OpenFOAM講習中級編：
非標準ライブラリswak4Foamの使い方

今野　雅 (東京大学)

自己紹介

- 所属
  - 東京大学 大学院工学系研究科 建築学専攻
- 専門
  - 建築環境工学 (温熱・空気環境、特に数値予測)
- 所属学会
  - 日本建築学会
  - 空気調和・衛生工学会
  - 日本流体力学会
  - 日本風工学会
  - オープンCAE学会(副会長)

目次

1. swak4Foamとは
2. goovyBCライブラリとは
3. funkySetFields, funkySetBoundaryFieldsとは
4. swakFunctionObjects系ライブラリとは
5. 適宜デモ

swak4Foamとは

- **主に以下のような機能を持つ非標準のライブラリ・ユーティリティ群**

  ✓ ***groovyBC***：ユーザ定義のカスタム境界条件をプログラミングしなくても、標準の境界条件では難しい複雑な境界条件を関数表現を用いて動的に指定できるライブラリ

  ✓***funkySetFields, funkySetBoundaryFields***：場や境界パッチの分布を関数で与えることができるユーティリティ

  ✓***simpleFunctionObjects系***：ソルバ実行時に、様々な場の統計量をログや別ファイルに出力できるライブラリ (標準以外の機能を補完)

  ✓***swakSourceFields, swakTopoSources, replayTransientBC***：今回は時間の都合により説明しません

- **名前の由来：***SWiss Army Knife for Foam* (Foam用の十徳ナイフ)

swak4Foamの作者

- **作者：Bernhard Gschaider**
  - ✓ ***Unofficial OpenFOAM Wiki***の管理者
  - ✓ OpenFOAMを使う上で便利なpythonツールや
    ライブラリ集である***pyFoam***の作者
- **勤務先：ICE @ Leoben, Austria**
  - ✓ LeobenのMONTAN大学内のChristian-Doppler
    (ドップラ効果の発見者) 記念研究所のメンバが起業した会社
  - ✓CAEコンサルティング、ソフトウェア開発、流体計測(PIV)
- **Bernhard による6th OpenFOAM Workshopの発表スライド**

  swak4Foamの現時点で一番詳しいと思われるドキュメント

http://www.openfoamworkshop.org/6th_OpenFOAM_Workshop_2011/Program/training.htm

swak4Foamのダウンロード下準備

- **Webページ**

  http://openfoamwiki.net/index.php/Contrib/swak4Foam

  (Bernhard が管理している*Unofficial OpenFOAM Wiki*にある)

- **ダウンロードに必要なソフトウェア**

  ✓ ***svn***：プログラムソース等を集中管理するバージョン管理システム

  Subversionの略称およびコマンド

Ubuntuでのインスール方法

(実線の囲みでは赤字をターミナルで打ってください。青字は説明です)

```
sudo apt-get install subversion
```

swak4Foamのダウンロード

- ここではOpenFOAMのバージョンを2.0.xと仮定するが、1.xも同様
- 非標準のライブラリを置くディレクトリを作成し移動

✓場所はどこでも良いのだが、推奨は以下

$WM_PROJECT_USER_DIR/Libraries(通常 ~/OpenFOAM/$USER-2.0.x)

```
mkdir $WM_PROJECT_USER_DIR/Libraries ディレクトリの作成
cd $WM_PROJECT_USER_DIR/Libraries ディレクトリへの移動
```

なお、上記のディレクトリへの移動は以下のほうが簡単

```
run
cd ../Libraries
```

- svnを用いたソースのダウンロード (チェックアウト)

```
svn checkout https://openfoam-extend.svn.sourceforge.net/
svnroot/openfoam-extend/trunk/Breeder_2.0/libraries/swak4Foam/
```

swak4FoamのREADME その1

- **READMEというのは目を通すべき重要な文章**

  ✓最初の部分は必ず読んでおいてほうが後々間違いが少なくなる

```
cd swak4Foam
more README もしくはemacsやvi、geditなどお好きなエディタで読みましょう
```

```
* Description
  A collection of libraries and tools that let the user handle
  OpenFOAM-data based on expressions
* Contributors etc
** Original Author
   Bernhard Gschaider (bgschaid@ice-sf.at)
** Current Maintainer
   Bernhard Gschaider (bgschaid@ice-sf.at)
** Contributors
   In alphabetical order of the surname
   - Peter Keller :: =sprinklerInlet=-case
   - Alexey Petrov :: =pythonFlu=-integration
```

swak4FoamのREADME その2

```
** Documentation
   See: http://openfoamwiki.net/index.php/contrib/swak4Foam
* Installation/Compilation
** Requirements
   - Version 2.0 of OpenFOAM (a version that works with 1.6, 1.6-ext
     and 1.7 is available separately)
     - The =finiteArea=-stuff will probably work with version 2.0-ext
       (once that is available)
   - the compiler generators =bison= and =flex=
** Building
: wmake all
   at the base directory should build all the libraries and
   tools.

  The main library =swak4FoamParsers= can't be built in parallel (no
  values of =WM_NCOMPPROCS= bigger than 1)
```

swak4Foamのビルド下準備

- **ビルドに必要なソフトウェアのインストール**

  ✓ ***bison, flex*** : groovyBC等で指定できる、関数等の複雑な構文を解析するパーサ・プログラムを作成するツール。yaccとlexという昔からUNIX界で使われていたツールのGNUプロジェクト版

```
sudo apt-get install bison lex ソフトウエア・パッケージは複数指定可能
```

- **並列ビルドをしない設定**

  ✓ 環境変数 WM_NCOMPPROCS が2以上の場合、wmake によるビルド時に並列ビルドを行うが、swak4FoamParsersライブラリが並列ビルドできないため、環境変数の値を1にする(値を1にしてももちろん可)

```
export WM_NCOMPPROCS=1 シェルがsh系(bash等)の場合。以下もsh系を仮定
```

swak4Foamのビルド

wmake all	READMEに書いてあるビルド法

または、ビルド時のログを取る時は以下のようにする。(ビルド時にエラーが出ることがあるので、後で確認できるよう出来るだけログを取っておいたほうが良い)

wmake all >& log &	コマンド名 > fileでfileにログが残るが、>&とするとエラー出力も含めてfileに書かれる。ライブラリのビルドは一般に時間がかかるので、最後に&を付けてバックグラウドで走らせて、他の仕事ができるようにする。
tail -f log	logファイルをtailコマンドでトレースする
Ctrl-C	コントロール+Cでコマンドは終了。tailは止めて良い
more README	ビルド終了まで残りのREADMEでも読んでましょう
jobs	ビルドのバックグラウンドのジョブの状態確認

[1]+ Done wmake all &>log これが出れば終了!

tail log	ログの末尾を見て、正常に終了したか確認する

groovyBCライブラリとは

- **複雑な境界条件を動的に指定できるライブラリ**

- **ドキュメント**

✓swak4Foamのページにはは書かれていない

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

✓ただし上記のページも一部古いのでどちらも併せて読む必要がある

- **ソルバでgroovyBCライブラリを使う場合の設定**

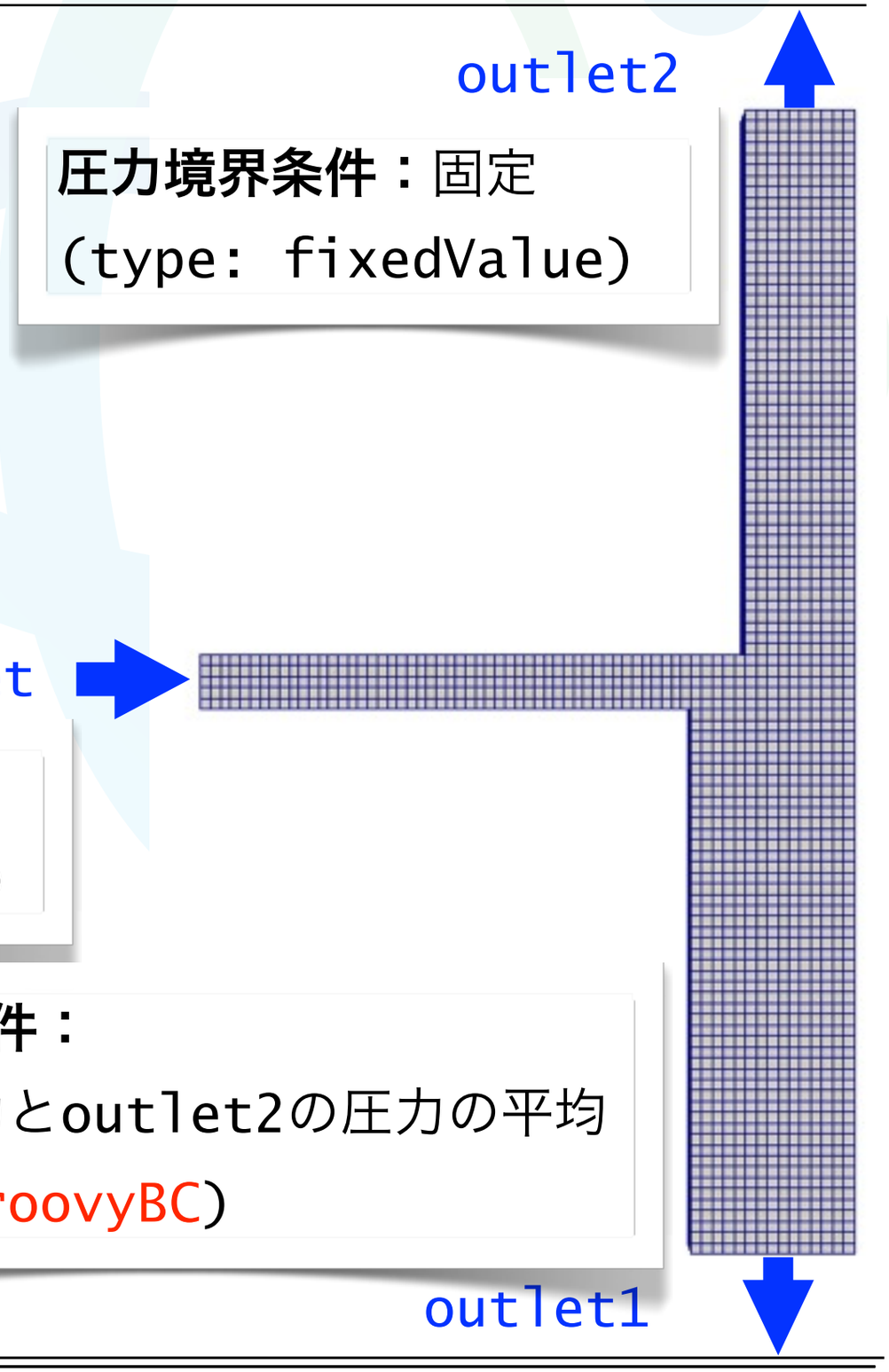
✓controlDictへ以下のライブラリの動的リンクの指示を追加

```
libs ( "libgroovyBC.so" );
```

✓ ソルバに予めリンクされていないライブラリを使う場合は、標準のライブラリであっても同様に設定する必要がある。

groovyBCの使用例の紹介

- **ソースにおける使用例集の場所**
  - ✓swak4Foam/Examples/groovyBC
- **その1例： average-t-junction**
  - ✓T字管流れ場解析 (pimpleFoam)



groovyBCを使用した境界条件 その1

```
outlet1
{
  type                groovyBC;

  variables           variablesで変数を定義する

    “pInlet{inlet}=sum(p*mag(Sf()))/sum(mag(Sf()));
     pOutlet2{outlet2}=p;";

     変数名{外部名}=関数(指定した外部名における値が取得される);
     {}が省略された場合には、同じ境界パッチ上での値が取得される。
     外部名は{エンティティ型‘エンティティ名/リージョン名}と書かれる。
     エンティティ型:internalField,patch,cellZone等。省略時はpatch。
     リージョン名が省略された場合は、境界と同じリージョン。
     セミコロンで複数の文が記述できる。
     関数は後述する。
(続く)
```

groovyBCを使用した境界条件 その2

(続く)

```
fractionExpression “1”;
```
1:ディレクレ(値指定)型，0:ノイマン(勾配指定)型。

fractionExpressionの定義が無いディフォルト時は1(ディレクレ型)。

```
gradientExpression "0";
```
境界の勾配を定義する。fractionExpressionが0の時のみ有効。

```
valueExpression "0.5*(pInlet+pOutlet2)";
```
境界の値を定義する。 fractionExpressionが1の時のみ有効。

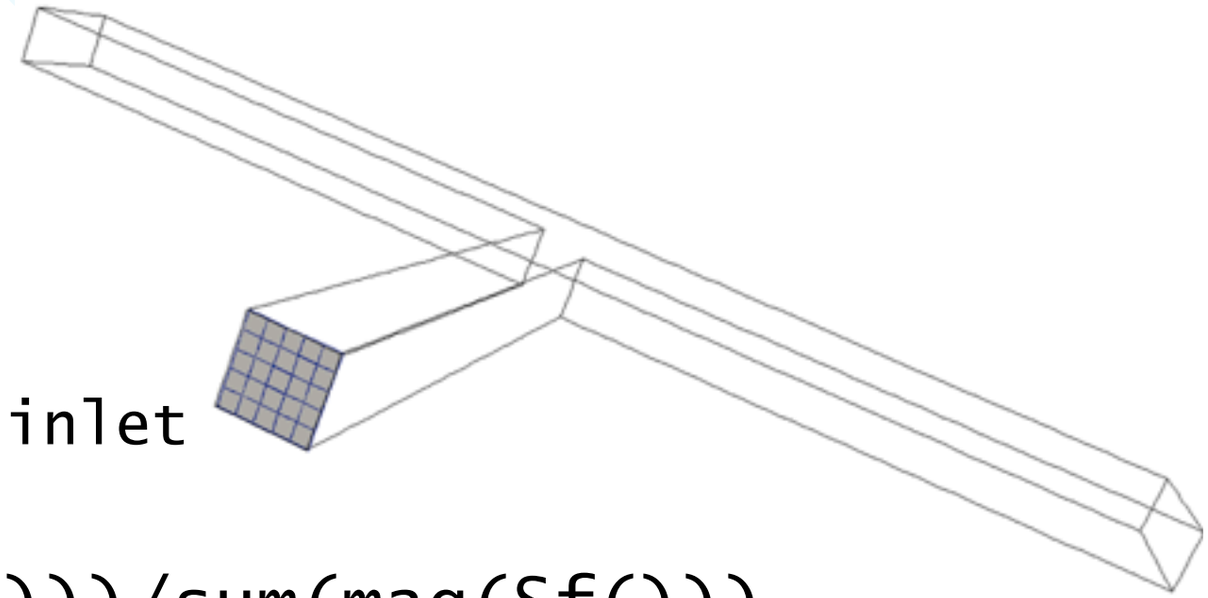
variablesで定義した変数が使える。

```
value                uniform 100010;
```
ここで指定された値は初期値としてのみ使用され、その後は意味を持たない。

```
}
```

groovyBCで定義される関数例と適用例

- **Sf** : 界面の面積ベクトル (さらに mag を取ると界面の面積)
- **mag** : 大きさ (スカラーとベクトル共に)
- **sum** : 積算値 (スカラーのみ)
- **min, max** : 最小値、最大値 (スカラーのみ)
- **normal** : 界面の法線ベクトル
- **pos** : 界面の中心座標ベクトル
- **pts** : 節点の座標ベクトル
- **time** : 時刻




```
pInlet{inlet}=sum(p*mag(Sf()))/sum(mag(Sf()))
```

inletパッチの圧力の平均値

inletパッチにおける圧力×界面面積の積算

inletパッチの面積

groovyBCを使用した境界条件 その3

Examples/groovyBC/wobbler/0/D：応力解析での変位

```
type groovyBC;
timelines (                                  timelinesにより時系列データが定義できる
    {
        name impulse;                        時系列データが代入される変数名が定義できる
        outOfBounds clamp;                   範囲外値の処理指定(timeVarying系と同様)
        fileName "$FOAM_CASE/impulse.data";      時系列データファイル名
    }
);
valueExpression "-impulse*normal()";     時系列データを用いて関数で定義可
```

Examples/groovyBC/wobbler/impulse.data：時系列データ

```
(
    (0 0)       (時刻  値)
    (1 0.1)     ここで指定されない時刻での値は、補間されて時系列変数に代入される
(略)
)
```

groovyBCを使用した境界条件 その4

Examples/groovyBC/wobbler/0/D：応力解析での変位

```
type groovyBC;
lookuptables (                          lookuptablesにより任意の表データが定義できる
    {
        name impulseLookup;             表データが代入される変数名が定義できる
        outOfBounds clamp;              範囲外値の処理指定(timeVarying系と同様)
        fileName "$FOAM_CASE/impulse.data";          表データファイル名
    }
);
valueExpression "-impulseLookup(time())*normal()";
```

lookuptablesの変数はスカラー型のパラメータを持ち、表データを元に値が返る。

よって、時系列だけでなく、鉛直プロファイル等の位置依存データ等を記述可能

Examples/groovyBC/wobbler/impulse.data：表データ

```
(
    (0 0)       (パラメータ  値)
    (1 0.1)
(略)
```

funkySetFieldsとは

- **場の分布を関数で与えることができるユーティリティ**

- **ドキュメント**

✓swak4FoamのページにはgroovyBC自体の詳細は書かれていない

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

- **ソースにおける使用例集の場所 (**swak4Foam/Examplesの下にある)
  - ✓ FunkySetFields/setLowerHalfAndATube.funkySetFieldsDict
  - ✓ InterFoamWithSources/mixingThing/system/funkySetFieldsDict
  - ✓ other/angledDuctImplicit/system/funkySetFieldsDict
  - ✓ other/capillaryRise/system/funkySetFieldsDict

funkySetFieldsの設定例

- **設定の仕方**
  - ✓ 通常辞書ファイルで設定: system/funkySetFieldsDict
  - ✓ 簡単な設定ならコマンドラインのオプションでも設定可能

コマンドライン例：

時刻0における場Uについて内部領域の全部を(0 0 0)に設定

```
funkySetFields -time 0 -field U -expression “vector(0,0,0)”
```

時刻0における場gammaについて、以下のように設定

格子中心のx座標が1以下かつy座標が2以下の場合1、それ以外0

```
funkySetFields -time 0 -field gamma
  -expression "pos().x <= 1 && pos().y <= 2 ? 1 : 0"
```

(コマンドラインは本来一行)

funkySetFields辞書による設定例 その1

Examples/FunkySetFields/setLowerHalfAndATube.funkySetFieldsDict :

```
    initLower                                   各設定の名前(任意)
    {
        field lowerPatch;                       設定する場の名前
        create true;                            場を新規に作成する場合true(通常false)
        expression "1";                         場に設定する値　(様々な関数式使用可)
        condition "pos().y<0";                  場に設定する条件(様々な関数式使用可)
        valuePatches ( zminus );                fixedValue値を設定するパッチ名のリスト
        dimension [0 1 -1 0 0];                 場の次元(新規に作成する場合に必要)
    }
    clearLower
    {
        field lowerPatch;                       createが無指定なので、既存の場
        expression "0";
        keepPatches true;                       元々のパッチの状態を保持する場合true
    }                                           (通常はtype zeroGradientになる)
```

funkySetFields辞書による設定例 その2

Examples/other/angledDuctImplicit/system/funkySetFieldsDict :

```
velClass
{
    field velClassFunky;
    create true;
    condition "Uy>=0";
    lookuptables (
      {
        name velClass;
        outOfBounds clamp;
        fileName "$FOAM_CASE/velClass.data";
      }
    );
    variables (
      "Usize=mag(U);"
      "Uy=U.y;"
    );
    expression "velClass(Usize)+1";
}
```

groovyBCと同様にlookuptablesも使用可能

groovyBCと同様にvariablesで変数定義が可能であるため、表データを用いて、関数を経由した複雑な場の分布を指定することが可能。

funkySetBoundaryFieldsとは

- **境界パッチの分布を関数で与えることができるユーティリティ**
  (funkySetFieldsと異なり場の内部は修正しない)

- **ドキュメント**
  - ✓swak4Foamのページに簡単な説明があるのみ
    http://openfoamwiki.net/index.php/Contrib_funkySetFields
  - ✓ソースのREADMEにも詳しい説明は無いので、以下の例かソースコードを参考にするしかない

- **ソースにおける使用例の場所 (**swak4Foam/Examplesの下にある)
  - ✓ FunkySetBoundaryFields/funkySetBoundaryDict.dambreak

funkySetBoundaryFieldsの設定例

tutorials/multiphase/interFoam/laminar/damBreak に適用できる例

Examples/FunkySetBoundaryFields/funkySetBoundaryDict.dambreak :

```
velocities {                ブロック名(任意)
  field U;                  書き換える場の名前
  expressions
  (
    {
      target value; 修正又は追加するターゲット名
                    (ターゲットが既存の場合は修正し、無い場合は追加される)
      patchName leftWall; パッチ名
      variables "maxY=max(pts().y);thres=0.25*maxY;";     変数定義
      expression "(pos().y<thres) ? vector(1,0,0)*(maxY-
pos().y) : vector(0,0,0)";                修正又は追加されるターゲット値
    }
  );
}
```

swakFunctionObjects系ライブラリとは

- **関数を用いた計算結果の後処理やモニター等が行える動的ライブラリ**

- **ドキュメント**

  ✓swak4Foamのページには詳細は書かれておらず、 Bernhard による 6th OpenFOAM Workshopの発表スライドにもあまり書かれていないので、以下の使用例を見るか、ソースを見るかしかない

- **ソースにおける使用例集の場所 (**swak4Foam/Examplesの下にある)

  ✓ other/angledDuctImplicit/system/controlDict

  ✓等々 (swakFunctionObjects系 を使っているケースは多数ある)

swakFunctionObjects系の使用例その1

Examples/other/angledDuctImplicit/system/controlDict :

```
libs ( "libsimpleSwakFunctionObjects.so"
       "libswakFunctionObjects.so" );
```

どちらも通常はソルバーにリンクされていないライブラリなので、動的にリンクする

```
pressureDrop   ブロック名(任意)
{   type patchExpression; ファンクション型名(パッチの値モニター)
  variables ( "pOut{patch'outlet}=sum(p*area())/sum(area());");
                変数の定義 'outletパッチでのpの(面積加重)平均値をpOutとする
  accumulations ( average ); 演算タイプ(min, max, sum, average)
  patches ( inlet );         パッチ名リスト
  expression "p-pOut";       出力演算式
            (inletの圧力平均値とoutletの圧力平均値の差、つまり圧力損失)
  verbose true;              ログに出力する時はtrue(ファイルには出る)
```

結果は”ファンクション型名_ブロック名/開始時刻/パッチ名”というファイルに出力される。ファイル例： patchExpression_pressureDrop/0/inlet

swakFunctionObjects系の使用例その2

```
pressureToFilter
{
  type patchExpression;
  variables "pFilter{cellZone'porosity}=average(p);";
  accumulations ( average );
  patches ( inlet outlet );
  expression "p-pFilter";
}
alternatePressureToFilter
{
  type swakExpression;
  valueType patch;
  patchName outlet;
  variables "pFilter{cellZone'porosity}=average(p);";
  accumulations ( average );
  expression "p-pFilter";
}
```

パッチ名リスト(複数パッチ可)

結果ファイルは”ファンクション型名_ブロック名/開始時刻/パッチ名”

patchExpressureより複雑な領域の型を処理できる

パッチ名(複数パッチ不可)

結果ファイルは”ファンクション型名_ブロック名/開始時刻/ブロック名”

swakFunctionObjects系の使用例その3

Examples/other/angledDuctImplicit/system/controlDict :

```
yPlusField  ブロック名(任意)
{
  type expressionField;    ファンクション型名(場データの演算後出力)
  outputControl timeStep; 出力制御
                          (timeStep:時刻反復毎、outputTime: データ出力時)
  outputInterval 1;  outputControlがtimeStepの時の出力インターバル
  fieldName yPlus;    出力する場の名称
  expression "pow(0.09,0.25)*sqr(k)*nearDist()/mu"; 演算式
  autowrite true;     自動的に出力する時はtrue
}
```

演算結果の場は、通常の計算結果が出力される時刻ディレクトリに、通常の場データと同様の形式で出力される。ファイル例: 10/yPlus

よって、paraView等を用いて通常の出力場と同様に可視化もできる。

Thank you

OpenCAE