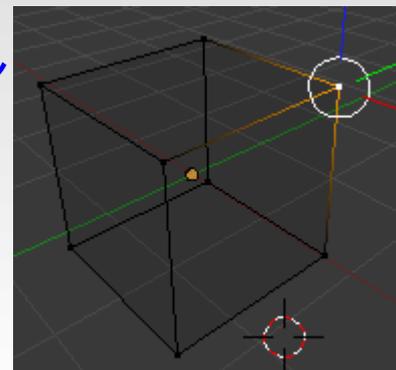
マウス  
右ボタン



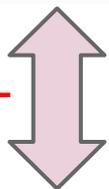
Aキー



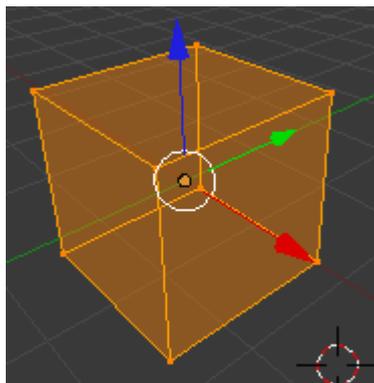
マウス  
右ボタン



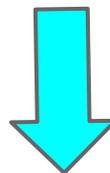
Aキー



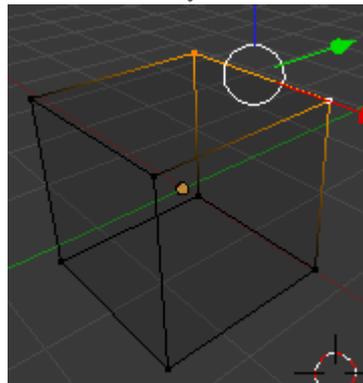
全選択



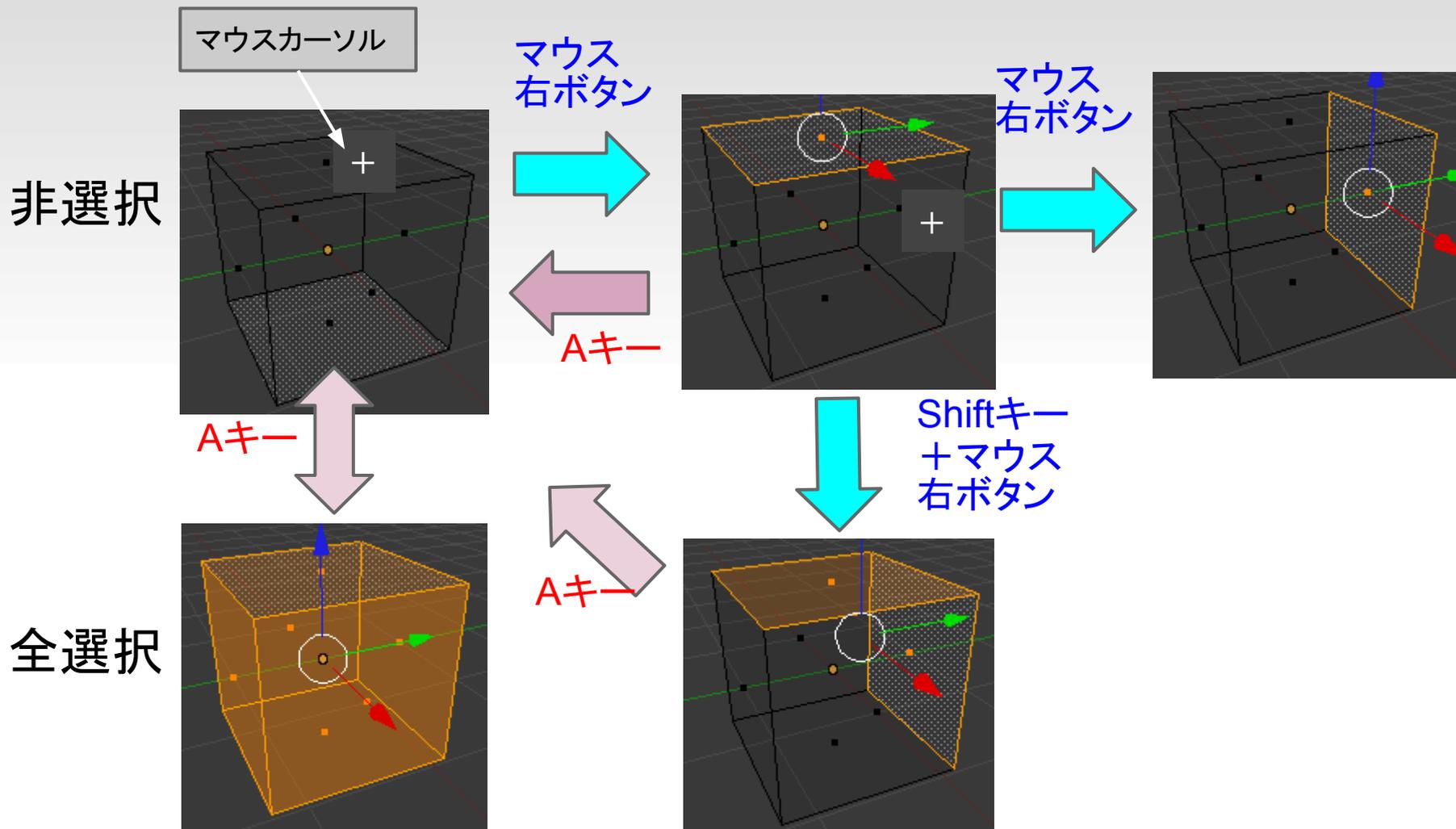
Aキー



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 the Blender 3D Viewport interface. On the left, the 'Object Tools' panel is visible with various manipulation tools like Move, Rotate, and Scale. The central 3D Viewport shows a cube in 'User Persp' view. On the right, a context menu is open, listing various actions such as 'Zoom to Border' and 'Toggle Perspective/Orthographic'. The 'Tools Shelf' (ツールシェルフ) is highlighted with a red box in the menu. At the bottom, the 'Tools Shelf' (ツールシェルフ) is also highlighted with a red box, and a red arrow points from it to the context menu. A pink box with the text 'Tキー' (T key) is positioned next to the 'Tools Shelf' label in the menu.

▼ 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  
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

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 N

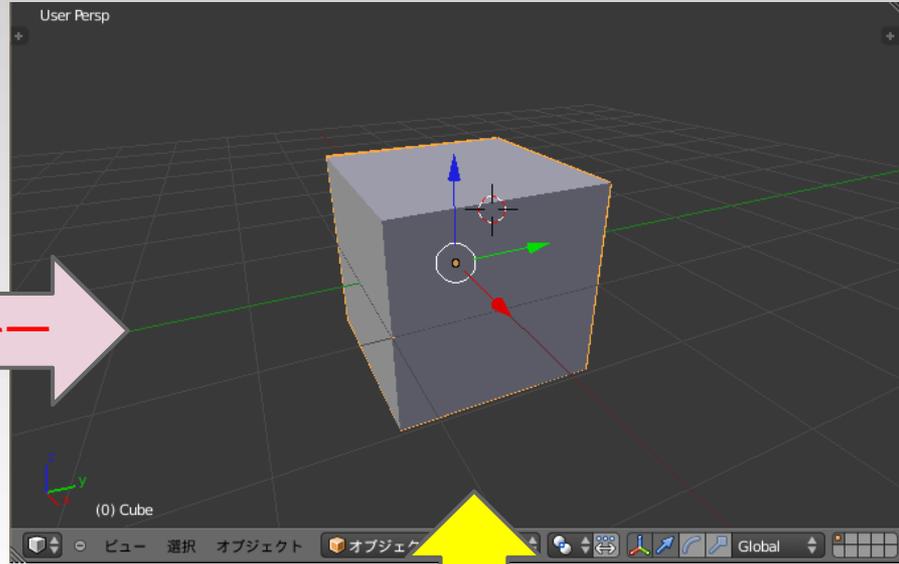
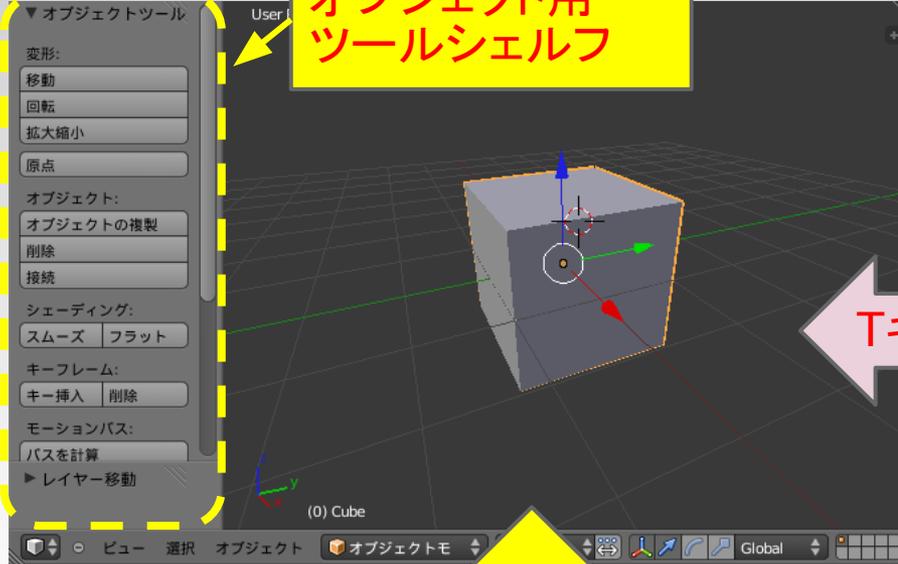
ビュー 選択 オブジェクト オ

ビュー 選択 オブジェクト オ

Global

Tキー

オブジェクト用  
ツールシェルフ

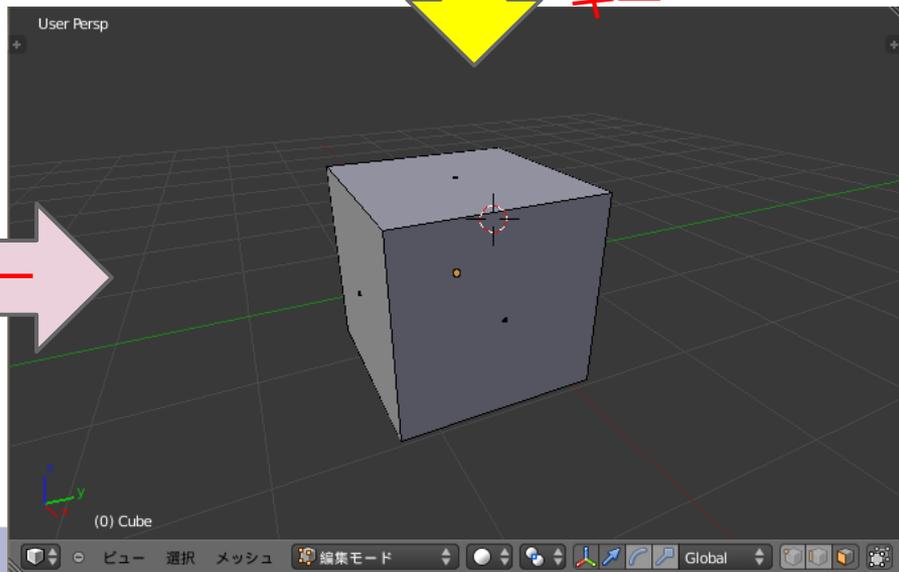
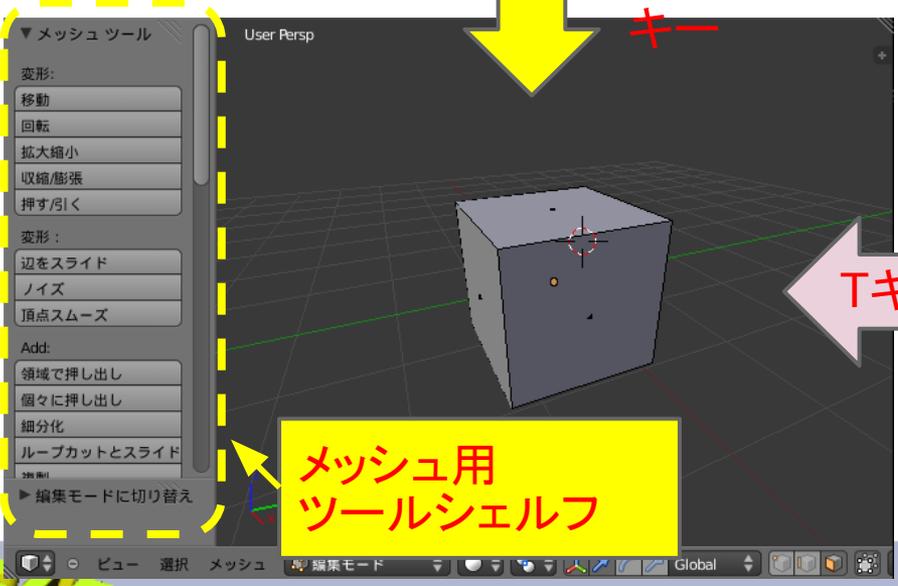


Tキー



Tab  
キー

メッシュ用  
ツールシェルフ



Tキー



Tab  
キー

# プロパティ(数値パッド)

The image shows a 3D software interface with a '3D View' window. On the left, there is a 'メッシュ ツール' (Mesh Tools) panel with various options like '移動' (Move), '回転' (Rotate), '拡大縮小' (Scale), etc. The main 3D view shows a cube on a grid. A context menu is open over the cube, listing various actions. The 'プロパティ' (Properties) option is highlighted with a red box. A red arrow points from the 'ビュー' (View) button in the bottom toolbar to the 'プロパティ' option. A pink box with the text 'Nキー' (N key) is located to the right of the context menu.

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

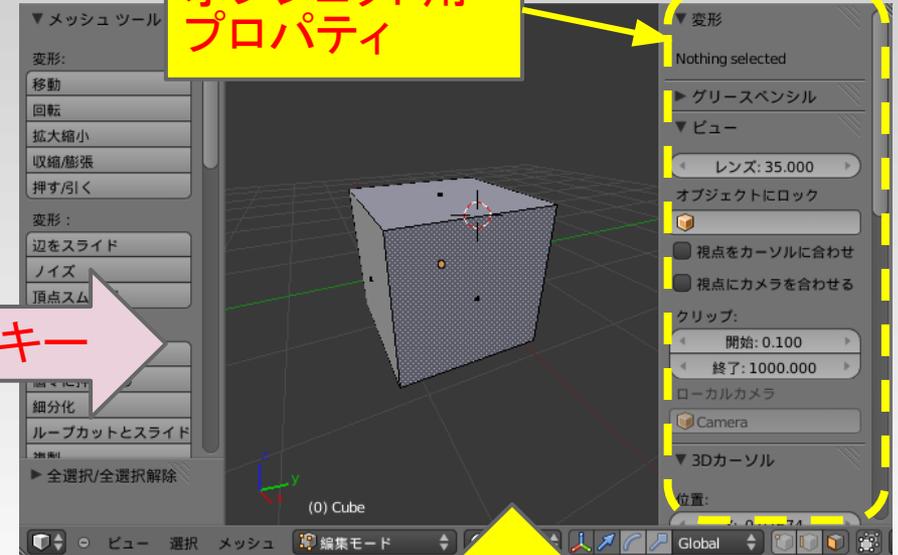
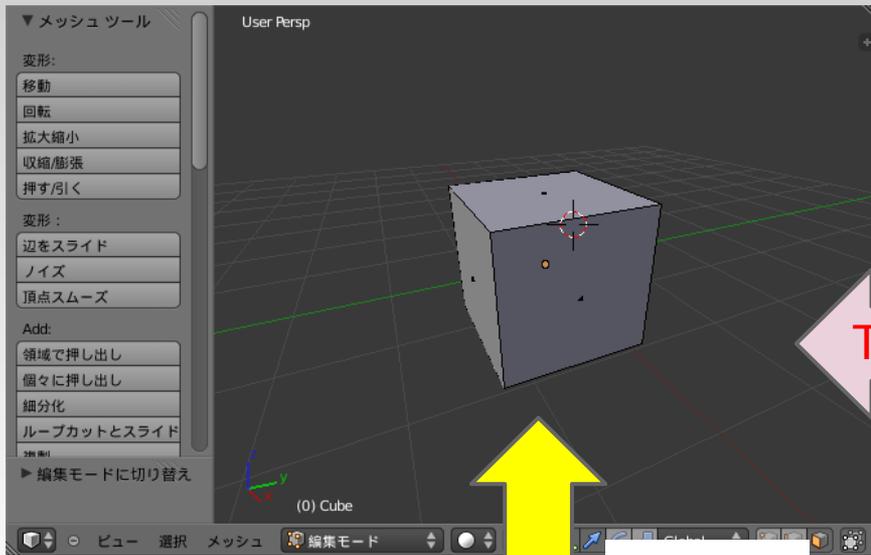
ツールシェルフ T

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

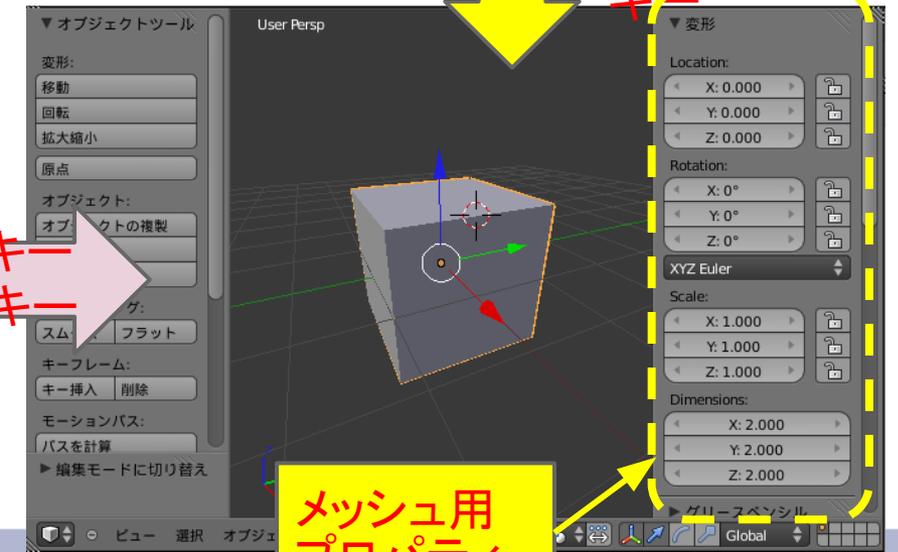
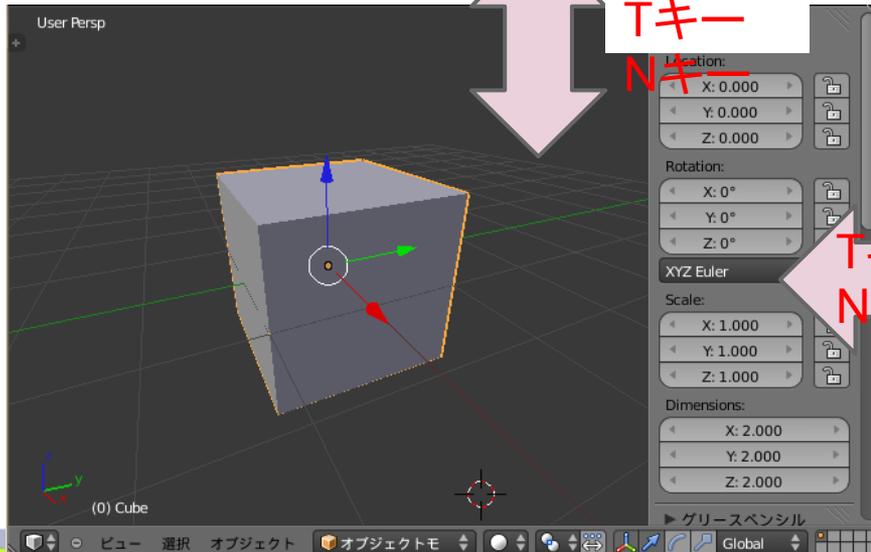
Nキー



オブジェクト用  
プロパティ

Tキー

Tab  
キー  
Tキー



メッシュ用  
プロパティ

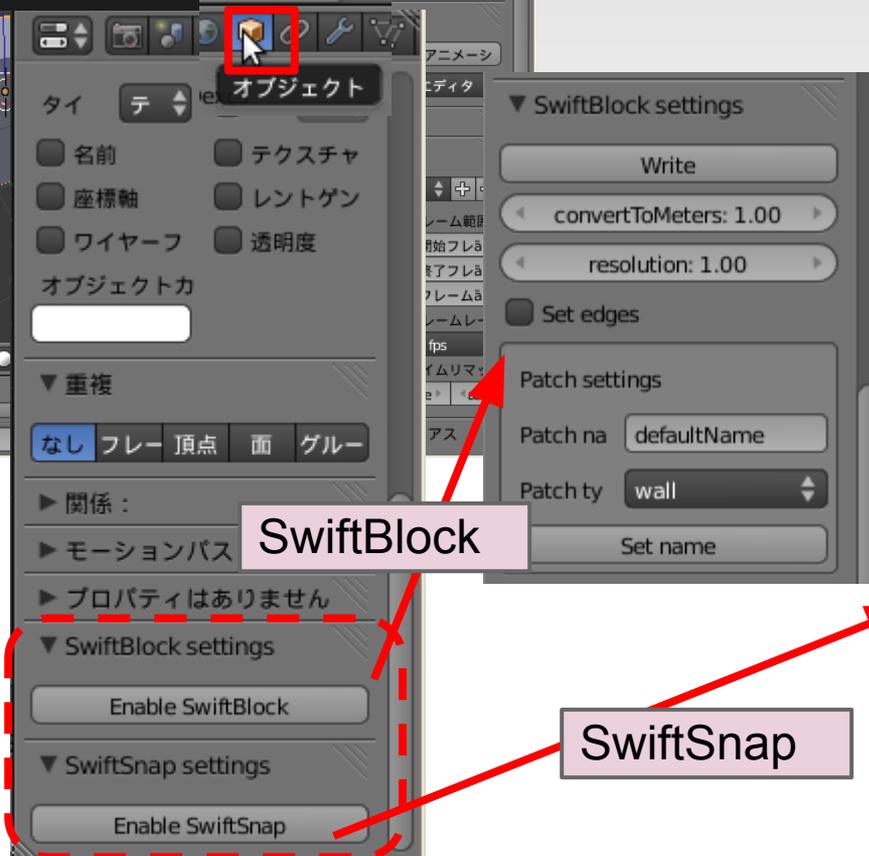
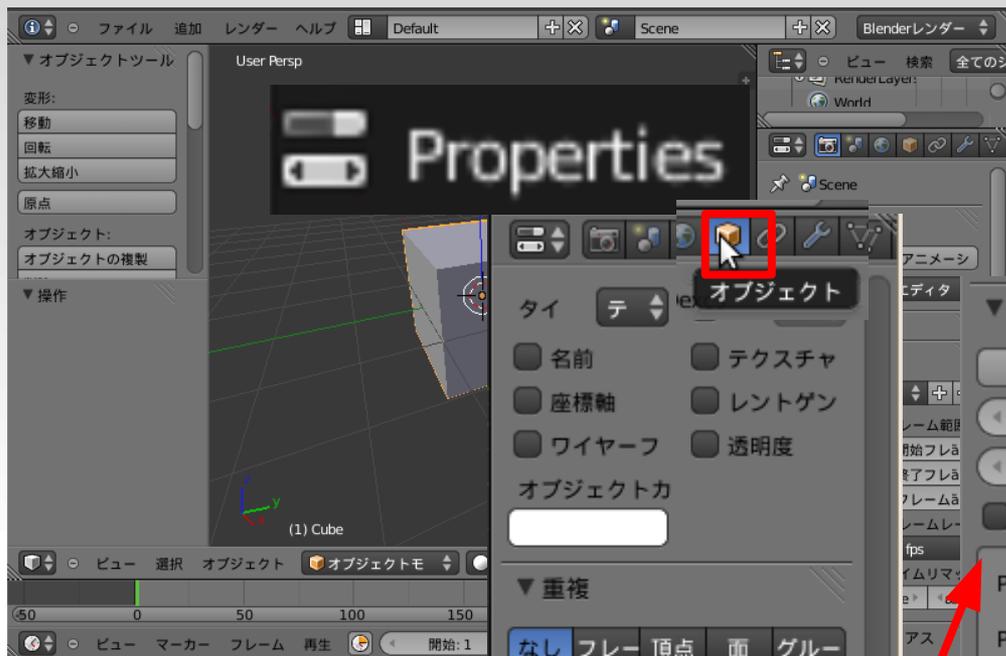
Tキー  
Nキー

Tab  
キー

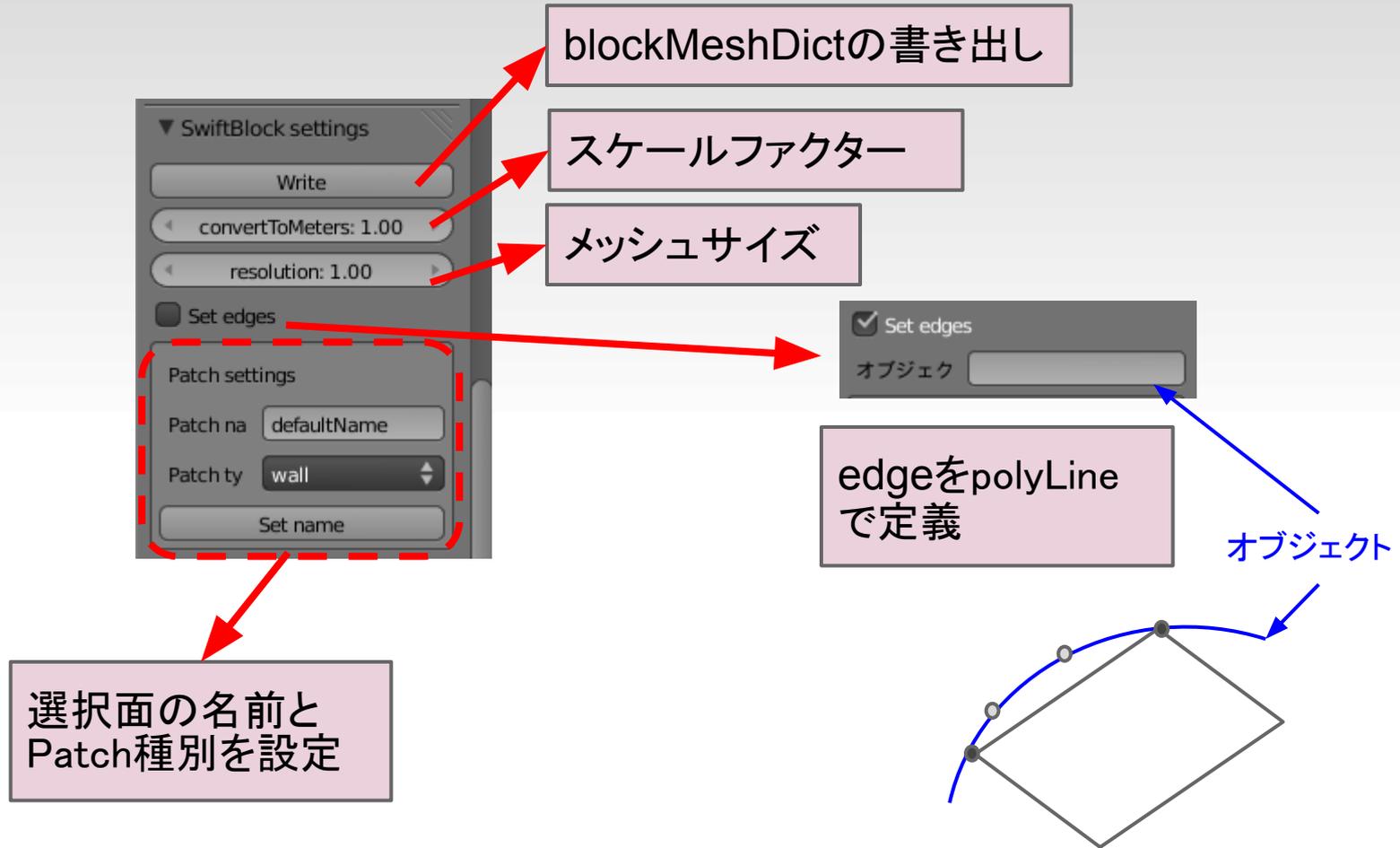
# swiftツール

- **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

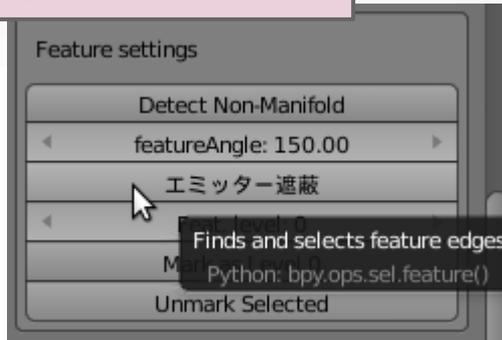


# SwiftSnap

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



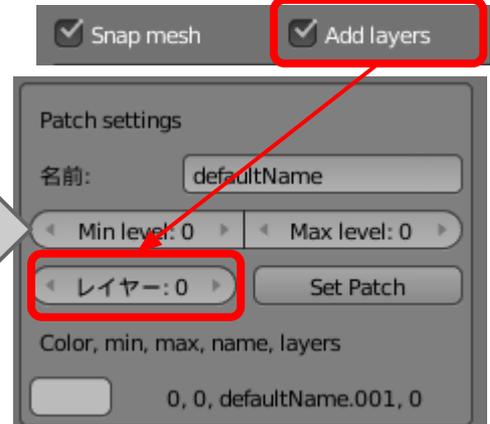
Region分割レベルの設定



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

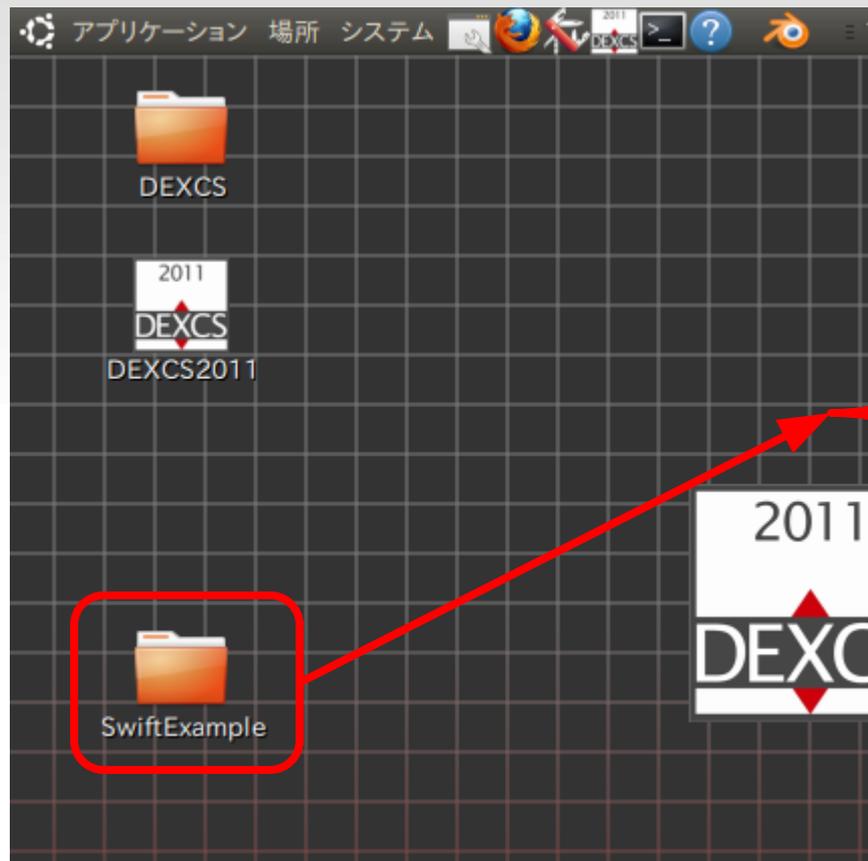
Surface分割レベルとレイヤー設定



# 実習例題

形状データのみ

形状データ  
+ Swiftパラメタ



Folder	*.blend	*Swift.blend
1_dexcs	○	○
2_pipeExampleSnap	○	◎
3_SwiftBlockExample	○	◎
4_Pipeblock	○	◎
5_cylinder	○	○
6_cube	○	

◎: 公開サンプル  
○: OCSE^2製作

# 実習手順の概要

Folder	*.blend	*Swift.blend	SwiftSnap ⇒ Write	SwiftBlock ⇒ Write	blockMesh 実行	snappyHexMesh 実行
1_dexcs	○	○ 1	2	3	4	5
2_pipeExampleSnap	○	◎ 1	2		3	4
3_SwiftBlockExample	○	◎ 1	-	2	3	-
4_Pipeblock	○	◎ 1	-	2	3	-
5_cylinder	○	○ 1	-	2	3	-
6_cube	○ 1	2	3	4	5	6

snappyHexMesh  
による外部流れ  
メッシュ作成

snappyHexMesh  
による内部流れ  
メッシュ作成

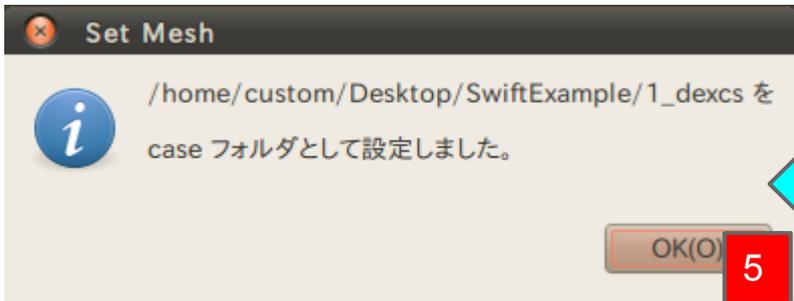
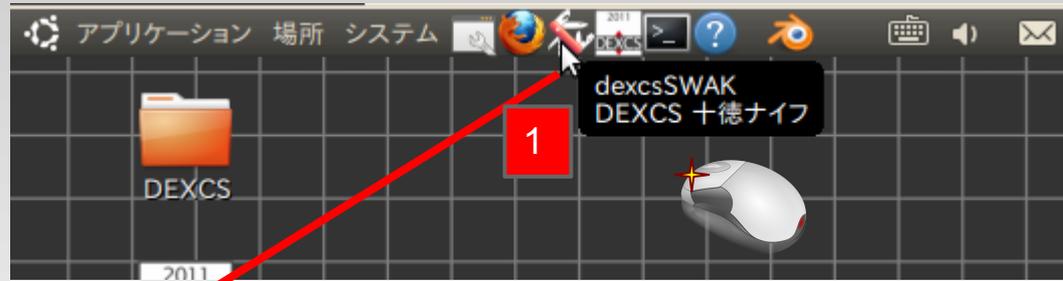
Swiftツールの  
機能確認

blockMesh  
(のみ)による  
メッシュ作成

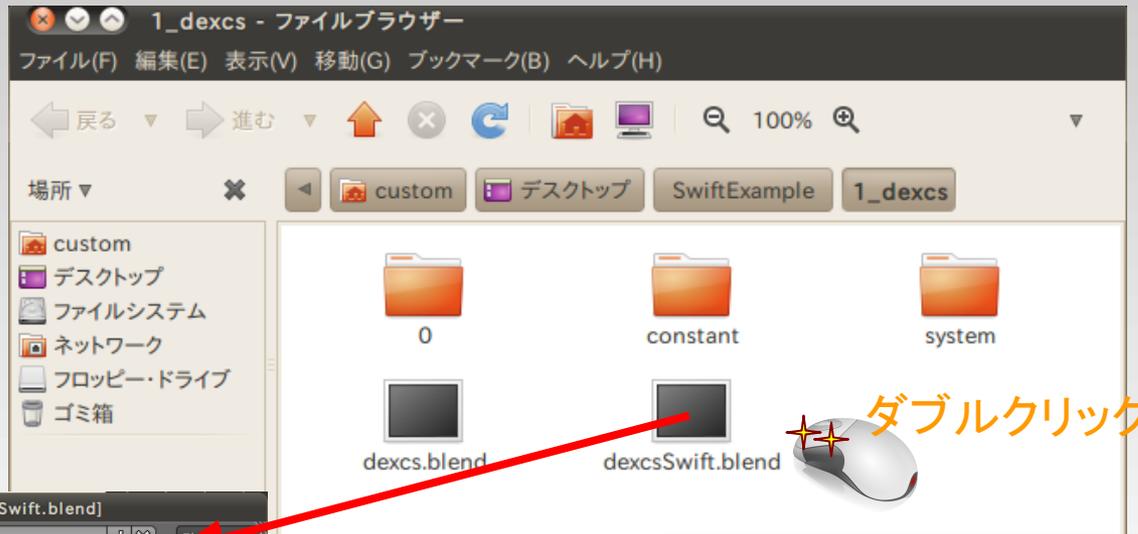
snappyHexMesh  
による外部流れ  
メッシュ作成

使用法  
演習

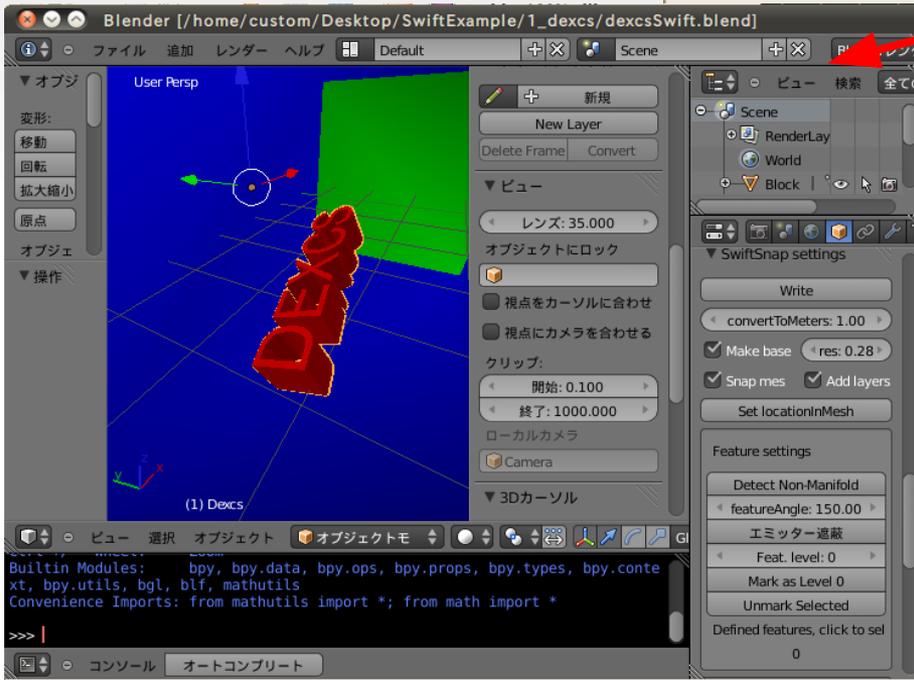
# 実習要領



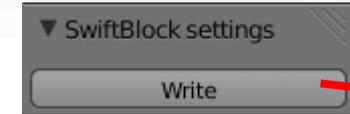
# Swiftツールの の実行



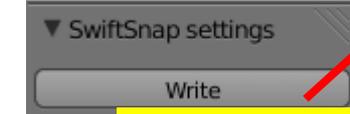
ダブルクリック



SwiftBlock



SwiftSnap

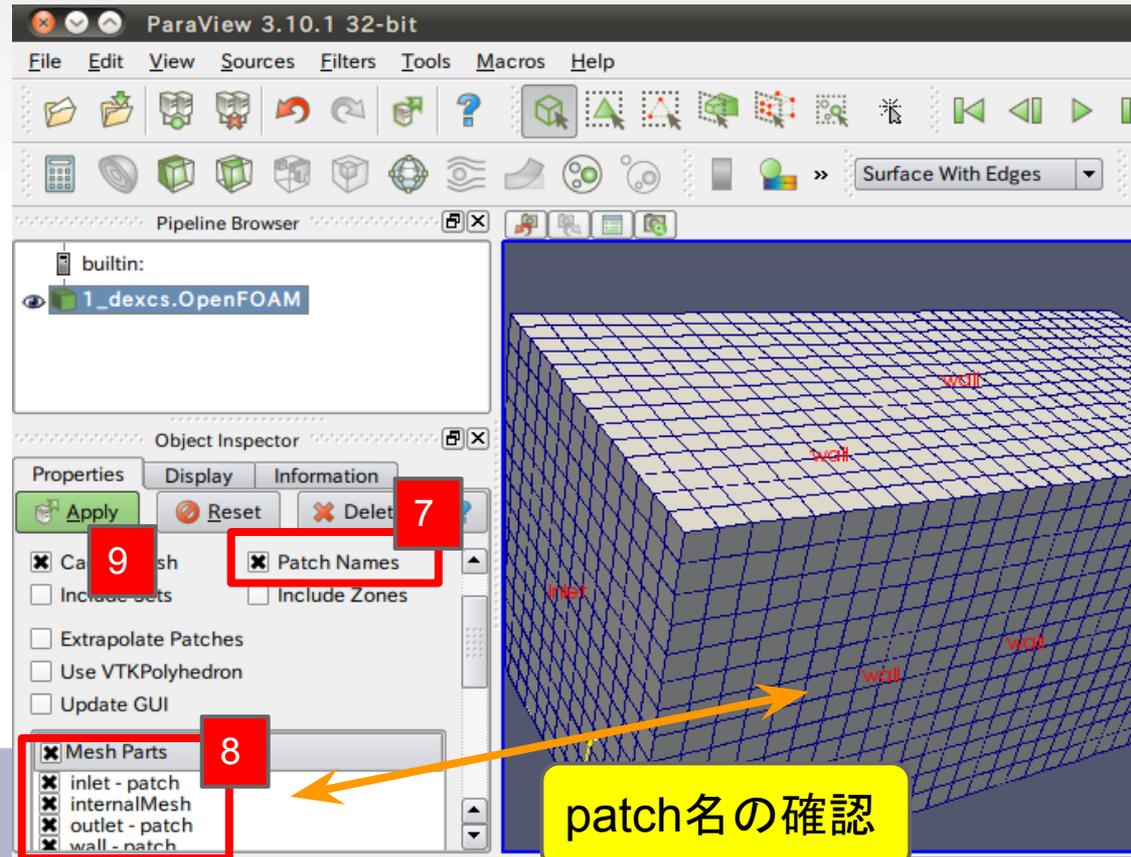
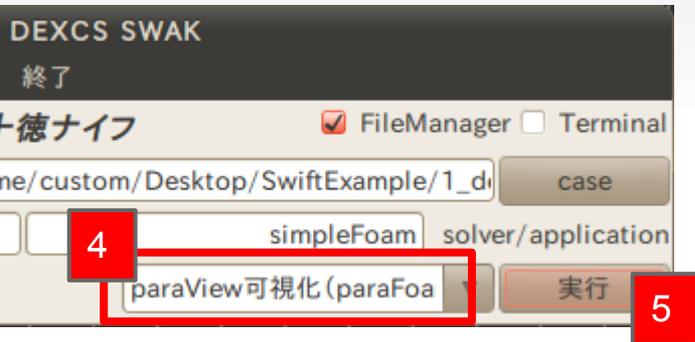


1_dexcs	5個のアイテム
0	0個のアイテム
constant	2個のアイテム
polyMesh	0個のアイテム
triSurface	0個のアイテム
system	5個のアイテム
triSurface	0個のアイテム
controlDict	1.3 KB
decomposePar...	1.4 KB
fvSchemes	1.6 KB
fvSolution	1.2 KB
dexcs.blend	533.4 KB
dexcsSwift.blend	511.9 KB

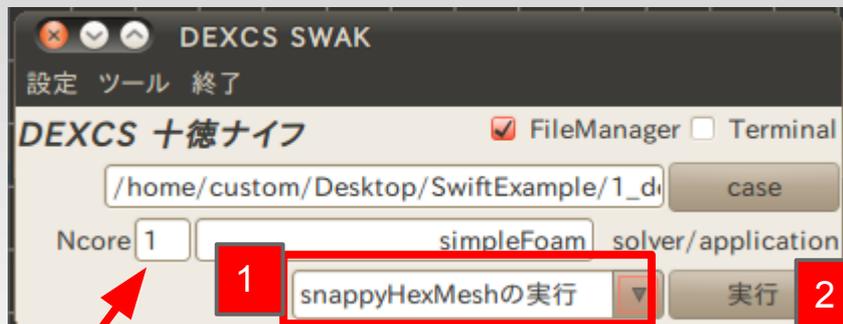
Dictファイルの  
出力先を間違  
えないよう注意

実習2では、  
もう一手間必  
要(後述)

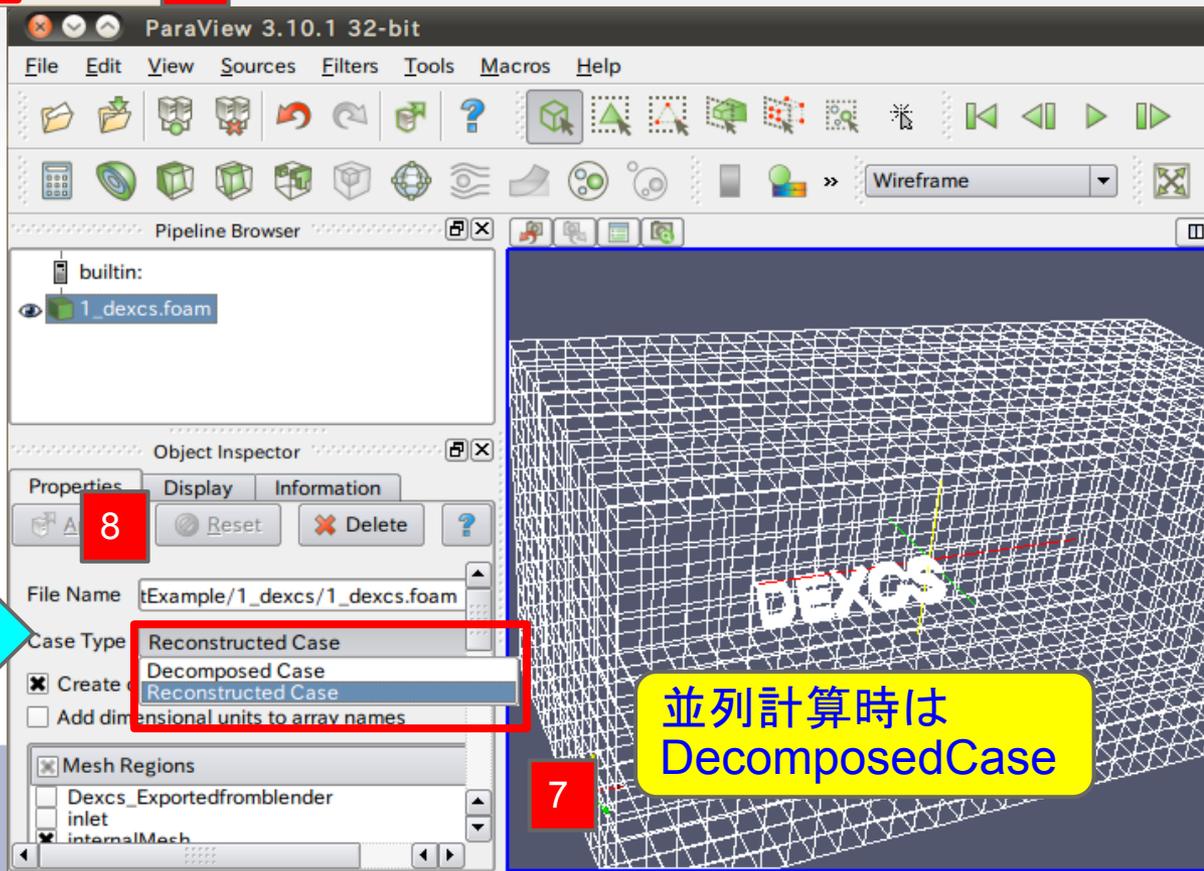
# メッシュ作成と確認 (blockMesh)



# メッシュ作成と確認(snappyHexMesh)



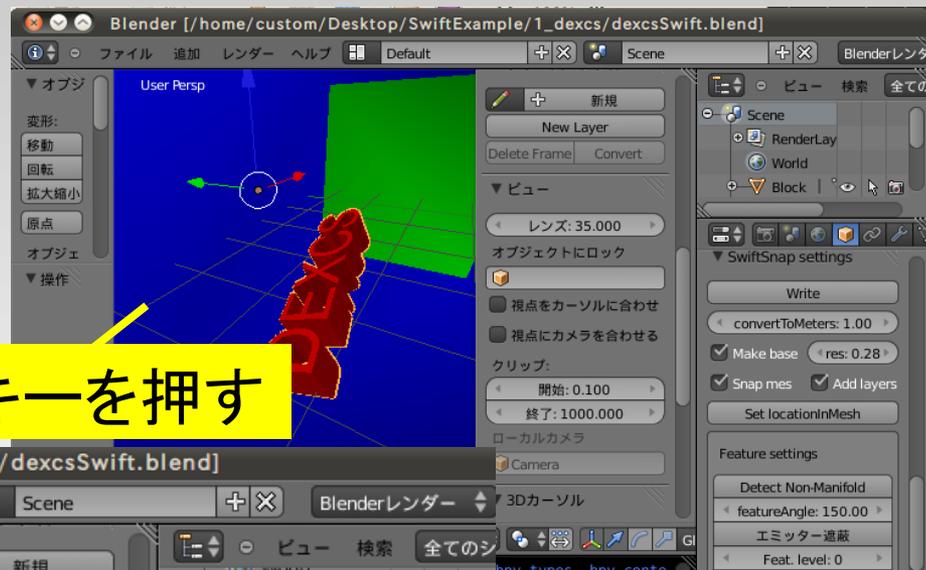
Ncore:並列数



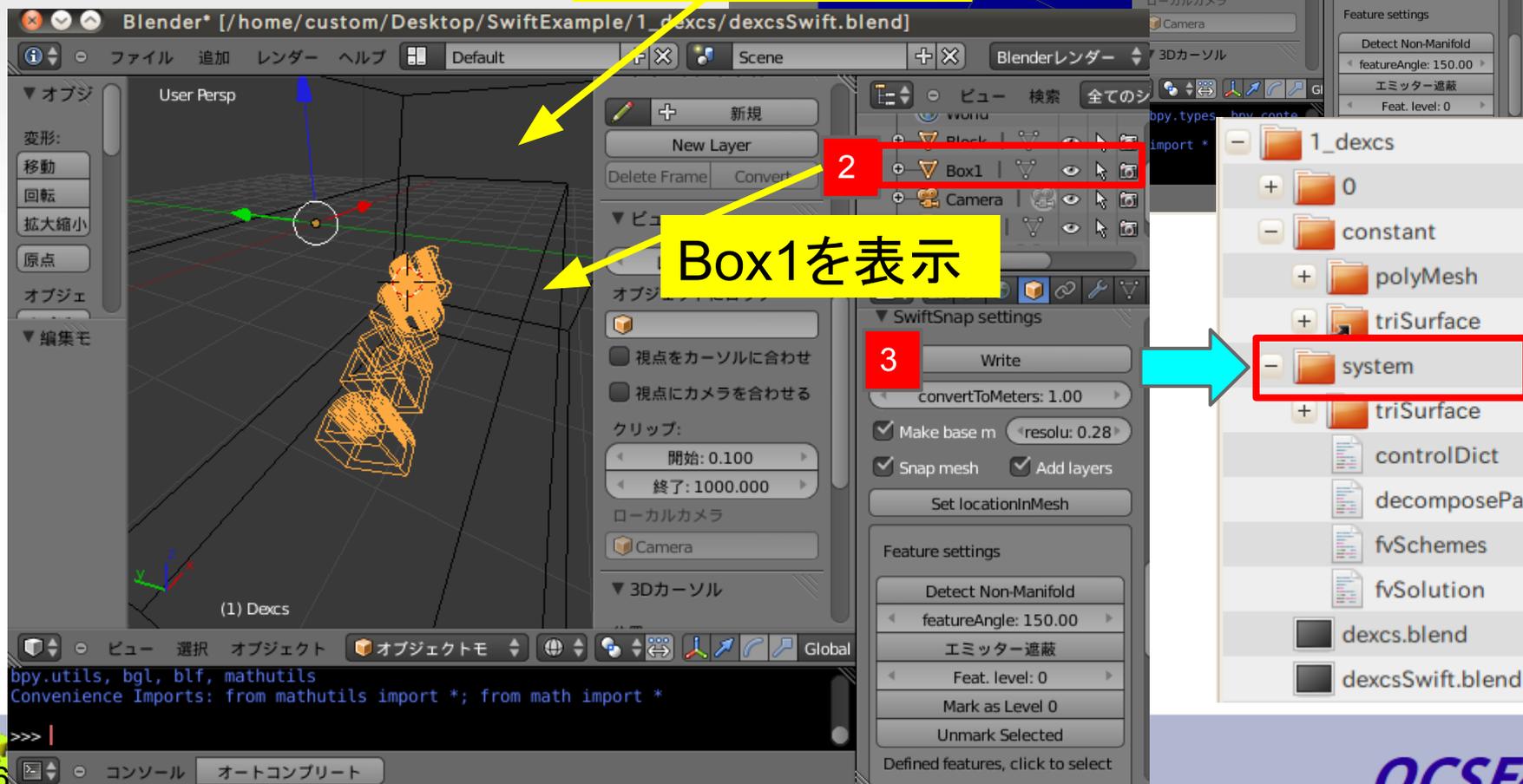
並列計算時は  
DecomposedCase

# 実習1

## (SwiftSnap)



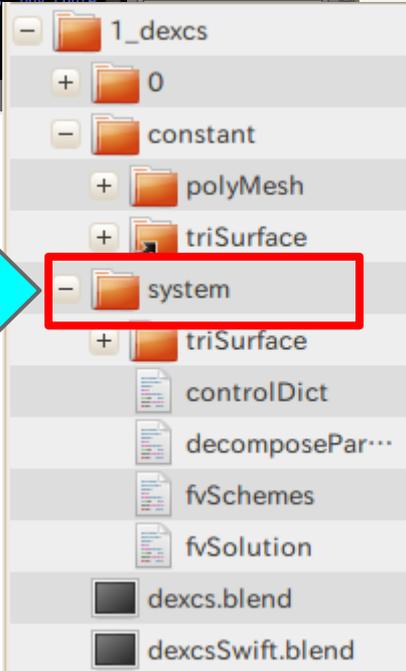
1 Tabキーを押す



2

Box1を表示

3



# 実習1 (SwiftBlock)

1 Blockを選択

2 patch名の確認

3

Tabキー

Blender\* [/home/custom/Desktop/SwiftBlock] Scene  
Default Blenderレンダー

オブジェクト: User Persp  
変形: 移動, 回転, 拡大縮小, 原点  
削除: すべ

ビュー: レンズ: 35.000, オブジェクトにロック, 視点をカーソルに合わせ, 視点到カメラを合わせる, クリップ: 開始: 0.100, 終了: 1000.000, ローカルカメラ, Camera

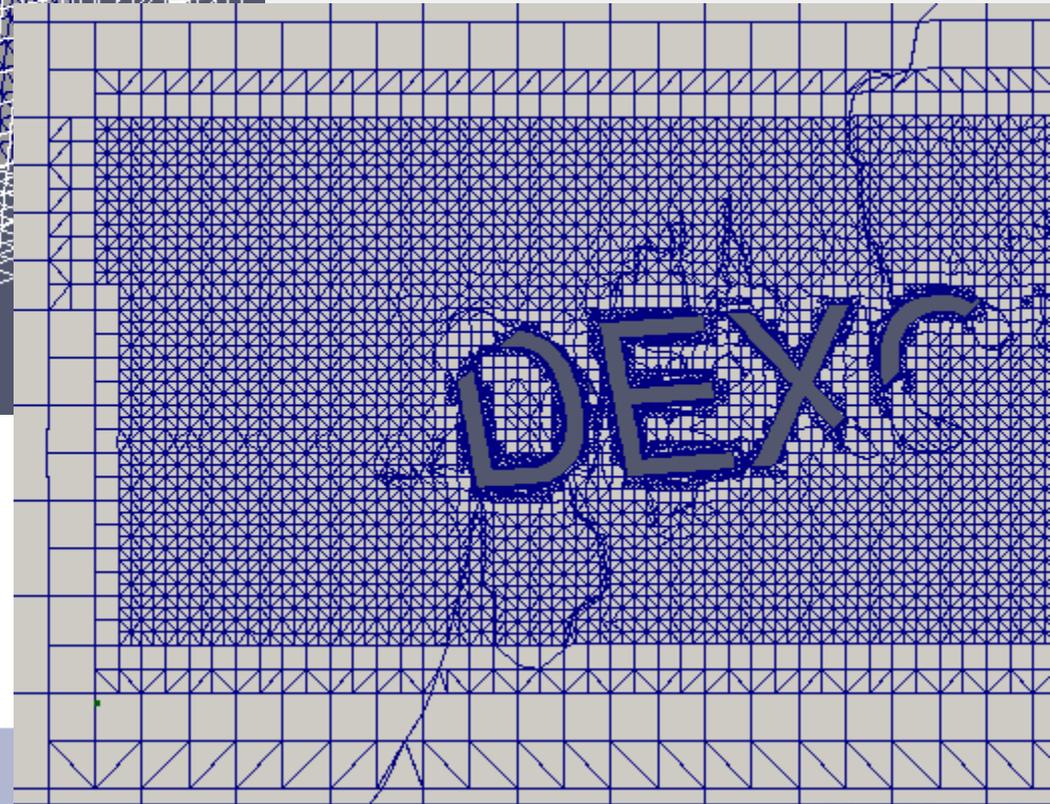
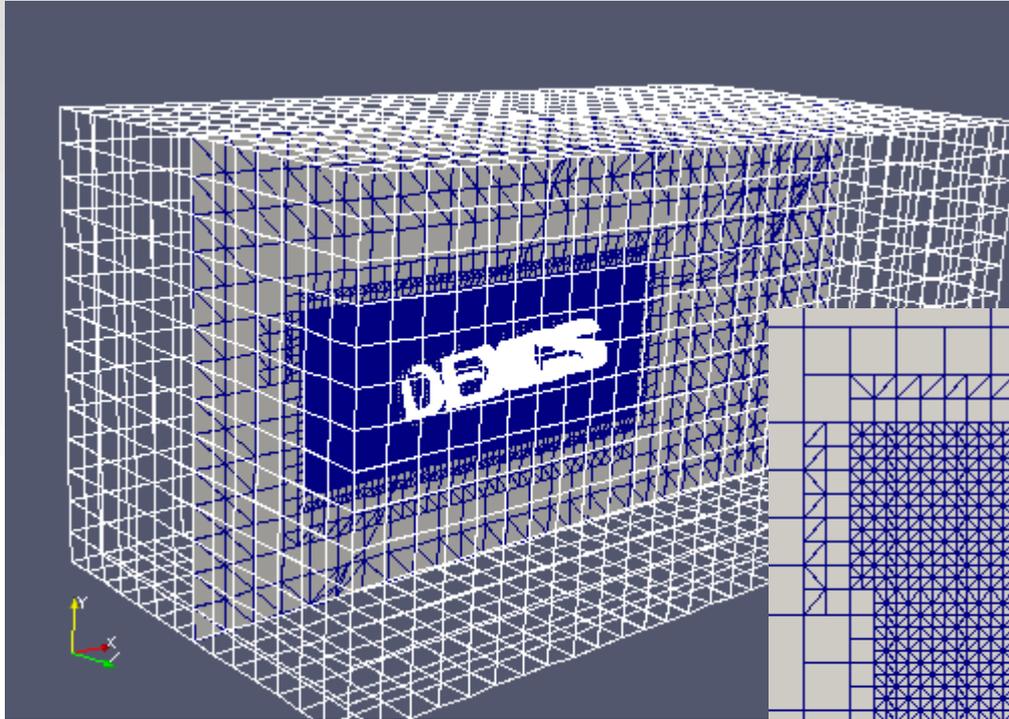
Write, convertToMeters: 1.00, resolution: 0.28, Set edges, Patch settings, Patch na: wall, Patch typ: patch, Set name, defaultName wall, inlet patch, outlet patch, wall patch

コンソール: bpy.utils, bgl, blf, mathutils, Convenience Imports: from mathutils

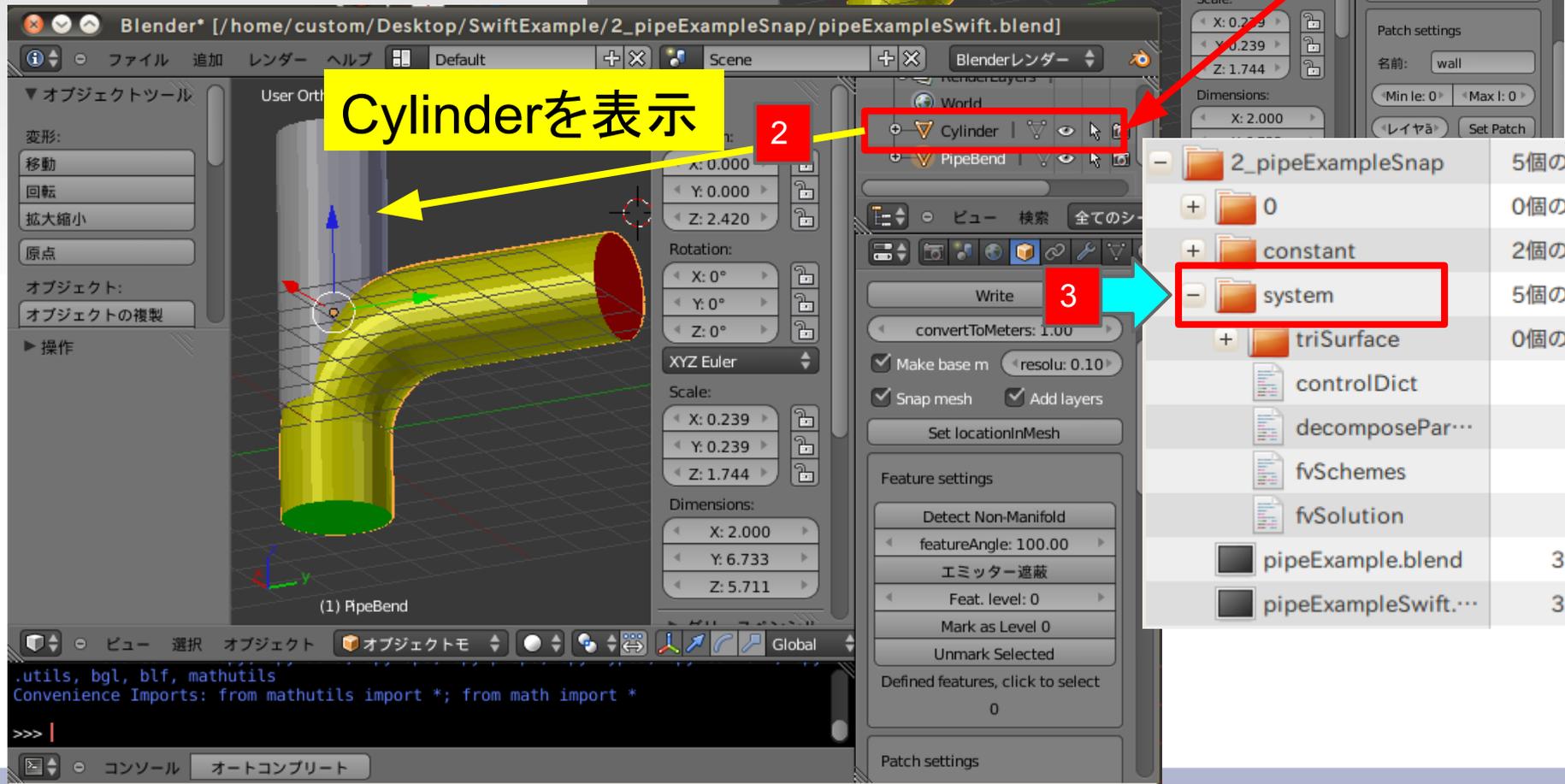
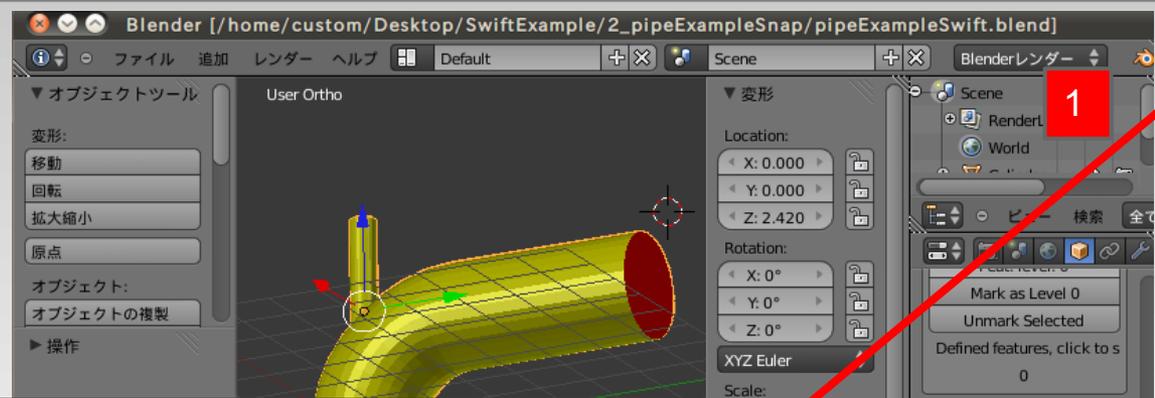
1_dexcs	5個のア
0	0個のア
constant	2個のア
polyMesh	0個のア
triSurface	0個のア
system	5個のア
triSurface	0個のア
controlDict	1
decomposePar...	1
fvSchemes	1
fvSolution	1
dexcs.blend	533
dexcsSwift.blend	511

# 実習1 (メッシュ確認例)

nPoints:378833  
nCells:344038



# 実習2 (SwiftSnap)



# 実習2 (SwiftSnap の続き)

SwiftSnap settings

Write **1**

The image shows a sequence of three file explorer windows illustrating the SwiftSnap process. The first window shows the 'SwiftSnap settings' dialog with the 'Write' button highlighted by a red box and labeled '1'. A cyan arrow points from this button to the second window. The second window shows a file explorer view of a project folder '2\_pipeExampleSnap'. A red box highlights the 'triSurface' folder, and another red box highlights the file 'blockMeshDict'. A blue arrow labeled '2' points from this 'blockMeshDict' file to the third window. The third window shows the same file explorer, but now a blue box highlights the file 'blockMeshD...' in the 'polyMesh' folder, indicating it has been moved there. A yellow box with red text '手作業にて移動' (Move manually) is positioned below the blue arrow.

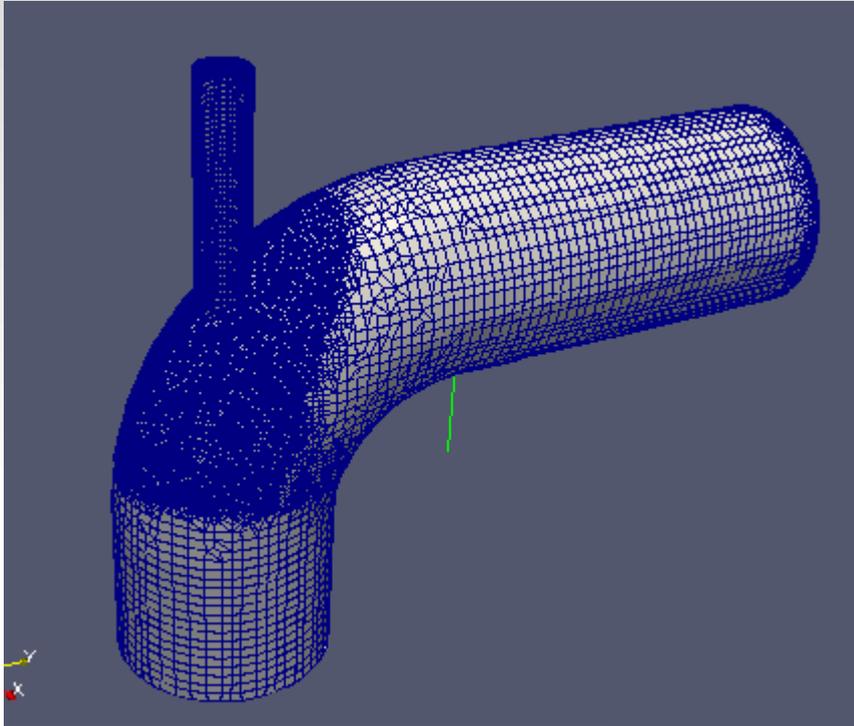
Folder/Item	Count
2_pipeExampleSnap	5個のアイテム
0	0個のアイテム
constant	2個のアイテム
system	5個のアイテム
triSurface	0個のアイテム
controlDict	1.3 KB
decomposePar...	1.4 KB
fvSchemes	1.6 KB
fvSolution	1.2 KB
pipeExample.blend	391.5 KB
pipeExampleSwift...	326.8 KB

Folder/Item	Count
2_pipeExampleSnap	5個のアイテム
0	0個のアイテム
constant	2個のアイテム
polyMesh	0個のアイテム
(空)	
triSurface	7個のアイテム
system	7個のアイテム
triSurface	7個のアイテム
blockMeshDict	1001 バイト
controlDict	1.3 KB
decomposePar...	1.4 KB
fvSchemes	1.6 KB
fvSolution	1.2 KB
snappyHexMes...	2.8 KB
pipeExample.blend	391.5 KB
pipeExampleSwift...	326.8 KB

Folder/Item	Count
2_pipeExampleSnap	5個のアイテム
0	0個のアイテム
constant	2個のアイテム
polyMesh	1個のアイテム
blockMeshD...	1001 バイト
triSurface	7個のアイテム
system	6個のアイテム
system	7個のアイテム
decomposePar...	1.4 KB
fvSchemes	1.6 KB
fvSolution	1.2 KB
snappyHexMes...	2.8 KB
pipeExample.blend	391.5 KB
pipeExampleSwift...	326.8 KB

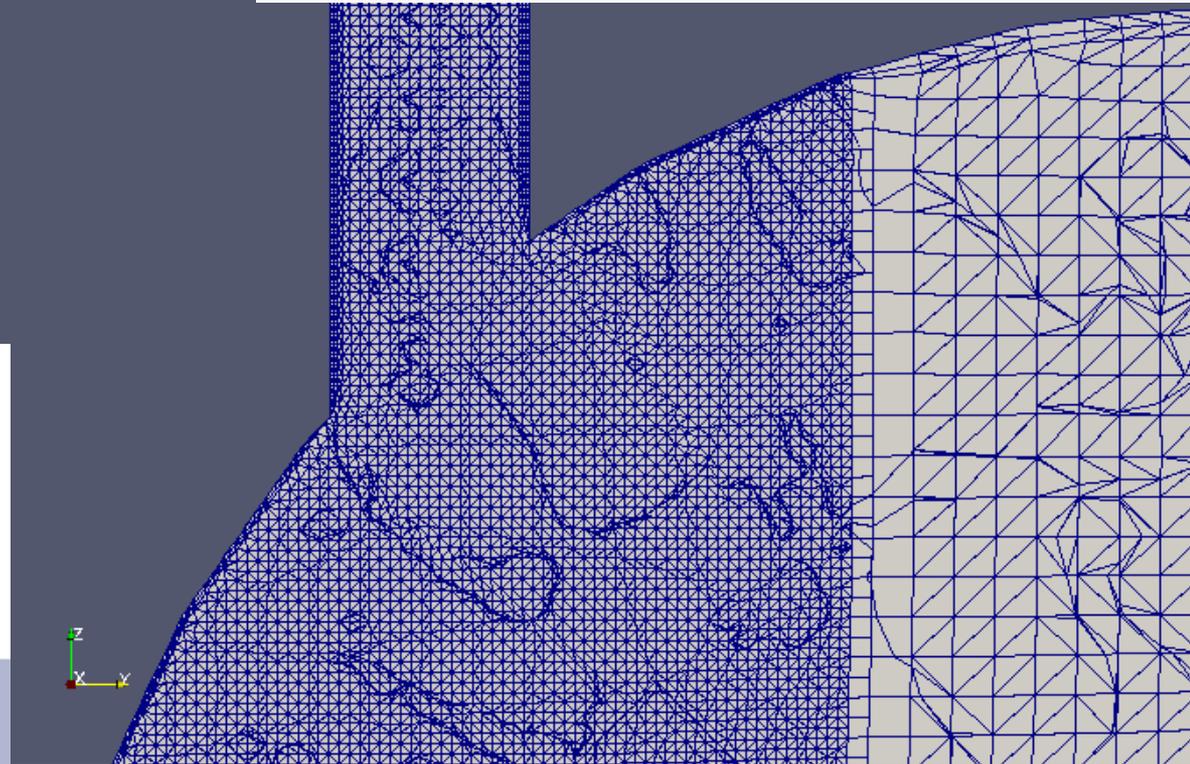
生成されたファイル

# 実習2 (メッシュ確認例)



nPoints:371879

nCells:346956



# 実習3

## (SwiftBlock)

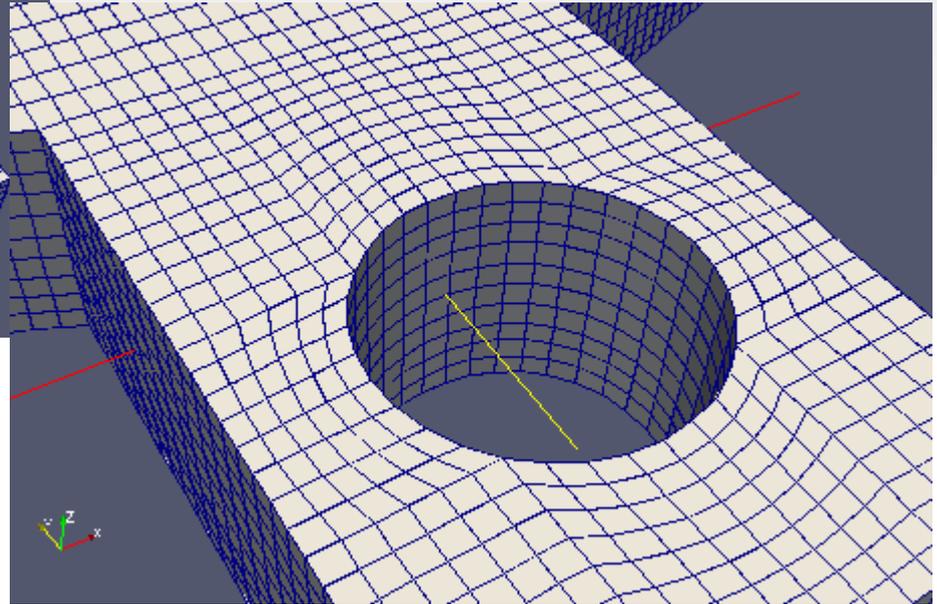
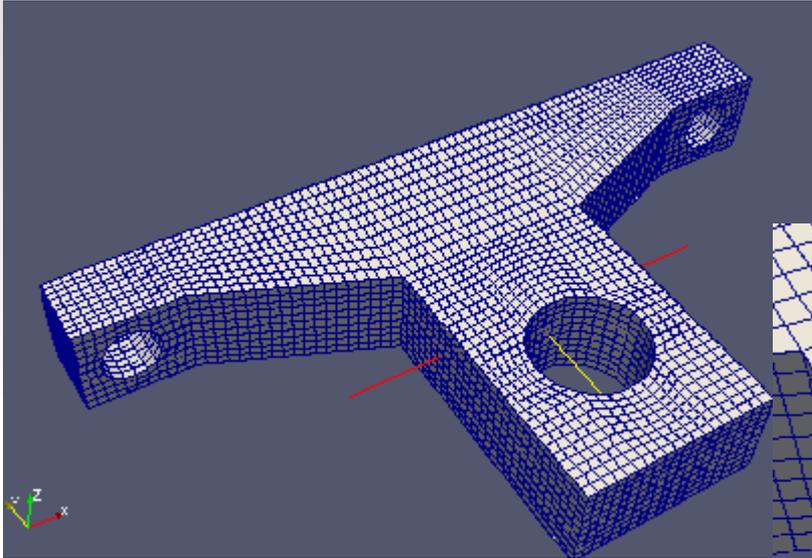
Blender 2.80 interface showing SwiftBlock settings and a file browser. A red box highlights the 'Write' button in the SwiftBlock settings (labeled '1') and the 'Write' button in the file browser (labeled '2'). A yellow box with the text '数分後' (After a few minutes) and a blue arrow points to a 3D model of a block structure. A blue arrow also points from the 'Write' button in the SwiftBlock settings to the 'polyMesh' folder in the file browser.

Folder/Item	Count
3_SwiftBlockExample	5個のアイテム
0	0個のアイテム
constant	2個のアイテム
polyMesh	0個のアイテム
triSurface	0個のアイテム
system	5個のアイテム
block.blend	297.5 KB
blockSwift.blend	313.5 KB

# 実習3 (メッシュ確認例)

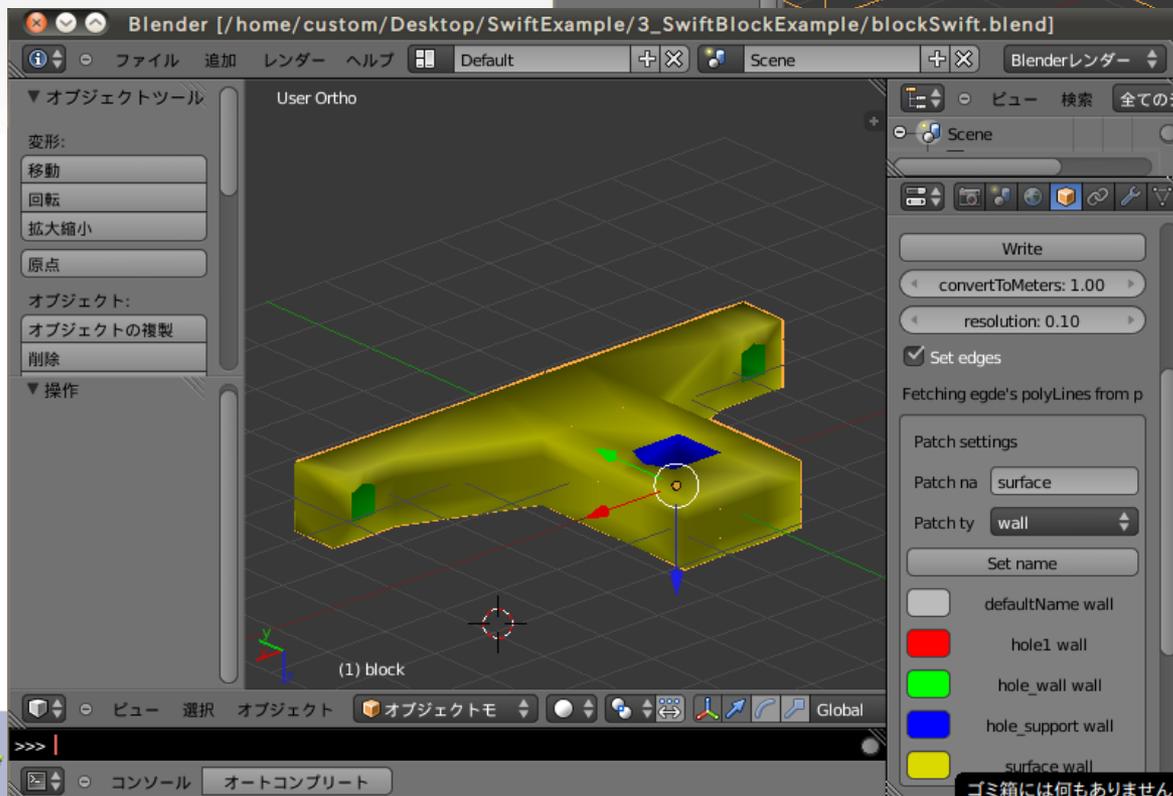
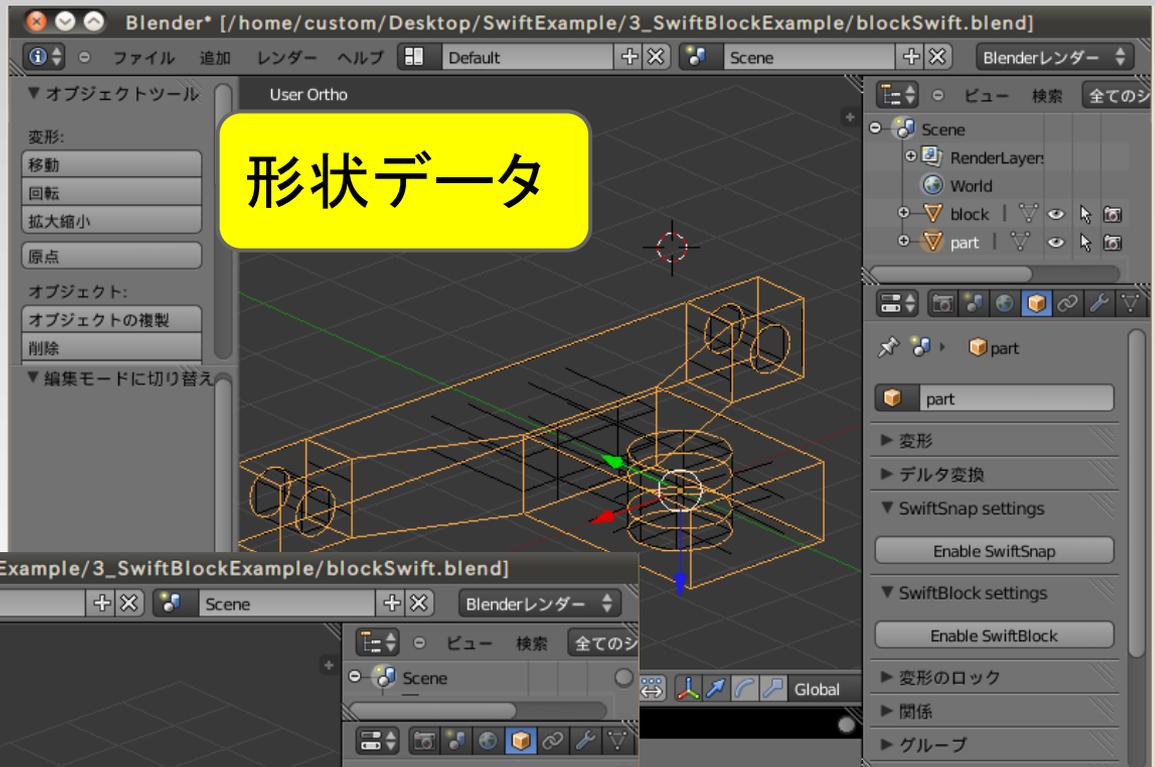
nPoints: 15345

nCells: 12470



# 実習3

## (SwiftBlock)



Swiftパラメタ  
の確認

# 実習4

## (SwiftBlock)

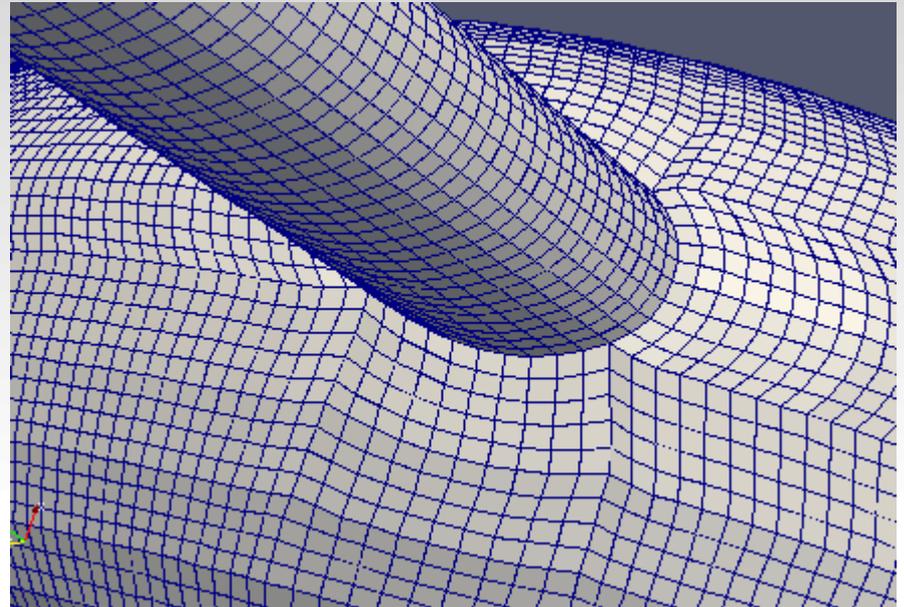
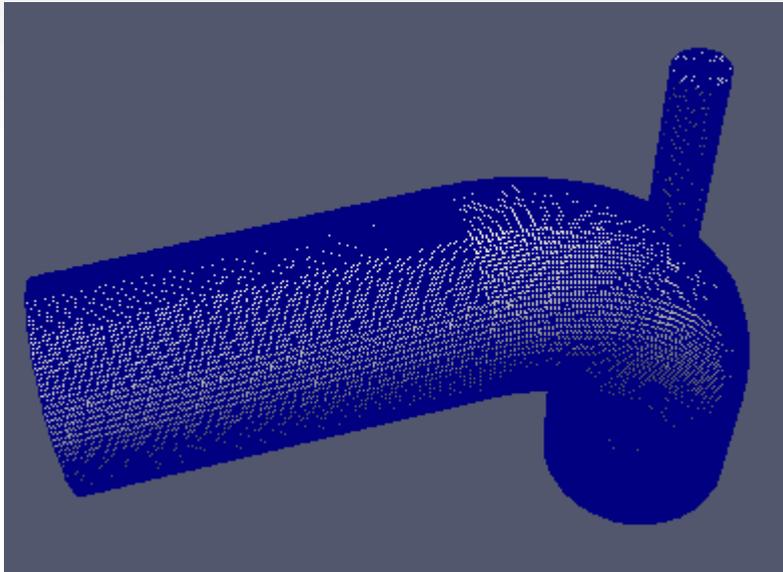
Blender 2.79 interface showing SwiftBlock settings and a file browser. A yellow box says "数分後" (After a few minutes). Red boxes and arrows highlight the "Write" button in the SwiftBlock settings (2), the "Write" button in the file browser (3), and the "inlet" patch in the SwiftBlock settings (1). A blue arrow points from the SwiftBlock settings to the 3D view. A red arrow points from the file browser to the SwiftBlock settings.

Item	Count	Size
4_Pipeblock	5個のアイテム	
0	0個のアイテム	
constant	2個のアイテム	
polyMesh	0個のアイテム	
triSurface	0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
decomposePar...		1.4 KB
fvSchemes		1.6 KB
fvSolution		1.2 KB
bend.blend		323.8 KB
bendSwift.blend		337.5 KB

SwiftBlock settings:

- Write
- convertToMeters: 1.00
- resolution: 0.03
- Set edges
- Fetching egde's polyLines from Pi
- Patch settings
- Patch na: inlet
- Patch ty: patch
- Set name
- defaultName wall
- stream patch
- outlet patch
- inlet patch

# 実習4 (メッシュ確認例)

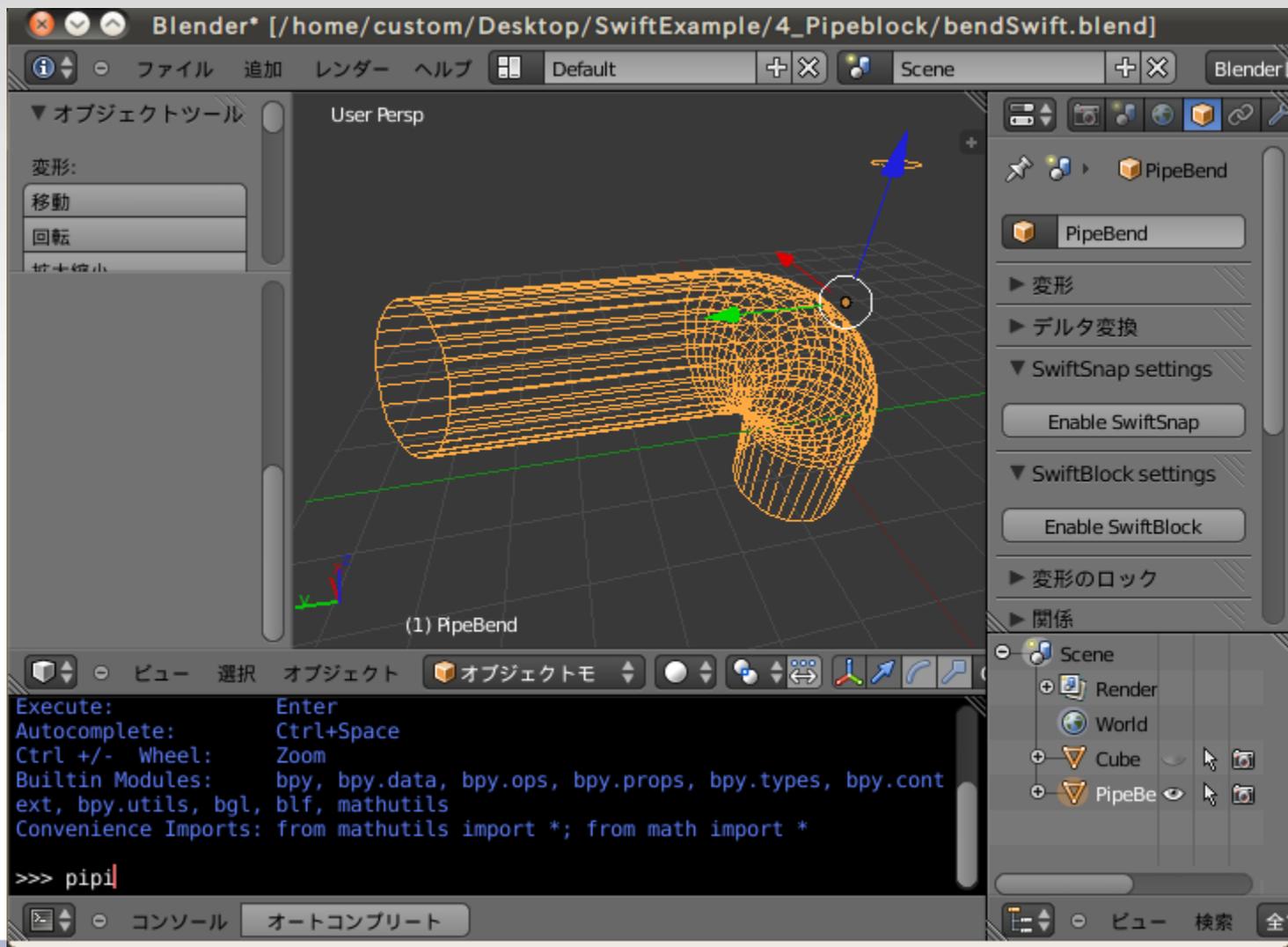


nPoints: 1075524

nCells: 1052784

メモリ1GB では計算不能  
1.5GBならOK

# 実習4(形状確認)



# 実習5

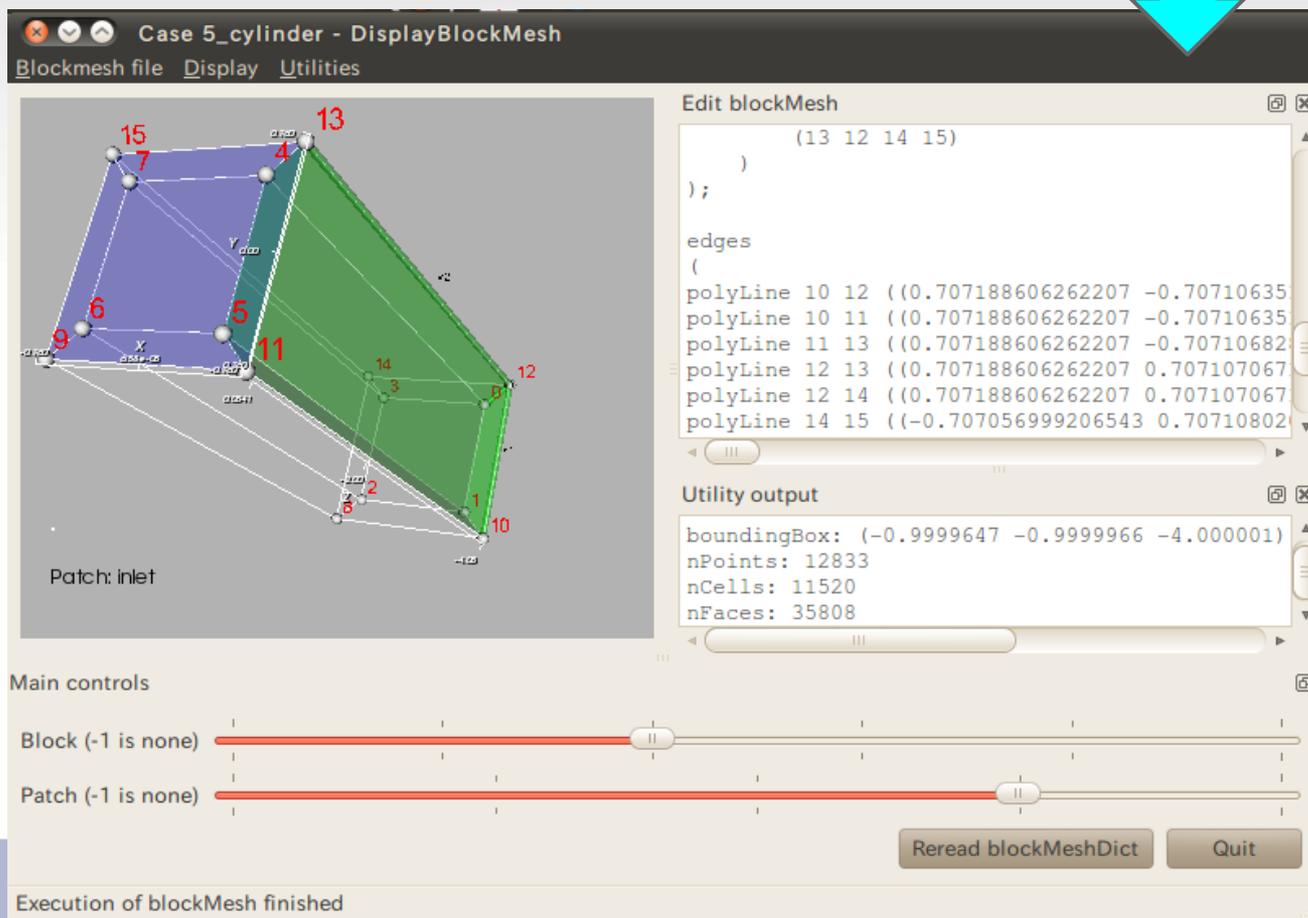
## (SwiftBlock)

The image shows a Blender 2.79 interface with a SwiftBlock workflow. The 3D viewport displays a cylinder mesh. The Properties panel on the right shows the SwiftBlock settings, including a 'Write' button and a 'Patch t' dropdown menu. A red box highlights the 'Write' button, and a yellow arrow points to the 'Patch t' dropdown menu. A red box also highlights the 'polyMesh' folder in the File browser, with a blue arrow pointing to it from the 'Write' button.

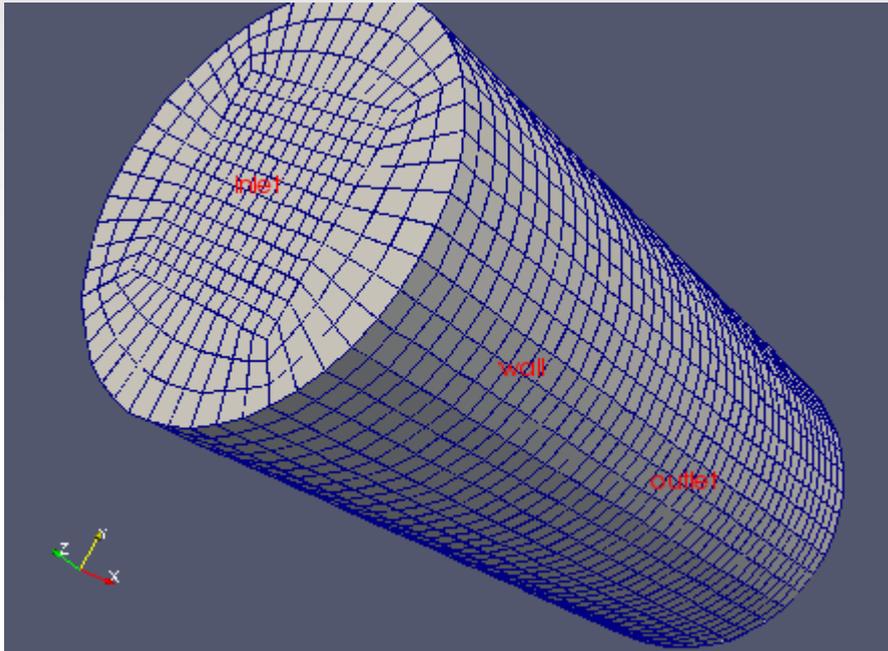
Folder Name	Item Count
5_cylinder	5個のアイテム
0	0個のアイテム
constant	2個のアイテム
polyMesh	0個のアイテム
triSurface	0個のアイテム
system	5個のアイテム
cylinder.blend	434.9 KB
cylinderSwift.blend	408.9 KB

# 実習5

## (blockMeshDictの確認)

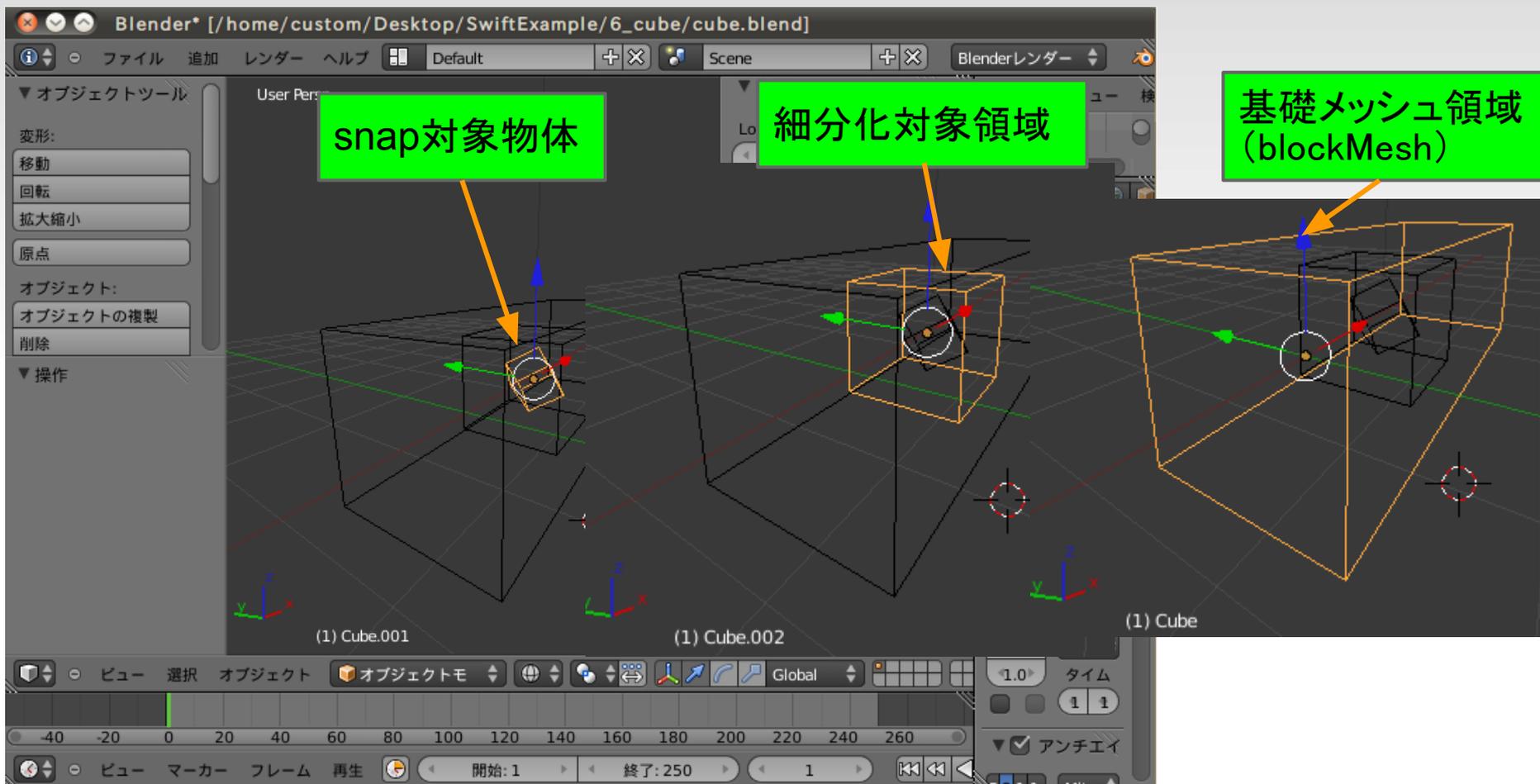


# 実習5 (メッシュ確認例)



nPoints: 12853  
nCells: 11520

# 実習6 (Swiftツール実践)



# 実習6 (SwiftBlock起動)

The image displays two screenshots of the Blender 2.79 interface. The top screenshot shows a 3D view of a cube with a yellow arrow pointing to a red box labeled '1' containing the number '1'. A yellow callout box above it says 'Blockを選択' (Select Block). A second yellow arrow points from the cube to a red box labeled '2' containing the number '2', which highlights the 'SwiftBlock' icon in the top toolbar. The bottom screenshot shows the 'SwiftBlock settings' panel with a red box labeled '3' containing the number '3' highlighting the 'Enable SwiftBlock' button. A yellow callout box below it says '面編集モード' (Face Edit Mode). A red box labeled '4' containing the number '4' highlights the 'Face Edit Mode' icon in the bottom toolbar.

1 Blockを選択

2

3

4

面編集モード

# 実習6 (SwiftBlock設定)

1 流入面を選択

2 選択面の名前入力  
Patch種別を選択

3 ボタンを押す

4 設定が反映

5 設定の確認

Zキー

Blender\* [/home/custom/Desktop/SwiftExample/6\_cube/cube.blend]

Scene

Blenderレンダー

変形

Median:

X: -1.000

Y: 0.000

Z: 0.000

Global Local

Mean Crease: 0.000

Mean Bevel Wei: 0.000

グリセペンシル

新規

New Layer

Delete Frame Convert

ビュー

レンズ: 35.000

オブジェクトにロック

視点をカーソルに合わせ

Patch settings

Patch n defaultName

Patch t wall

Set name

defaultName wall

inlet patch

SwiftBlock settings

Write

convertToMeters: 1.00

resolution: 1.00

Set edges

defa

SwiftSna

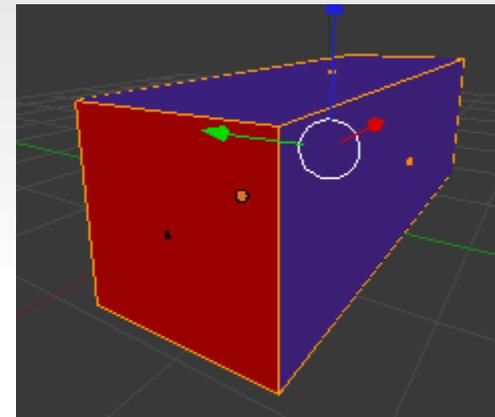
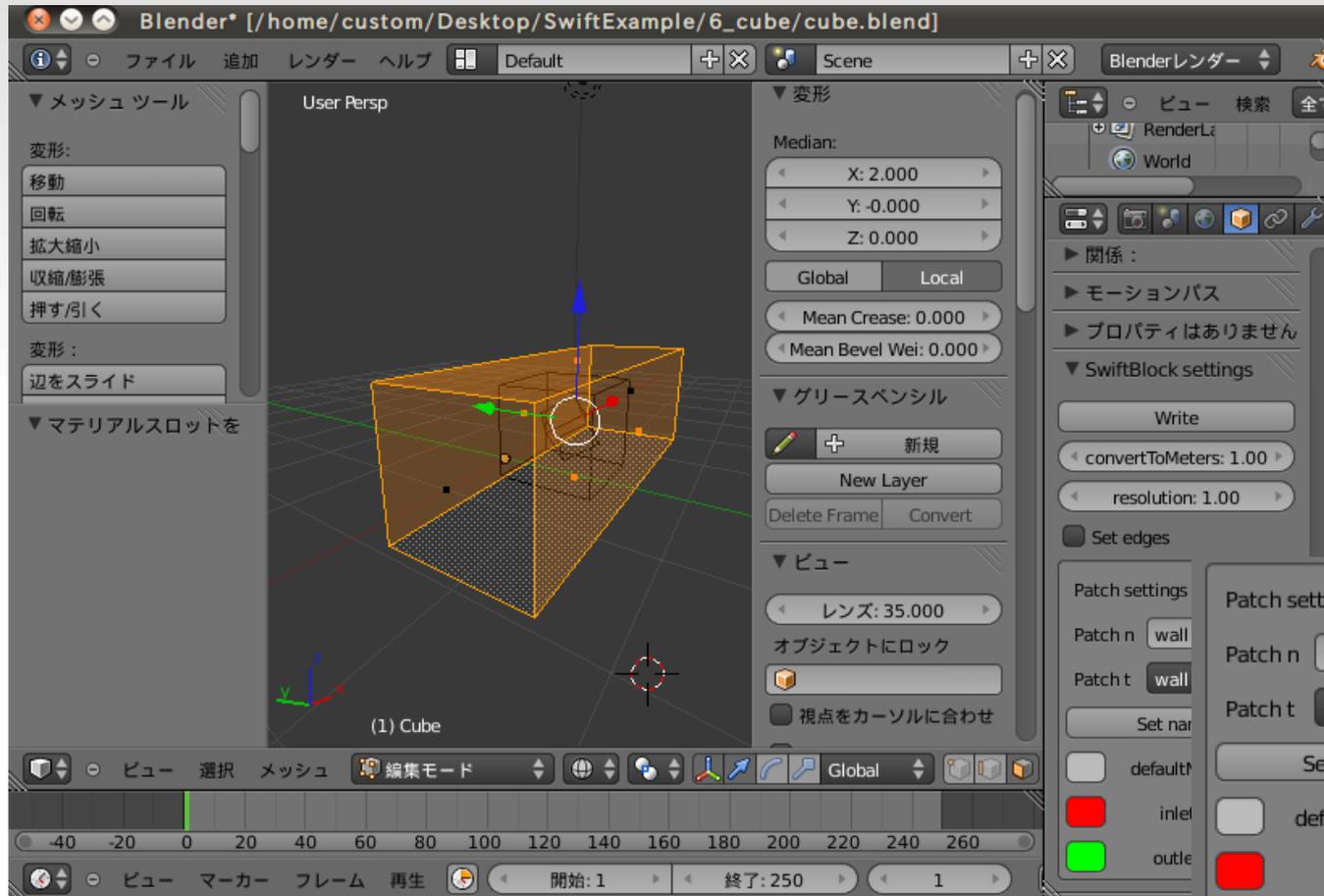
Enable

ビュー 選択 メッシュ 編集モード

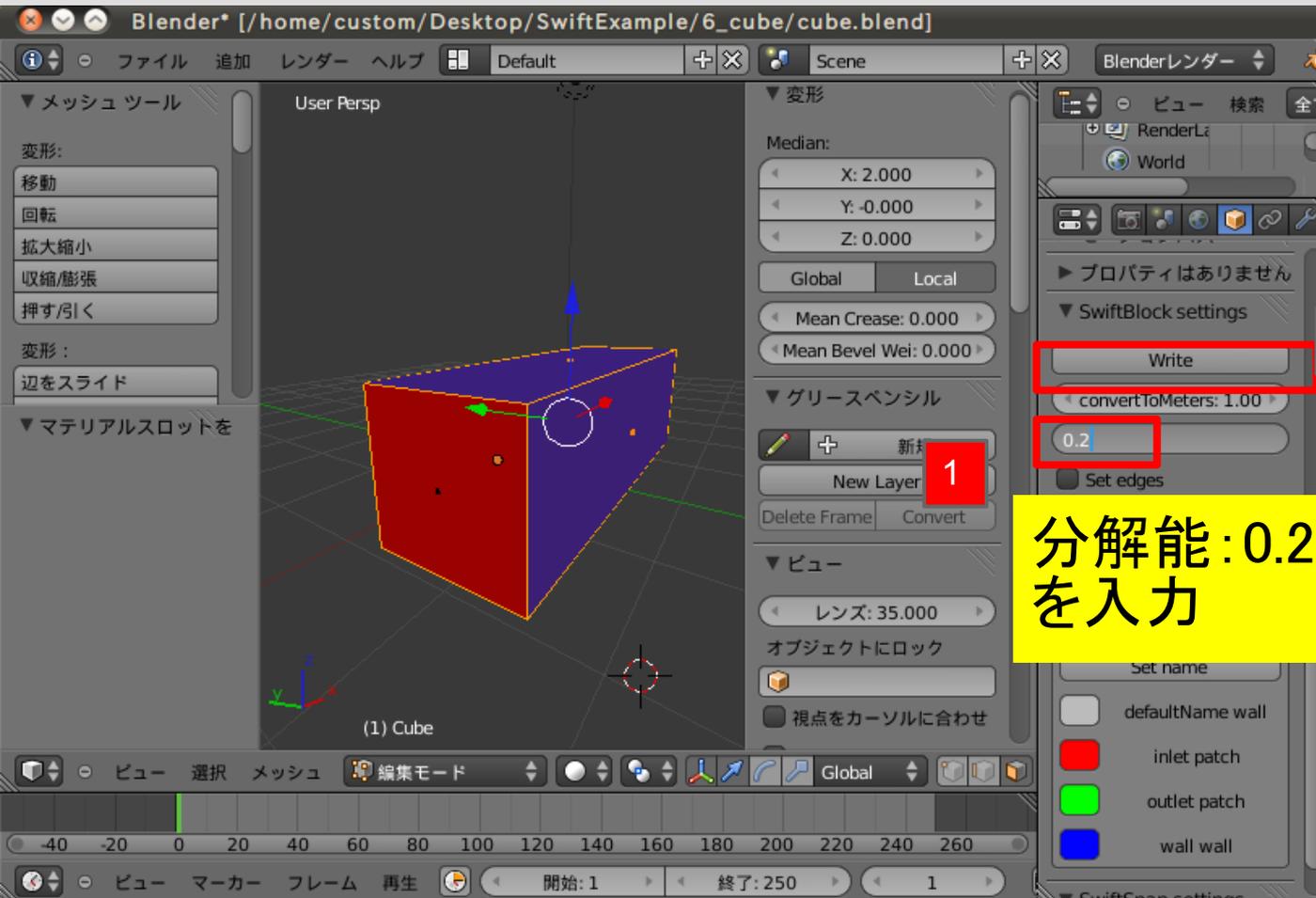
Global

開始: 1 終了: 250 1

# 実習6 (SwiftBlock設定完了)



# 実習6 (SwiftBlock出力)



6_cube	4個の
0	0個の
constant	2個の
polyMesh	0個の
triSurface	0個の
system	5個の
triSurface	0個の
controlDict	
decomposePar...	
fvSchemes	
fvSolution	
cube.blend	43

# 実習6

## (BlockMesh確認)

SwiftBlock settings

Write

-	6_cube	4個のアイテム
+	0	0個のアイテム
-	constant	2個のアイテム
+	polyMesh	0個のアイテム
+	triSurface	0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
	decomposePar...	1.4 KB
	fvSchemes	1.6 KB
	fvSolution	1.2 KB
	cube.blend	439.8 KB

6\_cube

0

constant

polyMesh

blockMeshDict

facesFound.obj

log.swiftblock

openInParaview.block

triSurface

system

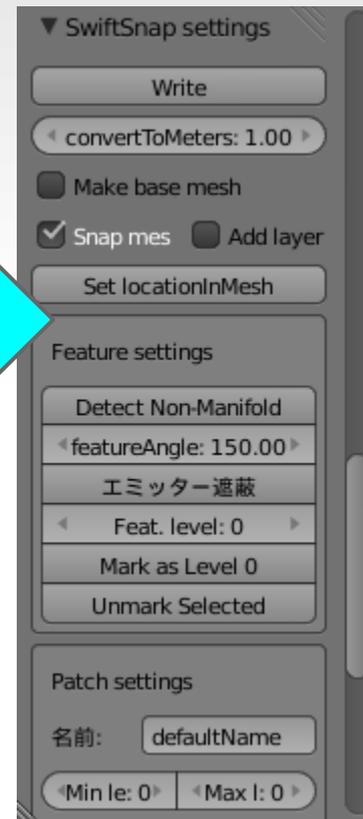
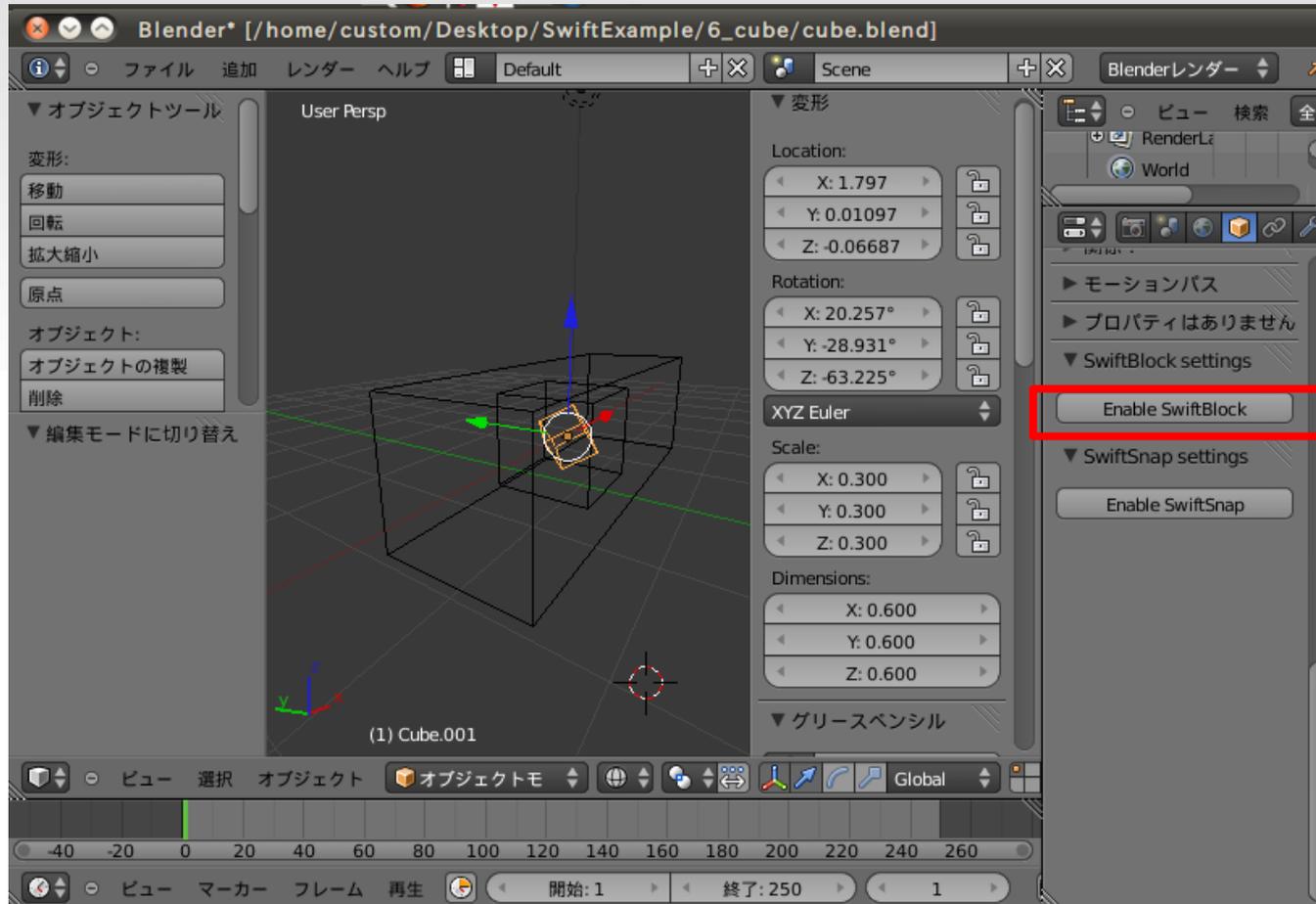
triSurface

controlDict

decomposeParDict

```
blockMeshDict ✕
vertices
(
  (5.0 0.9999999403953552 -1.0)
  (5.0 -1.0 -1.0)
  (-1.0000001192092896 -0.9999998211860657 -1.0)
  (-0.9999996423721313 1.0000003576278687 -1.0)
  (5.000000953674316 0.999999463558197 1.0)
  (4.999999523162842 -1.0000005960464478 1.0)
  (-1.0000003576278687 -0.9999996423721313 1.0)
  (-0.9999999403953552 1.0 1.0)
);
blocks
(
  hex (4 5 6 7 0 1 2 3) (10 30 10) simpleGrading
);
patches
(
  patch inlet
  (
    (2 6 7 3)
  )
  patch outlet
  (
    (0 4 5 1)
  )
)
```

# 実習6 (SwiftSnap起動)



# 実習6 (フィーチャーエッジの設定)

1

2

1

2

2

パラメタ  
設定された

Feature settings

Detect Non-Manifold

featureAngle: 150.00

エミッター遮蔽

Feat. level: 0

Mark as Level 0

Unmark Selected

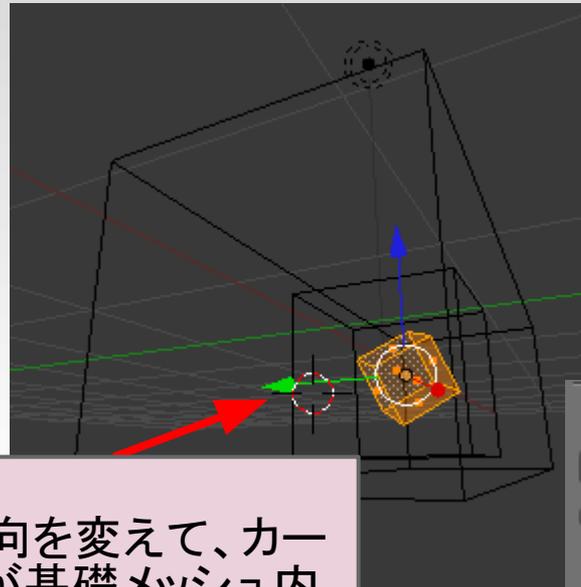
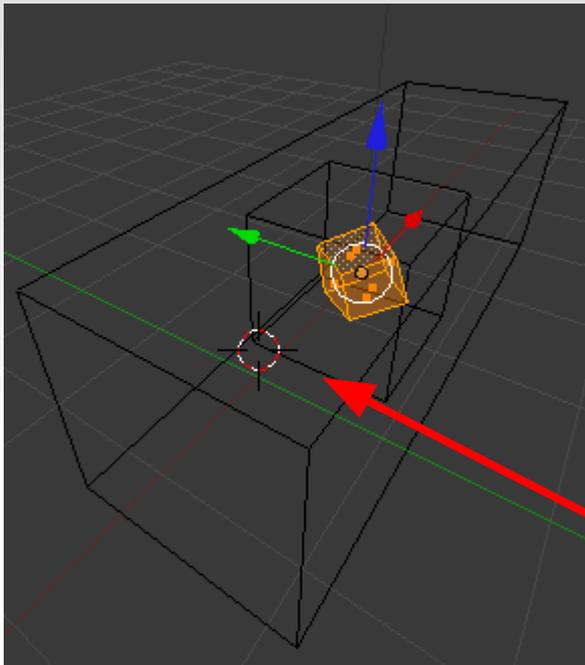
Defined features, click to s  
0

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

58

# 実習6 (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

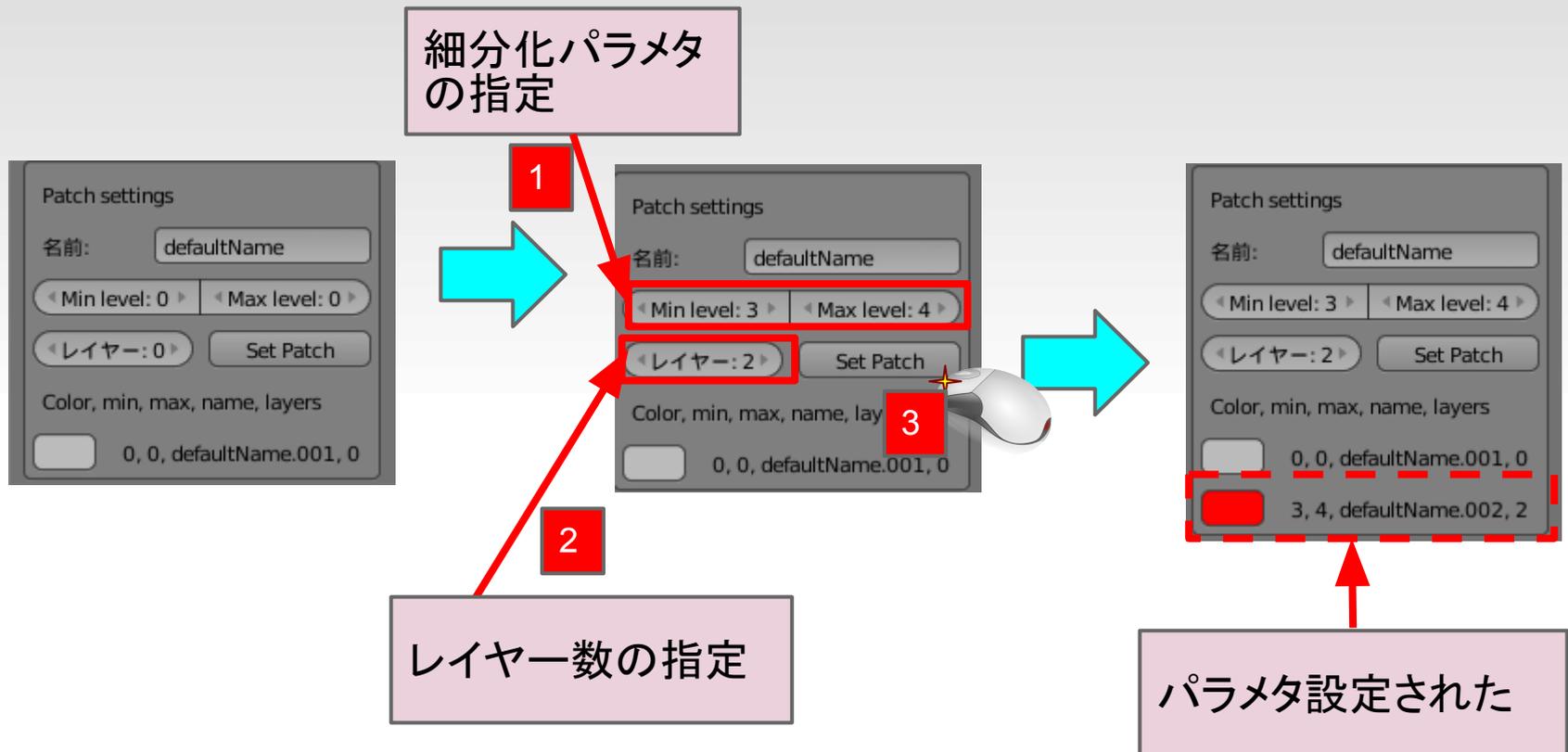
Patch settings

名前: defaultName

Min lev: 0 Max lev: 0

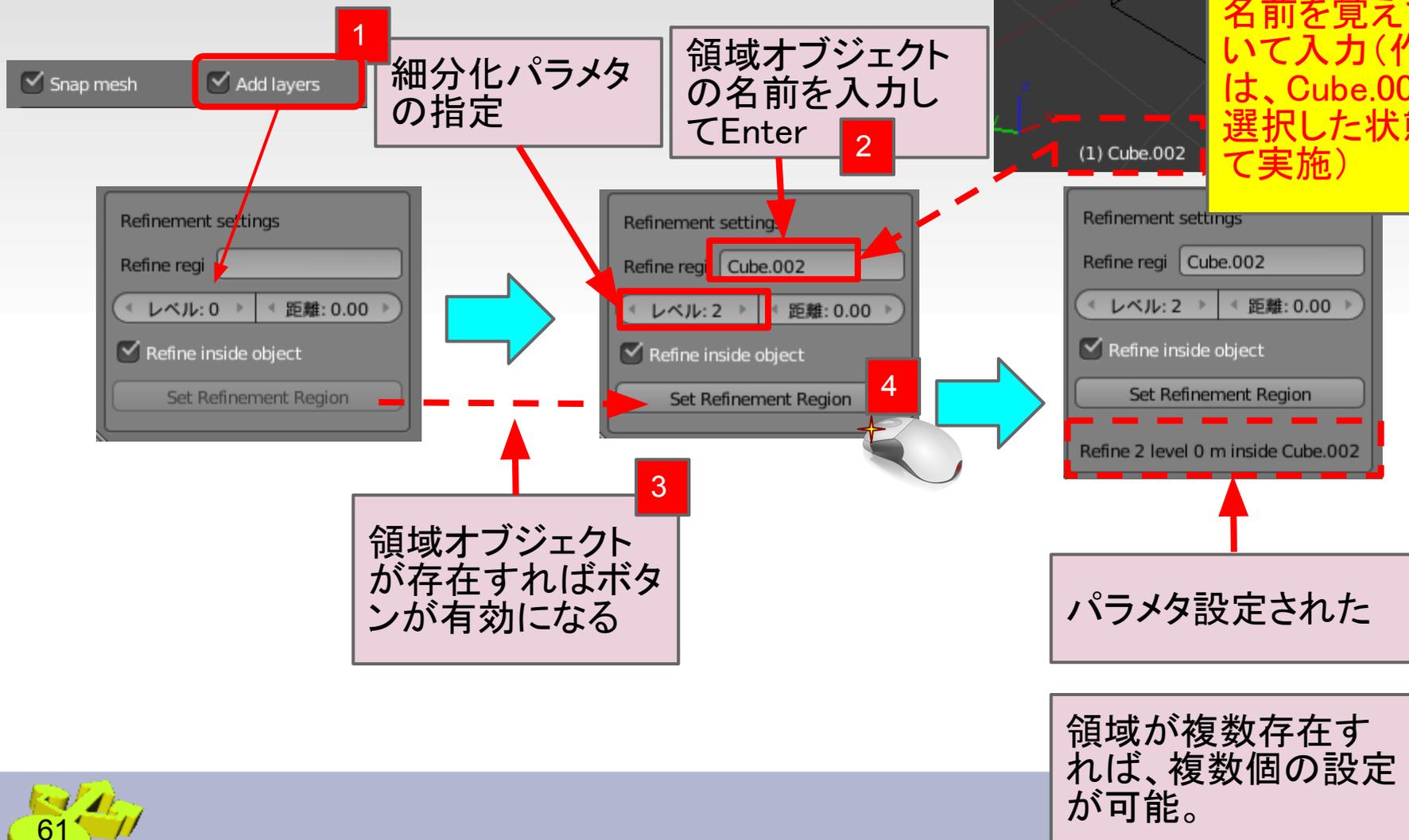
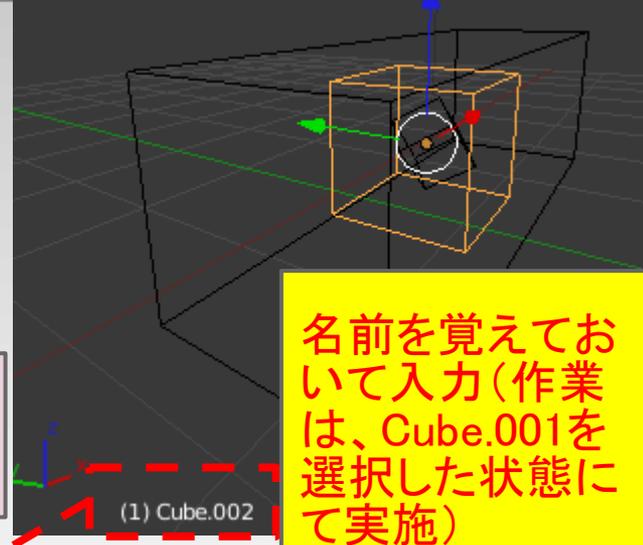


# 実習6 (表面細分化指定)

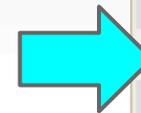
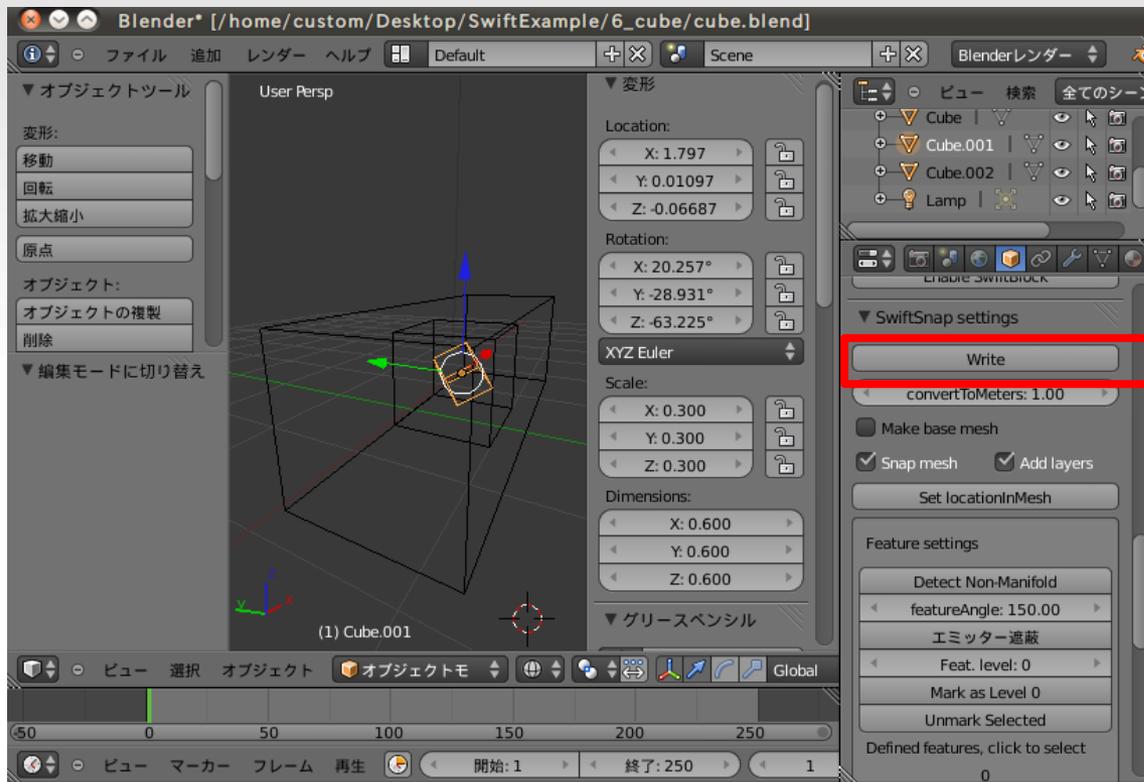


# 実習6

## (領域細分化の指定)



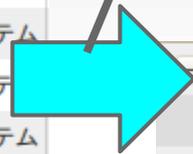
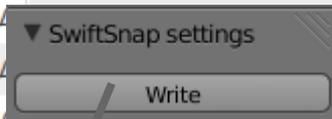
# 実習6 (SwiftSnap書き出し)



-	6_cube	4個のアイテム
+	0	0個のアイテム
-	constant	2個のアイテム
+	polyMesh	4個のアイテム
+	triSurface	0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
	decomposePar...	1.4 KB
	fvSchemes	1.6 KB
	fvSolution	1.2 KB
	cube.blend	439.8 KB

# 実習6 (snappyHexMeshDict確認)

-	6_cube	4個のアイテム
+	0	0個のアイテム
-	constant	2個のアイテム
+	polyMesh	4個のアイテム
+	triSurface	0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
	decomposePar...	1.4 KB
	fvSchemes	1.6 KB
	fvSolution	1.2 KB
	cube.blend	439.8 KB

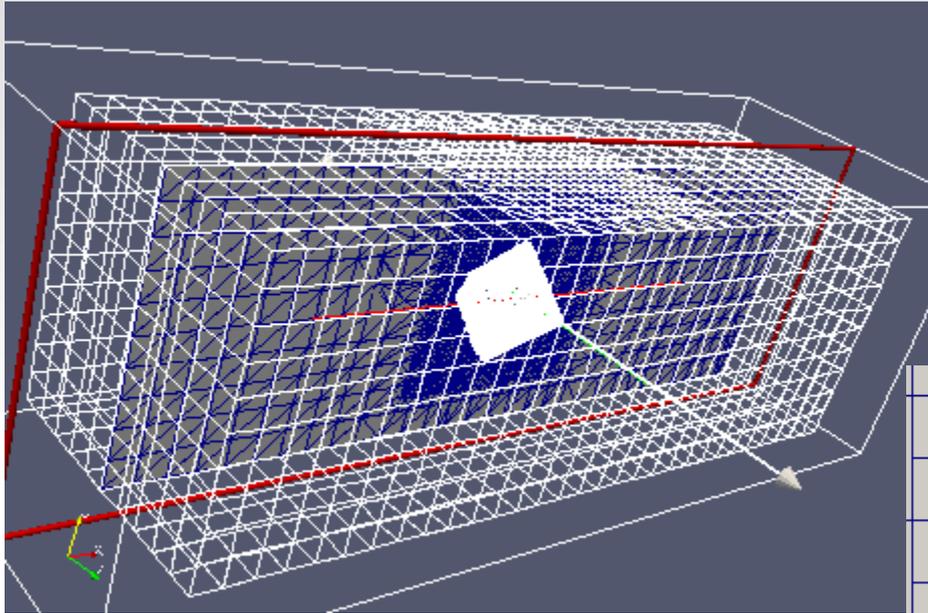


-	system
-	triSurface
	Cube.002.stl
	defaultName.002.stl
	level0.eMesh
	level0.obj
	controlDict
	decomposeParDict
	fvSchemes
	fvSolution
	snappyHexMeshDict

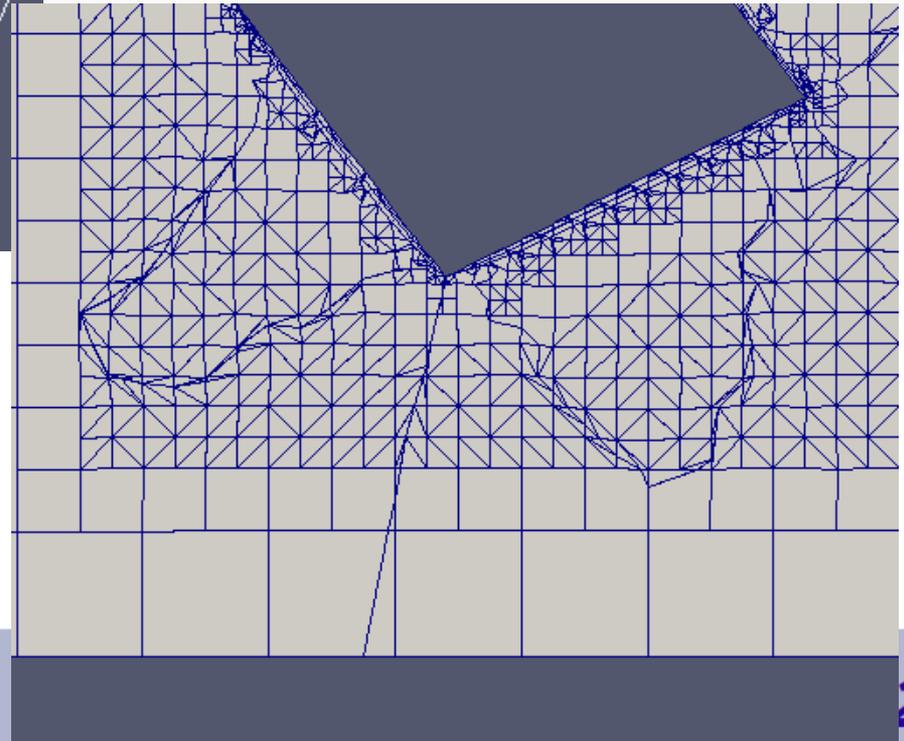
生成されたファイル

```
snappyHexMeshDict ✕  
  
// **** *  
castellatedMesh true;  
snap true;  
addLayers true;  
  
geometry  
{  
    defaultName.002.stl  
    {  
        type triSurfaceMesh;  
        name defaultName.002;  
    }  
    Cube.002.stl  
    {  
        type triSurfaceMesh;  
        name Cube.002;  
    }  
};  
  
castellatedMeshControls  
{  
    features  
    (  
        {  
            file "level0.eMesh";  
        }  
    )  
};
```

# 実習6 (メッシュ確認例)



nPoints:67248  
nCells:50965



# まとめ

- OpenFOAM用のGUIメッシュャとして、Swiftツールの機能を確認・実習した。
- Swiftツールは公開されたばかりの発展途上ツールで課題もあるが、致命的問題ではなさそう。
  - ファイル出力先の問題
  - 設定変更、やり直しの方法
  - edgeGradingには非対応 (SwiftBlock)
  - 曲線内挿はpolyLineのみ (SwiftBlock)
- フリーのGUIメッシュャとして期待は大。(次期DEXCS for OpenFOAM(R) には搭載予定)

# 補足

- Blender(3DView)・・・操作を間違えたらUndo (Ctrl+Z)
- 本資料の電子版は、後日アップロード予定
  - オープンCAE学会
  - <http://mogura7.zenno.info/~et/wordpress/ocse>
- ライブDVD⇒HDDへインストール、仮想マシンで使う方法
  - **DEXCS2011 for OpenFOAM®** リリースノート **+**
- DEXCS十徳ナイフ: **使用上の注意、改良版**
  - <http://mogura7.zenno.info/~et/wordpress/ocse/?p=481>
- SwiftBlock作成演習中級は、オープンCAE初心者勉強会の夏合宿(8/18, 19@岐阜)にて実施予定