bouyantBoussinesqSimpleFoamについて

TM

第18回OpenCAE初心者勉強会（岐阜）

2012/11/3

概要

OpenFOAM　ver.2.0.x tutorials

heatTransfer/bouyantBousinesqSimpleFoam
自然対流伝熱問題のチュートリアル
＜前提＞
- 定常問題
- 非圧縮性
- 乱流モデル
- 乱流伝熱
- ブジネスク近似（温度差が比較的小さい）

2012/11/3

支配方程式

連続の式（非圧縮）

$$div(\mathbf{v}) = 0$$

レイノルズ平均方程式

$$\mathbf{v}grad\mathbf{v} = -grad(p_{rgh} + \rho gh)/\rho_{ref} + \Delta(\nu_{eff}\mathbf{v})$$

エネルギー方程式

$$div(\mathbf{v}T) - Tdiv(\mathbf{v}) = \Delta(\kappa_{eff}T)$$

修正動粘性係数

渦動粘性係数

$$\nu_{eff} = \nu + \nu_t$$

修正熱拡散率

$$\kappa_{eff} = \frac{\nu}{\mathrm{Pr}} + \frac{\nu_t}{\mathrm{Pr}\,t}$$

プラントル数　乱流プラントル数

ブジネスク近似

$$\rho \cong \rho(T_{ref}) + \left(\frac{\partial\rho}{\partial T}\right)_p (T - T_{ref}) = \rho_{ref}\left\{1 - \beta(T - T_{ref})\right\}$$

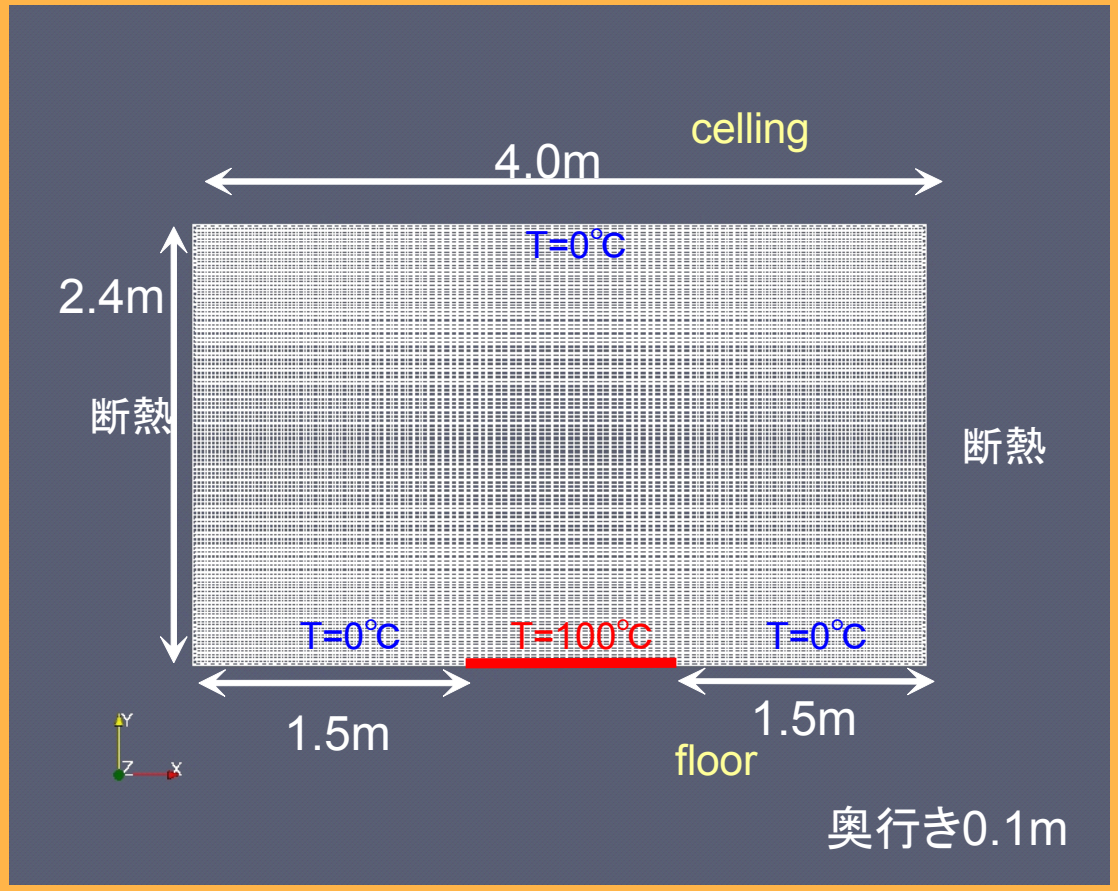
<<1

体積膨張係数

$$\beta = -\frac{1}{\rho_{ref}}\left(\frac{\partial\rho}{\partial T}\right)_p$$

2012/11/3

例題



tutorialのhotRoomを改造
2次元問題　　静止空気を加熱
100×100＝10,000メッシュ　　標準k-ε モデル

2012/11/3

物性値

```
transportModel Newtonian;
// Laminar viscosity
nu              nu [0 2 -1 0 0 0 0] 1.72e-05;
// Thermal expansion coefficient
beta            beta [0 0 0 -1 0 0 0] 3.7e-03;
// Reference temperature
TRef            TRef [0 0 0 1 0 0 0] 273.15;
// Laminar Prandtl number
Pr              Pr [0 0 0 0 0 0 0] 0.93;
// Turbulent Prandtl number
Prt             Prt [0 0 0 0 0 0 0] 0.9;
```

※空気の乱流プラントル数　0.9（境界層）　0.5（自由せん断流）
　流れによっていろいろなものがあるので注意

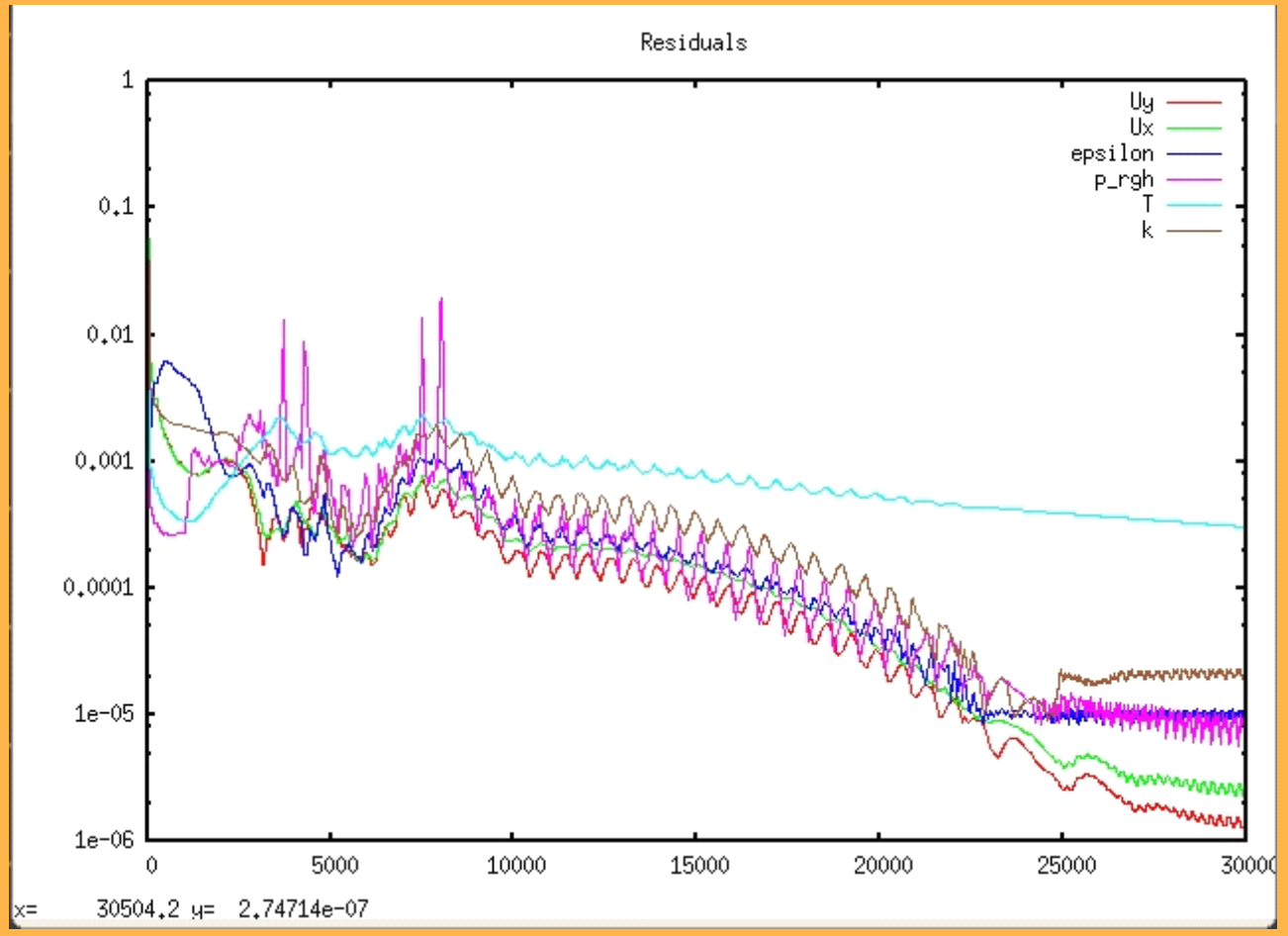
$$Ra = Gr \bullet \Pr = \frac{\beta \Delta T g L^3}{\nu^2} \times \frac{\nu}{\kappa} = 7.3 \times 10^{11}$$

乱流遷移条件　$Ra>10^9$

乱流を考慮する必要あり。

2012/11/3

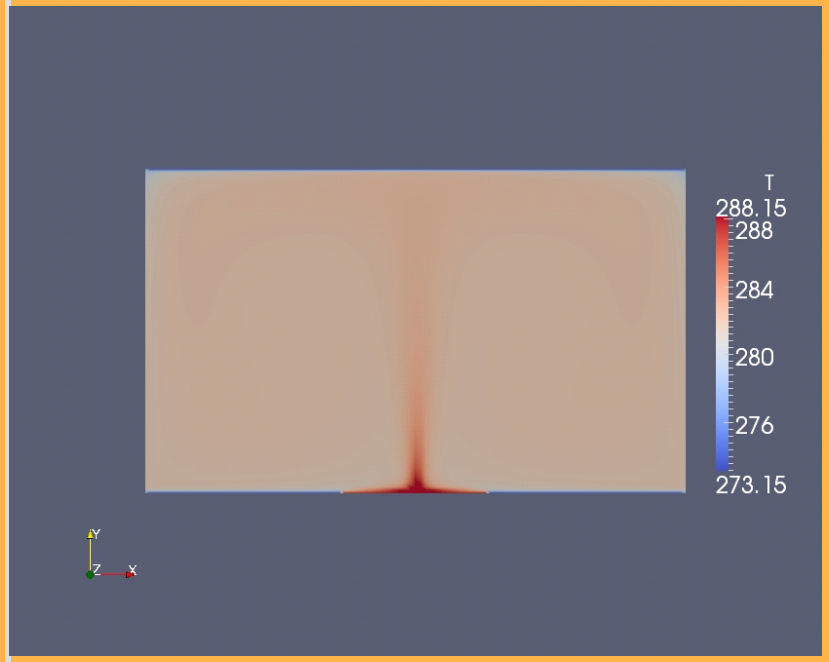
残差の履歴

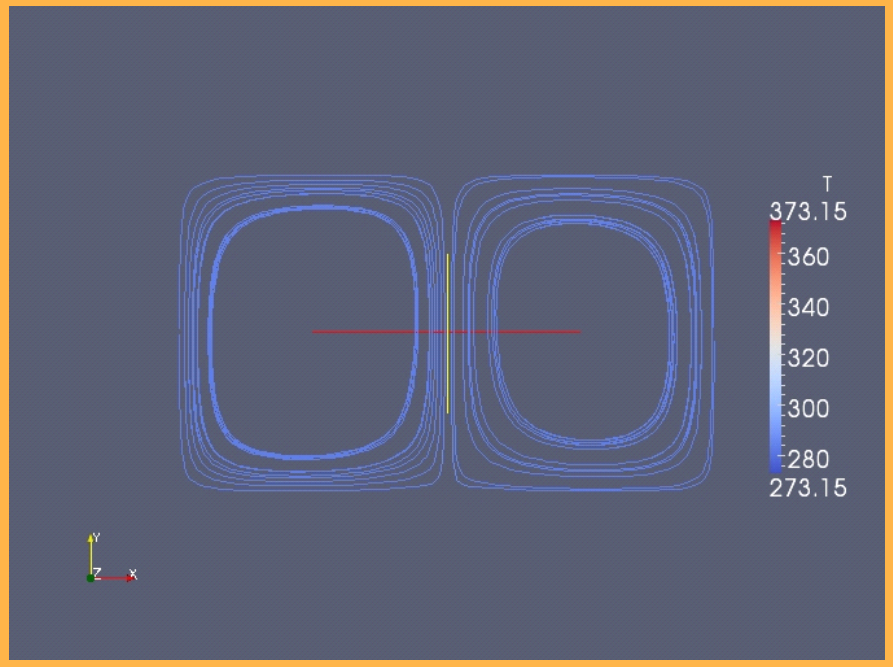


30,000ステップまで計算

2012/11/3

温度分布と流線





重ねて書きたいが、やり方がわからない

2012/11/3

壁面熱流束の計算

wallHeatFluxの改造

⇒buoyantSimpleFoamなど圧縮性流体に適用可

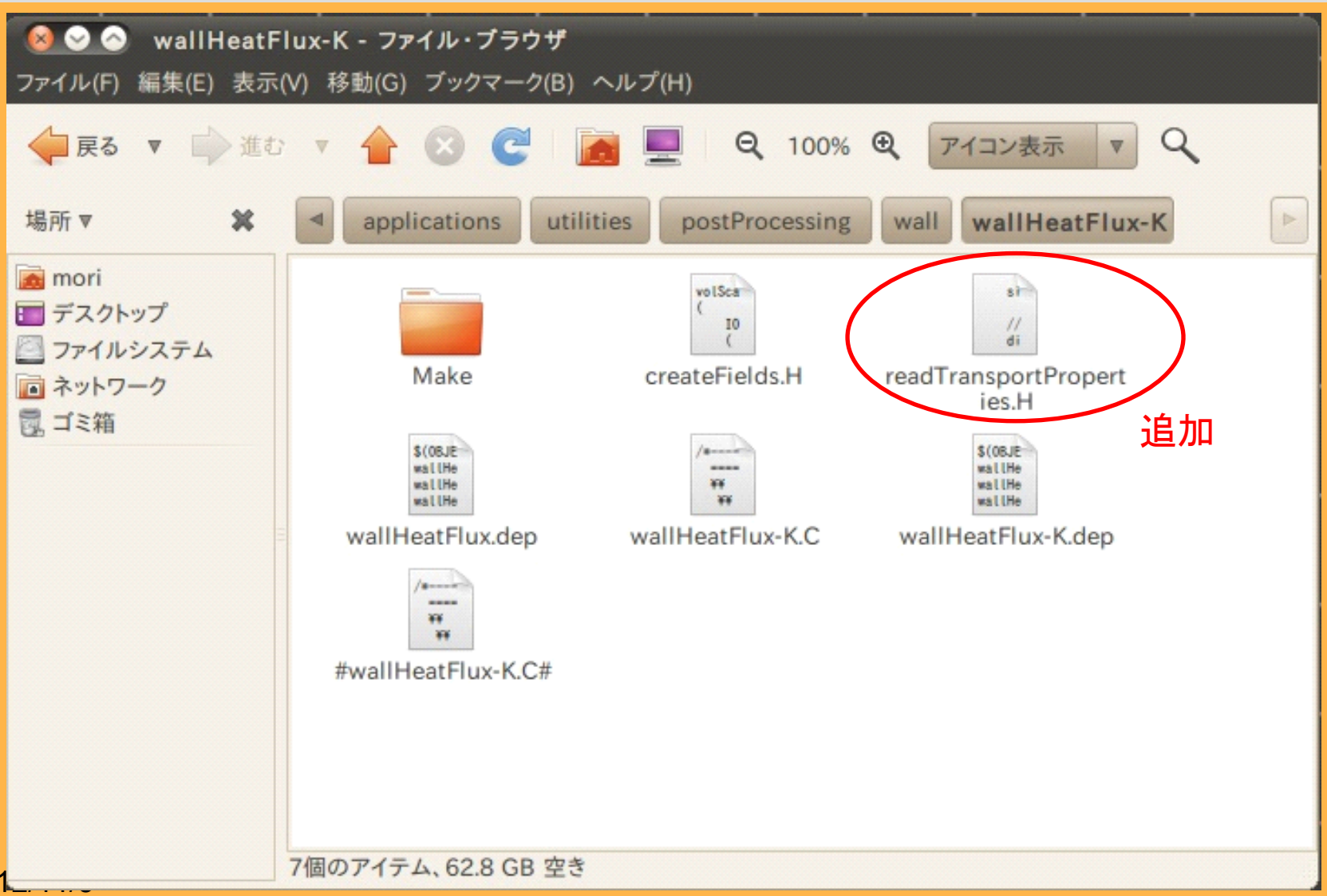
⇒buoyantBoussinesqSimleFoamは非圧縮性流体ソルバーなので使用不可

applications/utilities/postProcessing/wall/wallHeatFlux

これをカスタマイズ

2012/11/3

file


wallHeatFlux-K - ファイル・ブラウザ
ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ヘルプ(H)
戻る
進む
100%
アイコン表示
場所
applications
utilities
postProcessing
wall
wallHeatFlux-K
mori
デスクトップ
ファイルシステム
ネットワーク
ゴミ箱
Make
createFields.H
readTransportProperties.H
追加
wallHeatFlux.dep
wallHeatFlux-K.C
wallHeatFlux-K.dep
#wallHeatFlux-K.C#
7個のアイテム、62.8 GB 空き

wallHeatFlux-K/ make/options

```
EXE_INC = ¥
    -I$(LIB_SRC)/transportModels ¥
    -I$(LIB_SRC)/turbulenceModels ¥
    -I$(LIB_SRC)/turbulenceModels/incompressible/RAS/lnInclude ¥
    -I$(LIB_SRC)/turbulenceModels/incompressible/RAS/RASModel ¥
    -I$(LIB_SRC)/finiteVolume/lnInclude¥
    -I$(LIB_SRC)/transportModels/incompressible/singlePhaseTransportModel

EXE_LIBS = ¥
    -lincompressibleRASModels ¥
    -lincompressibleTransportModels ¥
    -lfiniteVolume ¥
    -lgenericPatchFields
```

2012/11/3

createFields.H

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

volVectorField U
(
    IOobject
    (
        "U",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```

```
#include "createPhi.H"
#include "readTransportProperties.H"

autoPtr<incompressible::RASModel> turbulence
(
    incompressible::RASModel::New(U,
phi,laminarTransport)
);

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


    kappat = turbulence->nut()/Prt;
    kappat.correctBoundaryConditions();

    volScalarField kappaEff("kappaEff", turbulence-
>nu()/Pr + kappat);
```

2012/11/3　　修正部分は青字

readTransportProperties.H

```
singlePhaseTransportModel laminarTransport(U, phi);

    // Thermal expansion coefficient [1/K]
    dimensionedScalar
beta(laminarTransport.lookup("beta"));

    // Reference temperature [K]
    dimensionedScalar
TRef(laminarTransport.lookup("TRef"));

    // Laminar Prandtl number
    dimensionedScalar Pr(laminarTransport.lookup("Pr"));

    // Turbulent Prandtl number
    dimensionedScalar Prt(laminarTransport.lookup("Prt"));
```

wallHeatFlux-K.C

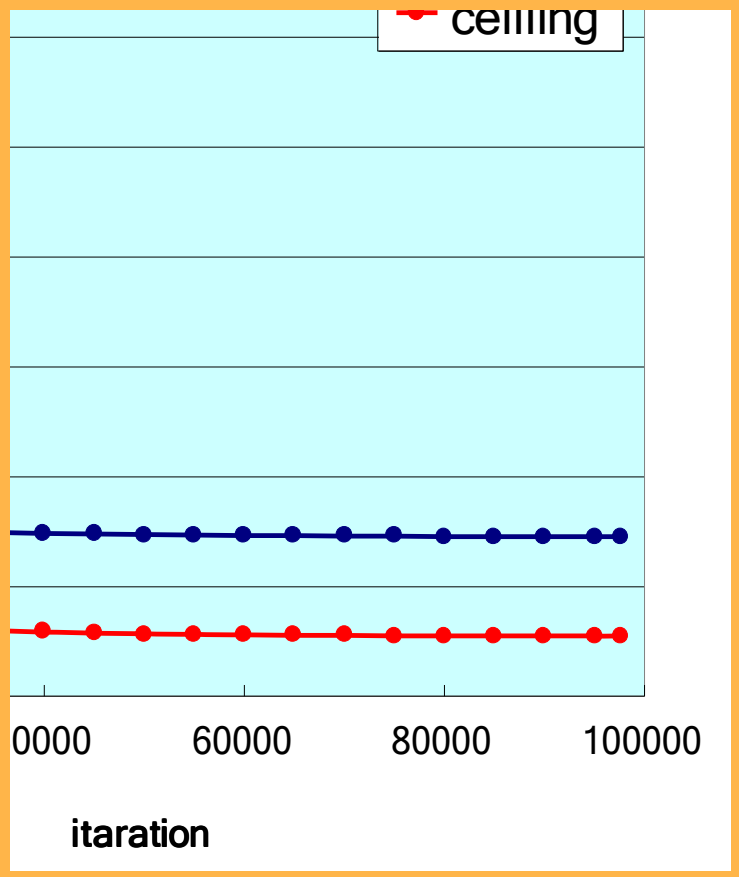
```
#include "fvCFD.H"
#include "singlePhaseTransportModel.H"
#include "RASModel.H"
#include "wallFvPatch.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

int main(int argc, char *argv[])
{
    timeSelector::addOptions();
    #include "setRootCase.H"
    #include "createTime.H"
    instantList timeDirs = timeSelector::select0(runTime, args);
    #include "createMesh.H"

    forAll(timeDirs, timeI)
    {
        runTime.setTime(timeDirs[timeI], timeI);
        Info<< "Time = " << runTime.timeName() << endl;
        mesh.readUpdate();

        #include "createFields.H"

        surfaceScalarField heatFlux
        (
            fvc::interpolate(kappaEff)*fvc::snGrad(T)
        );
```

$q = \kappa_{eff} \cdot gradT/(\rho \cdot Cp)$ [W]

修正部分は青字

HeatFluxの履歴

Cp=1.004[kJ/kgK]
ρ =1.293kg/m^3



<floor>
q=0.118[W]
<ceilling>
q=-0.117[W]

2012/11/3