

TreeFoam 操作マニュアル

目次

1. TreeFoam とは
2. 起動画面
3. TreeFoam 画面の説明
4. 基本的操作方法
5. コマンドボタンの説明
 - 5-1. ファイル関係
 - 5-2. 編集関係
 - 5-3. 計算関係
 - 5-4. ツール関係
6. ポップアップメニューの説明
7. その他
 - 7-1. gridEditor について
8. インストール

使用したアイコン :

gnome に標準で装備されているアイコンと、salomeMeca のアイコンを利用。これらアイコンは、path の設定を省くために、TreeFoam/icon フォルダ内にコピー保存して使っている。

利用したユーティリティプログラム :

以下の2本のユーティリティを使っている。この2本は、path の設定を省くために TreeFoam/bin フォルダ内にコピー保存して使っている。

- ・ unv2gmsh.py, unv2x.py : CAELinux で配布されているものを利用
- ・ ideasUnvToFoam : OpenFOAM で配布されているものを利用

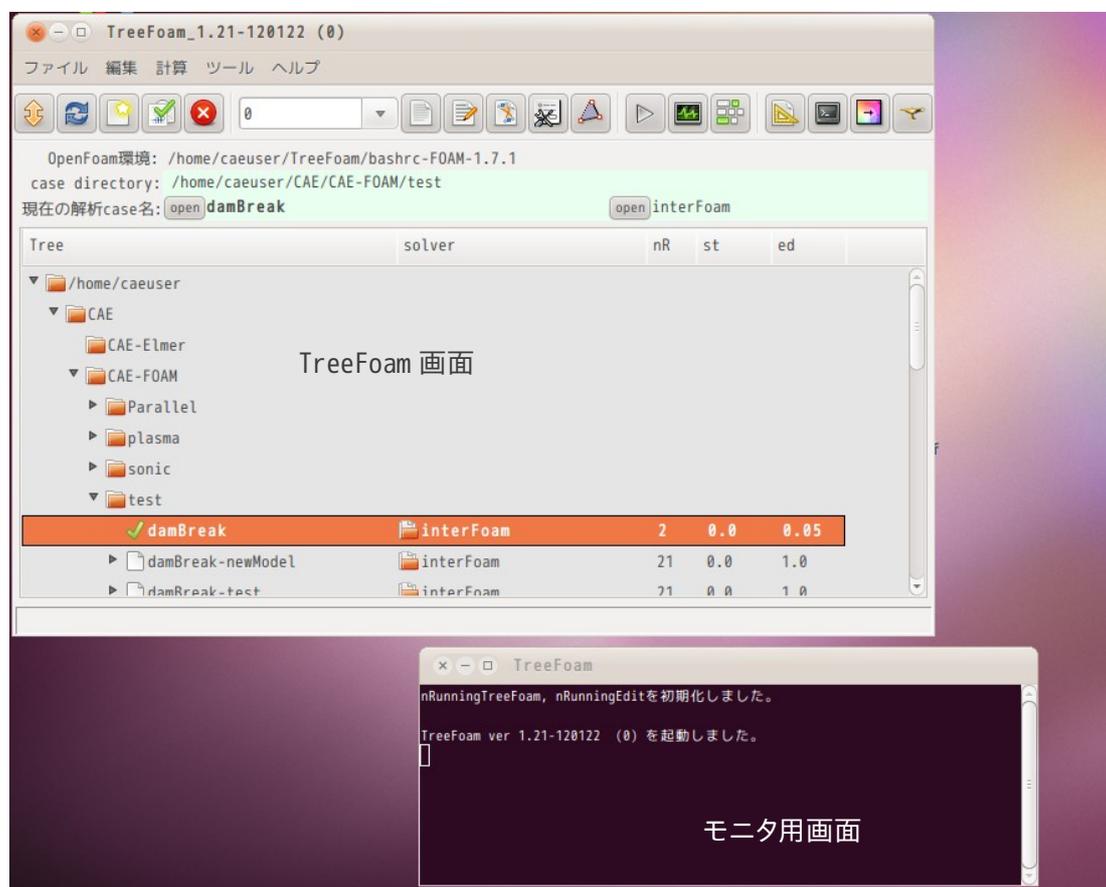
1. TreeFoamとは

OpenFOAMを使いやすくする為のツール。代表的には以下の機能を備えている。

- 1) case フォルダがツリー表示される
- 2) solver 名や解析結果フォルダ（数字フォルダ）の数が表示される
- 3) freeCAD、SalomeMeca、paraViewのランチャを備える
- 4) SalomeMeca で作成したメッシュを Foam 形式に領域名ごとに変換できる
メッシュ変換は、case フォルダ内の model/mesh.unv ファイルをメッシュ変換する
- 5) gridEditor を使うと OpenFOAM の境界条件を表形式で編集できる
- 6) TreeFoam や gridEditor はマルチタスク（各アプリのグローバル変数が取りこめる様、起動時に起動 No を付与している。）に対応しており、複数起動できる。gridEditor を複数起動し、gridEditor 間で「コピー」「貼り付け」ができる。

2. 起動画面

TreeFoam を起動すると以下の画面が現れる。この画面で基本的に操作する。TreeFoam 画面で操作し、エラー表示はモニタ用画面で確認する事が基本。

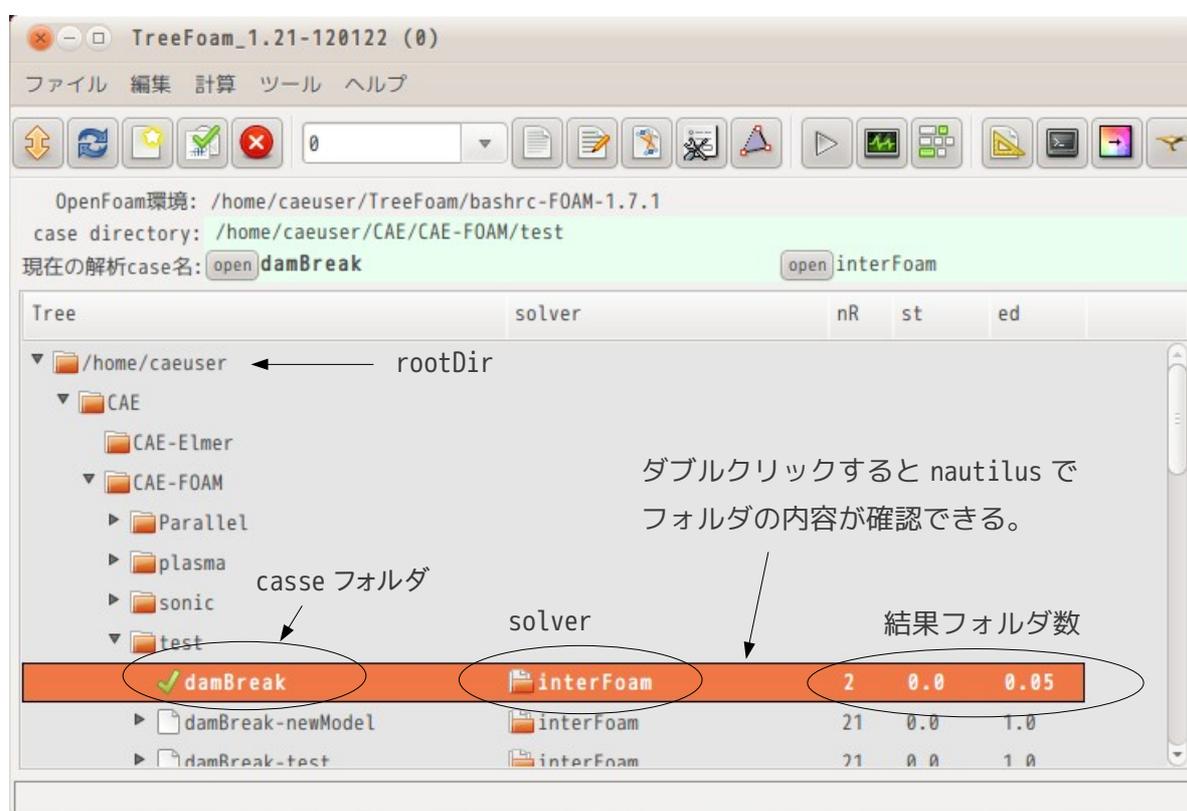


3. TreeFoam 画面の説明

フォルダ構成がツリーで表示され、OpenFOAM の case フォルダは、使っている solver や計算結果フォルダ数も表示される。

以下の画面では、「レ」マークの付いているフォルダ「damBreak」が解析する case フォルダとして設定されている。この case では、solver「interFoam」が設定され、計算結果フォルダが「2」ヶあり、「0.0~0.05」までの計算結果がある事が判る。

フォルダをツリー表示する時に、そのフォルダが OpenFOAM の case フォルダかどうかを確認 (system/controlDict が存在するかどうか) し、case フォルダの場合、solver と計算結果フォルダを表示する。



4. 基本的操作方法

TreeFoam を起動後、解析したい case をツリー上から選択し、 ボタンをクリックして解析 case として設定する。(クリックするとアイコンが「レ」マークに代わる。)

この後、



初期値や境界条件を設定する時間フォルダを設定する。



境界条件や初期値を設定。

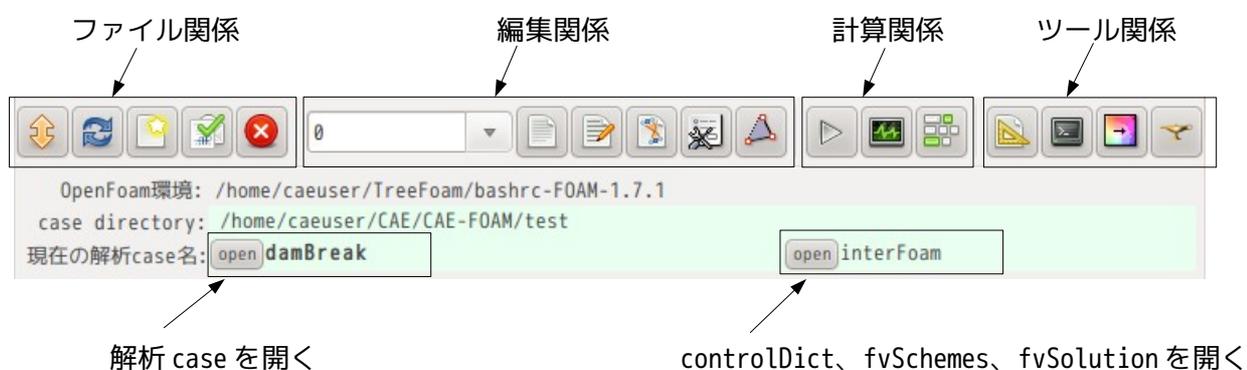
-  SetField 等で Field にデータをセットする。
-  解析 case を開く。
-  controlDict、fvSchemes、fvSolution を開く
-  計算開始。

で条件設定し、計算を開始する。

その他、詳細は「コマンドボタンの説明」を参照。

5. コマンドボタンの説明

以下のコマンドボタンを持っている。



5-1. 解析 case を開く

「open」ボタンをクリックする事で、解析 case のフォルダを nautilus で開くことができる。

尚、同様な事は、ツリー上のフォルダをダブルクリックしても同様に nautilus でフォルダの内容が確認できる。

5-2. controlDict、fvSchemes、fvSolution を開く

「open」ボタンをクリックする事で、これら 3 ヶのファイル (controlDict, fvSchemes, fvSolution) を gedit で開き編集する事ができる。これらのファイルは、頻繁に使用するなので、まとめて開ける様にしている。下図参照。



5-3. ファイル関係

5-3-1. rootDir の変更

ツリー表示させる為の最上位のフォルダを設定する。3項の例では、rootDir が「/home/caeuser」に設定されている。これ以下のフォルダがツリー表示される。

このrootDirを変更する場合は、このボタンをクリックして変更する。



5-3-2. Dir の再読み込み (reload)

フォルダ構成が変わった時（フォルダ追加、削除、名前変更）にツリーデータに反映させる。このボタンをクリックして、ツリーデータを更新する。

極稀に、ツリーデータが壊れフォルダ名がダブって表示される事があるが、この場合は、reload ボタンをクリックして、再読み込めば、復元できる。

5-3-3. 新規 case 作成

このボタンをクリックして、新規の case を tutorials からコピーして作成する。この部分は未作成。

5-3-4. 解析 case の設定

解析用 case として、定義する。ツリーからフォルダを選択後、このボタンをクリックすると、「レ」マークが付き、解析用 case として定義される。

5-3-5. 終了

終了処理を行なった後、TreeFoam を終了させる。終了は、必ず終了ボタン又は、メニューバー上から終了を選択して終了する。window 中の「X」ボタンでも終了できるがこの場合は、終了処理を行なわないので、この方法は避ける。

5-4. 編集関係

5-4-1. 編集時間フォルダの設定

境界条件を編集する時間フォルダを設定する。

ドロップダウンテキストボックス中の数字を編集する。この数字は、通常「controlDict」内の startTime に記述した時間が表示される。（startFrom が latestTime になっている場合は、latestTime の数字が表示される。）

5-4-2. field の編集

このボタンをクリックして、時間フォルダ内の field データを編集する。編集は、「gridEditor」か「gedit」を使って編集する。ここで編集する時間フォルダは、 で設定した時間フォルダ内の field になるので、実行前は、時間フォルダを確認しておく。編集する field は、下記ダイアログで選択する。



5-4-3. fieldへのデータセット

各 field の internalField や boundaryField にデータをセット、クリアする場合に使用する。

「setFields」コマンドでデータをセットする場合もこれを使う。使い方は、<setFieldsDict 編集> ボタンで setFieldDict ファイルを編集し、<setFields...> をクリックして setFields コマンドを実行する。

ここで編集する時間フォルダは、 で設定した時間フォルダの field を編集

(setFields、clearInternalFields、clearBoundaryFields) する事になるので、実行前は、時間フォルダを確認する。)

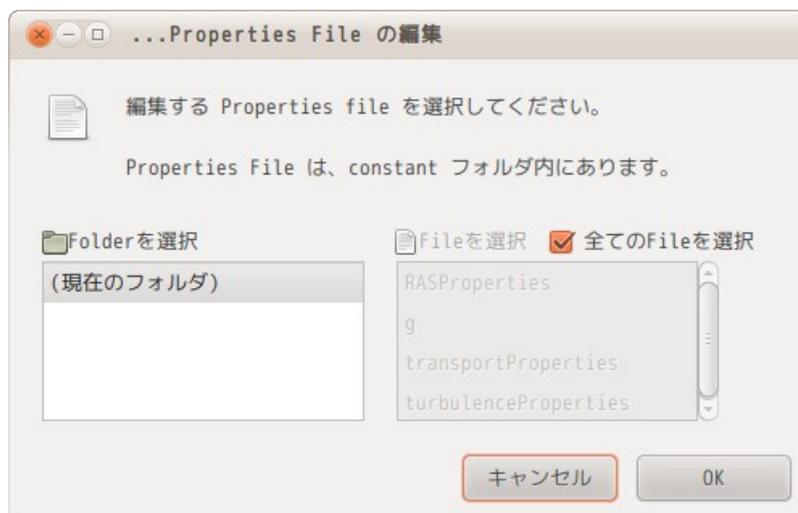
internalField や boundaryField のクリアは、これを使うよりも gridEditor の方が使いやすい。



5-4-4. properties ファイルの編集

constant フォルダ内にある「x x x properties」を編集する。(properties ファイルに限らず constant フォルダ内にある file を編集する。)

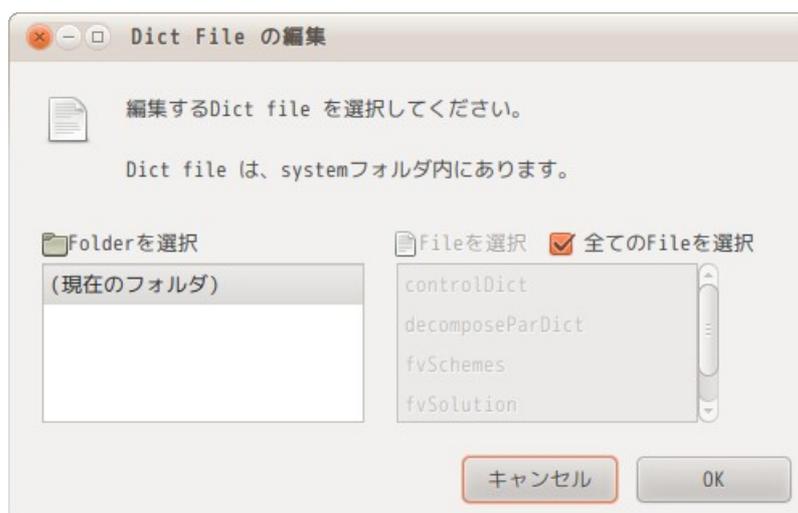
コマンドボタンをクリックすると下図の画面が現れるので、編集したい properties を選択して「OK」ボタンをクリックすると gedit が起動し編集できる。



5-4-5. Dict ファイルの編集

system フォルダ内にある「x x x Dict」ファイルを編集する。(Dict ファイルに限らず system フォルダ内にある faile を編集する。)

コマンドボタンをクリックすると下図の画面が現れるので、編集したい Dict ファイルを選択して「OK」ボタンをクリックすると、gedit が起動し騙取できる。



5-4-6. メッシュの編集 

メッシュに関する以下の操作を行なう。

- 1) メッシュ変換 (salomeMeca → OpenFOAM)
- 2) メッシュのスケール変更
- 3) faceZone 名、cellZone 名の変更
- 4) 隣接している faceZone、cellZone の結合
- 5) 内部パッチの作成 (faceZone から内部パッチを作成)
- 6) 領域分割 (cellZone から領域分割する)

これらの操作は、以下の画面から選択して操作する。



5-4-6-1. メッシュ変換



salomeMeca から吐き出した unv 形式のメッシュファイル (mesh.unv) を Foam 形式に変換する。メッシュ変換する unv ファイルは「model/mesh.unv」のファイルを変換するので、salomeMeca 側から吐き出した unv 形式のファイルは「model」フォルダ内に「mesh.unv」として保存しておく。

この変換には、下記 2 種を準備しているので、必要に応じて使い分ける。

unv2gmshToFoam salomeMeca 側で face と Volume を定義した場合は、こちらを選択する。

ideasUnvToFoam salomeMeca 側で face のみを定義した場合は、こちらを選択する。
 両変換とも SalomeMeca 側で定義した face と volume は、その名称が引き継がれ、それぞれ
 patch、faceZone、cellZone および、faceSet、cellSet として定義されるので、便利。

5-4-6-2. スケール変更 スケール変更...

salomeMeca 側で「mm 単位」で作成されたメッシュを「m 単位」に変換する場合等に使用する。
 <スケール変更> ボタンをクリックして、モデルのスケールを変更する。単位を mm→m に変換する場
 合は、0.001 を入力すれば、メッシュの座標を 1/1000 に変換できる。尚、変換したスケールは、保存
 されるので、表示されているモデルの大きさと現在のスケールからモデルを作成した時のスケールが
 判る。



5-4-6-3. faceZone 名、cellZone 名の変更 faceZone名変更 cellZone名変更

faceZone 名 (cellZone 名) を変更する場合に使用する。使い方は、以下の画面が表示されるので、
 faceZone 名 (cellZone 名) を選択して、新しい Zone 名を入力して<変更> ボタンをクリックする。こ
 の時点では、画面のリスト上は変更されているが、polyMesh フォルダ内の Zone 名は変更されていない。
 画面上の変更を適用させる為、最後に<適用> ボタンをクリックして、polyMesh フォルダ内に適用す
 る。



5-4-6-4. faceZone、cellZone の結合

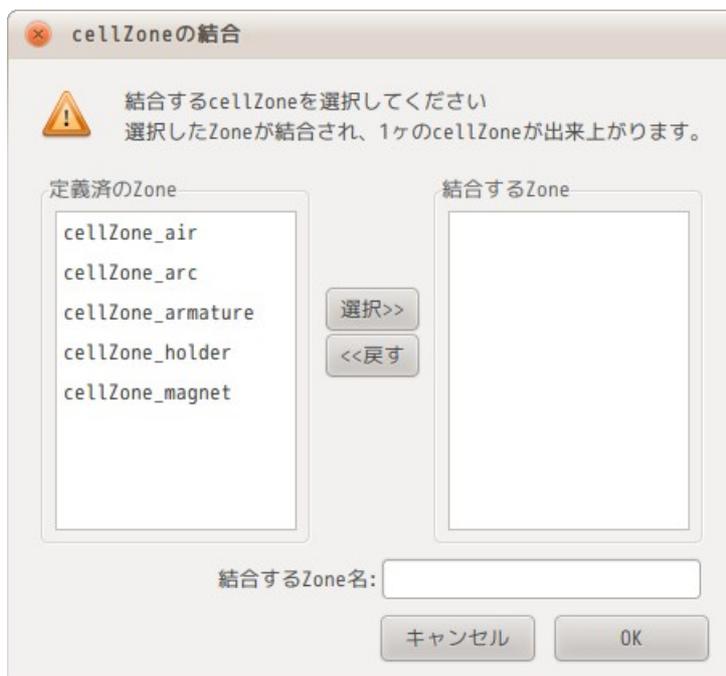
faceZone結合...

cellZone結合...

隣接してつながっている複数の faceZone (cellZone) を結合して、1ヶの faceZone (cellZone) を作成する時に使用する。

使い方は、結合したい faceZone (cellZone) を選択して、結合させるウインドに移動し、結合させる faceZone 名 (cellZone 名) を入力して、<OK> ボタンをクリックして結合させる。





5-4-6-5. 内部パッチの作成 内部patch作成...

faceZone から内部パッチ (internalPatch) を作成する時に使用する。

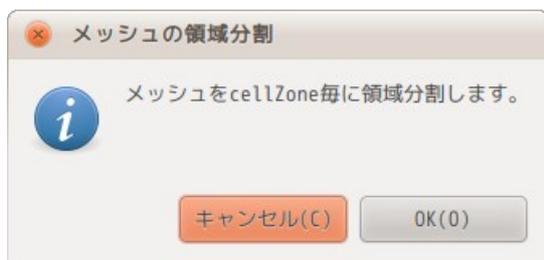
使い方は、以下の画面上から、内部パッチを作成したい faceZone を選択 (複数可) し、<OK> ボタンをクリックすることで、内部パッチが作成される。



5-4-6-6. 領域分割 (Region) 領域分割(Region)...

OpenFOAM の solver 「chtMultiRegionFam」等を使用する場合、個体と流体をそれぞれ領域分割する必要があるので、この領域分割を行なう。

使い方は、<領域分割 (region) ...> ボタンをクリックすると、以下の画面が現れるので <OK> ボタンをクリックする。

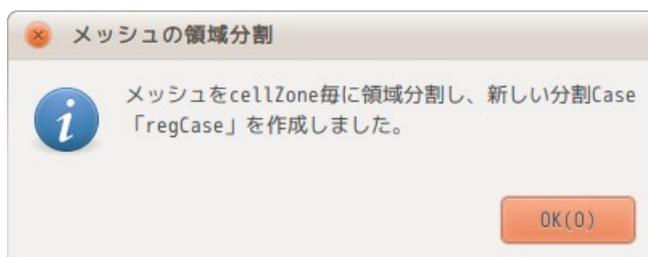


この後、領域分割が開始される。領域分割すると、メッシュが入れ替わるので、領域分割したメッシュはcaseフォルダ内に「regCase」フォルダが出来上がり、このフォルダを新規caseとして計算する構成をとっている。この為、このregCaseフォルダ内には、「0」「constant」「system」フォルダもコピーされている。

領域分割が完了すると、以下の画面が現れるので、各領域を流体、個体領域に分けて定義する。



流体、個体領域に分けた後は、<OK>ボタンをクリックすると、以下の画面が現れ、領域分割は完了する。



領域分割後は、regCase/system/changeDictionaryAllRegionsDict ができあがっているので、これを

使って（編集して）、changeDictionary コマンドにより境界条件を設定することができる。

5-4-6-7. 領域分割 (Conjugate) 領域分割(Conjugate)...

この領域分割は、OpenFOAM-ext の solver 「conjugateHeatFoam」等で使用される領域分割を行なう。この solver も前項と同様に固体、流体領域を分割する必要があり、この領域分割を行なう。本来 conjugateHeatFoam の solver は、各領域の節点を共有する必要がないので、固体、流体を別々にメッシュを切り、境界面を定義して計算させるが、カップリングする field やその境界面がたくさんある場合は、これを使う方がその定義を自動で設定できるので便利。

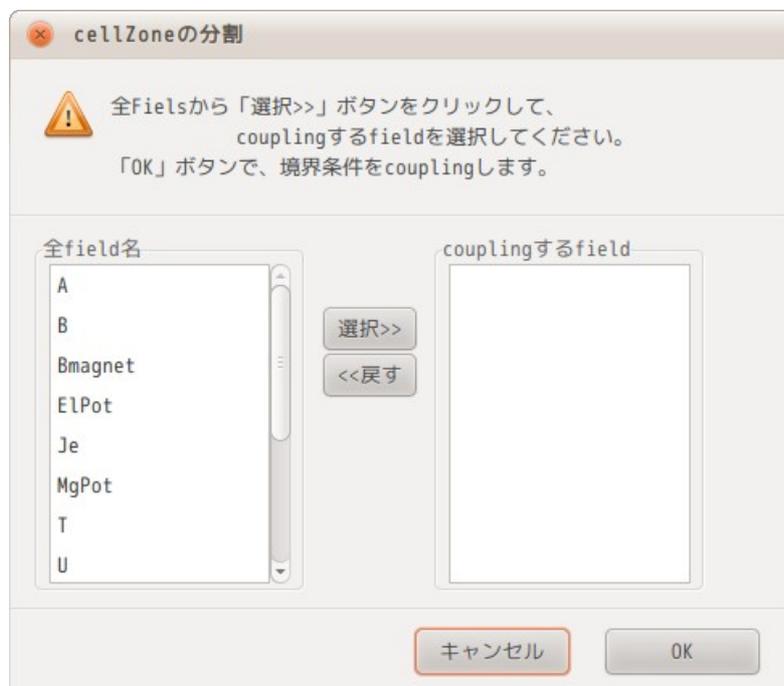
使い方は、<領域分割 (Conjugate) ...> ボタンをクリックすると、以下の画面が現れるので <OK> ボタンをクリックする。



この後、領域分割が開始され、分割が終了すると、以下の画面が現れるので、それぞれの領域を流体、固体領域に分けて定義する。この時、流体領域は1ヶの領域しか扱えないので、複数の流体領域がある場合は、予め、cellZone を結合し1ヶの領域として定義しておく。



この後、以下の画面が現れるので、couplingするfieldを定義する。この定義により、boundaryFieldやboundaryが自動的に修正される。（couplingするfieldで領域分割した各領域の境界面のpatchTypeとその内容が自動的に設定される。）



<OK>ボタンをクリックした後は、以下の画面が現れ、領域分割は終了する。領域分割後は、regCaseフォルダが出来上がり、このフォルダ内に分割後のメッシュやconstant、systemフォルダがコピーされているので、このcaseを解析caseとして、計算する。



5-5. 計算関係

5-5-1. solverの起動

このボタンをクリックすることで、controlDict内に記述されたsolverが起動する。実行時のlogは、

solve.logに残る。この為、実行中、あるいは実行後に pyFoamPlotWatcher を起動すれば、残渣を表示させることができる。

5-5-2. plotWatcher の起動

このボタンをクリックすることで、「plotWatcher」が起動して、残渣を表示させる事ができる。このボタンは、実行中、または実行後に起動させる。残渣は「solve.log」ファイルを読み込み、表示する。

5-5-3. 並列処理

この部分は未完成。

5-6. ツール関係

5-6-1. freeCAD の起動

このボタンをクリックする事で、freeCAD が起動する。起動する為には、当然 freeCAD がインストールされている必要がある。

5-6-2. FOAM 端末の起動

このボタンをクリックする事で、設定された解析 case をカレントディレクトリとして、FOAM 端末が起動する。この為、この端末から FOAM で提供されている各種ユーティリティを起動することができる。また、FOAM 端末は、ツリー上のポップアップメニューからでも起動できる。

5-6-3. paraFoam の起動

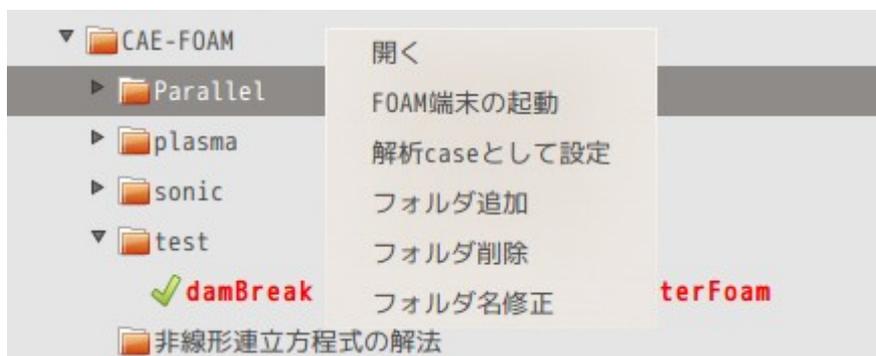
このボタンをクリックする事で、指定された解析 case の計算結果を paraView で確認できる。

5-6-4. salomeMeca の起動

このボタンをクリックする事で、salomeMeca が起動する。

6. ポップアップメニューの説明

メニューボタンの他に、ツリー上で右クリックした時にもメニューが現れる。



- 1) 開く
 選択しているフォルダの内容が確認できる。
 フォルダをダブルクリックしても同様にフォルダ内容が確認できる。
- 2) FOAM 端末の起動
 右クリックしたフォルダをカレントディレクトリとして FOAM 端末が起動する
- 3) 解析 case として設定
 右クリックしたフォルダを解析 case として設定する。（「レ」マークを付ける）
- 4) フォルダ追加
 フォルダを追加する。
- 5) フォルダ削除
 フォルダを削除する。
- 6) フォルダ名修正
 フォルダ名を修正する。

7. その他

7-1. gridEditor について

gridEditor は、各 field と各 patch の表を作成し、この表の中で Excel の様に境界条件を編集する事ができるツールである。この表は、polyMesh/boundary の patch を基準にして各 field の patch 内容を読み込んで表を作成している。

また、データ検索を高速化している為、gedit で全 field を編集する場合よりも早く起動する。特に

setFields でデータをセットしている場合は、その差が顕著にでる。(gedit で編集する場合、boundaryField がファイルの最後にあるので、スクロールに時間が掛かる。gridEditor は、高速検索している為に素早く表示できる。ただし、高速検索はコメント文を無視して「boundaryField」のキーワードを検索しているので、コメント文中に「boundaryField」という文字列あった場合には、ここで引っ掛かってしまうので、コメント文はなるべく避ける。)

編集する各 field 全てに、データがセットされている場合 (latestTime で途中から境界条件を変えて再開させる時は、全 field に計算結果がありデータがセットされている) は、gedit でも同じだが、さすがに読み込みに時間が掛かってしまう。

7-1-1. 起動方法 (TreeFoam からの起動)

起動する前に編集時間設定ボックス の時間を確認し、編集したい時間に設定する。この時間は、controlDict 内の startFrom が latestTime の場合、最終の計算時間に設定されている。起動は、TreeFoam 上の <field 編集> ボタン  をクリックして現れた画面上 (下図) で、<Grid で編集> ボタンをクリックすると gridEditor が起動する。起動後は、表形式で初期値や境界条件 (boundaryType、 boundaryField、 internalField) が編集できる。



7-1-2. 起動方法 (FOAM 端末からの起動)

FOAM 端末 (OpenFOAM 用に環境設定された端末) から gridEditor を起動するには、FOAM 端末上から gridEditor コマンドを入力する。gridEditor コマンドは、以下の形式で入力する。

```
gridEditor [caseDir] [数字フォルダ] [polyMeshDir]
```

コマンドに続くパラメータは省力可能なので、以下の様に入力しても全て同じ gridEditor が起動する。

```
$ gridEditor . 0 constant/polyMesh
$ gridEditor . 0
$ gridEditor .
$ gridEditor
```

7-1-3. 操作方法

gridEditor の表内データ (cell データ) の編集は、直接キー入力するか「ダブルクリック」または「F2」キーで編集モードに入り、直接表内で編集できる。直接編集できる編集可能領域は、下図の赤枠内だが、特別な方法 (ポップアップメニュー) で patch 名も編集できる。

cell データは、cell 選択後、マウスの右クリックで「コピー」「貼り付け」ができる。gridEditor を複数起動し、gridEditor 間でも「コピー」「貼り付け」ができる。

編集後は、「保存」ボタンで編集内容が各ファイルに適用される。(「保存」ボタンをクリックしないと編集内容が反映されない。)

	define patch (boundary)	R	U	alpha1	alpha2
field type dimensions		volSymmTensorField; [0 2 -2 0 0 0 0];	volVectorField; [0 1 -1 0 0 0 0];	volScalarField; [0 0 0 0 0 0 0];	volScalar [0 0 0 0
internal Field		uniform (0 0 0 0 0 0);	uniform (0 0 0);	nonuniform List<scalar> 2268 (1 1...	uniform 0
leftWall	type wall;	type kqRWallFunction; value uniform (0 0 0 0 0 0);	type fixedValue; value uniform (0 0 0);	type zeroGradient;	type zero
rightWall	type wall;	type kqRWallFunction; value uniform (0 0 0 0 0 0);	type fixedValue; value uniform (0 0 0);	type zeroGradient;	type zero
lowerWall	type wall;	type kqRWallFunction; value uniform (0 0 0 0 0 0);	type fixedValue; value uniform (0 0 0);	type zeroGradient;	type zero
atmosphere	type patch;	type inletOutlet; inletValue uniform (0 0 0 0 0 0); value uniform (0 0 0 0 0 0);	type pressureInletOutletVelocity; value uniform (0 0 0);	type inletOutlet; inletValue uniform 0; value uniform 0;	type inletOutlet; inletValue uniform 0; value uniform 0;
defaultFaces	type empty;	type empty;	type empty;	type empty;	type empty;

patch 名 patch type 編集可能領域

各 field の各 Patch が一覧で確認でき、cell の内容が編集でき、コピー、貼り付けが可能なので、初期値や境界条件の設定が楽になる。

internalField も編集できる。ただし、setFields でデータをセットした時、その形式が uniform 形式でなく、nonuniform List 形式でデータがセットされており、この場合は編集できない。（編集しても無視する。）ただし、nonuniform を uniform に表中で書き換えてやれば、uniform 形式で編集することができる。

7-1-2. メニューボタンの説明

メニューボタンは、以下のボタンを準備している。

- 1) 開く 別の gridEditor を起動する。（未作成）
- 2) 保存 編集した内容を boundary と各 field に反映する。
- 3) csv 保存 表形式のイメージをそのまま csv で保存する。
- 4) 再読込 boundary と各 field の内容を再読込し、表示する。
- 5) 閉じる gridEditor を終了する。
- 6) コピー 選択した cell の内容を clipBord にコピーする。
- 7) 貼り付け clipBord の内容を貼り付ける。

7-1-3 ポップアップメニューの説明

7-1-3-1. cell 領域のポップアップメニュー

cell 領域では、右クリックで以下のポップアップメニューが使える。

action;	type fixe
(0 cellコピー	value uni
act cell貼付	type fixe
(0 cell貼付	value uni
et;	type pres

- 1) コピー 選択した cell の内容を clipBord にコピーする。
- 2) 貼り付け clipBord の内容を cell に貼り付ける。

ここで言う clipBord は、OS が持っている clipBord ではなく、TreeFoam フォルダ内の clipBord ファイルを clipBord として使っており、ここにコピーしたデータを python の list 形式データとして保存している。貼り付け時は、この内容を読み込んで貼り付ける。

7-1-3-2. 行ラベル (patch 名) 領域のポップアップメニュー

行ラベル (patch 名) 領域では以下のポップアップメニューが使える。このメニューは、ラベル (patch 名) をクリックして行を選択した上で、右クリックするとポップアップメニューが現れる。



- 1) patch 名変更 patch 名を変更する。
- 2) patch 削除 patch を削除する。

両メニューとも、boundary と全ての field の boundaryField を書き換える。書き換えは、メニュー実行時に書き換えてしまうので、再読み込み (reload) しても元には戻らないので patch を削除する場合は注意が必要。

8. インストール

8-1. 動作環境

最低限必要な条件は、python と wxGlade がインストールされていること。さらに、フォント「VL ゴシック」がインストールされている事。フォントについては、基本的に system フォントを使っているが、太字にした時、フォントが変わってしまった為、太字の定義を「VL ゴシック」に設定している為。後は、TreeFoam が必要としているアプリ関係。

python	必須
wxGlade	必須
VL フォント	
OpenFOAM	
PyFoam	
freeCAD	

SalomeMeca

8-2. インストール方法

インストールは、\$HOME フォルダ直下に「TreeFoam」フォルダを作りこの中に全てのファイルをコピーする。

<フォルダ構成>

```
$HOME
  TreeFoam
    bin          実行ファイル
    icons        TreeFoam で使うアイコンを保存
    python       python スクリプト
    wxg          wxGlade のデータ
```

実行する為には、「configTreeFoam」の内容を書き換え、環境に合うように設定する。

黄色のハッチングは、TreeFoam 上で設定でき、終了時に書き換えるので、起動には問題ない。

赤色のハッチングは、正しく設定しないと OpenFOAM、paraFoam、FreeCad、salomeMeca が正常に起動しなくなるので正しく記述する。

-----configTreeFoam の内容-----

```
#
# TreeFoam の設定
# -----
#
# rootDir の設定
rootDir /home/caeuser

# 選択されている現在の case の設定
workDir /home/caeuser/CAE/CAE-FOAM/test/damBreak

# FOAM 端末の環境設定ファイル
# OpenFOAM の他、必要な箇所に PATH、PYTHONPATH を通しておく。
bashrcFOAM /home/caeuser/TreeFoam/bashrc-FOAM-1.7.1

# paraFoam の起動
paraFoam /home/caeuser/TreeFoam/runParaFoam-1.7.1

# SalomeMeca の起動
```

```
salomeMeca /home/caeuser/TreeFoam/runSalomeMeca.5.1.4
```

```
# CADの起動
```

```
CAD freecad
```
