**Clarence Centre Indoor CO2**

**Files**

Time series of 454 vtu files

Accessible on the HPC

/rds/general/user/lmottet/projects/predictivemodelling/live/lmottet/ClarenceCentre/Simu\_Fluidity

Or it can be downloaded here:

<https://www.dropbox.com/s/wmzopedxcelzpaf/ClarenceCentre.zip?dl=0>

1 vtu file = 1 time step ~ 2seconds

The files are named: ClarenceCentre\_n.vtu where n is between 0 and 454. NB: At n=0, t=0 sec.

Warning: the time step is not exactly constant between 2 vtu files. The file TimeVTU.dat gives you the exact time of each vtu file. For now, it is fine - you can work with this set of data assuming the time-step is constant. A better set of data will eventually come later to fix this issue.

The vtu files can be visualised into Paraview (open-source free software).

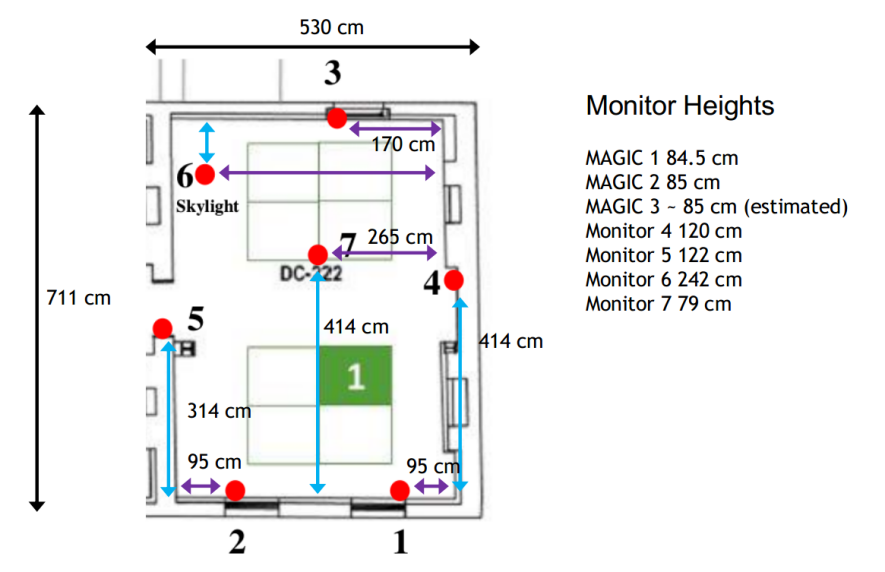
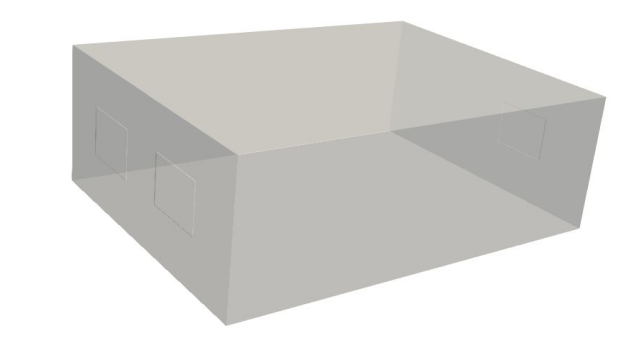
**Geometry**

The geometry is a room with 3 windows: 2 on one side and 1 on the other side.

In the vtu files, the x, y and z coordinates are in the range: 221.022 m < x < 229.514 m; 62.7509 m < y < 70.1622 m; 6.08 m < x < 8.5 m.

The room is rotated by a 110 deg. angle. The size of the room is 5.30 m x 7.11 m x 2.42 m

Here are schematics of the room and sensors locations (red dots).



**Mesh**

The mesh is unstructured. There is 148,906 nodes, i.e. 679,119 cells.

**Sensors**

During the field study experiment, 7 sensors were used. See figure above for the location in the room.

The coordinates (in meter) of the seven indoor CO2 sensors in Fluidity simulation are:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | x | y | z |
| MAGIC1 | 221.45593863 | 66.08811021 | 6.98 |
| MAGIC2 | 222.58460511 | 69.18909586 | 6.98 |
| MAGIC3 | 228.17923182 | 64.38595605 | 6.98 |
| MAGIC4 | 224.94447869 | 63.86062547 | 7.28 |
| MAGIC5 | 225.7490888 | 68.99507798 | 7.3 |
| MAGIC6 | 228.36720395 | 67.32916289 | 8.5 |
| MAGIC7 | 225.81663006 | 66.25684165 | 6.87 |

**Extract data**

The python library to extract data from vtk files is “vtk”. Vtu files are special format of vtk files. Paraview itself is based on vtk libraries to read the vtu files.

To install vtk version compatible with python 3 go: <https://vtk.org/download/>

Download the python wheel compatible with your python version X: [vtk-9.0.1-cp3X-cp3Xm-manylinux2010\_x86\_64.whl](https://www.vtk.org/files/release/9.0/vtk-9.0.1-cp36-cp36m-manylinux2010_x86_64.whl)

Note that the vtk library compatible with python3.9 does not exist yet. Replace the X with the python3 version you have. Vtk9 is compatible with python 3.5, 3.6, 3.7 and 3.8.

In the folder in which you have download the wheel:

pip3.X install vtk-9.0.1-cp3X-cp3Xm-manylinux2010\_x86\_64.whl

You will also need to have *vtktools.py* which is a useful script encapsulating some vtk libraries feature.

To extract all the data of the vtu file use *ExtractData.py* pythton script.

If you want to extract data at sensors location use the python script *ExtractSensors.py*

If you want to plot the data of sensors for the experiment and the simulation, run *Plot.py*

To find which nodes (from the unstructured mesh) are the closest in the mesh of the real sensor location use *ClosestPoints.py*. Note that the parameter to change the influence zone is influenceArea.

**Summary of the files in GAN\_CO2 folder**

***Documentation***

README\_CO2Data.docs: this file

influenceAreaSensors.pptx: explanation of the influence area as defined in ClosestPoints.py

***Python Script***

ExtractData.py: Extract all the data of the fluidity simulation

ExtractSensors.py: Extract the data into fluidity simulation at sensor location

Plot.py: Plot the experiment and fluidity data as a function of time at sensor location.

ClosestPoints.py: Find the nodes in the vtu file corresponding to the sensor location in the experiment

vtktools.py: script encapsulating some useful vtk features

***Output files***

CO2\_Experiment.dat: CO2 data from experiment. Input given.

CO2\_Fluidity.dat: Fluidity data at sensors location. Output of ExtractSensors.py

TimeVTU.dat: Time of each vtu files. Output of ExtractSensors.py

Sensor1\_NodesIDs.dat: IDs of the nodes in the vtu file corresponding to the location of sensor 1 in the experiment. Output of ClosestPoints.py

Sensor2\_NodesIDs.dat: IDs of the nodes in the vtu file corresponding to the location of sensor 2 in the experiment. Output of ClosestPoints.py

Sensor3\_NodesIDs.dat: IDs of the nodes in the vtu file corresponding to the location of sensor 3 in the experiment. Output of ClosestPoints.py

Sensor4\_NodesIDs.dat: IDs of the nodes in the vtu file corresponding to the location of sensor 4 in the experiment. Output of ClosestPoints.py

Sensor5\_NodesIDs.dat: IDs of the nodes in the vtu file corresponding to the location of sensor 5 in the experiment. Output of ClosestPoints.py

Sensor6\_NodesIDs.dat: IDs of the nodes in the vtu file corresponding to the location of sensor 6 in the experiment. Output of ClosestPoints.py

Sensor7\_NodesIDs.dat: IDs of the nodes in the vtu file corresponding to the location of sensor 7 in the experiment. Output of ClosestPoints.py

***Output plots***

CO2\_FluidityExperiment\_sensor1.svg: Comparison of fluidity and experiment CO2 data at sensor 1. Output of Plot.py

CO2\_FluidityExperiment\_sensor2.svg: Comparison of fluidity and experiment CO2 data at sensor 2. Output of Plot.py

CO2\_FluidityExperiment\_sensor3.svg: Comparison of fluidity and experiment CO2 data at sensor 3. Output of Plot.py

CO2\_FluidityExperiment\_sensor4.svg: Comparison of fluidity and experiment CO2 data at sensor 4. Output of Plot.py

CO2\_FluidityExperiment\_sensor5.svg: Comparison of fluidity and experiment CO2 data at sensor 5. Output of Plot.py

CO2\_FluidityExperiment\_sensor6.svg: Comparison of fluidity and experiment CO2 data at sensor 6. Output of Plot.py

CO2\_FluidityExperiment\_sensor7.svg: Comparison of fluidity and experiment CO2 data at sensor 7. Output of Plot.py

CO2\_FluidityExperiment.svg: Comparison of fluidity and experiment CO2 data for all sensors. Output of Plot.py

CO2\_Fluidity.svg: Fluidity data for all sensors. Output of Plot.py

CO2\_Experiment.svg: Experiment data for all sensors. Output of Plot.py