

TP1 : Comparaison de modèles de turbulence pour un écoulement turbulent sur une plaque plane

Objectif :

Ce sujet consiste à modéliser sous le logiciel libre **OpenFoam** un écoulement turbulent parmi ceux dits « canoniques » et plus particulièrement l'écoulement le long d'une plaque plane. Ce premier TP nous permettra de

1. nous familiariser avec le logiciel
2. comparer quelques modèles de turbulence parmi les plus utilisés
3. comprendre le rôle de la résolution du maillage en proximité des parois
4. comprendre le rôle des lois de paroi

0. Installation d'OpenFoam



Pour l'installation d'OpenFoam sur votre ordinateur, faites référence aux instructions contenues dans le fichier InstallOpenFoam disponible sur Moodle et, surtout, au site www.openfoam.com

1. Description du problème

Nous nous attachons à la simulation d'un écoulement de plaque plane largement utilisé pour la validation des modèles de turbulence et inclus dans la base de validation de la NASA :

<https://turbmodels.larc.nasa.gov/flatplate.html>

<https://www.grc.nasa.gov/www/wind/valid/fpturb/fpturb.html>

L'écoulement est schématisé sur la figure 1. L'écoulement incident est caractérisé par un nombre de Mach (basé sur la vitesse du son et les conditions thermodynamiques à l'infini) $Ma=0.2$. Le nombre de Reynolds par unité de longueur de plaque (semi-infinie), basé sur la viscosité cinématique de l'écoulement incident est $Re=5$ millions. D'autres détails sont fournis sur la figure 1.

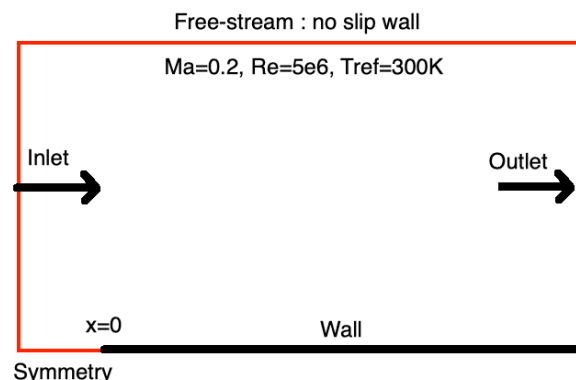


Figure 1

Le domaine de calcul est rectangulaire. L'écoulement, aligné avec l'axe x , procède de la gauche vers la droite. La frontière d'entrée (côté gauche du rectangle) est située en amont du bord

d'attaque de la plaque, qui est placé à l'abscisse 0. Compte-tenu de la symétrie du problème, on ne simule que la moitié supérieure de l'écoulement.

Le nombre de Mach étant faible, les effets de la compressibilité peuvent être négligés. Nous allons donc simuler l'écoulement comme étant incompressible avec propriétés constantes.

Nous résolvons donc les équations RANS incompressibles (il n'est pas nécessaire de prendre en compte l'équation de l'énergie) complétées par un modèle de turbulence :

$$\left\{ \begin{array}{l} \frac{\partial \bar{u}_i}{\partial x_i} = 0 \\ \frac{\partial \bar{u}_i}{\partial t} + \frac{\partial (\bar{u}_i \bar{u}_j)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{1}{\rho} \frac{\partial}{\partial x_j} \left(\mu \frac{\partial \bar{u}_i}{\partial x_j} - \overline{\rho u'_i u'_j} \right) \end{array} \right. \quad \begin{array}{l} \tau_{ij} = \mu \frac{\partial \bar{u}_i}{\partial x_j} = \text{contraintes visqueuses} \\ \tau_{ij}^R = -\overline{\rho u'_i u'_j} = \text{contraintes turbulentes} \\ \text{(tenseur de Reynolds)} \end{array}$$

1.1 Algorithme de résolution

Nous recherchons une solution stationnaire des équations RANS. Pour cela nous utilisons l'algorithme SIMPLE, qui consiste à résoudre de façon itérative l'équation de continuité et une équation de Laplace pour la pression, obtenue en prenant la divergence de l'équation pour la quantité de mouvement.

Cela veut dire que, parmi les nombreux solveurs disponibles dans OpenFoam, nous allons utiliser `simpleFoam`.

Pour exécuter un calcul, nous avons besoin de construire un répertoire contenant tous les scripts avec les informations nécessaires pour le calcul : maillage, modèles, conditions de fonctionnement, ...

Cela prend la structure d'un arbre, schématisé sur la figure 2.

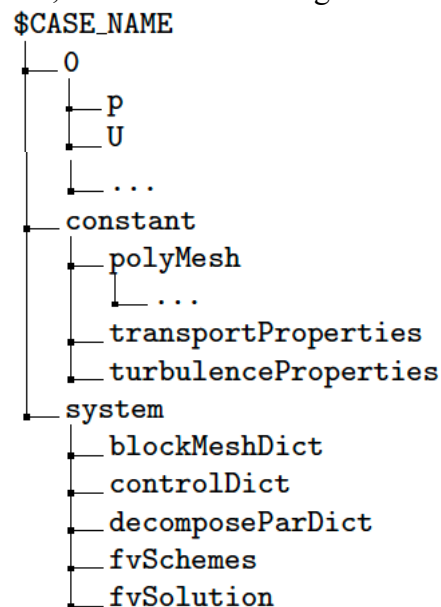


Figure 2 : Structuration typique d'un projet OpenFoam.

Pour la pression et la vitesse, on a les mêmes dictionnaires pour tout choix du modèle de turbulence.

```

/*-----* C++ -*-----*/
|=====|
|  \ \   /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
|  \ \   /  O p e r a t i o n | Version: v2006
|  \ \   /  A n d             | Website: www.openfoam.com
|  \ \   /  M a n i p u l a t i o n |
|-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        volScalarField;
    object       p;
}

// *****

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 0;

boundaryField
{
    inlet
    {
        type      zeroGradient;

/*
        type      totalPressure;
        rho        none;
        psi        none;
        p0         uniform 1.02828;
        value      uniform 1.02828;
*/
    }

    outlet
    {
        type      fixedValue;
        value      uniform 0;

/*
        type      fixedValue;
        value      uniform 1;
*/
    }

    topWall
    {
        type      zeroGradient;
    }

    bottomWall
    {
        type      zeroGradient;
    }

    symmetry
    {
        type      symmetryPlane;
    }

    frontAndBack
    {
        type      empty;
    }
}

// *****

```

```

/*-----*- C++ -*-----*/
|=====|
|  \ \ /  | F ield      | OpenFOAM: The Open Source CFD Toolbox
|  \ \ /  | O peration  | Version: v2006
|  \ \ /  | A nd        | Website: www.openfoam.com
|  \ \ /  | M anipulation|
|=====|
/*-----*/

FoamFile
{
    version      2.0;
    format       ascii;
    class        volVectorField;
    object       U;
}
// *****

dimensions      [0 1 -1 0 0 0];

internalField    uniform (69.4 0 0);

boundaryField
{
    inlet
    {
        type      fixedValue;
        value      $internalField;
    }

    outlet
    {
        type      zeroGradient;
    }

    topWall
    {
        type      slip;
    }

    bottomWall
    {
        type      fixedValue;
        value      uniform (0 0 0);
    }

    symmetry
    {
        type      symmetryPlane;
    }

    frontAndBack
    {
        type      empty;
    }
}
// *****

```

m/s
Condition
initiale vitesse
uniforme

Condition
d'entrée :
Valeur imposée

Condition de
sortie :
Extrapolation

Loin de la
plaque
comportement
non visqueux

Condition
d'adhérence

Symétrie

Non renseigné
en 2D

[0 2 -1 0 0 0 0]			
No.	Property	SI unit	USCS unit
1	Mass	kilogram (kg)	pound-mass (lbm)
2	Length	metre (m)	foot (ft)
3	Time	second (s)	second (s)
4	Temperature	Kelvin (K)	degree Rankine (°R)
5	Quantity	mole (mol)	mole (mol)
6	Current	ampere (A)	ampere (A)
7	Luminous intensity	candela (cd)	candela (cd)

Figure 3 : Exemple de définition des dimensions pour une viscosité cinématique (m²/s) et tableau des dimensions.

Dans le répertoire `constant` on trouve les dictionnaires permettant de préciser les propriétés de transport du fluide et les modèles de turbulence.

Dans le répertoire `system` on trouve les dictionnaires permettant de préciser les paramètres numériques pour la génération du maillage (`blockMeshDict`), la discrétisation des équations (`fvSchemes`), la méthode de résolution (`fvSolution`), les conditions d'arrêt/exploitation des résultats (`controlDict`) ou encore la parallélisation des calculs (`decomposeParDict`).

Dans `openFoam` il existe une multitude de modèles de turbulence : LRR, LamBremhorstKE, LaunderSharmaKE, LienCubicKE, LienLeschziner, RNGkEpsilon, SSG, ShihQuadraticKE, SpalartAllmaras, kEpsilon, kOmega, kOmegaSST, kOmegaSSTLM, kOmegaSSTSAS, kkLOmega, qZeta, realizableKE, v2f

Tous ne sont pas complètement validés et garantis ! C'est l'inconvénient de l'OpenSource...

Les plus utilisés par la communauté (et donc les mieux validés) sont : kOmegaSST, SpalartAllmaras et kEpsilon, sur lesquels nous allons nous focaliser dans ce TP.

Ces modèles sont décrits dans la suite.

1.2 Modèles de turbulence

1.2.1 Modèle kOmegaSST

Il s'agit d'un modèle à deux équations hybride combinant le kOmega de Wilcox et le kEpsilon standard. Il résout une équation pour l'énergie cinétique turbulente k et une pour la dissipation spécifique ω .

Pour ce modèle, le dossier « 0 » a la structure suivante :

```
0
├── P
├── U
├── nut
├── k
└── omega
```

Dans ces fichiers on trouve les conditions aux limites pour toutes les variables transportées.

1.2.1 Modèle kEpsilon

Il s'agit d'un modèle à deux équations qui résout une équation pour l'énergie cinétique turbulente k et une pour la dissipation spécifique ϵ . La clé kEpsilon correspond au modèle standard. Ce modèle n'est pas bien adapté pour des écoulements de paroi et il nécessite l'utilisation d'une loi de paroi.

Pour ce modèle, le dossier « 0 » a la structure suivante :

```
0
├── P
├── U
├── nut
├── k
└── varepsilon
```

Dans ces fichiers on trouve les conditions aux limites pour toutes les variables transportées, dont la loi de paroi pour nut, k et epsilon. Plusieurs lois de paroi sont disponibles dans OpenFoam.

1.2.3 Modèle Spalart-Allmaras

Il s'agit d'un modèle à une équation qui résout une équation de transport pour la viscosité

1.2.3 Modèle Spalart-Allmaras

Il s'agit d'un modèle à une équation qui résout une équation de transport pour la viscosité tourbillonnaire en écoulement libre, $\tilde{\nu}$ (nuTilda). C'est un modèle développé pour l'aérodynamique externe, capable de donner des résultats satisfaisants pour des écoulements avec gradients de pression.

Lorsqu'on utilise ce modèle, le dossier « 0 » prend la structure suivante :

```
0
├── P
├── U
├── nut
└── nuTilda
```

Avec la viscosité tourbillonnaire nut qui est le produit de nuTilda et d'une fonction d'amortissement.

Dans ces fichiers il faut remplir les conditions aux limites pour nuTilda. En entrée/sortie on peut imposer une petite valeur à l'infini ou bien une extrapolation. A la paroi on impose nuTilda=0.

2. Travail à effectuer

Le répertoire turbulentFlatPlate contient deux scripts permettant d'automatiser certaines opérations.

- Allclean : permet d'effacer les données des calculs

- **Allrun** : permet de lancer plusieurs calculs simultanément à partir d'une liste de modèles et de maillages. Pour ce faire, il utilise des dictionnaires-modèle (template) pour les différents modèles de turbulence à étudier.

Après avoir défini les paramètres nécessaires à la génération des maillages (pour plusieurs valeurs possibles de la taille adimensionnelle de la première cellule proche de la paroi,

$$y^+ = \frac{u_\tau}{\nu} \Delta y_{wall} \text{ avec } u_\tau = \sqrt{\frac{\tau_{wall}}{\rho}}.$$

Nous avons vu en cours que la taille de la première maille joue un rôle essentiel dans le traitement des modèles à la paroi.

- Pour $y^+ = O(1)$, on peut appliquer les conditions aux limites du modèle directement au point de paroi (sauf exceptions, comme la variable ω notamment).
- Pour $10 < y^+ < 100$, le premier point de paroi se trouve dans la zone logarithmique de la couche limite, et l'application d'une loi de paroi est nécessaire pour avoir le bon comportement du profil de vitesse.

Le répertoire contient aussi un script nommé **plot**, permettant de tracer la distribution du coefficient de frottement $C_f = \frac{\tau_{wall}}{\frac{1}{2}\rho U_{ref}^2}$, en fonction du nombre de Reynolds local basé sur la distance au bord d'attaque $Re_x = \frac{(x-x_0)U_{ref}}{\nu}$ à l'aide du logiciel gratuit **Gnuplot**. Les résultats sont comparés avec une corrélation empirique des données expérimentales de Wiegardt et Tillman (On the Turbulent Friction Layer for Rising Pressure, NACA TM-1314, 1951).

Activité 1 :

1.1 A l'aide de la documentation en ligne de OpenFoam, étudier et commenter brièvement la structure des dictionnaires suivants :

- `0 : nut, k, omega, epsilon`
- `constant:turbulenceProperties.orig` (template)
- `system:controlDict, fvSchemes, fvSolution`

1.2 A l'aide la documentation en ligne de OpenFoam, étudier et commenter brièvement la structure du script **Allrun**

Activité 2 : Travail sur le modèle komegaSST

2.1 Dans **Allrun**, commenter tous les maillages sauf celui correspondant à $y^+ = 10$; modifier le `endTime` dans `system/controlDict` et le fixer à la valeur 500, puis 1000, puis 2000, puis 3000. Regarder la sensibilité de la solution à ce paramètre et commenter. Choisir la valeur qui offre le meilleur compromis coût/précision.

2.2 Lancer **kOmegaSST** et **kEpsilonSST** sur les maillages $y^+ = 1, y^+ = 5, y^+ = 10, y^+ = 30, y^+ = 100$.

Etudier l'évolution du coefficient de frottement pariétal avec le raffinement du maillage. Si c'est trop long, vous pouvez vous répartir les calculs.

2.3 (facultatif) : ~~générez le champ du module de la vitesse à l'aide de la commande `postProcess -func "mag(U)"` et visualisez-le avec ParaView.~~

~~(facultatif) : tracer les profils de vitesse à l'aide de la commande `postProcess -func sampleDict` et les visualiser dans Gnuplot. Comparer avec la loi logarithmique.~~

Activité 3 :

3.1 en s'inspirant des exemples présents dans la base OpenFoam, préparer les dictionnaires pour un calcul avec le modèle Spalart-Allmaras.

3.2 Modifier et exécuter le script Allrun pour le modèle Spalart-Allmaras. Comparer les résultats obtenus pour le Cf sur des maillages de finesse différente avec ceux des autres modèles.

Activité 4 : Pour quelques-uns des maillages (par exemple $y^+=1$, $y^+=5$, $y^+=30$) superposer les résultats des trois modèles étudiés et commenter.