

DEXCS2011 for OpenFOAMについて

- DEXCS for OpenFOAMとは
- DEXCS2011の開発方針
- DEXCS2011の新機能概要
- 推奨利用方法
- 注意事項

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、64bit版もあります。

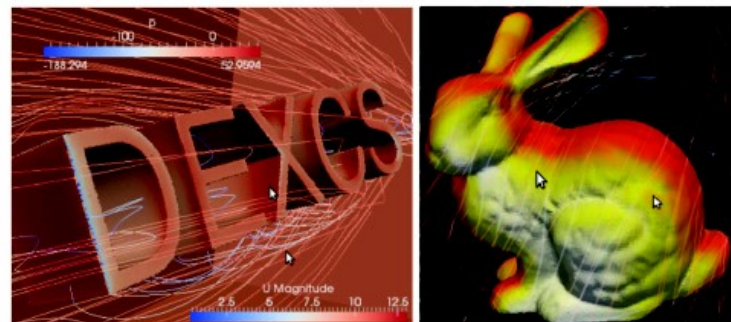
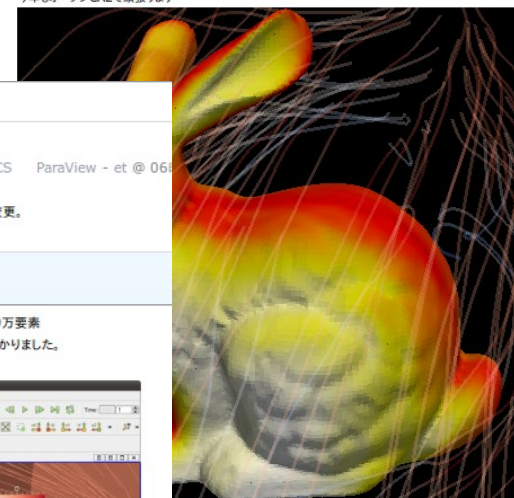


図4 DEXCS for OpenFOAMの解析例

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)

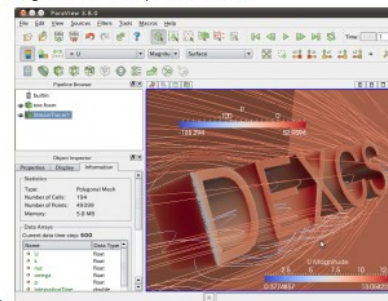
新マシンでは実用計算も十分可能

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

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

弾塑性、大変形

構造解析 Adventure
線形弾性

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)



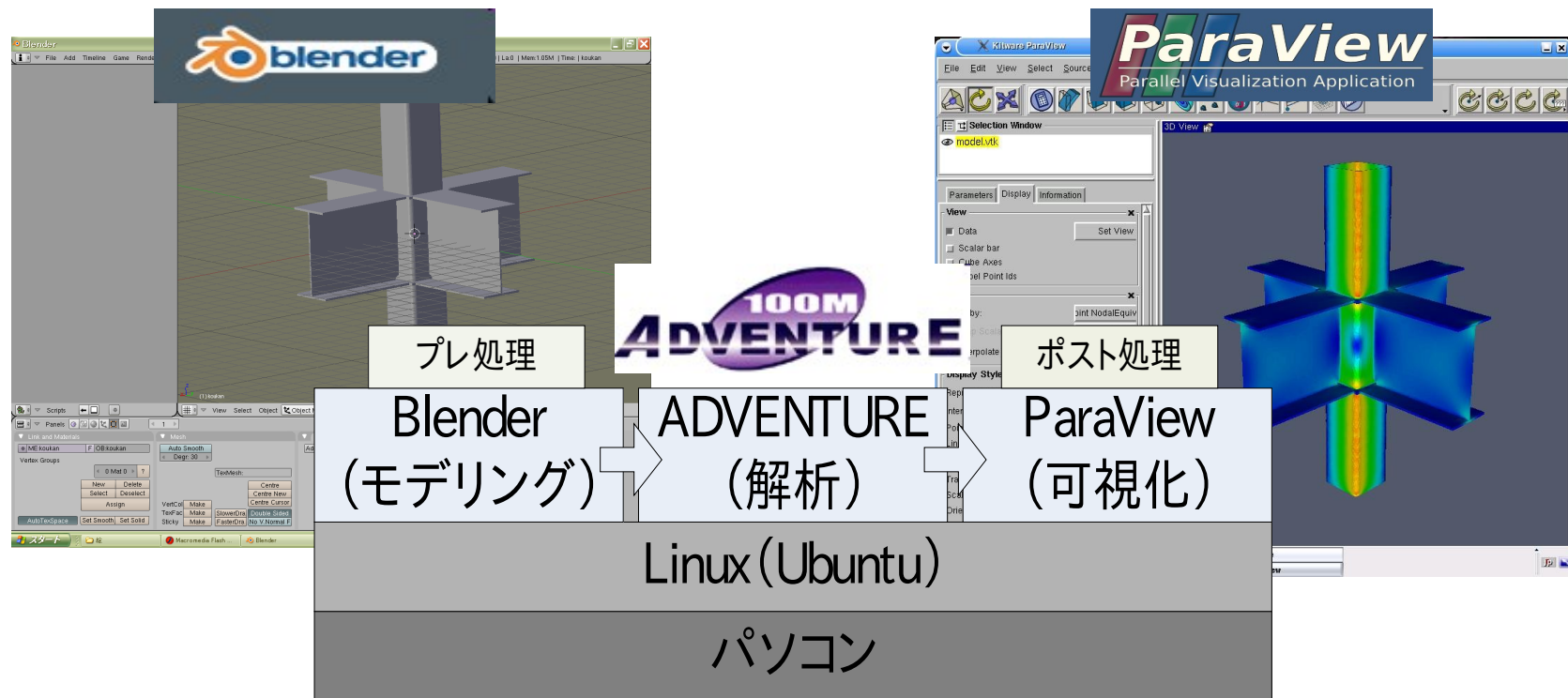
拡張性を持つ設計支援用解析システム

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

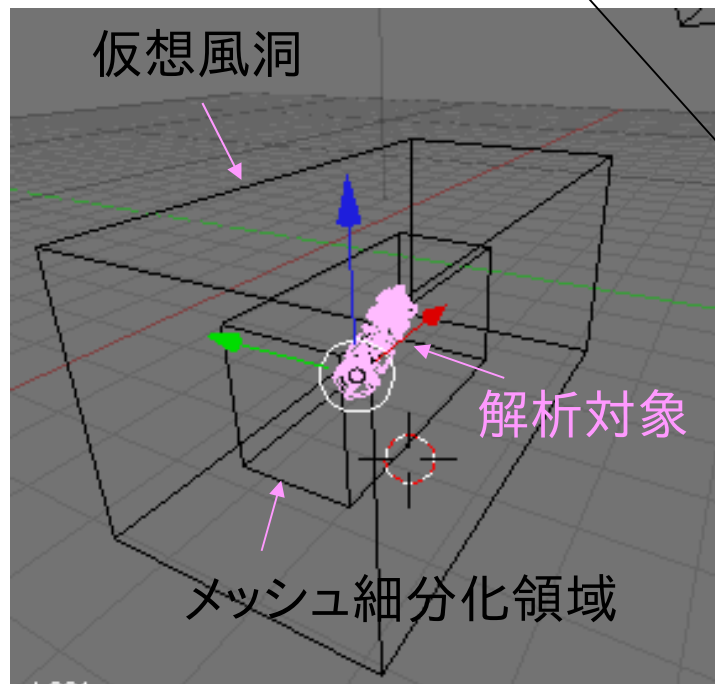


DEXCS2009-OpenFOAMの嬉しさ

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



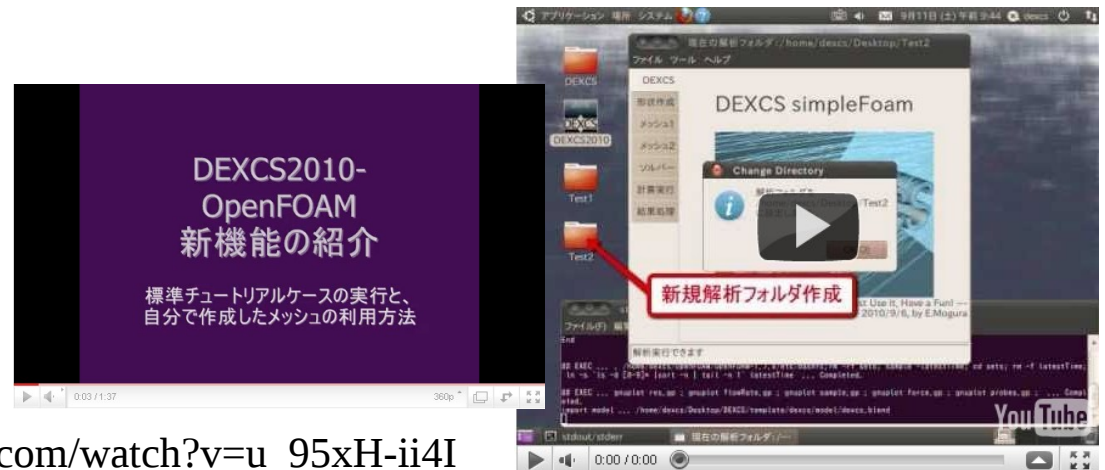
コマンドライン入力は一切不要
GUIエディタでパラメタ変更可能

(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)

DEXCS2010-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、結果処理サンプルの充実



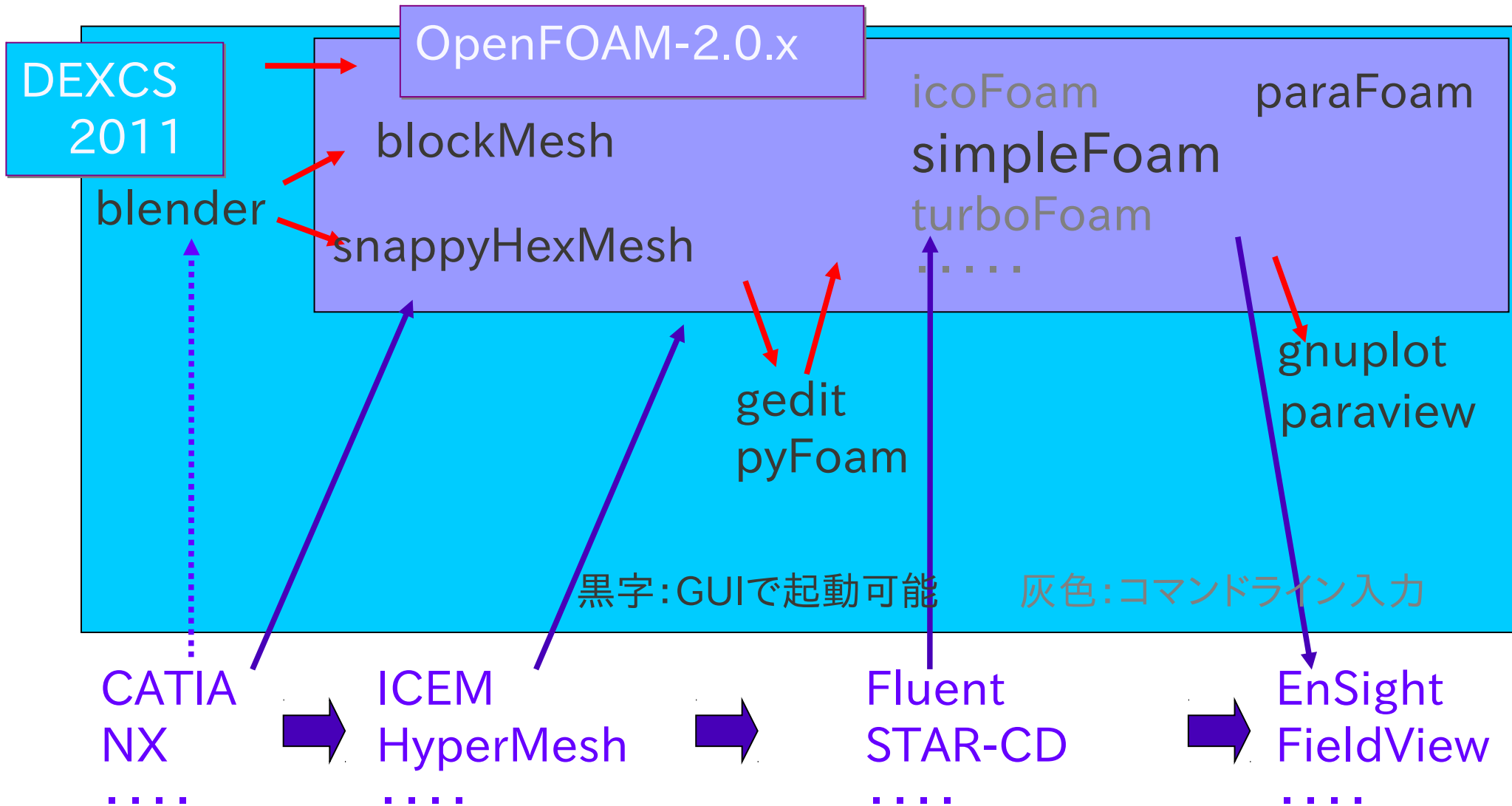
新機能

http://www.youtube.com/watch?v=u_95xH-ii4I

- OpenFOAM 標準チュートリアルのGUI操作
- 上記チュートリアルベースの簡単メッシュ変更

解析の流れ

形状作成 メッシュ作成 境界条件 計算実行 後処理



DEXCS2010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 の開発方針

課題

すぐに使える, 手軽, 可搬性
「DVD起動も可能」とうたいつつ……
仮想マシン (VMPlayer, VirtualBox) の構築は面倒臭い

DEXCS2009⇒2010

シンプル化した部分 と そうでない部分

DEXCSの解析フォルダ ⇒ OpenFOAMのcaseフォルダでの作業

課題と2011の方針

すぐに使える, 手軽, 可搬性
「DVD起動も可能」とうたいつつ……
仮想マシン (VMPlayer, VirtualBox) の構築は面倒臭い

ライブDVDでの
動作を前提とした
テスト・操作説明

DEXCS2009⇒2010

シンプル化した部分とそうでない部分

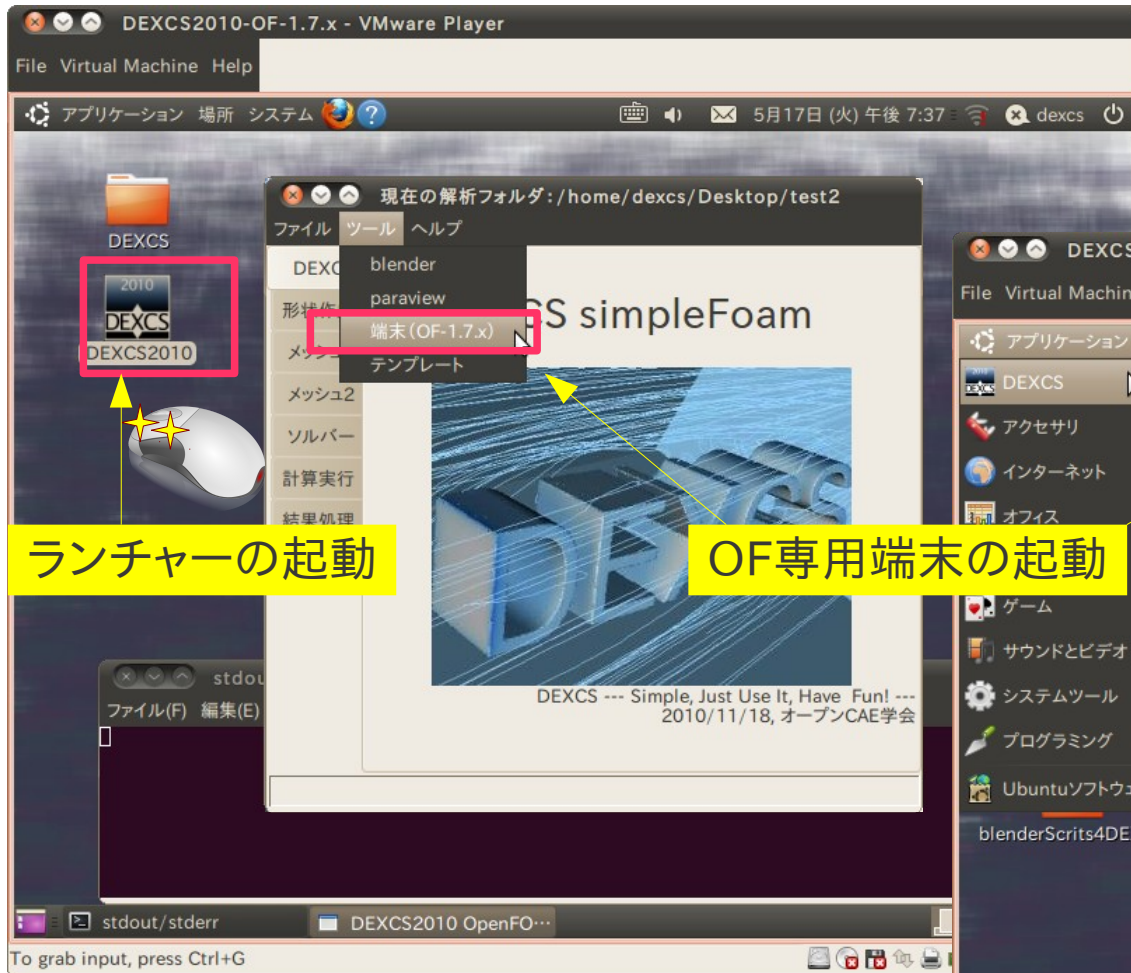
DEXCSの解析フォルダ ⇒ OpenFOAMのcaseフォルダでの作業

DEXCS 十徳ナイフ
(dexcsSWAK)

DEXCS2011の新機能概要

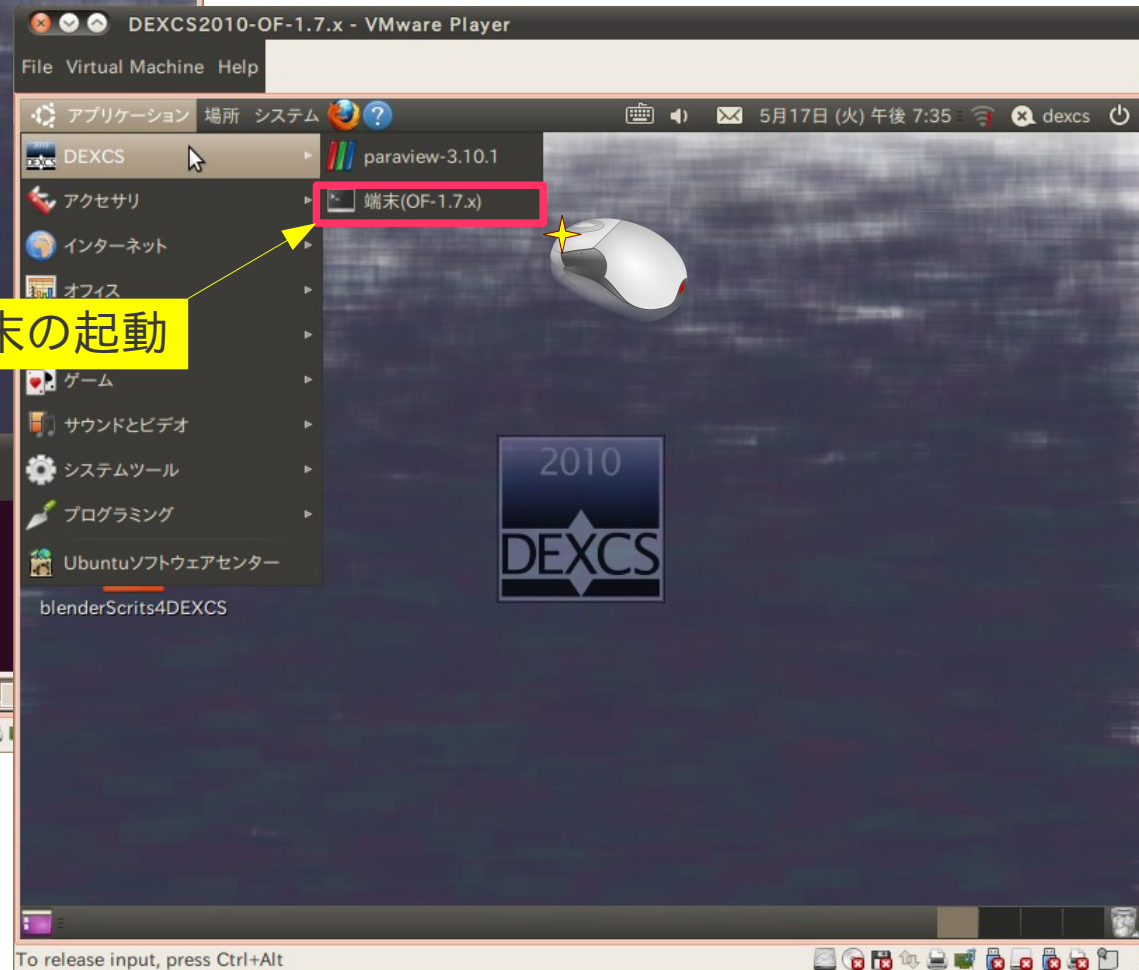
(DEXCS2010 ⇒ 2011の変化点)

DEXCS2010



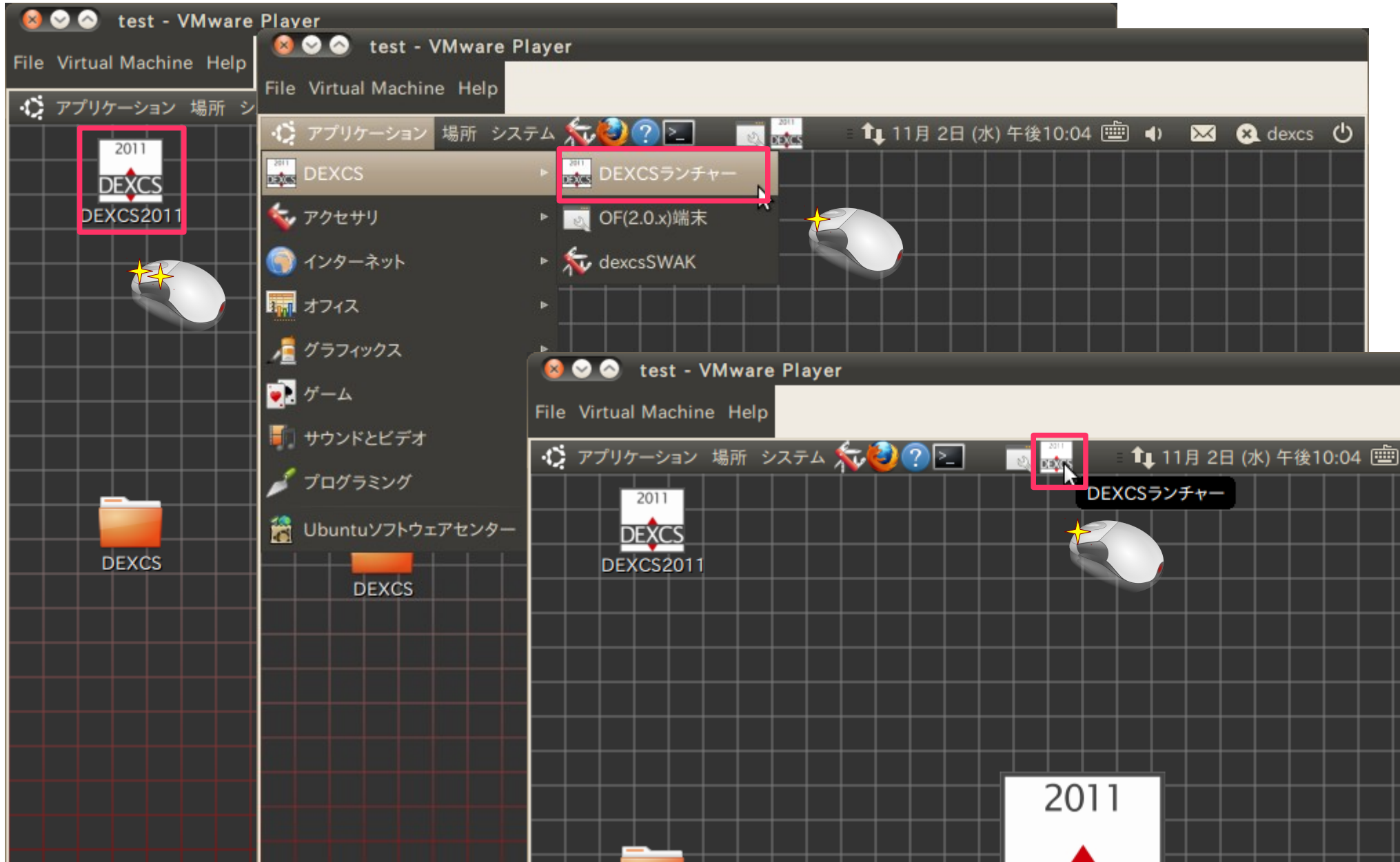
ランチャーの起動

OF専用端末の起動

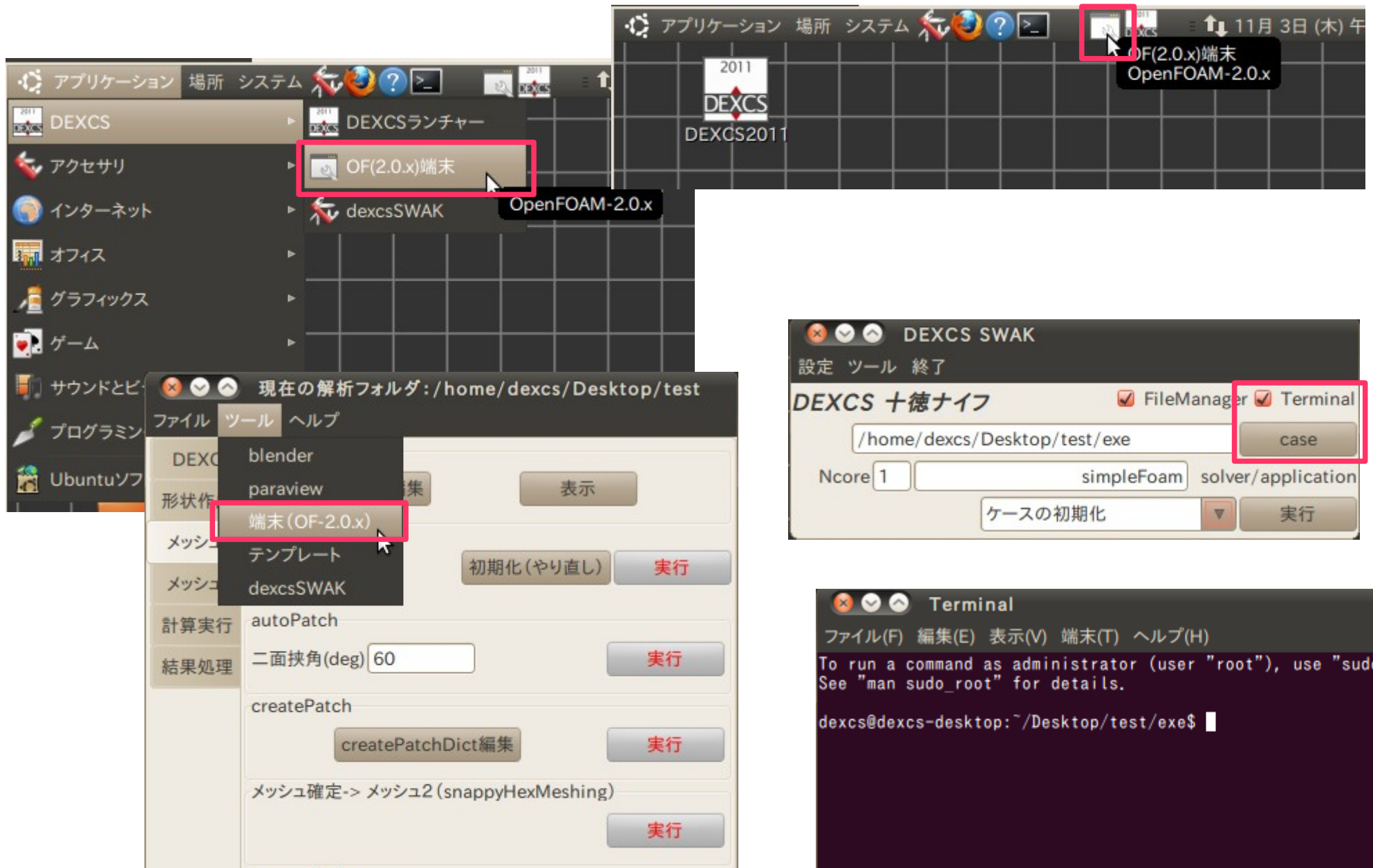


DEXCS2011

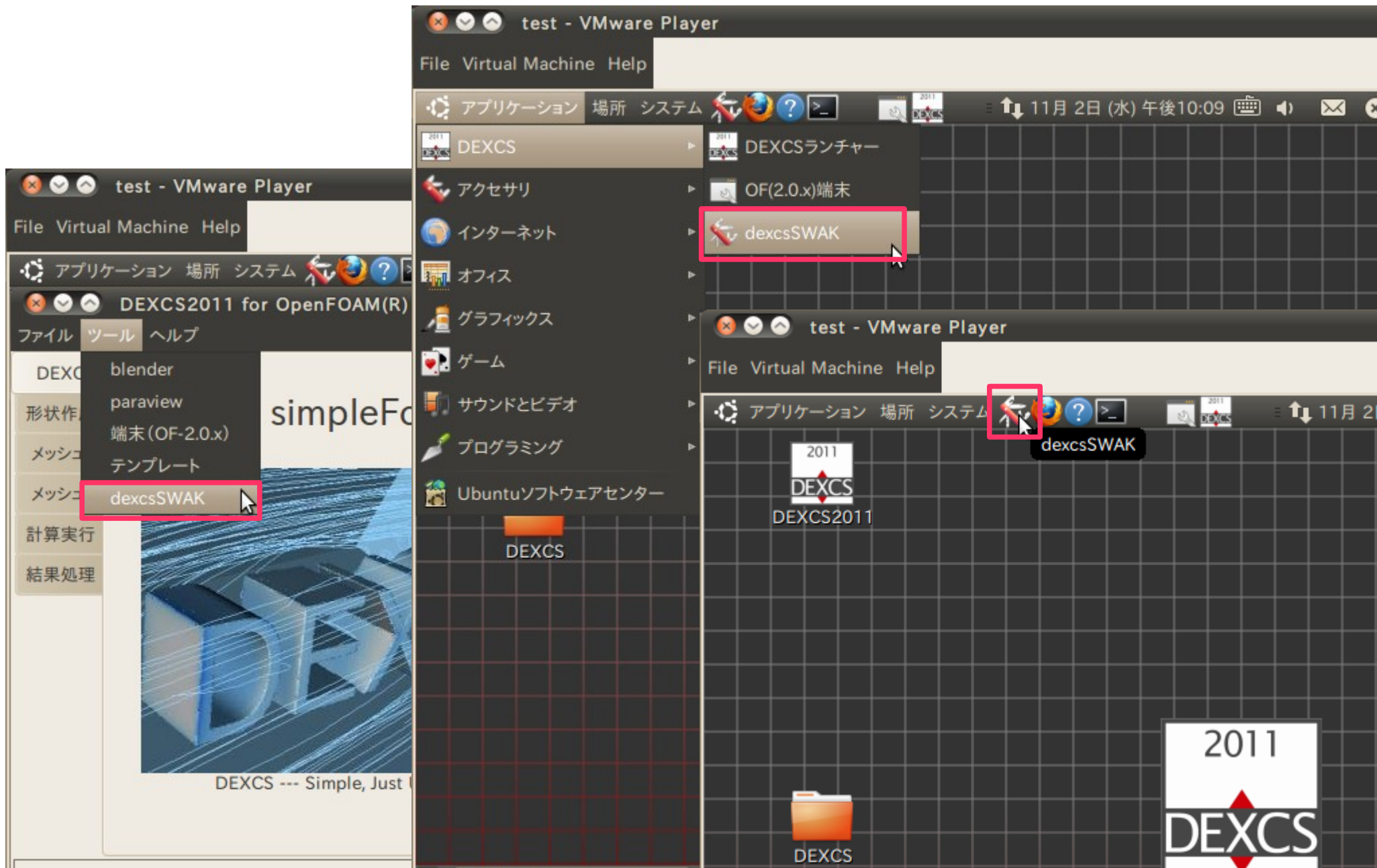
ランチャー起動方法(3通り)



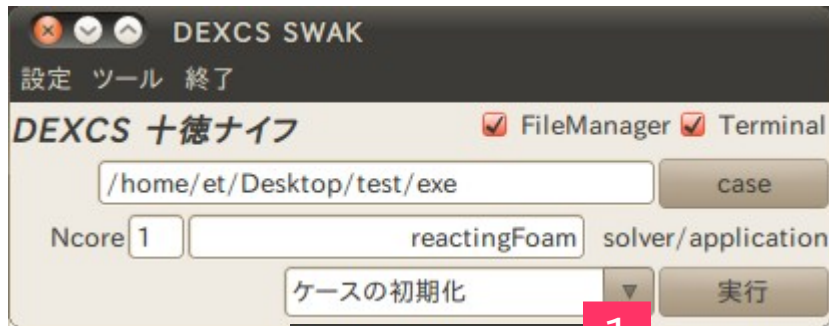
OpenFOAM専用端末の起動方法(4通り)



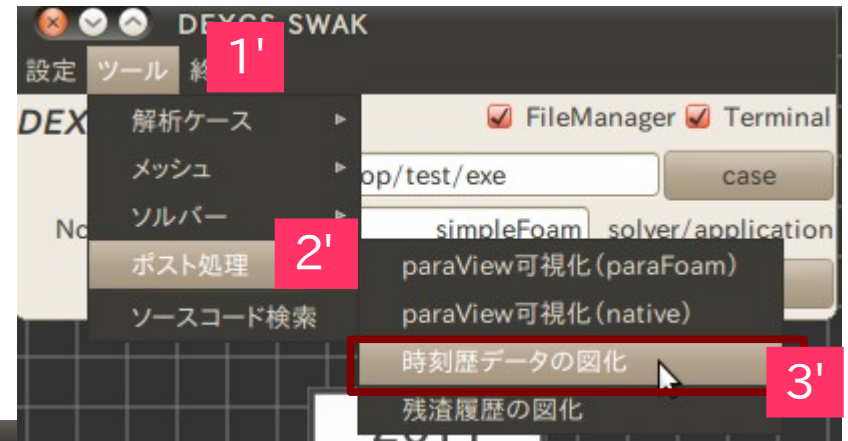
十徳ナイフ起動方法(3通り)



dexcsSWAK(Swiss Army Knife)



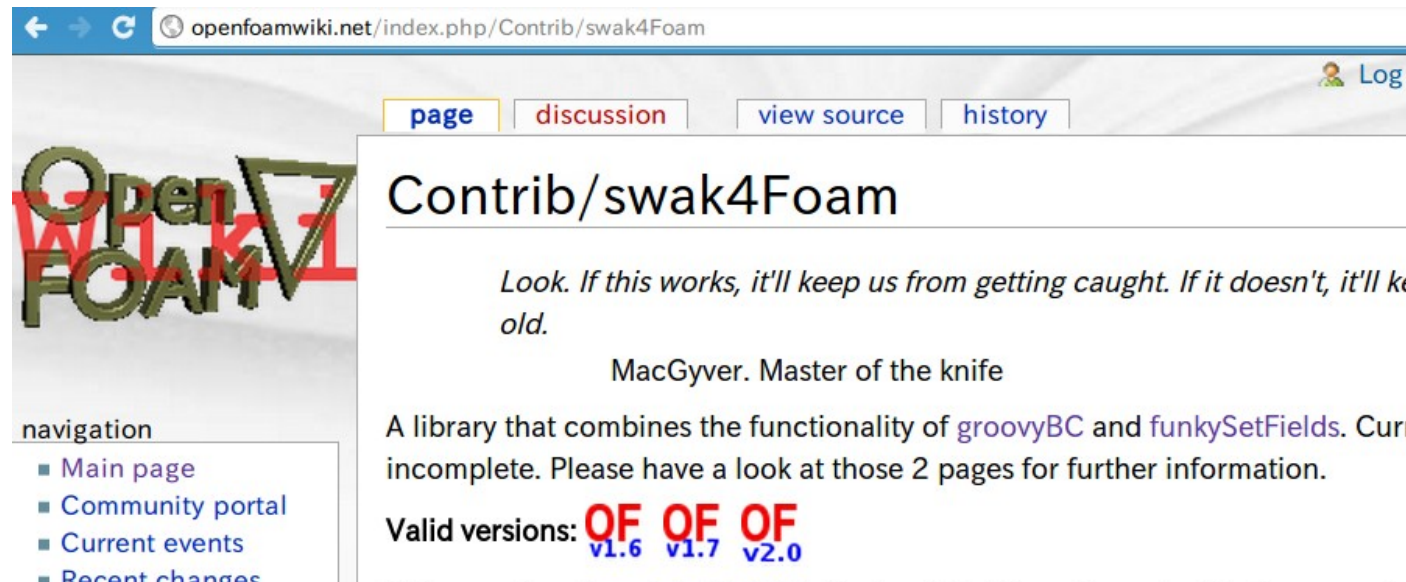
OR



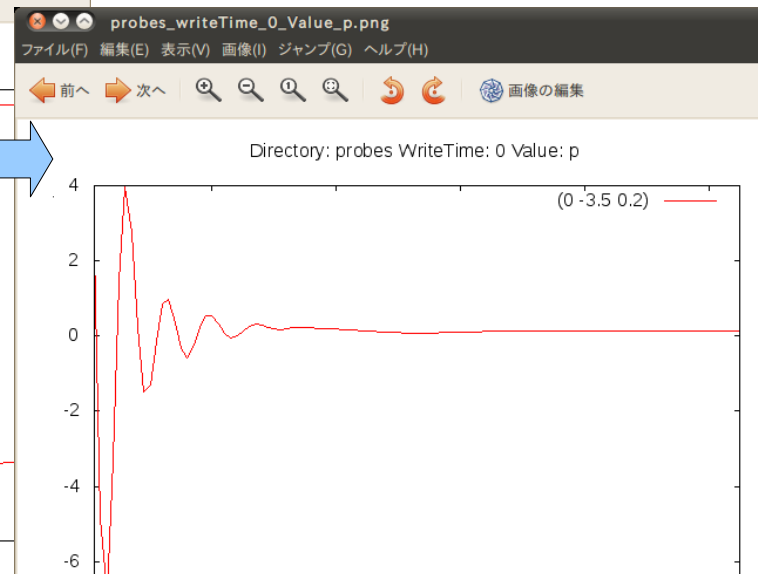
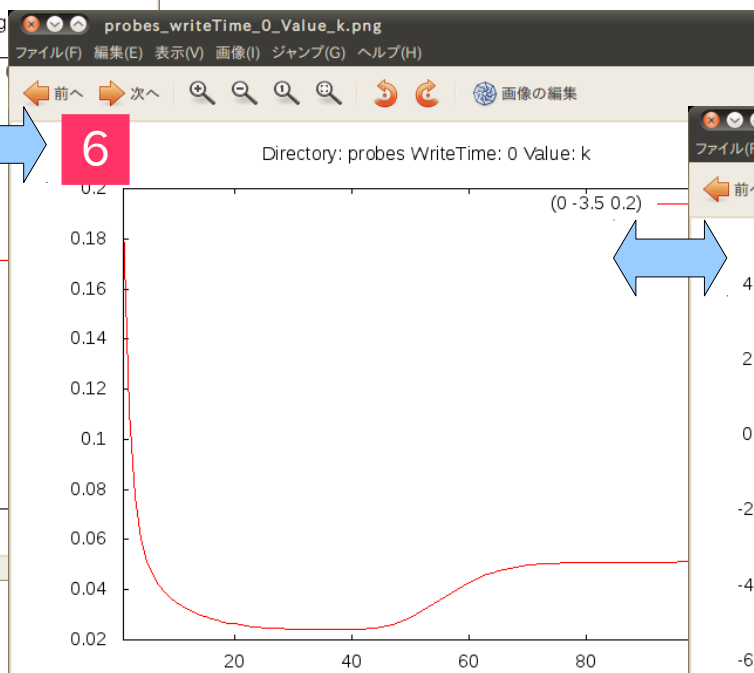
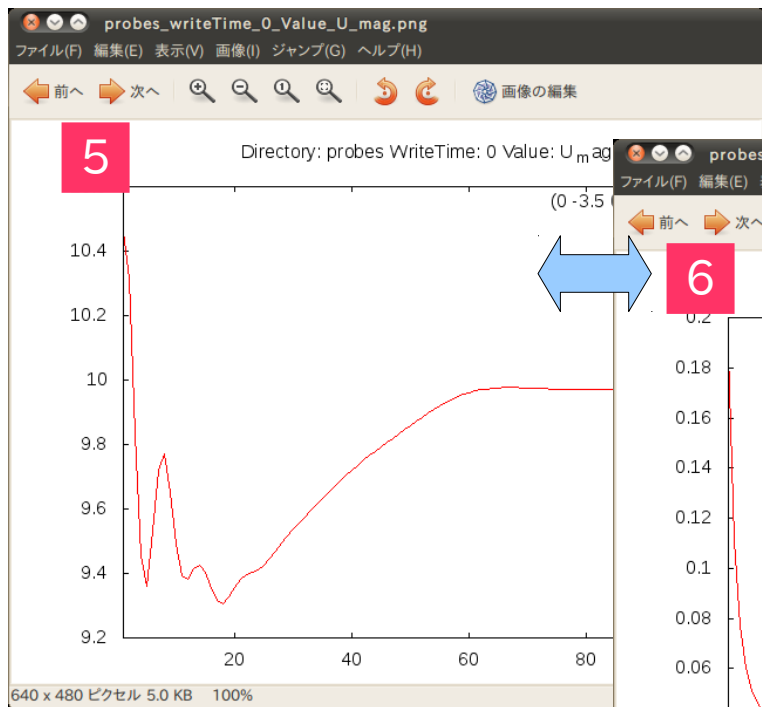
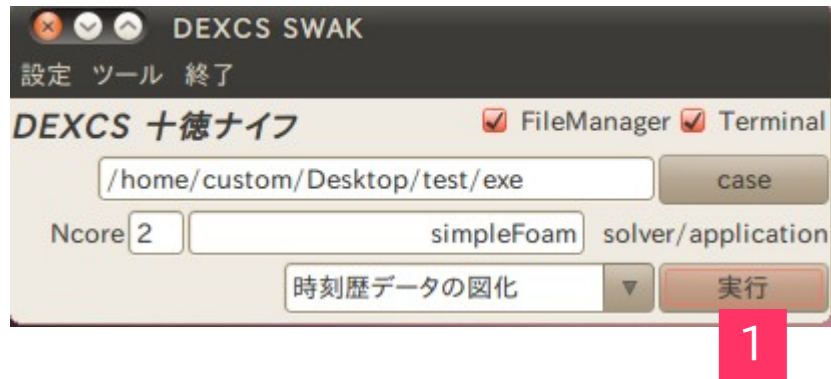
- ケースの初期化
- テンプレートケースの選択・変更
- blockMeshの表示・編集
- blockMeshの実行
- patch名の一括整合
- patch名の一括変更
- featureデータの作成
- snappyHexDictの簡易作成
- snappyHexMeshの実行
- 初期条件の一括参照・変更
- constデータの一括参照・変更
- systemデータの一括参照・変更
- ポテンシャル流れ場作成
- ソルバーの実行
- ソルバーの実行(プロット付)
- ソルバーの停止
- paraView可視化(paraFoam)
- paraView可視化(native)
- 時刻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



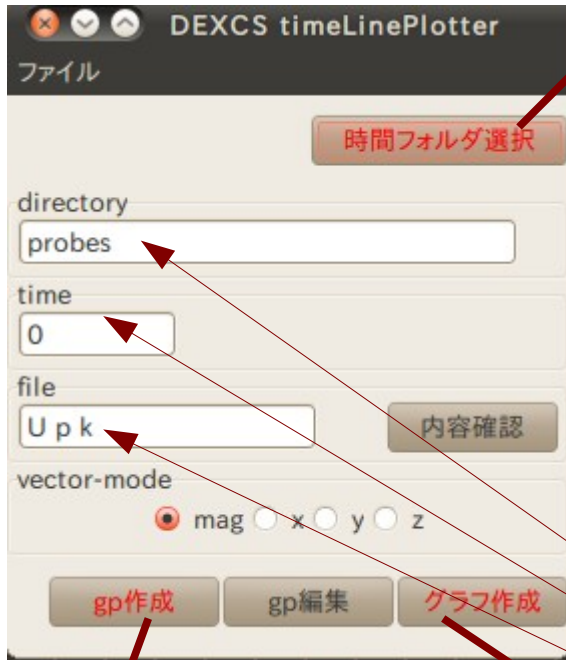
caseフォルダを対象に
pyFoamなど利用した
「便利ツール」の簡単起動



DEXCS timeLinePlotter



仕組み



```
def selectFile(self, event): # wxGlade: MyFrame.<event_handler>
    dlg = wx.DirDialog(self, message=u"プロットしたいファイルの存在する時
    style=wx.DD_DEFAULT_STYLE | wx.DD_NEW_DIR_BUTTON)
    if dlg.ShowModal() == wx.ID_OK:
        setPlotDir = dlg.GetPath()
        dlg.Destroy()
    else:
        dlg.Destroy()
        return False
    if os.path.isdir(setPlotDir):

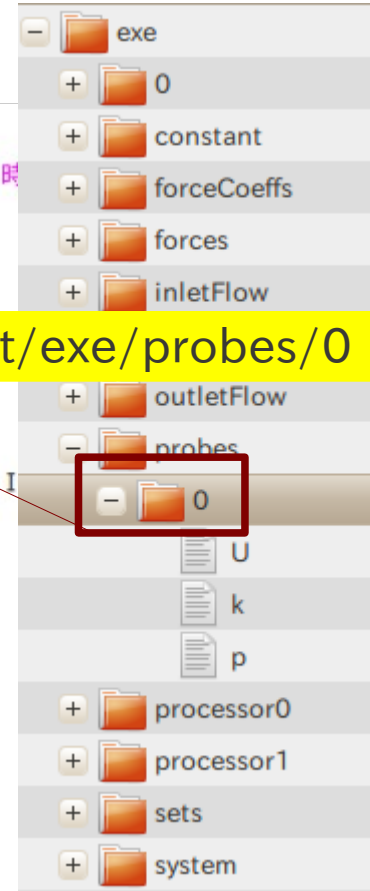
        message= setPlotDir + u" を\n" + u"\nプロットします。"
        dlg = wx.MessageDialog(self, message, 'Set Mesh', wx.OK | wx.I

        setPlotCmd = setPlotDir.split(os.sep)
        #self.gpFile = setPlotCmd[-1]
        self.gpTime = setPlotCmd[-1]
        self.gpDir = setPlotCmd[-2]

        os.chdir(setPlotDir)
        gpFile = glob.glob('*')
        self.gpFiles =
        for i in gpFile:
            self.gpFiles = self.gpFiles + i + " "

        self.caseDir = os.sep.join(setPlotCmd[0:-2])
        print self.gpDir
        self.text_ctrl_1.Value = self.gpDir
        self.text_ctrl_2.Value = self.gpTime
        self.text_ctrl_3.Value = self.gpFiles
        dlg.ShowModal()
        dlg.Destroy()
```

/home/dexcs/Desktop/test/exe/probes/0



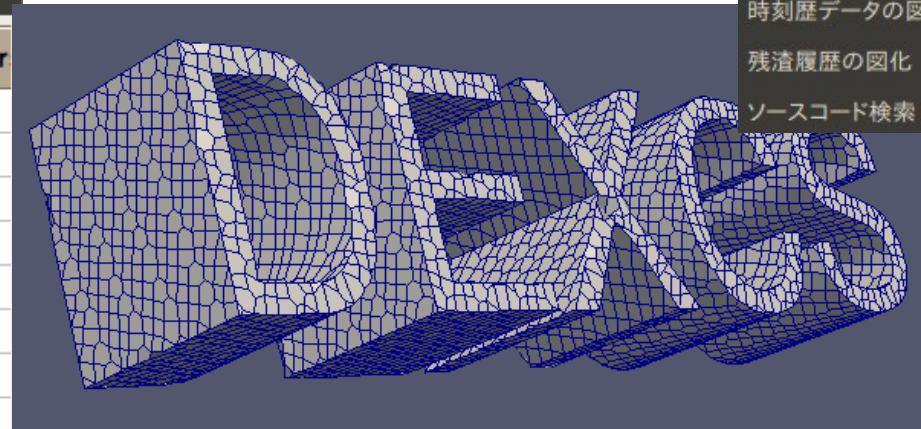
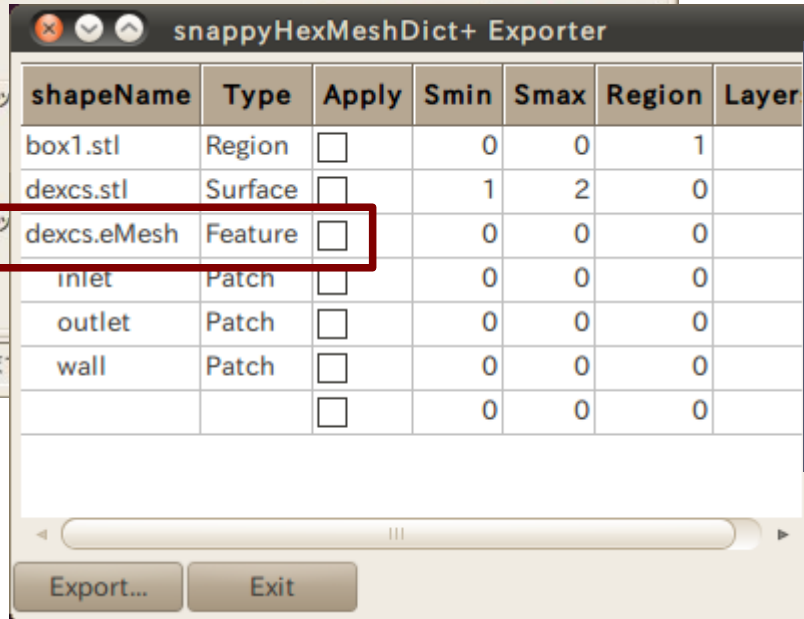
```
def makeGp(self, event): # wxGlade: MyFrame.<event_handler>
    os.chdir(self.postDir)
    title = "makeGp"
    msg = u"gpファイルを作成します"
    cmd = "pyFoamTimeLinePlot.py " + self.caseDir
    cmd += " --directory-name=" + self.gpDir
    cmd += " --basic-mode=lines"
    cmd += " --vector-mode=" + self.vectorMode
    cmd += " --time=" + self.gpTime
    cmd += " --gnuplot-file=" + self.gpDir + ".gp"
    self.runCommand(msg, title, cmd)
```

```
def plotFile(self, event): # wxGlade: MyFrame.<event_handler>
    os.chdir(self.postDir)
    title = "plotFile"
    msg = self.gpDir + u".gp を実行します"
    cmd = "gnuplot " + self.gpDir + ".gp"
    self.runCommand(msg, title, cmd)
    os.system("eog " + self.gpDir + "/*.png")
```

gnuplot, eog

pyFoamTimeLinePlot.py

snappyHexMesh Feature Edge



- 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

DEXCS2009⇒2010

シンプル化した部分と **そうでない部分**



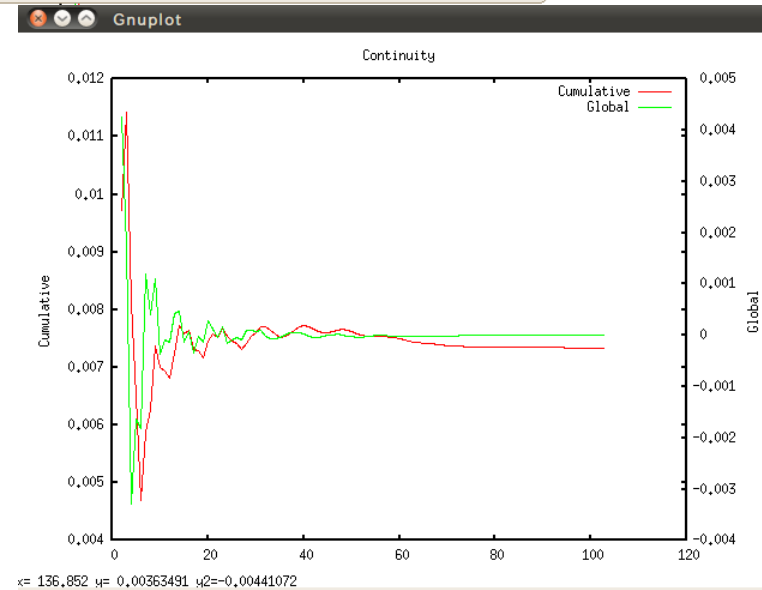
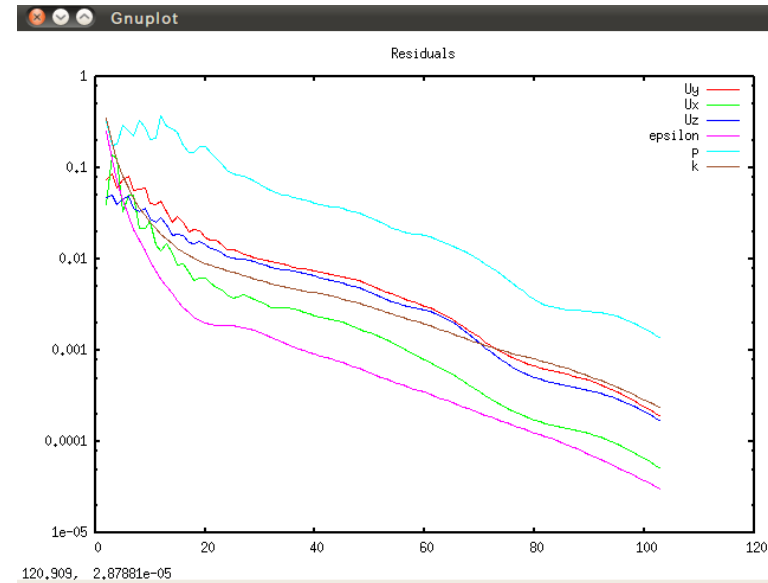
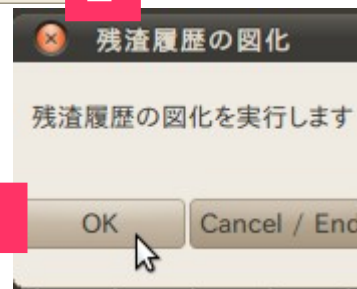
- ケースの初期化
- テンプレートケースの選択・変更**
- blockMeshの表示・編集
- blockMeshの実行
- patch名の一括整合
- patch名の一括変更
- featureデータの作成
- snappyHexDictの簡易作成
- snappyHexMeshの実行
- 初期条件の一括参照・変更
- constデータの一括参照・変更
- systemデータの一括参照・変更
- ポテンシャル流れ場作成
- ソルバーの実行
- ソルバーの実行(プロット付)
- ソルバーの停止

DEXCS for OpenFOAM(R) の機能一覧

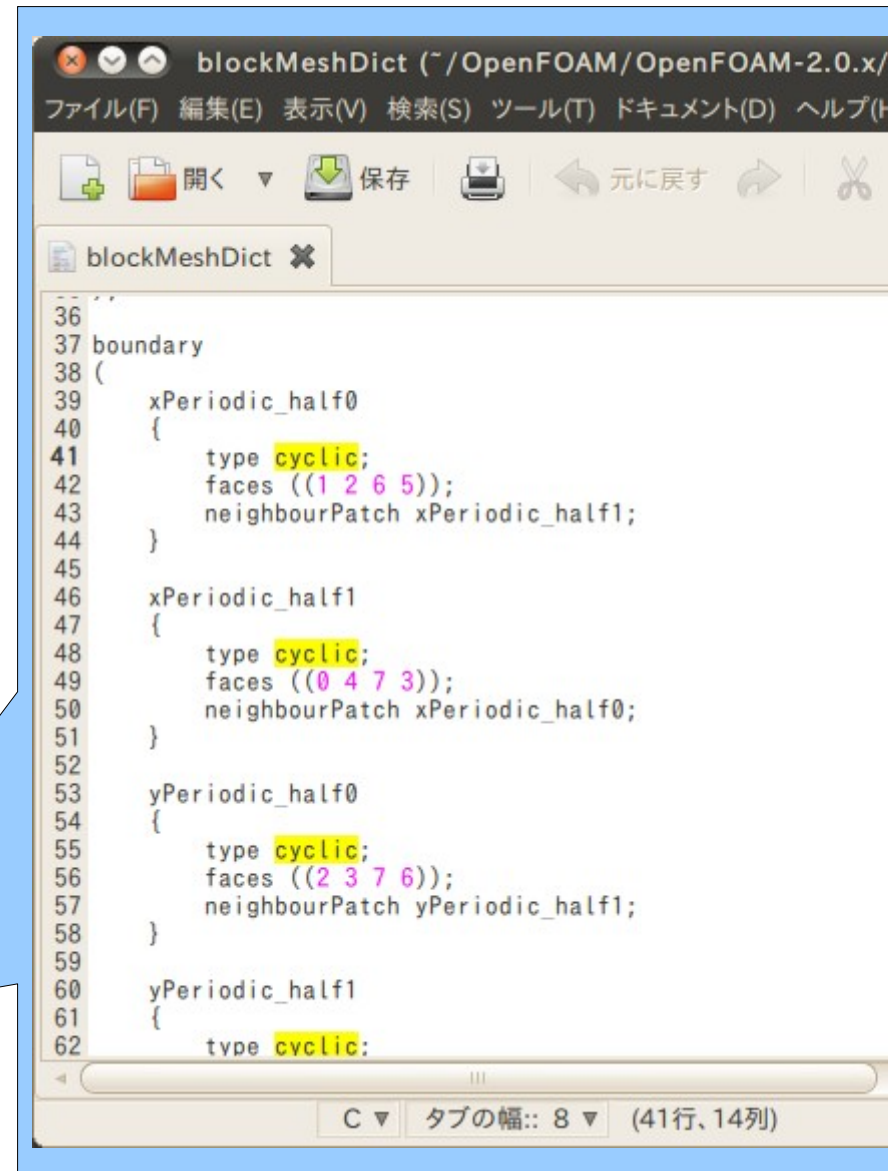
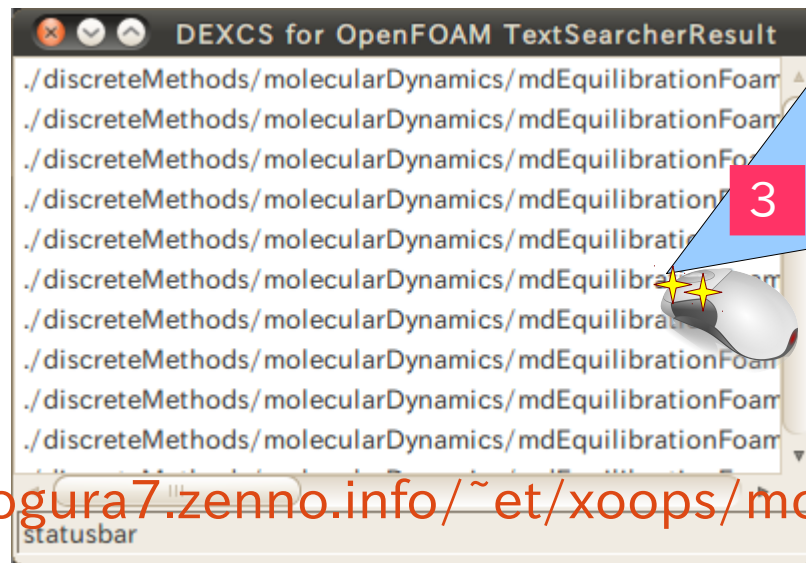
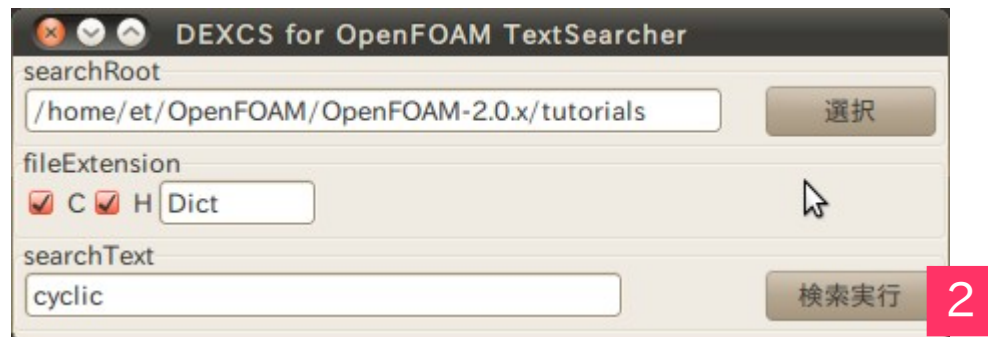
	2009	2010	2011	
	Launcher	Launcher	Launcher	SWAK
ケースの初期化				
テンプレートケースの選択・変更	◎	○	○	○
blockMeshの表示・編集		◎		◎
blockMeshの実行		◎	○	○
patch名の一括整合	◎	○	○	○
patch名の一括変更		◎	○	○
featureデータの作成			◎	◎
snappyHexDictの簡易作成	◎	○	○	○
snappyHexMeshの実行	◎	○	○	○
初期条件の一括参照・変更		◎	○	○
初期条件の一括参照・変更		◎	○	○
constデータの一括参照・変更		◎	○	○
ポテンシャル流れ場作成	◎	○	○	○
ソルバーの実行			○	○
ポテンシャル流れ場作成			○	○
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)		◎	○	○
ソルバーの停止		◎	○	◎
時刻歴データの図化				◎
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				
pyFoam Ver.	0.5.3	0.5.4		0.5.6
simpleFunctionObject		◎		
swak4Foam				◎

◎:新機能

残差履歴の図化



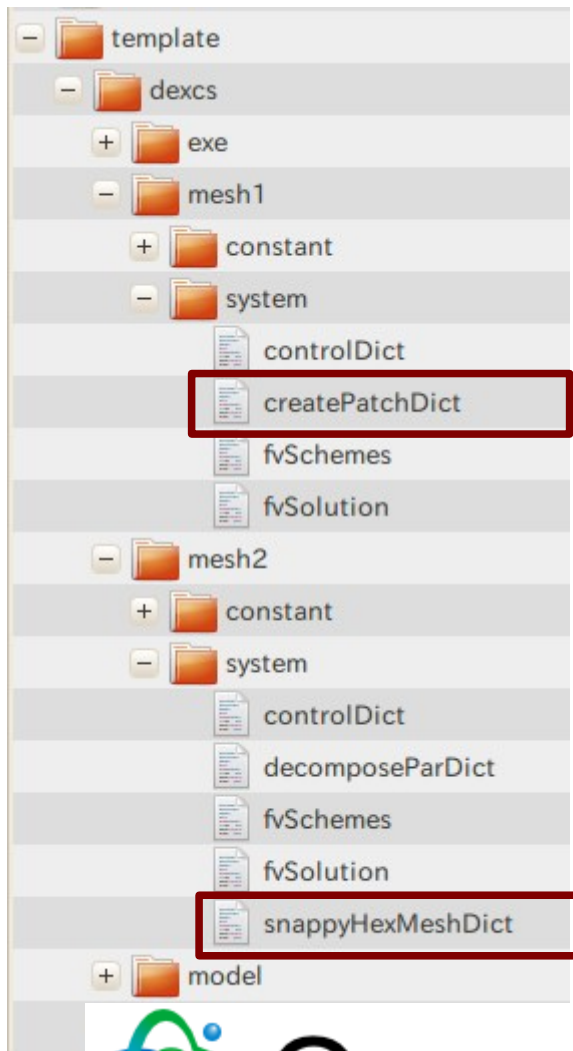
ソースコード検索




その他の変化点

- OF1.7.x ⇒2.0.x VerUp に伴うDict書式変更
- メッシュ選択時にOpenFOAMのファイル構成チェック
- snappyHexMeshDict (nGrow)
- 計算初期化で初期流れ計算の初期化
- 初期流れ計算の2重起動防止
- ボタン配置 (計算実行)
- 履歴データの結合と図化方法
- paraFoam、paraView切り替え

OF1.7.x ⇒ 2.0.x VerUp に伴うDict書式変更





```

34 // If no patches does print and
35 patchInfo
36 (
37 {
38     // Name of new patch
39     name inlet;
40
41     // Type of new patch
42     dictionary
43     {
44         type patch;
45     }
46
47     // How to construct: either 'patch
48     constructFrom patches;
49
50     // If constructFrom = patches : na
51     patches (auto0);
52
53     // If constsnappyHexDict
54     set f0;
55 }
56
    
```



```

34 // If no patches does
35 patches
36 (
37 {
38     // Name of new patch
39     name inlet;
40
41     // Type of new patch
42     patchInfo
43     {
44         type patch;
45     }
46
47     // How to construct: either '
48     constructFrom patches;
49
50     // If constructFrom = patches
51     patches (auto0);
52
    
```

1.7.x ⇒ 2.0.x 変化点

castellatedMesh	
snap	
addLayers	
geometry	
castellatedMeshControls	
maxLocalCells	
maxGlobalCells	
minRefinementCells	
maxLoadUnbalance	
nCellsBetweenLevels	
features	Featureデータ (*eMesh)を定義
refinementSurfaces	
resolveFeatureAngle	
refinementRegions	
locationInMesh	
allowFreeStandingZoneFaces	これがないと動作しない
snapControls	
nSmoothPatch	
tolerance	
nSolverIter	
nRelaxIter	
nFeatureSnapper	これがないとFeatureSnappが機能しない
addLayersControls	
relativeSizes	
layers	
expansionRatio	
finalLayerThickness	
minThickness	
nGrow	
featureAngle	
nRelaxIter	
nSmoothSurfaceNormals	
nSmoothNormals	
nSmoothThickness	
maxFaceThicknessRatio	
maxThicknessToMedialRatio	
minMedianAxisAngle	
nBufferCellsNoExtrude	
nLayerIter	
nRelaxedIter	
meshQualityControls	
maxNonOrtho	
maxBoundarySkewness	
maxInternalSkewness	
maxConcave	
minVol	
minTetQuality	
minArea	
minTwist	
minDeterminant	
minFaceWeight	
minVolRatio	
minTriangleTwist	
nSmoothScale	
errorReduction	
relaxed	
debug	
mergeTolerance	



メッシュ選択時にOpenFOAMのファイル構成チェック



snappyHexMeshDict (nGrow)

<http://mogura7.zenno.info/~et/xoops/modules/wordpress/index.php?p=415>

nGrowとは

Number of layers of connected faces that are not grown if points get not extruded; helps convergence of layer addition close to features

点がなければ生成されない面に結合されたレイヤの数. 輪郭に近いレイヤ追加の収束に役立つ.

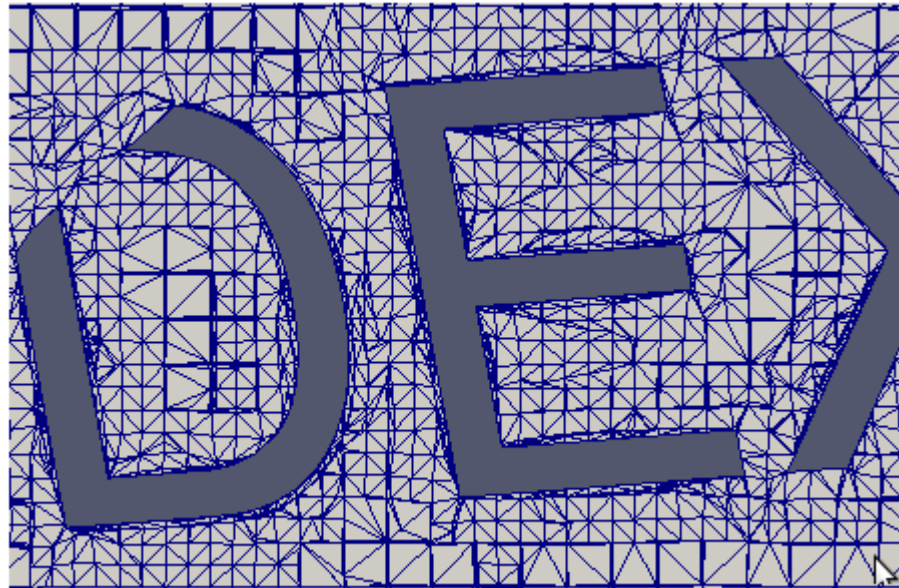
和文マニュアルの日本語訳が今ひとつ意味不明でしたが、

レイヤを生成出来なかった点につながる面の数で、この間はレイヤーを生成しない

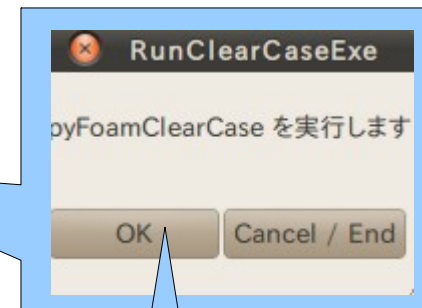
ということだと思います・・・と書いてみたところで、それでも意味不明だと言われそうですが(._.)

これまではnGrow=1としてきましたが、nGrow=0 とすると

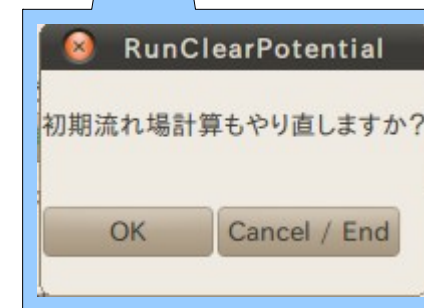
OF-2.0.x, nGrow=0



計算初期化で初期流れ計算の初期化



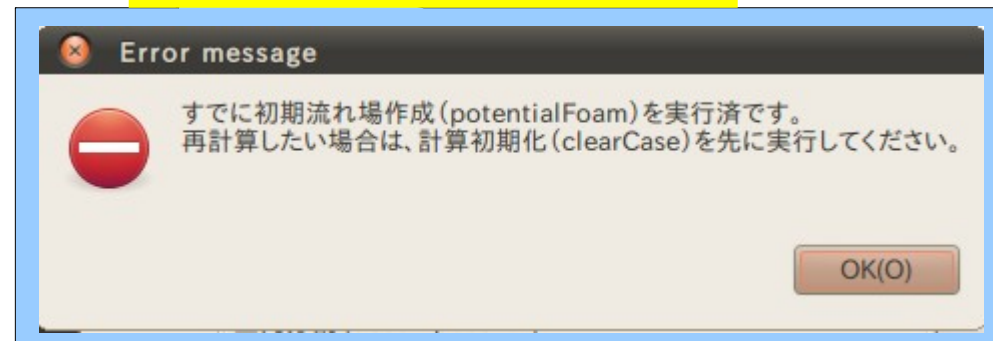
初期流れ場作成済みの場合



初期流れ計算の2重起動防止



初期流れ場作成済みの場合



ボタン配置(計算実行)

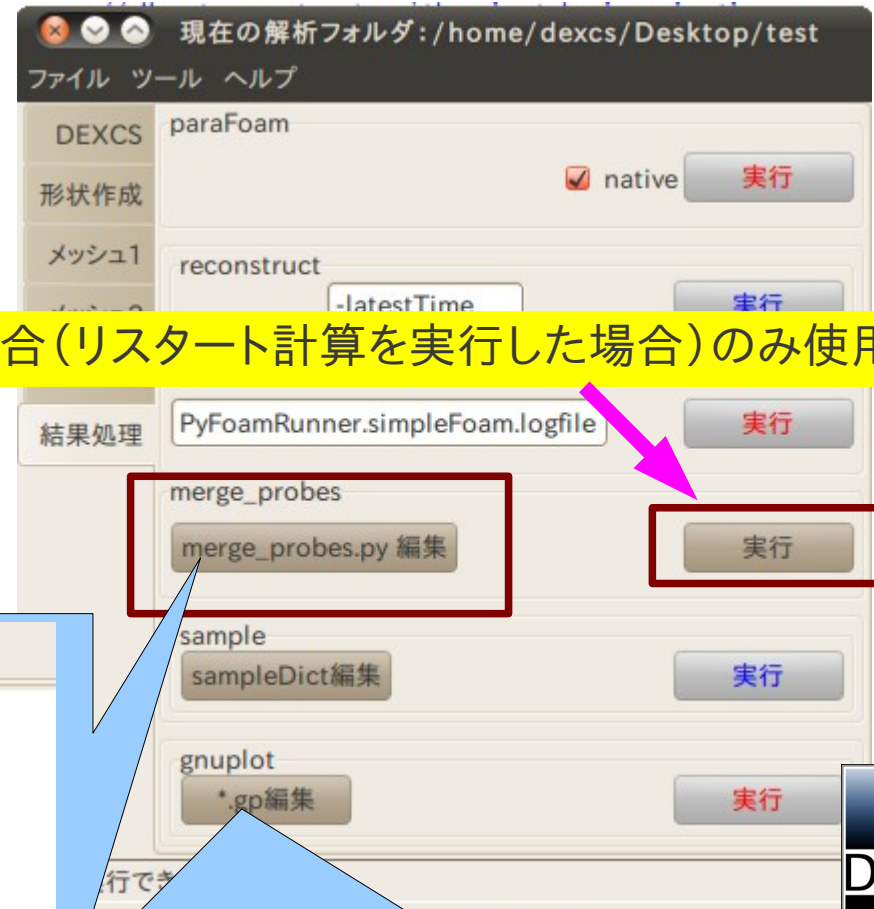
2010
DEXCS



2011
DEXCS



履歴データの結合と図化方法



必要な場合 (リスタート計算を実行した場合) のみ使用

```
merge_probes.py X
19 os.mkdir(probe_dir + 'all')
20 for prop in probe_prop:
21     outfile = file(probe_dir + 'all' + os.sep + prop, 'a')
22     for f in a:
23         infile = probe_dir + f + os.sep + prop
24         print "merging " + infile + " to all/" + prop + "..."
25         f_in = open(infile, "r")
26         data = f_in.read()
27         outfile.write(data)
28         f_in.close()
29     outfile.close()
30 if os.path.exists(probe_dir + '0'):
31     shutil.rmtree(probe_dir + '0')
32     shutil.move(probe_dir + 'all', probe_dir + '0')
33     return True
34 else:
35     print probe_dir + " does not exists"
36     return False
37
38 merge_probes( '../exe/probes', 'p', 'U', 'k')
39 merge_probes( '../exe/forces', 'forces.dat')
40 merge_probes( '../exe/patchExpression_massFlowSwak', 'inlet',
```



```
force.gp X
1 set terminal png
2 set output "forces.png"
3 set xlabel "Iteration number"
4 set ylabel "forceData"
5 set grid
6 set key box
7 set style data line
8 plot '< sed -e "s/(//g" -e "s/)//g" ../exe/forces/0/forces.dat' using 1:2
9     '< sed -e "s/(//g" -e "s/)//g" ../exe/forces/0/forces.dat' using 1:3
10    '< sed -e "s/(//g" -e "s/)//g" ../exe/forces/0/forces.dat' using 1:4
11 # EOF
```



paraFoam、paraView切り替え

The image displays three overlapping windows of the DEXCS software interface, illustrating the process of switching between paraFoam and paraView.

- Top-left window (paraFoam):** Shows the 'メッシュ確認' (Mesh Check) button highlighted with a red box. A blue callout box above it contains the 2010 DEXCS logo and the text 'paraView', and the 2011 DEXCS logo and the text 'paraFoam'.
- Top-right window (paraFoam):** Shows the 'native' checkbox checked and the '実行' (Execute) button highlighted with a red box. A blue callout box to its right contains the text 'paraFoam' and 'paraView'.
- Bottom window (paraView):** Shows the 'native' checkbox checked and the '実行' (Execute) button highlighted with a red box. A blue callout box to its right contains the text 'paraFoam' and 'paraView'.

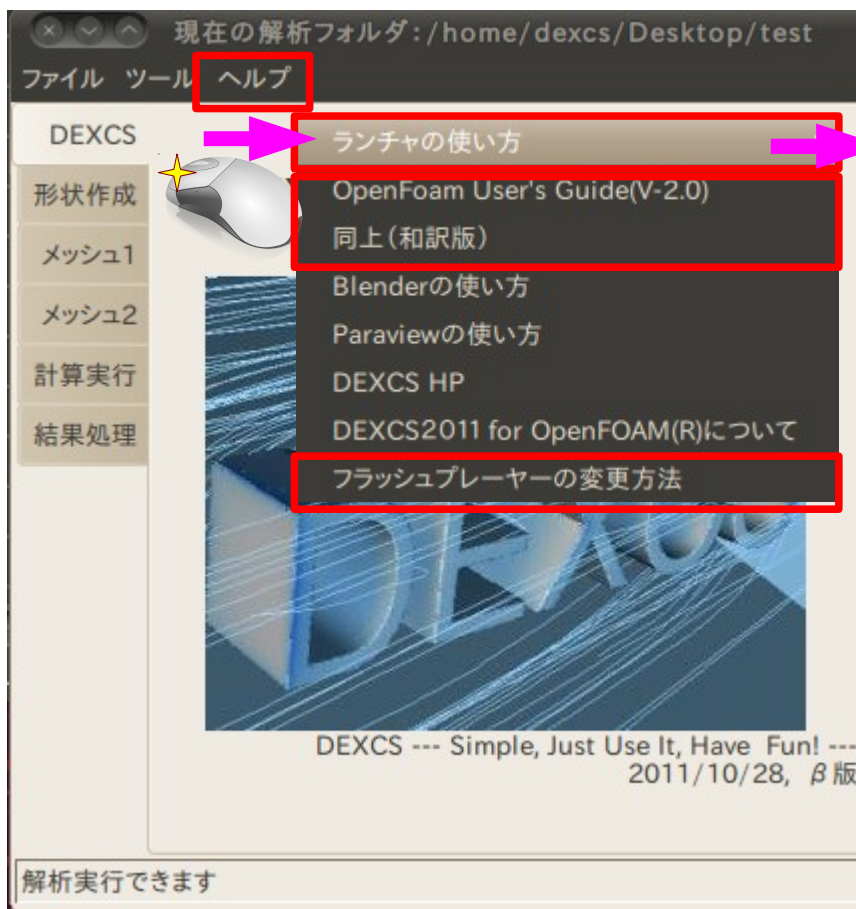
A blue arrow points from the '実行' button in the top-right window to the '実行' button in the bottom window, indicating the transition.

推獎利用方法

推奨利用方法1

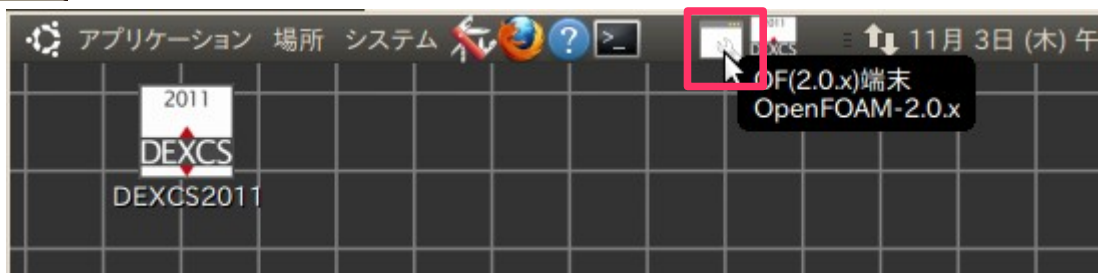
OpenFOAM初学者向け

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
まずは使ってみる
形状作成
メッシュ1(基礎メッシュ)
メッシュ2(詳細メッシュ)
計算実行
結果処理
- 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)
- 3 OpenFOAMの動作とファイル構成の概要理解
- 4 OpenFOAMユーザーズガイド

コマンド入力に不便を感じない人



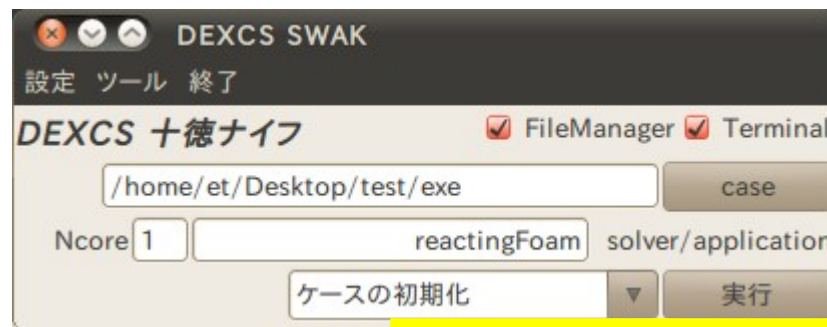
推奨利用方法2



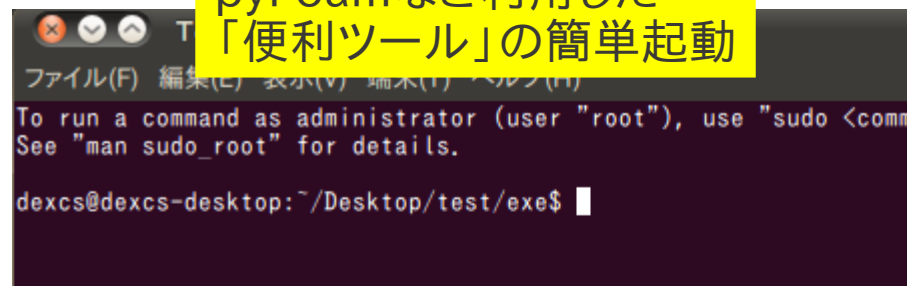
OpenFOAM利用経験者

OpenFOAMの動作と
ファイル構成を理解している

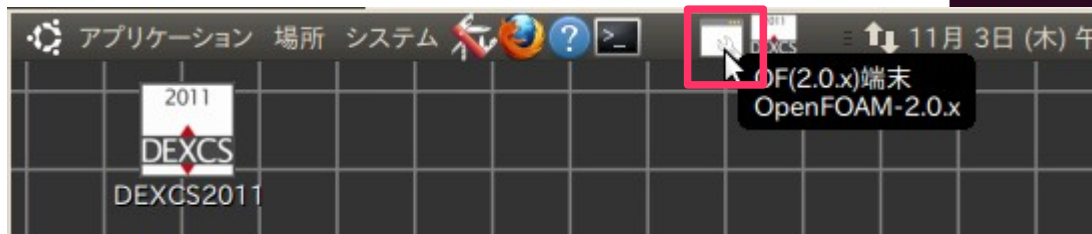
Linux(ubuntu)のコマンド入力が不得手な人



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



コマンド入力が不便を感じない人
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

The image illustrates the steps to customize the DEXCS launcher GUI:

- Step 1:** Open the file `Desktop/DEXCS/dexcsw.xg` in wxGlade.
- Step 2:** Edit the `blockMeshDict編集` label in the GUI.
- Step 3:** Set the event handler for the button to `EditBlockMeshDict`.

The screenshots show the wxGlade interface with various panels like 'Tree', 'Properties', and 'Events'. A pink arrow indicates the flow from the initial file opening to the final GUI configuration.

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

SPE (Stani's Python Editor)

```
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()
```

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

公開版リリースに向けて

- git 最新版のエラー問題
- OFマニュアル(v.2.0)和訳版の完成
- OF-(R)問題
- 十徳の機能追加
 - PyFoamStandardLogAnalyzer → plot
- 十徳の利用方法説明
- ランチャーカスタマイズ方法のチュートリアル

ご意見・ご要望・バグレポート

<http://mogura7.zenno.info/~et/xoops/>

E.Mogura's Site

XOOPS Cube Legacy 2.1 [NEWS](#) [FORUM](#) [DOWNLOAD](#) [LINK](#)

ACCOUNT

ログイン

ユーザ名:


et

パスワード:

ログイン

[パスワード紛失](#)

新規登録

 E. Mogura
mogura77x

ようやく完成。DEXCS2011 for
OpenFOAM(R) プロトタイプ。
「第7回オープンCAE初心者会
会」<http://t.co/Le01Q2x> の報
26 seconds ago · reply · retweet

最後のチャンス、いけ坂本
<http://t.co/B5WnI8u>
8 days ago · reply · retweet · fav

今度は駄目 <http://t.co/IT05f7I>
8 days ago · reply · retweet · fav

またまた、ピンチ <http://t.co/OzI>
8 days ago · reply · retweet · fav

 Join the

メインメニュー

[ホーム](#)
[お問い合わせ](#)

[昔のコンテンツ](#)

PRIVATE MESSAGE

お問い合わせ

名前:

メールアドレス:

ホームページ:

ICQ:

会社名:

お住まい:

本文:

送信