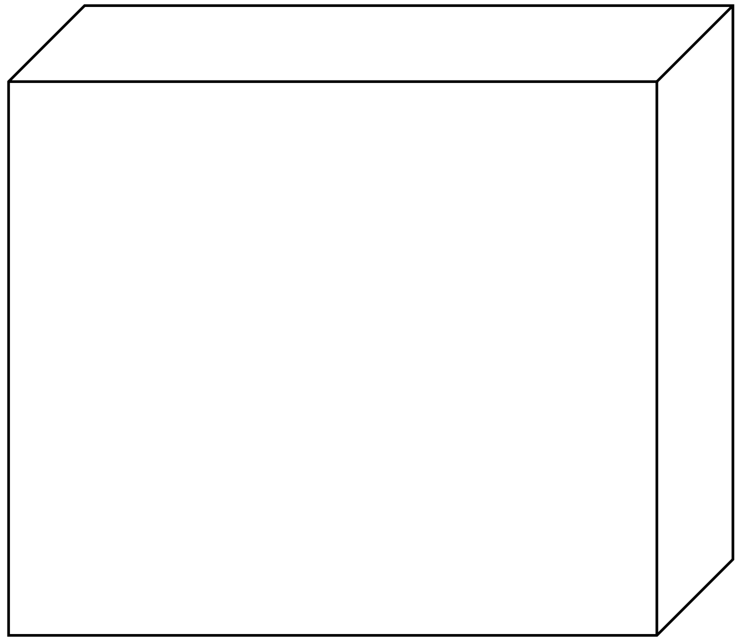
Miguel MARTÍNEZ VALERO

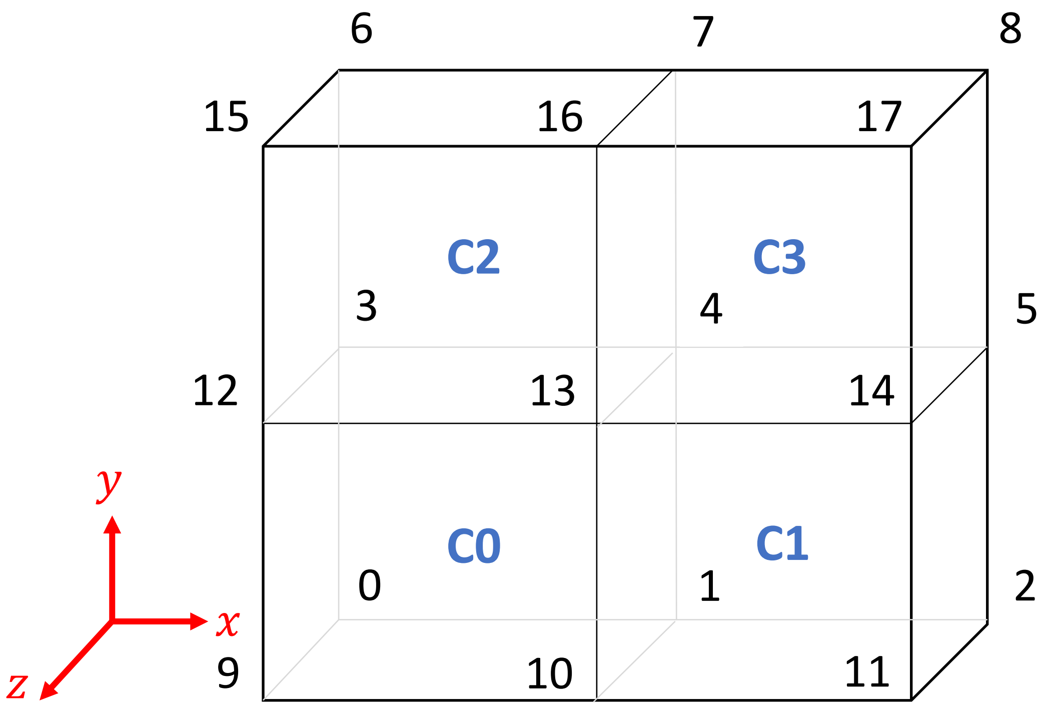
FIRST STEPS WITH CWIPI

Supervisor: Dr. Marcello MELDI

April 22nd 2022

1. COUPLE A CASE WITH OPENFOAM





CAVITY CASE (OPENFOAM)

this file must send the y coordinate and Receive the x coordinate from the other file

SIMPLE C++ FILE WITH THE SAME GEOMETRY

this file must send the x coordinate and Receive the y coordinate from the other file

In both files → $L_x = L_y = 1\,m$ $L_z = 0.1\,m$

2. WORKING DIRECTORY

- **Inside my CWIPI directory:**

  I. Makefile

  II. My C++ file (in the future this C++ file will stand for the EnKF)

  III. My OpenFOAM solver

  IV. Directories *0*, *constant* and *system* of the case we are coupling (at least it's necessary that the folder *system* is here, otherwise the command will not detect the *controlDict* file)

- **To use CWIPI once the files are compiled:**

$$mpirun\ -np\ cores.1st.file\ \ ./exec\_1st\_file : -\ np\ cores.2nd.file\ \ /.exec\_2nd\_file$$

3. DEFINE MESH

- **HOW TO DEFINE THE MESH IN CWIPI?**

```
void cwipi_define_mesh       (const char *coupling_id,
                              const int n_vertex,
                              cont int n_element,
                              double coordinates [ ],
                              int connectivity_index [ ],
                              int connectivity [ ])

void cwipi_add_polyhedra     (const char *coupling_id,
                              const int n_element,
                              int face_index [ ],
                              int cell_to_face_connectivity [ ],
                              const int n_faces,
                              int face_connectivity_index [ ],
                              int face_connectivity [ ])
```

4. STEPS TO DO NEXT

1. Make the 3D mesh work
2. Check the coupling with the *IcoFoam* solver of OpenFOAM
3. Increase the number of variables (i.e. x and y coordinates) to be sent and received ($stride > 1$ and save corresponding space in memory)
4. Introduce everything in a temporal loop
5. Change the variables and start trying an exchange of the components of the velocity $\boldsymbol{U}(u, v, w)$

   ➡ possibly here we are going to have several problems because we will need to understand the way in which OpenFOAM creates the mesh

6. Change the C++ file by the Ensemble Kalman Filter (start only with one ensemble)

   ➡ possibly here we are going to have several problems because we will need to switch from synchronous exchange primitive to asynchronous exchange primitive

7. Try with several ensembles (look at the possibility to run several simulations with only one solver?)