

# 新TreeFoamの紹介

(TreeFoam ver 3.13.220906)

## <変更内容>

1. stlファイルのチェック追加
2. gridEditor：長文のpatch内容の編集を修正
3. OpenFOAM-v2206への対応（バグ修正）

## 1. stlファイルのチェックを追加

画像処理ソフトを使って3Dデータを取得する場合などは、正しくstlデータが取得できていない事があり、これが原因でメッシュが作成できなくなる事がある。

- faceの重複定義（厚さ「0」のsolid）

- faceの向きが不揃い

- solidデータにも関わらず、隣接faceを持たないfaceが存在する（stlが閉じていない）

この様なエラーを検出して修正ファイルを保存する様にした。

この処理の為、stl編集画面上に「stlCheck」「solid抽出」ボタンを追加。

- stlCheckボタン： faceの重複定義があれば、削除

- face向きの不揃いがあれば、向きを修正

- solid抽出ボタン： faceの重複定義、face向きをチェック修正

- 隣接faceを持たないfaceを削除し、solidデータを抽出

# stlファイル編集画面の起動方法

snappyHexMeshによるmesh作成

stlファイル

開く ./model

stlチェック...

特徴線の抽出 クリック

150 (px)

Dict編集...

surfaceFeatureExtractDictを作成し

csvファイルによるmesh作成

並列処理 nCPU 4 分割数(

csv作成... csv編集 sna

snappyDict編集

csvファイルを作成・編集して、cs  
snappyHexMeshDictを作り出す。多  
csvファイルは、stlファイルと同  
並列処理にチェックすると、snapp

stl viewer

edge表示  原点表示

face表裏面の表示  
 両面  表面  裏面

stlファイルの編集

stlファイルのsolid名やscaleを変更する

| stl         | format | solid               | size(xyz) |      |      |
|-------------|--------|---------------------|-----------|------|------|
| baseHex.stl | ascii  | fineReg,halfSp,outW | 0.08      | 0.04 | 0.04 |
| fineReg.stl | ascii  | fineReg             | 0.08      | 0.02 | 0.02 |
| halfSp.stl  | ascii  | halfSp              | 0.005     | 0.01 | 0.01 |
| inW.stl     | ascii  | inW                 | 0.0       | 0.04 | 0.04 |
| outW.stl    | ascii  | outW                | 0.0       | 0.04 | 0.04 |
| sideW.stl   | ascii  | box                 | 0.1       | 0.04 | 0.04 |

stlファイルを選択し、各ボタンをクリックする事で変更できる。  
複数選択が可能。

ascii変換 solid名設定 solid名変更...

scale変更... 移動... 各軸回転... stlCheck solid抽出

face向き反転 stl結合... 内部solid名詳

ボタンを追加

形状確認(paraview) folder開く 閉じる

選択しているstlファイルを表示する。

## 実行例 (stlCheckボタン)

意識的に1個のface向きを反転させたstlを作成し、stlCheckした結果

stlCheckの出力log内容

```
reading '/home/caeuser/CAE/CAE-FOAM/OF-10/03_normalMesh/model/halfSp.stl' file...
  number of triangles }
  11944 triangles.   } 三角形の数、節点の数
  number of points  }
  5974 points.      }
  stl size
  xyz: 0.00500000000000000044 0.01 0.01 } stl形状の大きさ (XYZ) 、stl形状の表面積
  area of all triangles
  area: 0.00023554606076079247
deleting same faces... faceの重複定義
  deleted 0 faces.
checking face direction... faceの向きチェック→1個のface向き修正
  changed 1 faces directions.
checking stl, closed or not... stlが閉じているかチェック
  stl file is closed.
creating stl file...
--> checked stl file was saved at '/home/caeuser/CAE/CAE-FOAM/OF-10/03_normalMesh/model/halfSp_out.stl'.
```

**修正した結果のstlファイルを保存**

stlファイルをチェックし、

修正箇所があれば、修正したstlファイルを保存する。

修正箇所が無ければ、保存しない。

# 実行例 (solid抽出ボタン)

## 直方体にfaceを接続したstlのsolid抽出

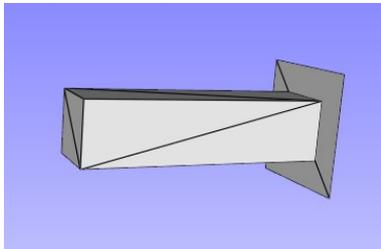
### solid抽出の出力log内容

```
reading '/home/caeuser/CAE/CAE-FOAM/OF-10/03_normalMesh/model/baseHex.stl' file...
number of triangles
  14 triangles.
number of points
  12 points.
stl size
  xyz: 0.08 0.04 0.04
area of all triangles
  area: 0.008799999999999999
deleting same faces...
  deleted 0 faces.
checking face direction...
  changed 0 faces directions.
deleting no neighbour face...
  deleted 2 faces.
checking stl, closed or not...
  stl file is closed.
creating stl file...
--> checked stl file was saved at '/home/caeuser/CAE/CAE-FOAM/OF-10/03_normalMesh/model/baseHex_out.stl'.
```

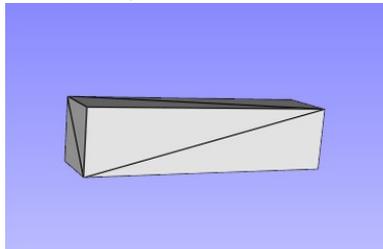
隣接faceを持っていないfaceを2個削除

修正した結果のstlファイルを保存

元のstl形状



修正後のstl形状



修正後では、余分なfaceが削除され、  
solid (閉じたstl) のみが抽出されている

## 2. gridEditor : 長文のpatch内容の編集を修正

patchTypeが「codedFixedValue」の場合、patch内にcodeを記述する為、patch内容が長文になってしまう。元々、gridEditorでは、patchの数値data以外で長文になってしまう事を想定していなかった。この為、数値data以外でpatch内容が長文になった場合は、

gridEditorの表示は問題ないが、保存すると内容が壊れてしまう事が判明し、修正。

### 元の内容

```
boundaryField
{
    :
    model_S
    {
        type            codedFixedValue;
        name            powerLaw;
        value            uniform (0 0 0);
        code
        #{
            vectorField f
            ( patch().lookupPatchField<volVectorField, vector>("U") );
            const scalar Ub = 5.4;
            const scalar Zb = 47.8;
            const scalar n = 5.0;
            forAll (f, i)
            {
                const scalar Z = patch().Cf()[i][2];
                f[i] = vector(0.0, Ub*pow(Z/Zb, 1.0/n), 0.0);
            }
            operator==(f);
        };
    }
    :
}
```

### 保存後の内容 (誤った結果が保存される)

```
boundaryField
{
    :
    model_S
    {
        type            codedFixedValue;
        name            powerLaw;
        value            uniform (0 0 0);
        code
        #{
            vectorField f
            ( patch().lookupPatchField<volVectorField, vector>("U") );
            const scalar Ub = 5.4;
            const scalar Zb = 47.8;
            const scalar n = 5.0;
        )
        ;
    }
    :
}
```

# 長文のpatch内容の修正方法

長文の場合は、cell内容の文末に「...」が表示される。

ここを編集する場合は、該当cell or 該当fieldをダブルクリックして、editorを起動して編集する。

gridEditor: 2\_model\_test\_pimple\_hotroom\_Uvar\_copy1/0 (0:0)

ファイル(F) 編集(E) 表示(V)

該当fieldをダブルクリック

|                | define patch<br>at constant<br>(boundary)    | T<br>(1)                               | U<br>(2)   |   |
|----------------|--|--|--|---|
| field type     |  | volScalarField;                        | volVectorField;  | volScalarField                            |
| dimensions     |  | [0 0 0 1 0 0 0];                       | [0 1 -1 0 0 0 0];  | [1 -1 -1 0 0 0]                           |
| internal Field |  | uniform 300;                           | uniform (0 0 0);   | uniform 0;                                |
| floor          | type wall;<br>inGroups List<word> 1(wall);   | type fixedValue;<br>value uniform 310; | type noSlip;   | type compressible;<br>value uniform 1;    |
| model_E        | type wall;<br>inGroups List<word> 1(wall);   | type zeroGradient;                     | type slip;   | type compressible;<br>value uniform 1;    |
| model_N        | type patch;<br>inGroups List<word> 1(patch); | type zeroGradient;                     | type inletV;<br>value 1;   | type calculate;<br>value \$internalField; |
| model_S        | type patch;<br>inGroups List<word> 1(patch); | type fixedValue;<br>value uniform 300; | type codedFixedValue;<br>name powerLaw;<br>value uniform (0 0 0);<br>code<br>#{<br>vectorField f<br>( patch().lookupPatchField<volVectorField, vector>("U") );<br>const scalar Ub = 5.4;<br>const scalar Zb = 47.8;<br>const scalar n = 5.0;...<br>} | type calculate;<br>value \$internalField; |
| model_W        | type wall;<br>inGroups List<word> 1(wall);   | type zeroGradient;                     | type slip;   | type compressible;<br>value uniform 1;    |

該当cellをダブルクリック

## 該当cellをダブルクリックした時

```
0:0:U.model_S.temp
~/TreeFoamUser/temp 保存(S)

1 type          codedFixedValue;
2 name          powerLaw;
3 value         uniform (0 0 0);
4 code
5 #{
6   vectorField f
7   ( patch().lookupPatchField<volVectorField, vector>("U") );
8   const scalar Ub = 5.4;
9   const scalar Zb = 47.8;
10  const scalar n = 5.0;
11  forAll (f, i)
12  {
13    const scalar Z = patch().Cf()[i][2];
14    f[i] = vector(0.0, Ub*pow(Z/Zb, 1.0/n), 0.0);
15  }
16  operator==(f);
17 #};|
```

なし ▾ タブ幅: 4 ▾ (17行、4列) ▾ [挿入]

## 該当fieldをダブルクリックした時

```
U
~/TreeFoamUser/temp/U.0 保存(S)

54   value         uniform (0 0 0);
55   }
56   model_S
57   {
58     type          codedFixedValue;
59     name          powerLaw;
60     value         uniform (0 0 0);
61     code
62     #{
63       vectorField f
64       ( patch().lookupPatchField<volVectorField, vector>("U") );
65       const scalar Ub = 5.4;
66       const scalar Zb = 47.8;
67       const scalar n = 5.0;
68       forAll (f, i)
69       {
70         const scalar Z = patch().Cf()[i][2];
71         f[i] = vector(0.0, Ub*pow(Z/Zb, 1.0/n), 0.0);
72       }
73       operator==(f);
74     #};
75   }
76   model_W
77   {
78     type          slip;
79   }
80   model_ground
81   {
82     type          noSlip;
83   }
84 }
```

C ▾ タブ幅: 4 ▾ (48行、17列) ▾ [挿入]

editor上で編集し、保存することで、編集内容がfieldに保存される。

## 数値dataを含む場合（計算結果があるfolder）

該当cellをダブルクリックした時

```
0:1:U.model_S.temp
~/TreeFoamUser/temp
保存(S)

1 type          codedFixedValue;
2 value         nonuniform List<vector>
3 800
4 (
5 (0 2.93058 0)
6 (0 2.95566 0)
7 (0 2.95171 0)
8 (0 2.94955 0)
9 (0 2.94907 0)
10 (0 2.94881 0)
11 (0 2.9487 0)
12 (0 2.94862 0)
13 (0 2.94863 0)
14 (0 2.9499 0)
15 (0 2.9508 0)
16 (0 2.9499 0)
17 (0 2.94863 0)
18 (0 2.94862 0)
19 (0 2.94867 0)
20 ...U.2...
21 )
22 ;
23 name          powerLaw;
24 code          #{
25     vectorField f
26     ( patch().lookupPatchField<volVectorField, vector>("U") );
27     const scalar Ub = 5.4;
28     const scalar Zb = 47.8;
29     const scalar n = 5.0;
30     forAll (f, i)
31     {
32         const scalar Z = patch().Cf()[i][2];
33         f[i] = vector(0.0, Ub*pow(Z/Zb, 1.0/n), 0.0);
34     }
35     operator==(f);
36 #};
```

該当fieldをダブルクリックした時

```
U
~/TreeFoamUser/temp/U.0
保存(S)

74 )
75 ;
76 }
77     model_S
78     {
79         type          codedFixedValue;
80         value         nonuniform List<vector>
81 800
82 (
83 (0 2.93058 0)
84 (0 2.95566 0)
85 (0 2.95171 0)
86 (0 2.94955 0)
87 (0 2.94907 0)
88 (0 2.94881 0)
89 (0 2.9487 0)
90 (0 2.94862 0)
91 (0 2.94863 0)
92 (0 2.9499 0)
93 (0 2.9508 0)
94 (0 2.9499 0)
95 (0 2.94863 0)
96 (0 2.94862 0)
97 (0 2.94867 0)
98 ...U.2...
99 )
100 ;
101     name          powerLaw;
102     code          #{
103         vectorField f
104         ( patch().lookupPatchField<volVectorField, vector>("U") );
105         const scalar Ub = 5.4;
106         const scalar Zb = 47.8;
107         const scalar n = 5.0;
108         forAll (f, i)
109         {
110             const scalar Z = patch().Cf()[i][2];
111             f[i] = vector(0.0, Ub*pow(Z/Zb, 1.0/n), 0.0);
112         }
113         operator==(f);
114     #};
115 }
```

数値dataを含む場合でも、数値データ以外はeditorで編集できる。

### 3. OpenFOAM-v2206への対応（バグ修正）

OF-v2206については、TreeFoam-3.12で対応したが、その後のOF-10への対応で、  
topoSetEditorが起動しない  
multiRegionCaseの材料設定ができない  
事が発生した。

当初は動いていたが、OF-10への対応で、OpenFOAMのバージョン判断を追加したが、この設定を誤り動かなくなってしまった。

これを修正。