Setting up the watersprinkler case

The modifications should be done using the `damBreak` case, therefore we should copy it to a new directory

```
run
mkdir sprinkler
cp -r $FOAM_TUTORIALS/multiphase/interFoam/laminar/damBreak sprinkler
cd sprinkler/damBreak
```

The `blockMeshDict` file should be replaced with the one downloaded from the webpage. Copy it to the directory

```
cp blockMeshDict constant/polyMesh
```

We can now run `blockMesh` to construct the mesh by writing

```
blockMesh
checkMesh
```

Boundary conditions

The following two boundary conditions should be added in the `0/U` file

```
inlet
{
        type            rampedFixedValue;
        refValueLow     uniform (0 -0.5 0);
        refValueHigh    uniform (0 0 0);
        startRamp       0.1;
        endRamp         0.15;
        value           uniform (0 0 0);
}
column
{
        type            fixedValue;
        value           uniform (0 0 0);
}
```

Boundary conditions

The following two boundary conditions should be added in the `0/p_rgh` file

```
inlet
{
    type            buoyantPressure;
    value           uniform 0;
}

column
{
    type            buoyantPressure;
    value           uniform 0;
}
```

Boundary conditions

The following two boundary conditions should be added in the `0/alpha1` file

```
inlet
{
    type            fixedValue;
    value           uniform 1;
}

column
{
    type            zeroGradient;
}
```

Boundary conditions/Calculations

We are using the `rampedFixedValue` boundary condition. Therefore we need to add the dynamic library in the `system/controlDict` file. Just copy-paste it in the end of the file

```
libs ("libMyBCs.so");
```

We can now execute the solver `interFoam` by running the following

```
interFoam
```

The utility AlphaCalc.C

Make a directory in the case map and copy the utility as below

```
cp -r AlphaCalc tutorials/sprinkler/damBreak/AlphaCalc
cd AlphaCalc
```

A look inside `AlphaCalc.C` gives us an interresting loop to calculate the desired sums

```
forAll(centres,nIter)
{
     if(centres[nIter][1] < 0)
     {
          if(centres[nIter][0] < 0.4*0.146) //  first collector
          {
               waterLevel = waterLevel + alpha1[nIter]*volumes[nIter];
          }
     }
}

Info << "waterLevel: " << waterLevel << endl;
```

The utility AlphaCalc.C / foamLog

The `forAll` loop multiplies the calculated `alpha1` values from the `interFoam` solver with the calculated volume for each cell that lies in the interval that is defined by the if-loops. In order to compile the utility, write

```
wmake AlphaCalc
```

Now run the utility using the following command in the terminal window

```
AlphaCalc >& log
```

To extract the wanted information from the log-file, we use `foamLog`. First we need to edit the database where the information what `foamLog` should extract is defined.

```
locate foamLog.db
sudo gedit <LocationOfFoamLog.db>
```

Now add the following in the beginning of the file

```
waterLevel/waterLevel: /waterLevel:
```

It now extracts the 12 different sums that was calculated from `AlphaCalc.C`.

foamLog

Write the following line

```
foamLog log
```

The extracted values are now put in the directory `sprinkler/damBreak/logs` directory. Now write

```
sed -e s/"waterLevel"/""/g logs/waterLevel_0 > waterLevel
```

Now we got the file `sprinkler/damBreak/waterLevel`. This consists of 12 sums corresponding to each water collector.

Python

It would be nice to visualize the water distribution. We make a script that plots the distribution as a histogram. First of all we need to install Python. To do that, type the following

```
cd $HOME/OpenFOAM
mkdir linuxSrc
cd linuxSrc
svn co https://openfoam-extend.svn.sourceforge.net/svnroot/ \
openfoam-extend/trunk/Breeder/other/scripting/PyFoam/
python setup.py install --prefix=$HOME/OpenFOAM
```

Now, add the following lines in the end of your `etc/apps/paraview3/bashrc` file

```
alias PF=export FOAM_INST_DIR=$HOME/OpenFOAM; \
export PYTHONPATH=$FOAM_INST_DIR/PyFoam/lib/python-2.6/
\site-packages:$PYTHONPATH; \
export PATH=$FOAM_INST_DIR/PyFoam/bin:$PATH
```

Two packages are needed to make the script work, `python-matplotlib` and `python-numpy`. Enter the ”Synaptic Package Manager” and download the packages for Python

Python script

Type `gedit histogram.py` in the `sprinkler/damBreak` directory. It is later executed with the commando `python histogram.py` . It should contain the following lines

```
#!/usr/bin/env python
import numpy.numarray as na
from pylab import *
import matplotlib.mlab as mlab

X = mlab.load('waterLevel')
labels = ["1","2","3","4","5","6","7","8","9","10","11","12"]
xlocations = na.array(range(len(X)))+0.8
width = 0.8
bar(xlocations, X, width=width)
xticks(xlocations+ width/2, labels)
xlim(0, xlocations[-1]+width*2)
title("Histogram over the watercollectors")
gca().get_xaxis().tick_bottom()
gca().get_yaxis().tick_left()
show()
```