

# 第9回OpenFOAM勉強会for beginner

## ソルバー改良事例発表

id:oga\_shin

小縣信也

ogata shinya

- ・ 事例① laplacianFoamの改良  
多種材料対応
- ・ 事例② laplacianFoamの改良  
温度依存熱伝導率の設定

## ジロ・デ・解析屋

<http://giropenfoam.web.fc2.com/openfoam/basic/basic.html>

The screenshot shows a web browser window with a terminal-like interface. On the left, there's a sidebar with a 'メインリスト' (Main List) link. The main content area is divided into two columns. The left column contains a terminal window with the following text:

```
1 Created by SharpDevelop.
2
3 User: 6gi87jiro
4 Date: 2005/12/02
5 Time: 12:12
6
7 Imports System
8 Imports System
9 Imports System
10
11 Imports System
12 Imports Micro
```

The right column contains a terminal window with the following text:

```
INIT MASTER(S)
ID GiroD'Kaisekiya
SOL 101
TIME 10000
CEND
ECHO = NONE
DISPLACEMENT(PLOT) = ALL
SPCFORCE(PLOT) = ALL
```

In the center, there's a QR code and a text overlay that reads: '解析屋の険しい道のりを皆で一緒に 超えるのか転がるのか...' (Together, we will overcome the steep path of the analysis house, whether by climbing or falling...). Below the QR code, it says 'Powered by FC2携帯変換' and '閉じる' (Close).

The main article is titled 'ソルバー改造の基礎' (Basics of Solver Modification). The text reads:

**はじめに**

本当に微々たる改造しかしてませんか (私の夫刀が・・・)、私のほうで実施している改造について質問がくる事があります。スロッシングの所で実施している重力項の取り扱いとか。そこで、ホントに極基本的な点についてのみですが、改造の手引きもどき程度の内容を以下に記述してみる事としました。皆さん、がんばっているんな改造を施し、OpenFOAMを活用してみてください。

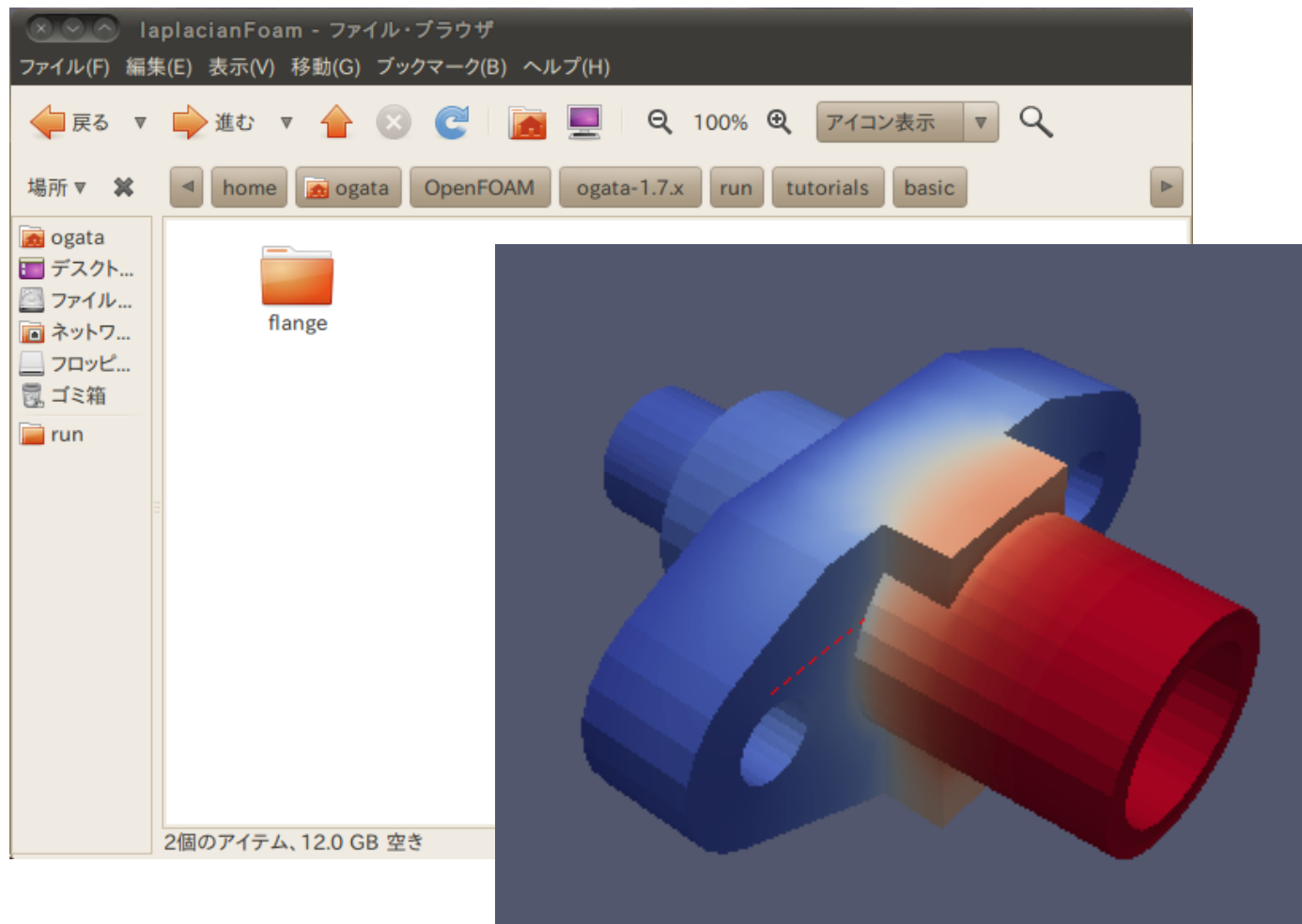
**導入：OpenFOAMによる熱伝導解析 (laplacianFoam) での解析手法の確認**

ホント少しですが、ソルバーに手を入れたい事があると思います。ソースコードが余りにも膨大で、全部を把握する事はできませんが、単純な改造であれば、全部を把握する必要が無い作りになっているのがOpenFOAMの売りの一つであるはずで。

以下、最も単純なlaplacianFoamに対して、その構成を (何となく) 理解すると共に、ほんの少しだけ拡張改造してみます。

まずは、「laplacianFoam=熱伝導解析」で実施している解析について、基本的な所から押さえていきます。

# laplacianFoamのケースフォルダ



tutorials/basic/laplacianFoam/flange

laplacianFoamを使うと非定常の熱伝導方程式が解ける。

# 熱伝導方程式

**熱拡散率 (温度伝導率) : thermal diffusivity**

一次元の熱伝導方程式を考える。

$$\rho c \frac{\partial T}{\partial t} = \lambda \frac{\partial^2 T}{\partial x^2}$$

$T$  : 温度、 $t$  : 時間、 $x$  : 距離

$\rho$  : 密度、 $c$  : 比熱、 $\lambda$  : 熱伝導率

熱拡散率  $\alpha$  を使用すると、以下の形で表現される。

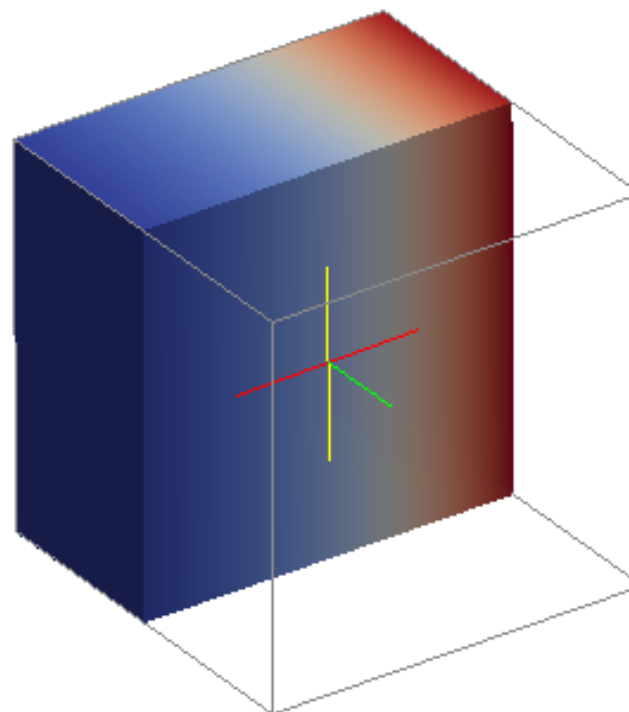
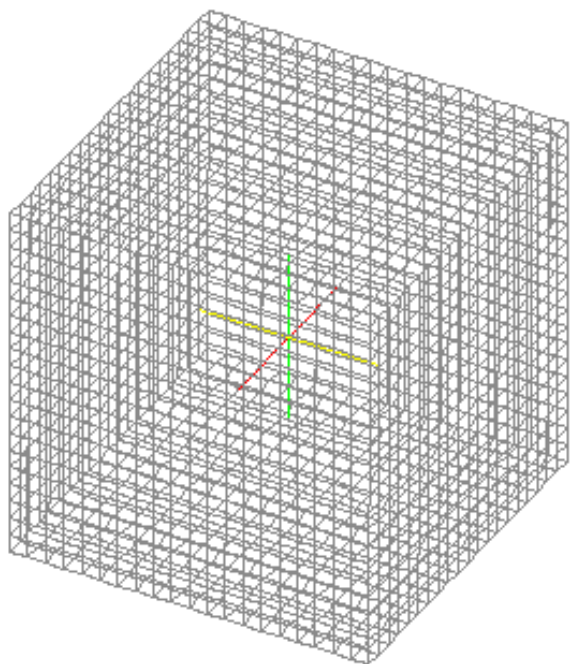
$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad , \quad \boxed{\alpha} = \frac{\lambda}{\rho c}$$

↓  
OpenFOAMではDTになっている

# laplacianFoamの計算

端末に以下を入力

```
run  
cd box  
blockMesh  
laplacianFoam  
paraFoam
```



## blockMeshDict 1/2

```
convertToMeters 1;

vertices
(
  (0 0 0)
  (1 0 0)
  (1 1 0)
  (0 1 0)
  (0 0 1)
  (1 0 1)
  (1 1 1)
  (0 1 1)
);

blocks
(
  hex (0 1 2 3 4 5 6 7) (20 20 20) simpleGrading (1 1 1)
);
```

## blockMeshDict 2/2

```
patches
(
  wall Walls
  (
    (3 7 6 2)
    (0 3 2 1)
    (4 5 6 7)
    (1 5 4 0)
  )
  patch HotWall
  (
    (2 6 5 1)
  )
  patch ColdWall
  (
    (0 4 7 3)
  )
);
```



# transproperties

```
FoamFile
{
  version    2.0;
  format     ascii;
  class      dictionary;
  location   "constant";
  object     transportProperties;
}
// *****

DT [ 0 2 -1 0 0 0 0 ] 4e-04; 熱拡散率
```

```
dimensions [0 0 0 1 0 0 0];  
internalField uniform 273; 初期値0°C  
boundaryField  
{  
  Walls  
  {  
    type zeroGradient; 断熱境界  
  }  
  HotWall  
  {  
    type fixedValue;  
    value uniform 373; 100°C固定  
  }  
  ColdWall  
  {  
    type fixedValue;  
    value uniform 273; 0°C固定  
  }  
}
```

## laplacianFoam改良①

多種材料の熱伝導計算ができるようにlaplacinFoamを改良

## laplacianFoamのソースの場所

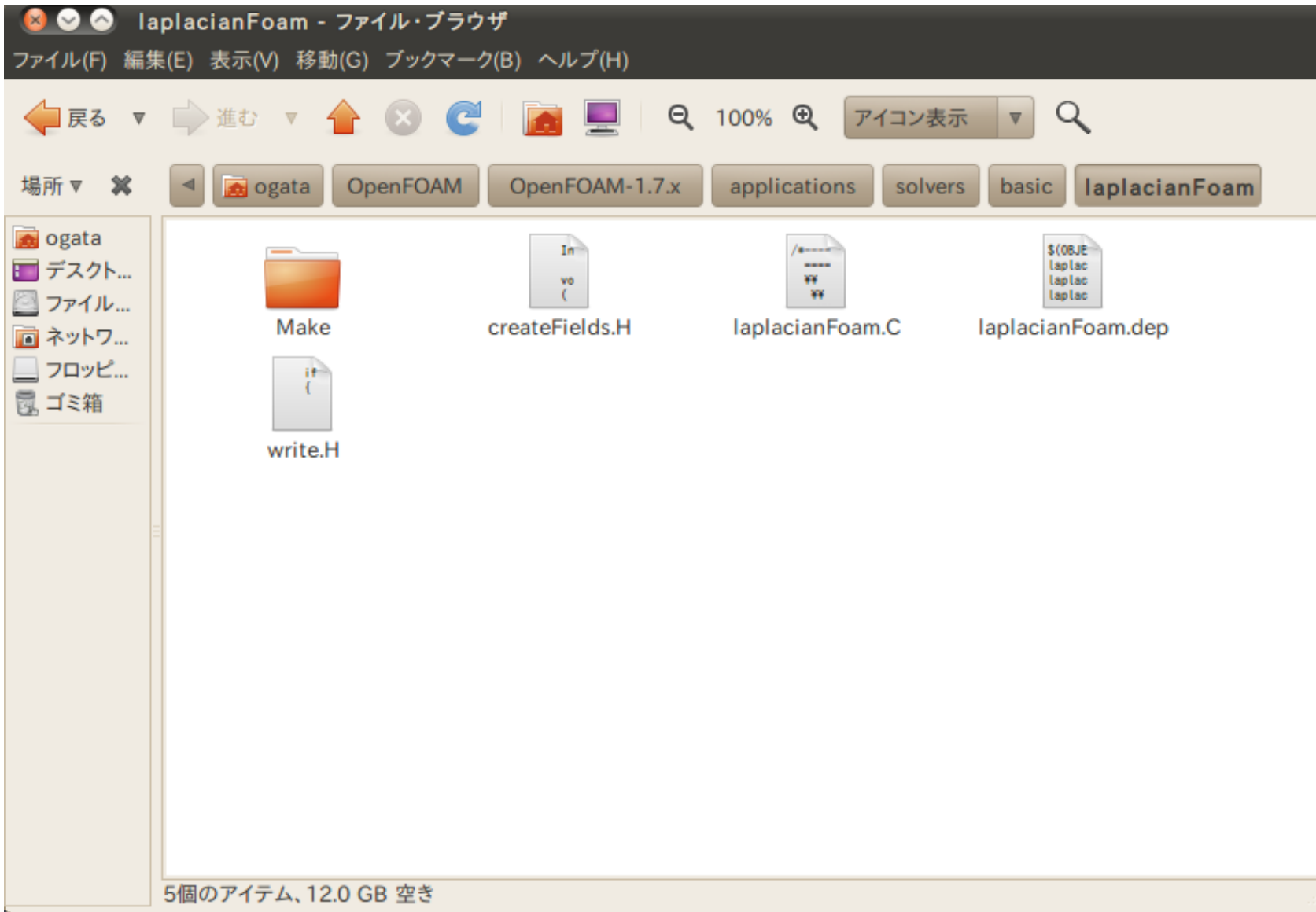
端末

```
app
find -name laplacianFoam
./bin/linuxGccDPOpt/laplacianFoam
./solvers/basic/laplacianFoam

cd solvers/basic/
ls
laplacianFoam potentialFoam scalarTransportFoam

cd laplacianFoam
ls
Make createFields.H laplacianFoam.C write.H
```

# laplacianFoamの構成ファイル



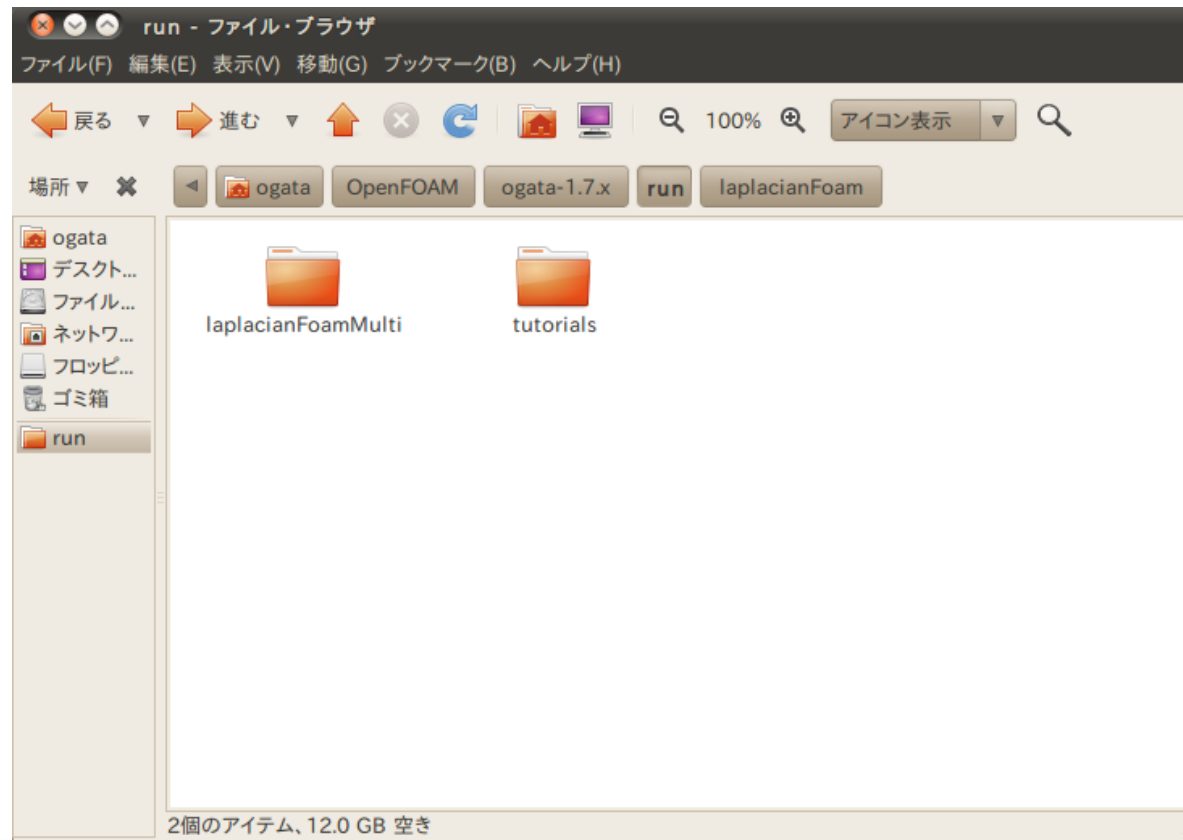
## laplacianFoamのソースをコピー

```
cd ..
```

```
ls
```

```
laplacianFoam potentialFoam scalarTransportFoam
```

```
cp -r laplacianFoam $FOAM_RUN/laplacianFoamMulti
```



# laplacianFoamMultiの構成ファイル



## laplacianFoam.C

```
for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
{
    solve
    (
        fvm::ddt(T) - fvm::laplacian(DT, T)
    );
}
```

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial x^2} = 0$$

熱伝導方程式は、solveの中に記述されている。  
基礎式と見比べる事で、DT=熱拡散率だと理解することができる。  
DTは、createFields.H で定義されている。



# createFields.H

```
Info<< "Reading field T\n" << endl;
```

```
volScalarField T  
(  
    IOobject  
    (  
        "T",  
        runTime.timeName(),  
        mesh,  
        IOobject::MUST_READ,  
        IOobject::AUTO_WRITE  
    ),  
    mesh  
);
```

略

```
Info<< "Reading diffusivity DT\n" << endl;
```

```
dimensionedScalar DT  
(  
    transportProperties.lookup("DT")  
);
```

温度 T については、“MUST\_READ”で  
“AUTO\_WRITE”になっている。  
毎ステップ、物理量の読み書きが行われ  
るという意味である  
これより、任意の物理量を入出力したい  
場合、createFields.H で宣言して、“  
MUST\_READ”“AUTO\_WRITE”に設定す  
ればよいと推測できる。

DTについては、transportProperties  
で読み出すだけになっている。

このコードだと、同一材料で構成された  
物モデルしか解けない。

## createFields.Hの変更

```
/*
dimensionedScalar DT
(
    transportProperties.lookup("DT")
);
*/

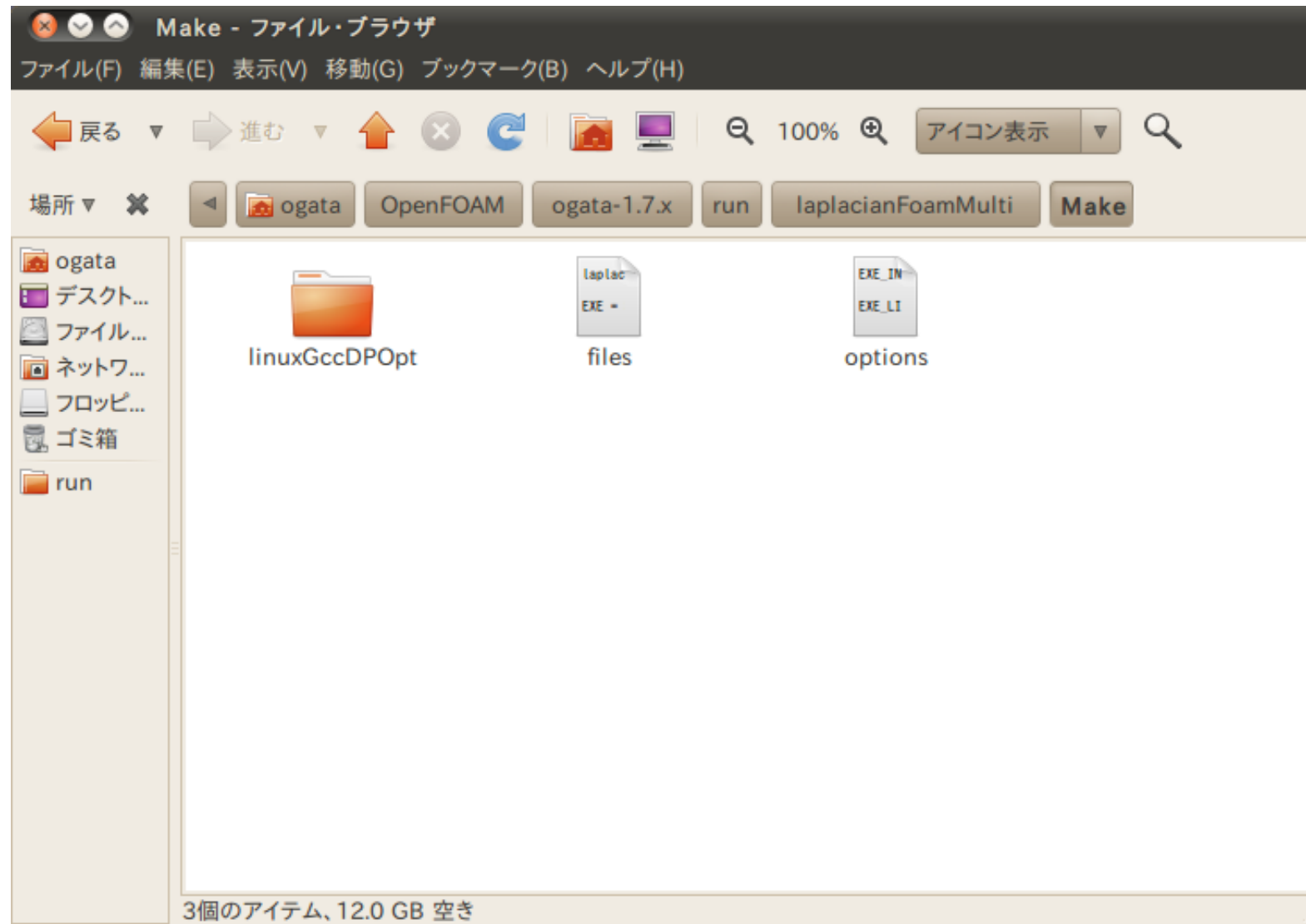
volScalarField DT
(
    IOobject
    (
        "DT",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::NO_WRITE
    ),
    mesh
);
```

既存のDT定義部分はコメントアウト

volScalarField Tをコピーし、  
volScalarField DTを作る。

“DT”という物理量ファイルを毎ステップ読み込むようにする。  
“DT”は時間によって変化しないので、  
“NO\_WRITE”でよい。

# laplacianFoamMulti/Make/files



## filesの変更

laplacianFoamMulti.C

```
EXE = $(FOAM_USER_APPBIN)/laplacianFoamMulti
```

既存ソルバーとの混在をさけるため、  
ユーザーディレクトリにコンパイルする。

FOAM\_USER\_APPBINの場所

```
ogashin@node12:~/OpenFOAM/ogashin-1.7.0/applications/bin/linuxGccDPOpt
```

## wmake

wmakeする前に、linuxGccDPOptディレクトリの所有者を自分にしておく。

```
app
sudo su
chmod -R username bin/linuxGccDPOpt/
run
cd laplacianFoamMulti
wmake
```

※DEXCS2010\_OF17xでは、src/finiteVolume/fieldsが存在しないため、wmake時にエラーが出る。

以下通りになれば成功！

```
laplacianFoamMulti -help
Usage: laplacianFoamMulti [-parallel] [-case dir] [-
help] [-doc] [-srcDoc]
```

## laplacianFoamMulti用ケースディレクトリの作成

- ①boxディレクトリをコピーし、box\_multiディレクトリを作成。
- ②0.orgの中のTをコピーし、DTファイルを作成。

```
cp -r box box_multi  
cd box_multi/0.org  
cp -r T DT
```

# 0/DT

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        volScalarField;
  object       DT;
}
// ***** //
dimensions    [ 0 2 -1 0 0 0 ];
internalField uniform 1e-03;
boundaryField
{
  Walls
  {
    type      fixedValue;
    value     uniform 1e-03;
  }
  HotWall
  {
    type      fixedValue;
    value     uniform 1e-03;
  }
  ColdWall
  {
    type      fixedValue;
    value     uniform 1e-03;
  }
}
```

→ setFieldsによって再定義される。

## setFieldsDictのコピー

複数のDTの定義は、setFieldsで行う。  
まず、setFieldsDictを探し、コピーしてくる。

```
util
find -name setFieldsDict
./preProcessing/setFields/setFieldsDict

cd preProcessing/setFields/
cp -r setFieldsDict $FOAM_RUN/box_multi/system
run
cd box_multi/system
ls
controlDict fvSchemes fvSolution setFieldsDict
```



# setFieldsDict

```
defaultFieldValues
(
    volScalarFieldValue DT 1e-03
);

regions
(
    boxToCell
    {
        box (0.3 0.3 0.3) (0.7 0.7 0.7);

        fieldValues
        (
            volScalarFieldValue DT 1e-06
        );
    }
);
```

初期値

領域を指定

その領域内のDTを指定

# setFields実行

端末

```
blockMesh
setFields
```

0/DT

```
dimensions      [0 2 -1 0 0 0 0];

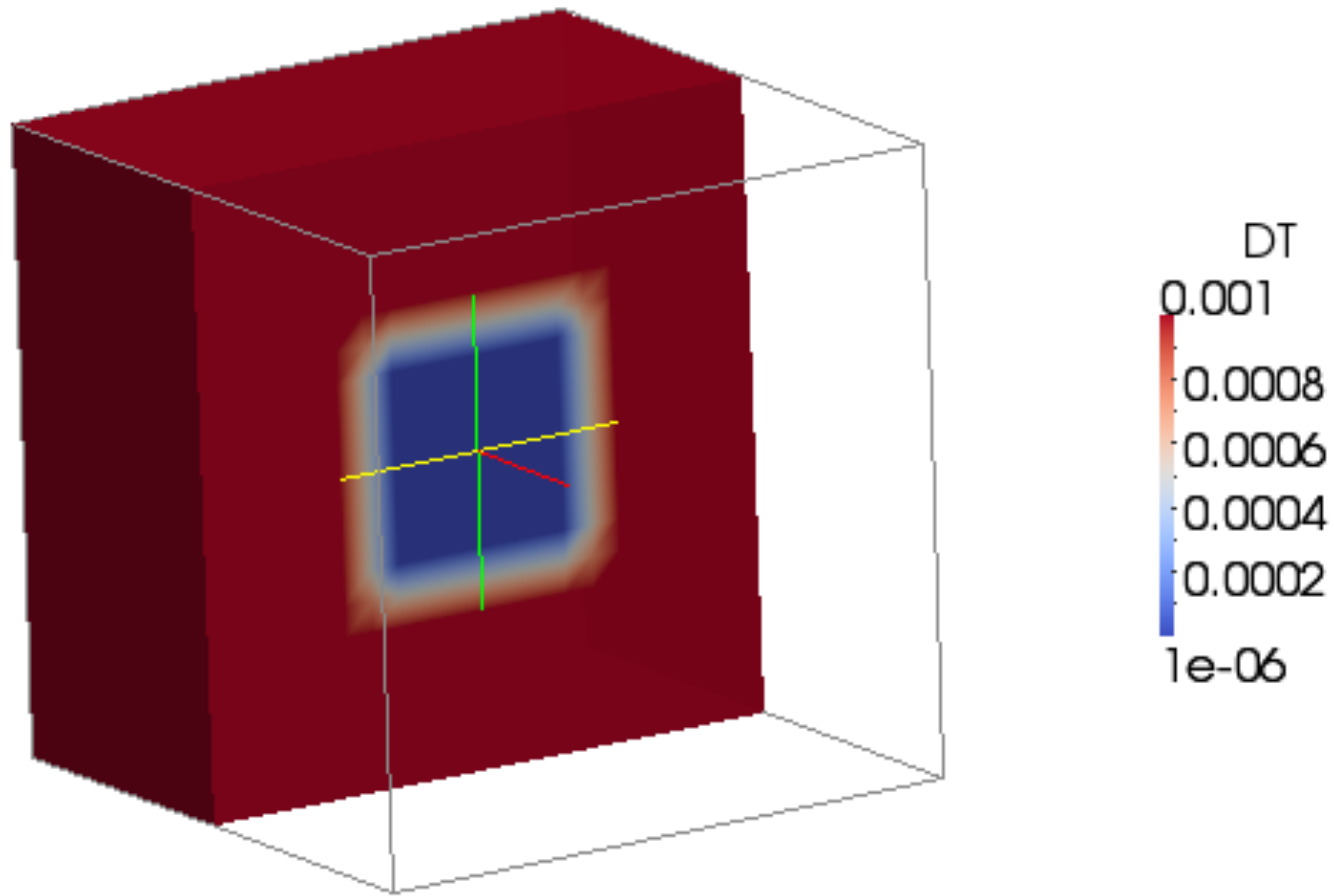
internalField   nonuniform List<scalar>
8000
(
0.01
0.01
0.01
0.01

略

0.01
0.01
0.01
1e-06
1e-06
1e-06
1e-06
1e-06
```

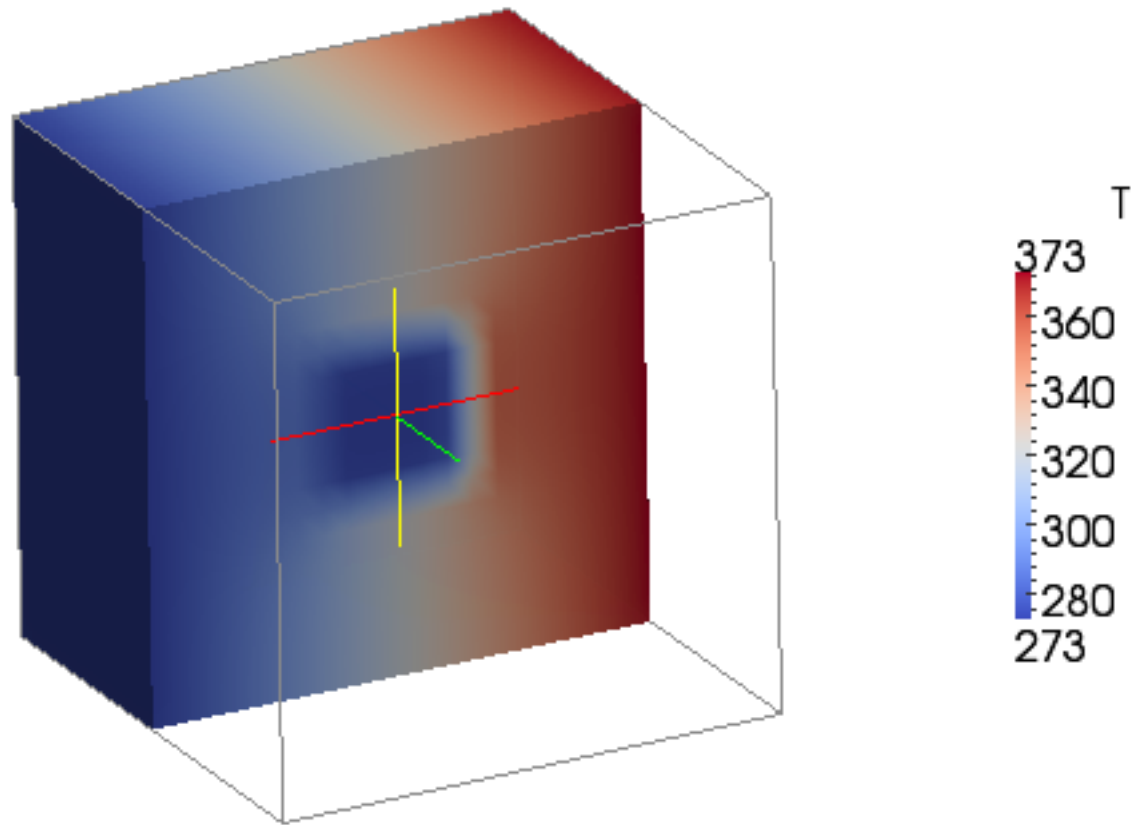
# DT分布の表示

paraFoam



# laplacianFoamMulti 計算実行&結果

laplacianFoamMulti



## laplacianFoam改良②

温度依存性熱伝導率を設定できるようにlaplacianFoamを改良

## laplacianFoamのソースのコピー

```
app
cd solvers/basic/
ls
laplacianFoam potentialFoam scalarTransportFoam

cp -r laplacianFoam $FOAM_RUN/laplacianFoamDepT
```

# laplacianFoamDepTの構成ファイル



ファイル名の変更  
laplaciaFoam.C → laplacianFoamDepT.C

## laplacianFoamDepT.C

```
for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
{
    DT=A1*T*T+A2*T+A3;
    solve
    (
        fvm::ddt(T) - fvm::laplacian(DT, T)
    );
}
```

solveの手前に1行追加。熱伝導方程式を解く前にDTを計算することになる。  
熱伝導率(拡散率)は、以下の二次式に従うと考えている。

$$y = ax^2 + bx + c$$



# laplacianFoamDepT/createFields.H

```
dimensionedScalar A1
(
    transportProperties.lookup("A1")
);
dimensionedScalar A2
(
    transportProperties.lookup("A2")
);
dimensionedScalar A3
(
    transportProperties.lookup("A3")
);

Info<< "Reading diffusivity DT\n" << endl;
volScalarField DT
(
    IOobject
    (
        "DT",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    A1*T*T+A2*T+A3
);
```

まず、A1,A2,A3の係数を読む。  
A1,A2,A3は、transProperties  
で定義。

volScalarField Tをコピーし、  
volScalarField DTに変更

DTは、毎ステップ計算をおこな  
うので、“NO\_READ”  
AUTO\_WRITE”。

初期値の設定

## laplacianFoamDepT/Make/files

laplacianFoamDepT.C

EXE = \$(FOAM\_USER\_APPBIN)/laplacianFoamDepT

# wmake

```
cd laplacianFoamDepT  
wmake
```

以下通りになれば成功！

```
laplacianFoamDepT -help  
Usage: laplacianFoamDepT [-parallel] [-case dir] [-  
help] [-doc] [-srcDoc]
```

## laplacianFoamDepT用ケースディレクトリの作成

- ①boxディレクトリをコピーし、box\_DepTディレクトリを作成。
- ②0.orgの中のTをコピーし、DTファイルを作成。

```
cp -r box box_DepT  
cd box_DepT/0.org  
cp -r T DT
```

# constant/transProperties

```
FoamFile
{
  version      2.0;
  format       ascii;
  class        dictionary;
  location     "constant";
  object       transportProperties;
}
// ***** //

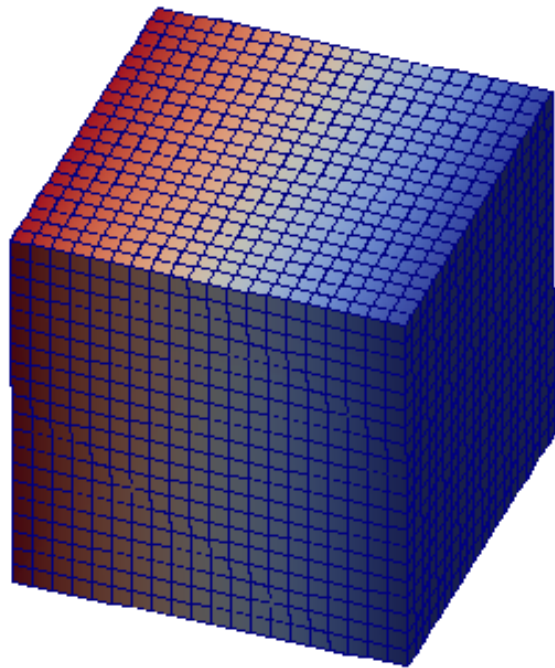
DT          DT [ 0 2 -1 0 0 0 0 ] 4e-05;
A1          A1 [ 0 2 -1 -2 0 0 0 ] 1e-08;
A2          A2 [ 0 2 -1 -1 0 0 0 ] -3e-05;
A3          A3 [ 0 2 -1 0 0 0 0 ] 0.0245;

// ***** //
```

# laplacianFoamDepT 計算実行&結果

laplacianFoamDepT  
paraFoam

DT



T

