Assignment 2 - Channel flow using different LES models

In this project we will study **(usikker på bra fluidmekanisk beskrivelse)** using openFOAM using different Large Eddy Simulation (LES) models.

As a starting point you should use the

```
/.../tutorials/incompressible/channelFoam/channel395
```

example.

For every new LES model you study, create a new folder structure for it. You are going to need the results from all models in order to compare the results later.

One Equation Eddy

The standard choice of LES model in the channelFoam tutorial is *One Equation Eddy*. All properties of the LES model is set in the

```
/.../constant/LESProperties
```

dictionary. As a starting point you should open the file and familiarize with it.

a)

Before you start, take a look at the time steps etc. in the various files located in the `control` and `system` folders. Then build the mesh as you did in the previous assignment, and run the `channelFoam` application. The runtime of the simulation varies from machine to machine. On older laptops, the runtime can be as much as 2 hours on a single CPU.

If you wish to run openFOAM in parallel, check out the following guide

```
www.openfoam.org/docs/user/running-applications-parallel.php
```

for details.

b)

Open the simulation in paraFOAM. Select U, p and UMean values, and make a slice in the middle of the channel with a *Z*-normal, choose to display the interpolated velocity field and play the simulation to the endpoint.

On the slice, apply a *calculator* tool by either going through

`filters→alphabetical→calculator`
or press the calculator icon next to the slice-tool. Choose a reasonable name

and select to calculate

` U - UMean.`

Once you have applied the calculator tool, go to the display tab and select *Colour by* the name of the value you created with the calculator. Select magitude and play the simulation again[1].

More exercises needs to be added

Smagorinsky

In order to do the switching you need to change the `LESProperties` file

`LESModel        Smagorinsky;`

Retrace the steps of the One Equation Eddy section.

---

[1]You might run into problems with segmentation faults at this point. OpenFOAM does not store the averages for the zeroth time. If this is a problem, at the initial properties tab, select *Skip Zero Time*, move the simulation to the first time step, and then select the preferred values for the visualization.

Dynimic One Equation Eddy

In order to change to this case we need something else besides a simple name change. Add the following to the `LESProperties` file:

```
LESModel        dynOneEqEddy;

...

dynOneEqEddyCoeffs
{
    ce              1.05;
    filter          simple;
}

need the one eq eddy coeffs too?
```

Retrace the steps of the One Equation Eddy section.

Homogeneous Dynimic Smagorinsky

Add the following to the `LESProperties` file:

```
LESModel        homogeneousDynSmagorinsky;

...

homogeneousDynSmagorinskyCoeffs
{
    ce 1.048;
    filter simple;
}

the same as dynOneEqEddy??
```

Retrace the steps of the previous sections.

Dynamic Lagrangian

This time we need to give additional information beyond the single file. Add the following to the `LESProperties` file:

```
LESModel        dynLagrangian;

...

dynLagrangianCoeffs
{
    filter simple;
    ce 1.048;
    theta 1.5;
}

CE same for all?
```

In addition you will need some initial conditions for the `fmm` and `flm` values. Enter the `0` dictionary (after mesh is built) and add the files on the following pages.

```
fml:

/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\ / F ield     | OpenFOAM: The Open Source CFD Toolbox           |
| \\ / O peration | Version: 2.0.0                                 |
| \\ / A nd       | Web: www.OpenFOAM.com                          |
| \\/ M anipulation |                                               |
\*---------------------------------------------------------------------------*/
FoamFile
{
version 2.0;
format ascii;
class volScalarField;
object flm;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions [0 4 -4 0 0 0 0];

internalField uniform 0;

boundaryField
{
    bottomWall
    {
        type            fixedValue;
        value           uniform 0;
    }
    topWall
    {
        type            fixedValue;
        value           uniform 0;
    }
    sides1_half0
    {
        type            cyclic;
    }
    sides2_half0
    {
        type            cyclic;
    }
    inout1_half0
    {
        type            cyclic;
    }
    inout2_half0
    {
        type            cyclic;
    }
    sides2_half1
    {
        type            cyclic;
    }
    sides1_half1
    {
        type            cyclic;
    }
    inout1_half1
    {
        type            cyclic;
    }
    inout2_half1
    {
        type            cyclic;
    }
}

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

fmm:

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      / F ield           | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    / O peration        | Version: 2.0.0                                  |
|   \\  / A nd               | Web: www.OpenFOAM.com                           |
|    \\/ M anipulation       |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
version 2.0;
format ascii;
class volScalarField;
object fmm;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions [0 4 -4 0 0 0 0];

internalField uniform 1;

boundaryField
{
    bottomWall
    {
        type            fixedValue;
        value           uniform 1;
    }
    topWall
    {
        type            fixedValue;
        value           uniform 1;
    }
    sides1_half0
    {
        type            cyclic;
    }
    sides2_half0
    {
        type            cyclic;
    }
    inout1_half0
    {
        type            cyclic;
    }
    inout2_half0
    {
        type            cyclic;
    }
    sides2_half1
    {
        type            cyclic;
    }
    sides1_half1
    {
        type            cyclic;
    }
    inout1_half1
    {
        type            cyclic;
    }
    inout2_half1
    {
        type            cyclic;
    }
}

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

You should now be able to run the application. Retrace the steps of the previous sections.

Comparing results with DNS

Need more info

...