



## 第13回オープンCAE初心者勉強会 進捗報告

---

秋山善克

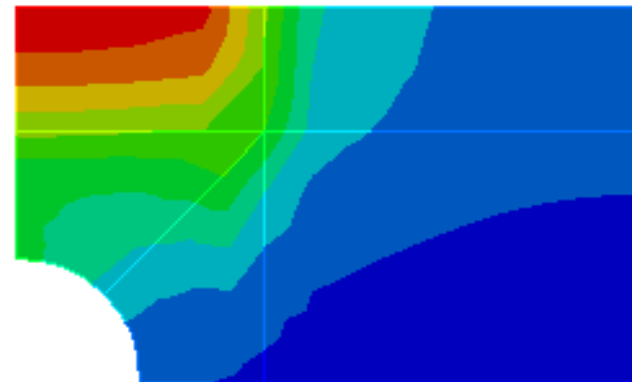
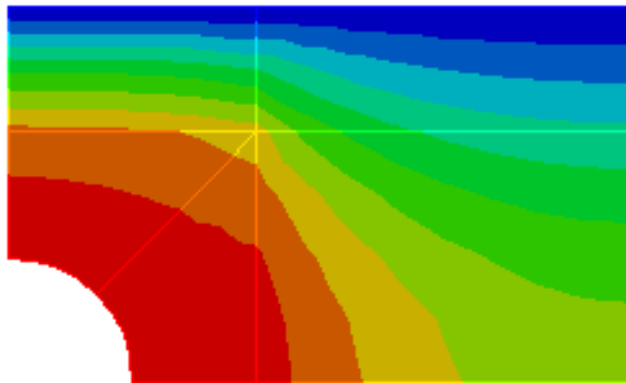
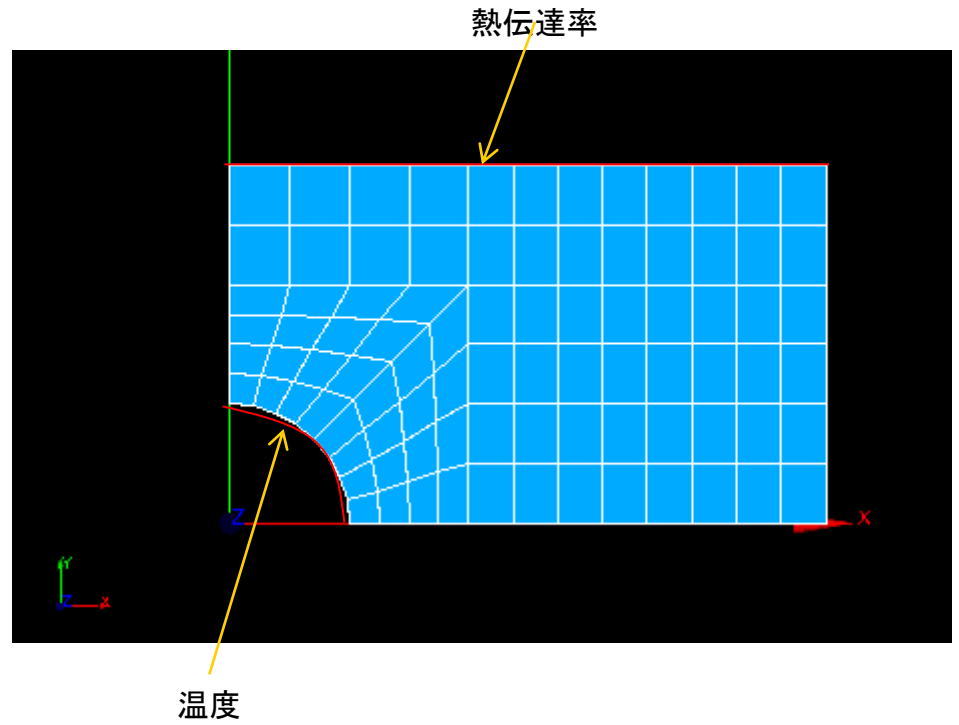
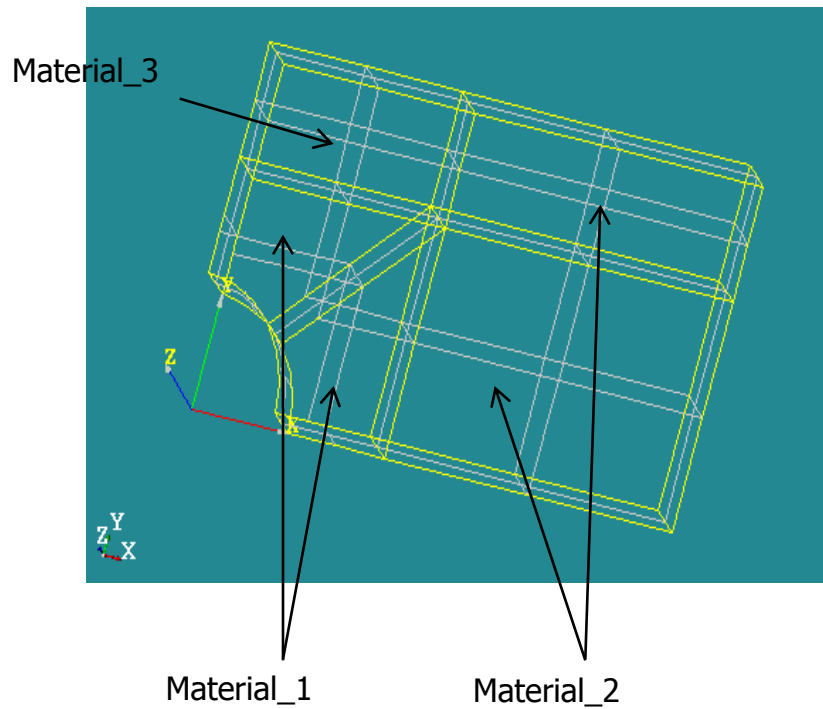


## 発表内容

---

- 熱流束出力について
- ArrheniusLaw組込時のエラーについて
- せん断発熱の組み込みについて

# 熱伝導解析



# 熱流束の出力

```
list=DEFI_LIST_REEL(DEBUT=0,  
    INTERVALLE=_F(JUSQU_A=30,  
    NOMBRE=60,,));
```

←非定常計算をするための設定(30sを60stepで計算)

```
TEMP=THER_LINEAIRE(MODELE=MODEL,  
    CHAM_MATER=MATFIELD,  
    EXCIT=_F(CHARGE=LOADING,,),  
    INCREMENT=_F(LIST_INST=list,,),  
    ETAT_INIT=_F(VALE=190,,),  
    ARCHIVAGE=_F(PAS_ARCH=2,,));
```

←非定常計算の設定

←初期値(この場合は温度)

←出力間隔(60stepを2step間隔で出力)

```
TEMP=CALC_ELEM(reuse =TEMP,  
    RESULTAT=TEMP,  
    OPTION='FLUX_ELNO',);
```

←熱流束の出力 要素出力?

```
TEMP=CALC_NO(reuse =TEMP,  
    RESULTAT=TEMP,  
    OPTION='FLUX_NOEU',);
```

←熱流束の出力 節点出力?

```
IMPR_RESU(FORMAT='MED',  
    RESU=_F(RESULTAT=TEMP,,));
```

←ウィザードからのデフォルトの出力(この場合は節点温度が出力される)

```
FIN();
```

# OpenFOAMに組み込まれている非ニュートンモデル

Src¥transportModels¥incompressible¥viscosityModels内

5

(

BirdCarreau

CrossPowerLaw

HerschelBulkley

Newtonian

powerLaw

)

BirdCarreau

$$\eta = \eta_{\infty} + (\eta_0 - \eta_{\infty}) \left\{ 1.0 + (k\dot{\gamma})^2 \right\}^{\frac{n-1.0}{2.0}}$$

CrossPowerLaw

$$\eta = \frac{(\eta_0 - \eta_{\infty})}{1.0 + (m\dot{\gamma})^n} + \eta_{\infty}$$

# CrossLawの作成

Src¥transportModels¥incompressible¥viscosityModels内のCrossPowerLawをコピー  
名前をCrossPowerLawからCrossLawに変更(フォルダ、ファイル名)

CrossLaw.C内  
粘性を計算しているところのnuInf\_を削除  
CrossPowerLawをCrossLawに置き換え

$$\text{CrossLaw } \eta = \frac{(\eta_0)}{1.0 + (m\dot{\gamma})^n}$$

CrossLaw.H内  
CrossPowerLawをCrossLawに置き換え  
dimensionedScalar nuInf\_を削除

CrossLaw.dep内  
CrossPowerLawをCrossLawに置き換え

Src¥transportModels¥incompressible¥Make内のfiles  
viscosityModels/CrossLaw/CrossLaw.Cを追加

Src¥transportModelsで./Allmake

6

(

BirdCarreau

CrossLaw ←CrossLawが追加される

CrossPowerLaw

HerschelBulkley

Newtonian

powerLaw

)



## ArrheniusLawの作成

---

Src¥transportModels¥incompressible¥viscosityModels内のCrossPowerLawをコピー  
名前をCrossPowerLawからArrheniusLawに変更(フォルダ、ファイル名)

CrossLaw.C内

```
const volVectorField& T,  
viscosityModel(name, viscosityProperties, U, T, phi), ←Tを追加
```

ArrheniusLaw.H内

```
ArrheniusLaw  
(  
    const word& name,  
    const dictionary& viscosityProperties,  
    const volVectorField& U,  
    const volVectorField& T, ←Tを追加  
    const surfaceScalarField& phi  
);
```

Src¥transportModels¥incompressibleフォルダー内  
引数にTを追加

Src¥transportModels¥incompressible¥Make内のfiles  
viscosityModels/ArrheniusLaw/ArrheniusLaw.Cを追加

Src¥transportModelsで./Allmake

※プログラムを作成したがエラーによりコンパイル不可→エラーが不明

## せん断発熱の作成

T方程式内に組み込めばよい？

熱伝導方程式 
$$\rho C_p \left( \frac{\partial T}{\partial t} + \mathbf{v} \cdot \nabla T \right) = \nabla \cdot (\kappa \nabla T) + Q$$

$$Q = \eta \dot{\gamma}^2$$

fvScalarMatrix TEqn

```
(  
    fvm::div(phi, T)  
    - fvm::Sp(fvc::div(phi), T)  
    - fvm::laplacian(kappaEff, T)
```

```
==laminarTransport.nu()*pow(Foam::sqrt(2.0)*mag(symm(fvc::grad(U))),2.0) );
```

次元が合わない

--> FOAM FATAL ERROR:

incompatible dimensions for operation

```
[T[0 0 -1 1 0 0 0] ] - [(nu*pow((1.41421*mag(symm(grad(U))))),2))[0 2 -3 0 0 0 0] ]
```