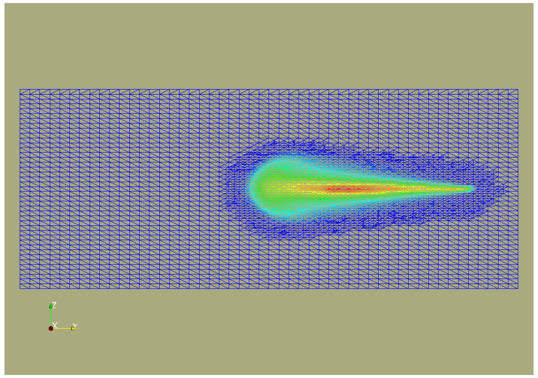
dieselDyMFoam: dynamic mesh refinement in dieselFoam

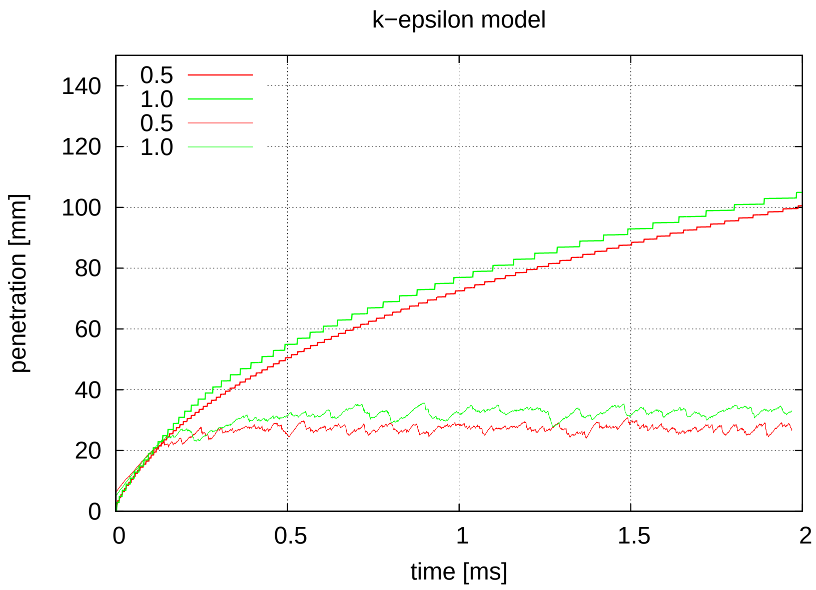
CFD with OpenSource Software, Assignment 3
Anne Koesters
Chalmers University of Technology
2010



OpenFOAM 1.6-x

grid dependency

- results of spray simulations are grid dependent
- a good grid with acceptable computation times is needed
- dynamic mesh refinement would allow to start with a coarser grid and just refine the grid where finer cells are needed



mesh refinement in *interDyMFoam*

```
mesh.update()
```

- read the dynamicMeshDict (located in the constant folder of the case)
- definition of all cells that will be refined
- refinement of the cells
- definition of cells that should be unrefined
- unrefinement of the cells

dieselDyMFoam

```
sol
cp -r combustion/dieselFoam/ $WM_PROJECT_USER_DIR/applications/solvers/
cd $WM_PROJECT_USER_DIR/applications/solvers
mv dieselFoam dieselDyMFoam
cd dieselDyMFoam
rename dieselFoam dieselDyMFoam *
sed -i s/dieselFoam/dieselDyMFoam/g *
sed -i s/dieselFoam/dieselDyMFoam/g Make/*
```

change the library path in Make/files to:

```
EXE = $(FOAM_USER_APPBIN)/dieselDyMFoam
```

and also change in Make/options:

```
    -I../dieselEngineFoam \
```

to

```
-I$(LIB_SRC)/../applications/solvers/combustion/dieselEngineFoam \
```

compile the solver:

```
wmake
```

dieselDyMFoam.C

Finally the .C file of our new solver needs to be modified. We need to do several modifications:

1. change the line

   ```
   #include "createMesh.H"
   ```

   to

   ```
   #include "createDynamicFvMesh.H"
   ```

2. add in the beginning of the file following line:

```
#include "dynamicFvMesh.H"
```

3. before *runTime++;* following lines need to be added:

```
 //*****************************************
// Make the fluxes absolute
fvc::makeAbsolute(phi, U);
 //*****************************************
```

dieselDyMFoam.C

4. before the line *evolve.spray()* the function calling the mesh refinement has to be added:

```
      //**********************************************************
      //start mesh refinement
      //**********************************************************
scalar timeBeforeMeshUpdate = runTime.elapsedCpuTime();
// Do any mesh changes
        mesh.update();
        if (mesh.changing())
  {
            Info<< "Execution time for mesh.update() = "
                << runTime.elapsedCpuTime() - timeBeforeMeshUpdate
                << " s" << endl;
  }
      //**********************************************************
      //end mesh refinement
      //**********************************************************
```

Make/files

The last step is to change the *Make/files*. Following lines need to be added:

1. add in EXE INC following three lines:

```
-I$(LIB_SRC)/dynamicMesh/lnInclude \
-I$(LIB_SRC)/meshTools/lnInclude \
-I$(LIB_SRC)/dynamicFvMesh/lnInclude \
```

2. add in EXE LIBS following lines:

```
-lmeshTools \
-ldynamicFvMesh \
-ltopoChangerFvMesh \
```

createFields.H and YEqn.H

We also need to define the field the refinement should act on. This is done by copying the *createFields.H* and the *YEqn.H* file from the *dieselEngineFoam* solver and doing some modifications.

Now download both files that are already modified from the course homepage and put it in the new solver dieselDyMFoam.

Finally the new solver can be compiled by doing:

```
wclean
wmake
```

aachenBomb

Download the case aachenBomb from the course homepage.
Modifications were done as follows:

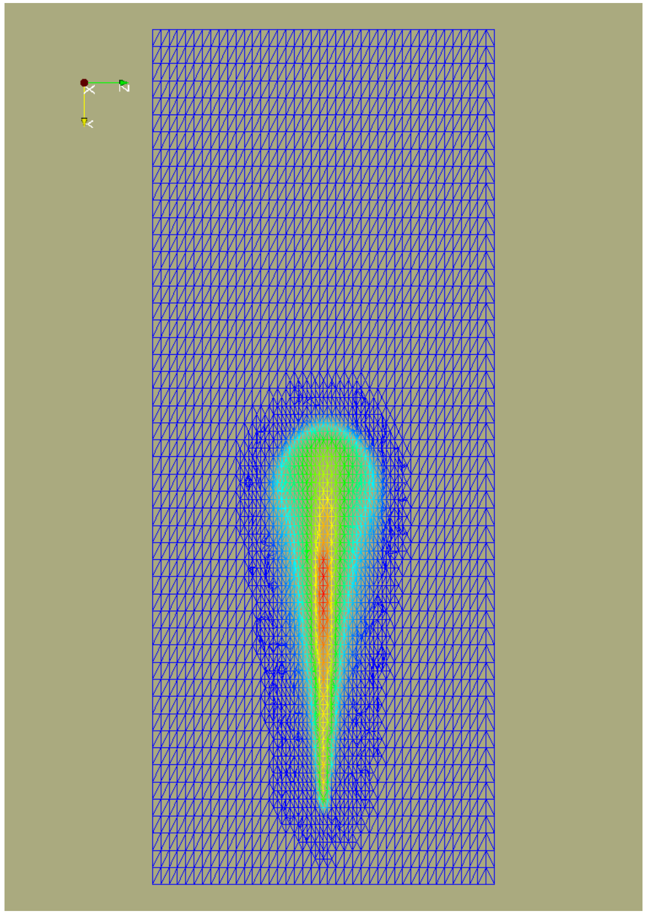
- setting up the dictionary *dynamicMeshDict* in the constant folder
- modification of system/fvSolution
- adding the field Ytf in the 0 directory
- modifying the blockMeshDict in the constant/polyMesh directory

Now start the case:

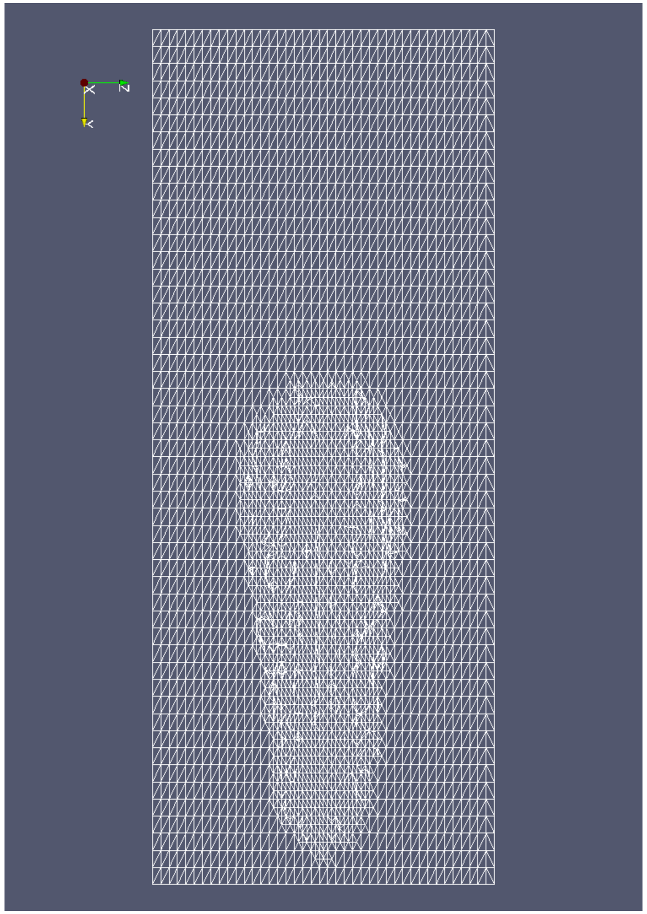
```
dieselDyMFoam >& log &
```

wait some time, then start paraFoam and have a look on the results!

results



Ytf field



mesh