

# Tutoriel 3: Ajout de l'équation de transport de la chaleur

Asmaa Hadane

March 4, 2025

# Introduction

Dans ce tutoriel, nous nous intéressons à la modification d'un solveur OpenFoam déjà existant en l'occurrence `icoFoam` afin de pouvoir, en plus de la résolution de l'équation de Navier-Stokes qui nous a donné la répartition des champs de vitesse et de pression dans la cavité, résoudre l'équation de transport de la chaleur afin d'avoir également une répartition du champ de température.

Le solveur icoFoam que nous rebaptisons my\_icoFoam resoudra donc le couplage Navier-Stokes et équation de transport de la chaleur, défini par les équations suivantes:

$$\vec{\nabla} \cdot \vec{U} = 0$$

$$\frac{\partial \vec{U}}{\partial t} + \vec{\nabla} \cdot (\vec{U}\vec{U}) = \vec{\nabla} \cdot (\nu \vec{\nabla} \vec{U}) - \frac{\vec{\nabla} P}{\rho}$$

$$\frac{\partial T}{\partial t} + \vec{\nabla} \cdot (\vec{U}T) = \vec{\nabla} \cdot (D_T \vec{\nabla} T)$$

# Ajout de l'équation de transport de la température

Afin de pouvoir résoudre l'équation de la chaleur dans la cavité entraînée par le solveur **icoFoam**, on procède à une série de modifications sur le solveur que nous allons résumer dans les diapositives qui suivent.

Le lecteur désirant davantage de détail pourra se rendre sur le site suivant:  
[http://openfoamwiki.net/index.php/How\\_to\\_add\\_temperature\\_to\\_icoFoam](http://openfoamwiki.net/index.php/How_to_add_temperature_to_icoFoam)

# Etape 1: Nouveau répertoire et modification des fichiers du Make

Afin de rajouter la résolution de l'équation de transport, il faut d'abord copier le dossier `icoFoam` contenu dans

*`OpenFOAM/applications/solvers/incompressible`*

vers notre répertoire de travail

*`run/solvers/applications/incompressible/.`*

et on choisit de le renommer `my_icoFoam`.

# Etape 1: Nouveau répertoire et modification des fichiers du Make

On modifie les fichiers contenus dans le dossier Make, en renommant `icoFoam.C` en `my_icoFoam.C` et en l'ouvrant à l'aide d'un éditeur de texte on y remplace partout `icoFoam` par `my_icoFoam`. Dans le terminal on exécute deux commandes:

- Effacer tous les fichiers devenus obsolètes:

*wclean*

- Tester que notre solver renommé marche correctement avec notre installation OpenFOAM:

*wmake*

## Étape 2 : Ajout du champ de température

On modifie le fichier `createFields.H` en définissant une nouvelle propriété de transport caractérisant la diffusion thermique, qui est le coefficient de diffusion DT:

```
createFields.H x
Info<< "Reading transportProperties\n" << endl;

IOdictionary transportProperties
(
    IOobject
    (
        "transportProperties",
        runTime.constant(),
        mesh,
        IOobject::MUST_READ_IF_MODIFIED,
        IOobject::NO_WRITE
    )
);

dimensionedScalar nu
(
    "nu",
    dimViscosity,
    transportProperties.lookup("nu")
);

//---- On ajoute le coefficient de diffusion ici----|
dimensionedScalar DT
(
    transportProperties.lookup("DT")
);
//-----
```

## Étape 2 : Ajout du champ de température

On définit le champ de température  $T$ , tout comme les variables de pression  $P$  et de vitesse  $U$  ont déjà été défini:

```
//_____Creation d'un champ de temperature_____

Info<< "Reading field T\n" <<endl;
volScalarField T
(
    IOobject
    (
        "T",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
//_____

Info<< "Reading field p\n" << endl;
volScalarField p
(
    IOobject
    (
        "p",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);
```



## Étape 3 : Ajout de l'équation de température

Dans le fichier `my_icoFoam.C` nous allons rajouter l'équation de transport de la température. Cependant comme le transport de la température dépend du champ de vitesse, il est nécessaire que dans l'ordre de résolution des équations, l'équation du transport de température vienne en dernier, juste après la correction du terme de pression existant et avant l'écriture du temps de la simulation.

## Étape 3 : Ajout de l'équation de température

```
// Pressure corrector

fvScalarMatrix pEqn
(
    fvm::laplacian(rAU, p) == fvc::div(phiHbyA)
);

pEqn.setReference(pRefCell, pRefValue);

pEqn.solve(mesh.solver(p.select(piso.finalInnerIter())));

if (piso.finalNonOrthogonalIter())
{
    phi = phiHbyA - pEqn.flux();
}

}

#include "continuityErrs.H"

U = HbyA - rAU*fvc::grad(p);
U.correctBoundaryConditions();
}

// _____ Equation de la chaleur _____
fvScalarMatrix TEqn
(
    fvm::ddt(T)
    + fvm::div(phi, T)
    - fvm::laplacian(DT, T)
);

TEqn.solve();

// _____
```

## Étape 3 : Ajout de l'équation de température

Après avoir sauvegardé, on ouvre le terminal dans le répertoire `my_icoFoam.C` et on exécute la commande suivante:

*wmake*

On s'assure ainsi qu'il n'y a pas d'erreur avant de passer à l'étape suivante.

## Étape 4 : Création d'un fichier contenant les conditions initiales et aux limites

Dans le fichier constant/transportProperties, après la ligne introduisant "nu", on ajoute la ligne du coefficient de diffusion de chaleur DT:

```
DT DT [0 2 -1 0 0 0 0] 0.002;
```

## Étape 4 : Création d'un fichier contenant les conditions initiales et aux limites

Dans le dossier 0 de cavity\_temperature on crée le fichier T:

```
/*===== -*- C++ -*- =====*/
//
// \ / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
//  W / O p e r a t i o n | Version: 2.2.2
//   W / A n d          | Web: www.OpenFOAM.org
//  W / M a n i p u l a t i o n
//
FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    object       T;
}
//=====//

dimensions      [0 0 0 1 0 0 0];

internalField   uniform 300;

boundaryField
{
    movingWall
    {
        type      fixedValue;
        value      uniform 350;
    }

    fixedWalls
    {
        type      fixedValue;
        value      uniform 300;
    }

    frontAndBack
    {
        type      empty;
    }
}
//=====//
```

Dans cette configuration on impose une température différente entre le mur mobile 350K et les murs fixes 300K.

# Étape 5 : Schéma, résidus pour la résolution de l'équation de transport

Puis dans le dossier `system` on modifie les fichiers `fvSchemes` et `fvSolution`. Dans `fvSchemes` on rajoute les opérateurs de divergence et laplacien apparaissant dans l'équation de transport dans les cases correspondantes:

```
divSchemes
{
    default            none;
    div(phi,U)         Gauss linear;
    div(phi,T)         Gauss upwind; // ligne à ajouter pour l'équation de temperature
}

laplacianSchemes
{
    default            none;
    laplacian(nu,U)     Gauss linear orthogonal;
    laplacian((1/A(U)),p) Gauss linear orthogonal;
    laplacian(DT,T)     Gauss linear corrected; // ligne à ajouter pour l'équation de temperature
}
```

# Étape 5 : Schéma, résidus pour la résolution de l'équation de transport

Dans `fvSolution` on rajoute:

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

```
solvers  
{  
    p  
    {  
        solver          PCG;  
        preconditioner   DIC;  
        tolerance        1e-06;  
        relTol           0;  
    }  
    // _____ solver de T rajouté _____  
    T  
    {  
        solver          BICCG;  
        preconditioner   DILU;  
        tolerance        1e-7;  
        relTol           0;  
    };  
    // _____  
    U  
    {  
        solver          PBiCG;  
        preconditioner   DILU;  
        tolerance        1e-05;  
        relTol           0;  
    }  
}
```

# Exécution du solver

Une fois tout ceci réalisé, on sauvegarde le travail et dans le répertoire `cavity_température` sur le terminal on peut lancer les commandes suivantes:

*blockMesh*

*my\_icoFoam*