OpenFOAMによる
電子機器シミュレーション
その1

オープンCAE学会　　西　剛伺

電子機器では何が必要か？

- bloackMeshDictによるモデル作成
- laplacianFoamソルバーの確認
- 発熱源の設定（均一発熱）
- 境界条件の設定（第三種境界条件）
- 異方性熱伝導率への対応
- 異方性熱伝導率を用いる際のソルバー記述（laplacianのままで良いのか？）

BlockMeshDictを記述する

```
vertices
(
    (-25 -25 0.0)                   // 0
    ( 25 -25 0.0)                   // 1
    ( 25  25 0.0)                   // 2
    (-25  25 0.0)                   // 3
    (-25 -25 5.0)                   // 4
    ( 25 -25 5.0)                   // 5
    ( 25  25 5.0)                   // 6
    (-25  25 5.0)                   // 7
);

blocks
(
    hex (0 1 2 3 4 5 6 7) (50 50 10) simpleGrading (1 1 1)

);
```

```
patches
(
    patch BottomSide
    (
        (0 1 2 3)
    )
    patch TopSide
    (
        (4 5 6 7)
    )
);
```

ソルバー（laplacianFoam.C）の記述

```
solve
(
        fvm::ddt(T) - fvm::laplacian(DT, T)
);
```

$$\frac{\partial T}{\partial t} - DT\nabla^2 T = 0$$

つまり，DTは熱拡散率である．

$$DT = \frac{k}{c\rho}$$

熱源の設定

```
solve
(
        fvm::ddt(T) - fvm::laplacian(DT, T)
        - 発熱
);
```

$$\frac{\partial T}{\partial t} - DT\nabla^2 T - \frac{\dot{q}_v}{c\rho} = 0$$

$\dot{q}_v$：単位体積当たりの発熱量 $[W/m^3]$

もしくは面発熱にするなら，

```
solve
(
        fvm::ddt(T) - fvm::laplacian(DT, T)
        - 発熱
);
```

$$\frac{\partial T}{\partial t} - DT\nabla^2 T - \frac{\dot{q}}{c\rho\partial z} = 0$$

$\dot{q}$：単位面積当たりの発熱量 $[W/m^2]$

zはメッシュサイズに合わせて計算する.

熱源の設定

```
solve
(
    fvm::ddt(T) - fvm::laplacian(DT, T)
    - ST
);
```

ソルバーの式に発熱項の“変数”の記述を追加すれば良い.

STという“変数”をcreateFields.Hに定義.
値をsystem/setFieldsDictで設定.

createFields.H

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

setFieldsDict

```
defaultFieldValues
(
                volScalarFieldValue ST 0
);
regions
(
    boxToCell
    {
        box (x- y- z-) (x+ y+ z+);
        fieldValues
        (
                    volScalarFieldValue ST        xxx
        );
    }
                :
```

境界条件の設定(swak4foam/groovyBC)

熱伝達境界を設定する: $-k_z \frac{\partial T}{\partial z} = h(T - T_{ambient})$

0/T

```
boundaryField
{
      :
   TopSide
   {
      type                    groovyBC;
      value                   uniform 300;                    // Initial value of T
      gradientExpression      "gradT";
      fractionExpression      "0";
      variables "Tout=300;h_conv=200;cond=30;gradT=h_conv*(Tout-internalField(T))/cond;";
   }
}
```

controlDict

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

異方性熱伝導率の使用

```
solve
(
        fvm::ddt(T) - fvm::laplacian(DT, T)
);
```

$$\frac{\partial T}{\partial t} - \nabla(DT(\nabla T)) = 0$$

上記の式は熱伝導率kを等方性として扱っている（$k_x=k_y=k_z$）.

$$\frac{\partial T}{\partial t} - \frac{k}{c\rho}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right) = 0 \qquad DT = \frac{k}{c\rho}$$

電子機器では有効熱伝導率を用いて材質を簡略化してモデル化すること
が多い．つまり，熱伝導率は異方性として扱う必要がある（$k_x \neq k_y \neq k_z$）.

$$\frac{\partial T}{\partial t} - \frac{1}{c\rho}\left(\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right)\right) = 0$$

それでは，DTはどのように設定すれば良いのか？

異方性熱伝導率の使用

DTという“テンソル変数”を
createFields.Hに定義.
値をsystem/setFieldsDictで設定.

volScalarield DT
(
    volSymmTensorField DT
    (
        IOobject
        (
            "DT",
            runTime.timeName(),
            mesh,
            IOobject::MUST_READ,
            IOobject::AUTO_WRITE
        ),
        mesh
    );
);

transportPropertiesの
記述からはDTを削除.

setFieldsDict

```
defaultFieldValues
(
        volScalarFieldValue ST 0
        volSymmTensorFieldValue DT (xxx 0 0 xxx 0 xxx)
);

regions
(
   boxToCell
   {
      box (x- y- z-) (x+ y+ z+);

      fieldValues
      (
            volSymmTensorFieldValue DT (xxx 0 0 yyy 0 zzz)
      );
   }
            :
```

＜ソルバ記述＞laplacianのままで良いのか

2.1.4 Laplacian
The Laplacian is an operation that can be defined mathematically by a combination of the divergence and gradient operators by $\nabla^2 \equiv \nabla \bullet \nabla$.
However, **the Laplacian should be considered as a single operation that transforms a tensor field into another tensor field of the same rank**, rather than a combination of two operations, one which raises the rank by 1 and one which reduces the rank by 1.

Gammaはテンソルでも良いということでそのまま異方性熱伝導率を持つ場合にも使用できるはず！！

From
“Open∇FOAM The Open Source CFD Toolbox Programmer’s Guide”, Version 2.1.1, 16th May 2012.

Term description	Implicit / Explicit	Text expression	fvm::/fvc:: functions
Laplacian	Imp/Exp	$\nabla^2\phi$ $\nabla \bullet \Gamma\nabla\phi$	laplacian(phi) laplacian(Gamma, phi)
Time derivative	Imp/Exp	$\frac{\partial\phi}{\partial t}$ $\frac{\partial\rho\phi}{\partial t}$	ddt(phi) ddt(rho,phi)
Second time derivative	Imp/Exp	$\frac{\partial}{\partial t}\left(\rho\frac{\partial\phi}{\partial t}\right)$	d2dt2(rho, phi)
Convection	Imp/Exp	$\nabla \bullet (\psi)$ $\nabla \bullet (\psi\phi)$	div(psi,scheme)* div(psi, phi, word)* div(psi, phi)
Divergence	Exp	$\nabla \bullet \chi$	div(chi)
Gradient	Exp	$\nabla\chi$ $\nabla\phi$	grad(chi) gGrad(phi) lsGrad(phi) snGrad(phi) snGradCorrection(phi)
Grad-grad squared	Exp	$\lvert\nabla\nabla\phi\rvert^2$	sqrGradGrad(phi)
Curl	Exp	$\nabla \times \phi$	curl(phi)
Source	Imp Imp/Exp†	$\rho\phi$	Sp(rho,phi) SuSp(rho,phi)

†fvm::SuSp source is discretised implicit or explicit depending on the sign of rho.
†An explicit source can be introduced simply as a vol<Type>Field, *e.g.*rho*phi.
Function arguments can be of the following classes:
phi: vol<Type>Field
Gamma: scalar volScalarField, surfaceScalarField, volTensorField, surfaceTensorField.
rho: scalar, volScalarField
psi: surfaceScalarField.
chi: surface<Type>Field, vol<Type>Field.

Table 2.2: Discretisation of PDE terms in OpenFOAM

検証数値実験

異方性熱伝導率を用いた際にfvm::laplacian()によって正しく演算できるか確認するため，laplacianFoamにソース項を追加したソルバ（以下，ソルバA）

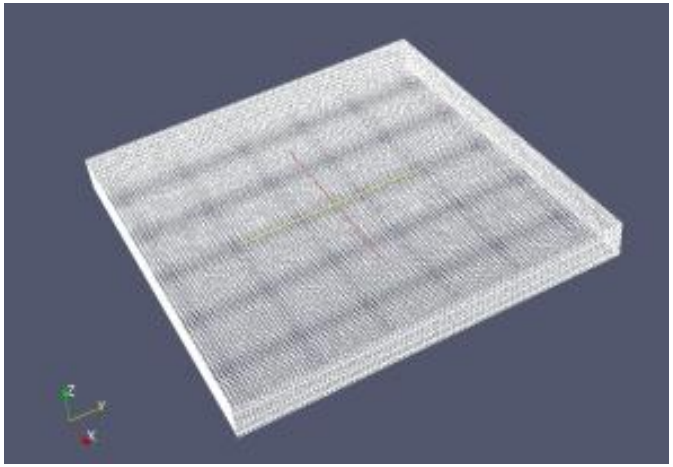
$$\frac{\partial T}{\partial t} - \frac{k}{c\rho}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2}\right) - \frac{\dot{q}_v}{c\rho} = 0$$

とSymmTensor（対象テンソル）を用いたソルバ（以下，ソルバB）

$$\frac{\partial T}{\partial t} - \frac{1}{c\rho}\left(\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z \frac{\partial T}{\partial z}\right)\right) - \frac{\dot{q}_v}{c\rho} = 0$$

の結果を比較することとした.

モデルは右図のような異方性熱伝導率（kx=ky≠kz）を有するブロックの下面中央部付近の矩形領域のみ発熱するものを使用.
上面は第三種境界条件，それ以外の境界は断熱.



断面図（ピンク色の部分が発熱部）

検証数値実験　－　ソルバAを用いるための変数変換

ソルバAに関しては，変数変換

$$\xi = z\sqrt{\frac{k_{xy}}{k_z}}$$

により，

$$\frac{\partial T}{\partial t} - \frac{1}{c\rho}\left(\frac{\partial}{\partial x}\left(k_{xy}\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_{xy}\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_z\frac{\partial T}{\partial z}\right)\right) - \frac{\dot{q}_v}{c\rho} = 0$$

を

$$\frac{\partial T}{\partial t} - \frac{k_{xy}}{c\rho}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial \xi^2}\right) - \frac{\dot{q}_v}{c\rho} = 0$$

に変換して用いた（つまり，z方向の寸法を変化させることで「等価」な問題を解く）．
第三種境界条件も変数変換に合わせて値を調整．

$$-k_z\frac{\partial T}{\partial z} = h(T - T_{ambient})$$

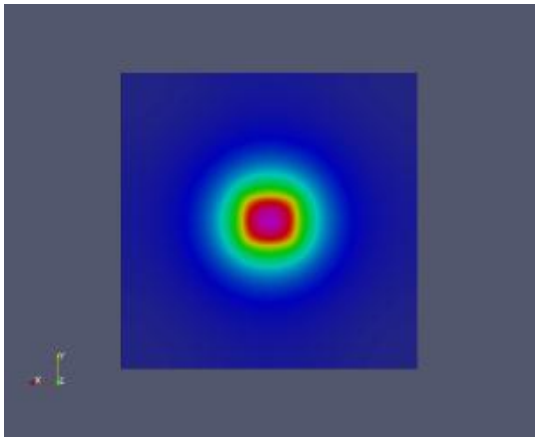
$$\Rightarrow \quad -\sqrt{k_{xy}k_z}\frac{\partial T}{\partial \xi} = h(T - T_{ambient})$$

検証数値実験　ー　結果

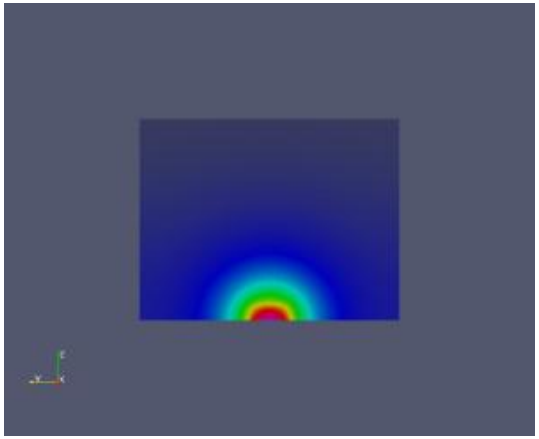
＜10秒後の結果＞　両者の結果が一致することを確認

ソルバA

底面の温度分布

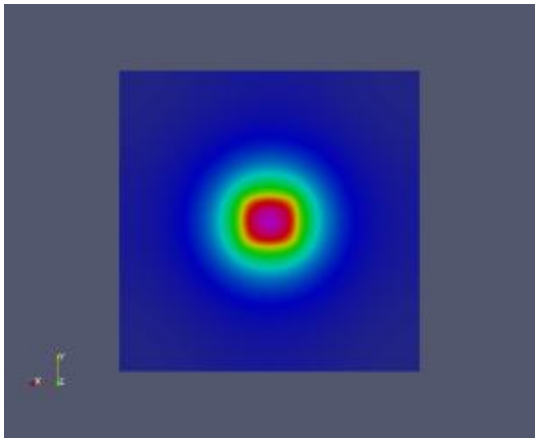


中央部断面の温度分布



ソルバB

底面の温度分布



中央部断面の温度分布