An overview of OpenFOAM

Arash Alizad Banaei/PDC

What is OpenFOAM?

Open Source **F**ield **O**peration **A**nd **M**anipulation

A computational fluid dynamics (CFD) software

Solves equations of fluid motion with finite volume method

First developed by OpenCFD Ltd

Open source

The source code is written in C++

OpenFOAM versions

- There are two main variants of OpenFOAM:

    - OpenCFD version :
        - New versions released twice a year (June-December)
        - Versions are named as (vYYMM) like v1912
        - Website: https://www.openfoam.com

    - OpenFOAM foundation version :
        - No defined date for new release
        - Versions are named with with two digits like 4.1
        - Website: https://openfoam.org

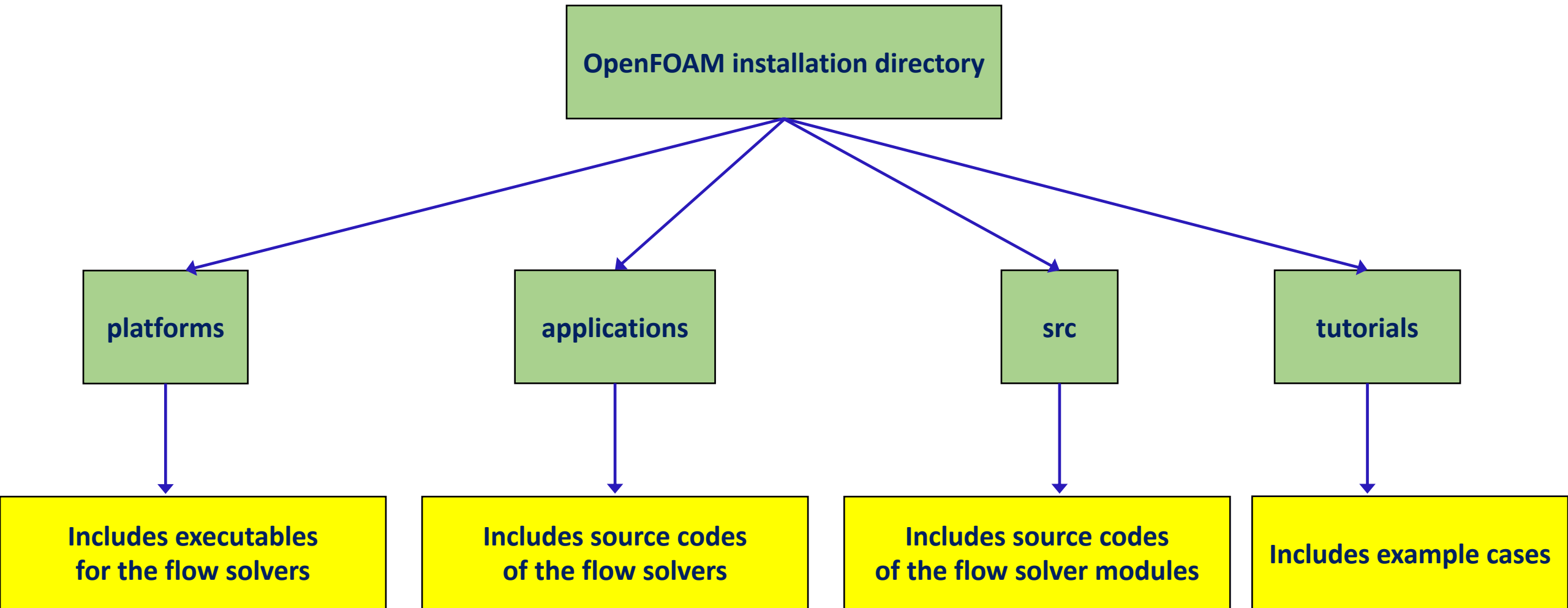# OpenFOAM versions
What is the story behind this?

- First version of OpenFOAM was released by OpenCFD in 2004

- OpenCFD kept issuing new releases up to 2.0.1 (2011)

- Silicon Graphics International (SGI) acquired OpenCFD

- OpenFOAM foundation was created with permission of using OpenFOAM trademark

- Since 2016, OpenCFD has started issuing OpenFOAM by itself

- For more information visit : https://www.openfoam.com/news/history

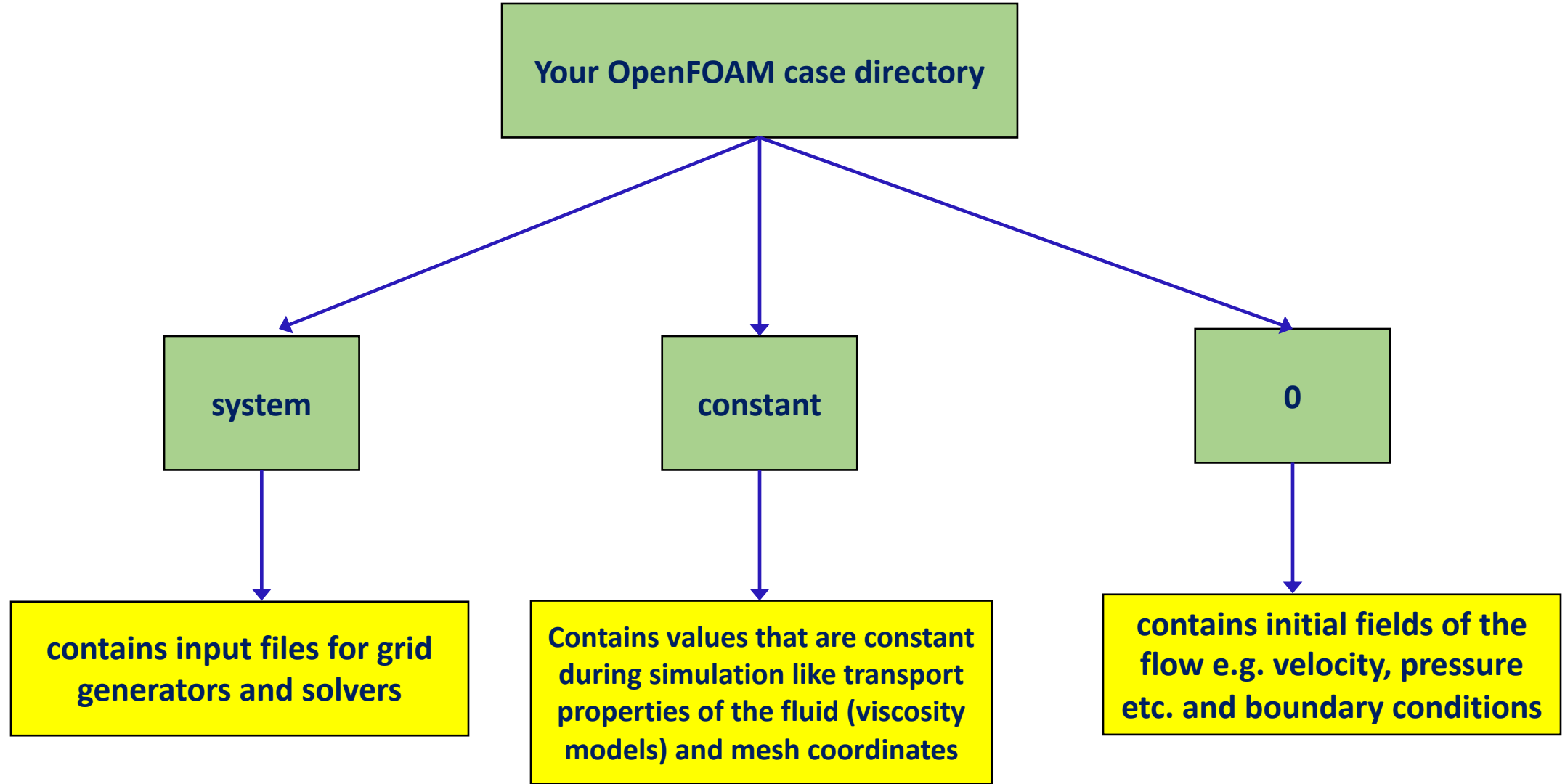# OpenFOAM versions
Which version to use?

- Depends on the feature you want to use

- Check the website and release notes to see which one fits better to your framework

- If both include the features you need, do some performance and accuracy benchmarks to see which one is better

- Otherwise it is just matter of taste!

- We will be using v1912 for the training

OpenFOAM environment

OpenFOAM input files

To setup a case, you need to have 3 directories in your case directory

OpenFOAM input files
Input file format

- For input files, OpenFOAM uses a plain text dictionary format with keywords and values

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

scale   0.1;

vertices                keyword
(
    (0 0 0)
    (1 0 0)
    (1 1 0)
    (0 1 0)                 value
    (0 0 0.1)
    (1 0 0.1)
    (1 1 0.1)
    (0 1 0.1)
);
```

OpenFOAM executables

- Unlike many other software, OpenFOAM does not have a unique executable. For every solver, mesh generation etc. there is a separate executable!

- You should run the right executable according to the solver you are using!
    - 'simpleFoam': if you use SIMPLE algorithm
    - 'icoFoam': if you use PISO algorithm for laminar flow
    - …
    - Check the documentation to see recommended solvers for different cases

OpenFOAM output files

- Similar to the input files, the output files are also in plain text dictionary format

```
internalField    nonuniform List<scalar>
400
(
1.66908e-08
-0.00577259
-0.0128362
-0.0182176
-0.0201885
-0.0179911
-0.0115712
-0.00137173
```

Do not worry if you have not understood anything! We will go through the files with an example

Postprocessing of the output data

- **ParaView** is the main tool for postprocessing of the output data

- ParaView is a general visualization tool (not part of OpenFOAM)

- It can be used in several ways to visualize OpenFOAM results:
  - Load the output data using ParaView data loader for OpenFOAM
  - Load the output data using the reader module supplied by OpenFOAM
  - Convert output data to VTK format by OpenFOAM conversion module and load it to ParaView
  - Convert output data to VTK in runtime using OpenFOAM function objects

- '**paraFoam**' is an OpenFOAM script to launch ParaView with OpenFOAM reader module
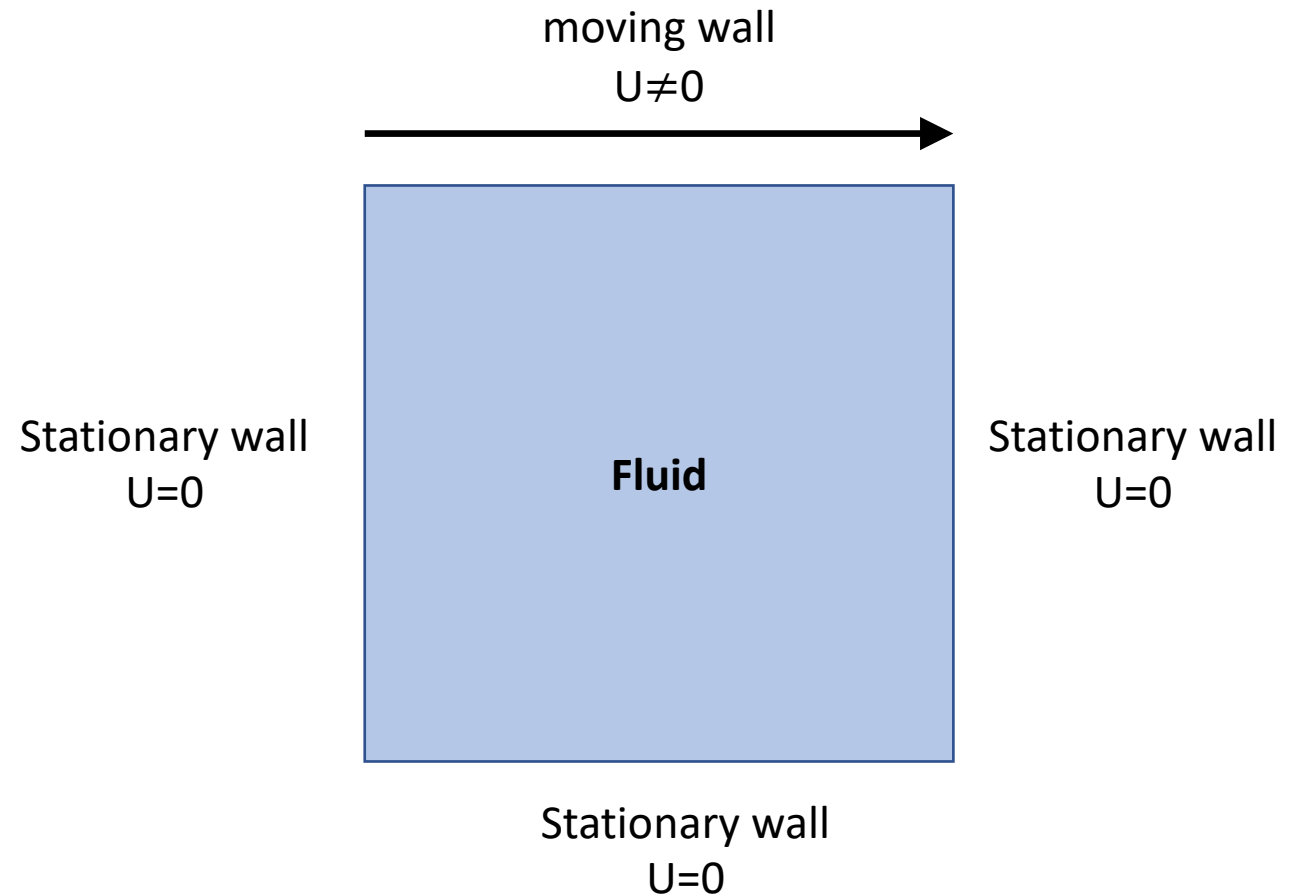
Parallelisation in OpenFOAM

- Parallelisation in OpenFOAM is Message Passing Interface (MPI)
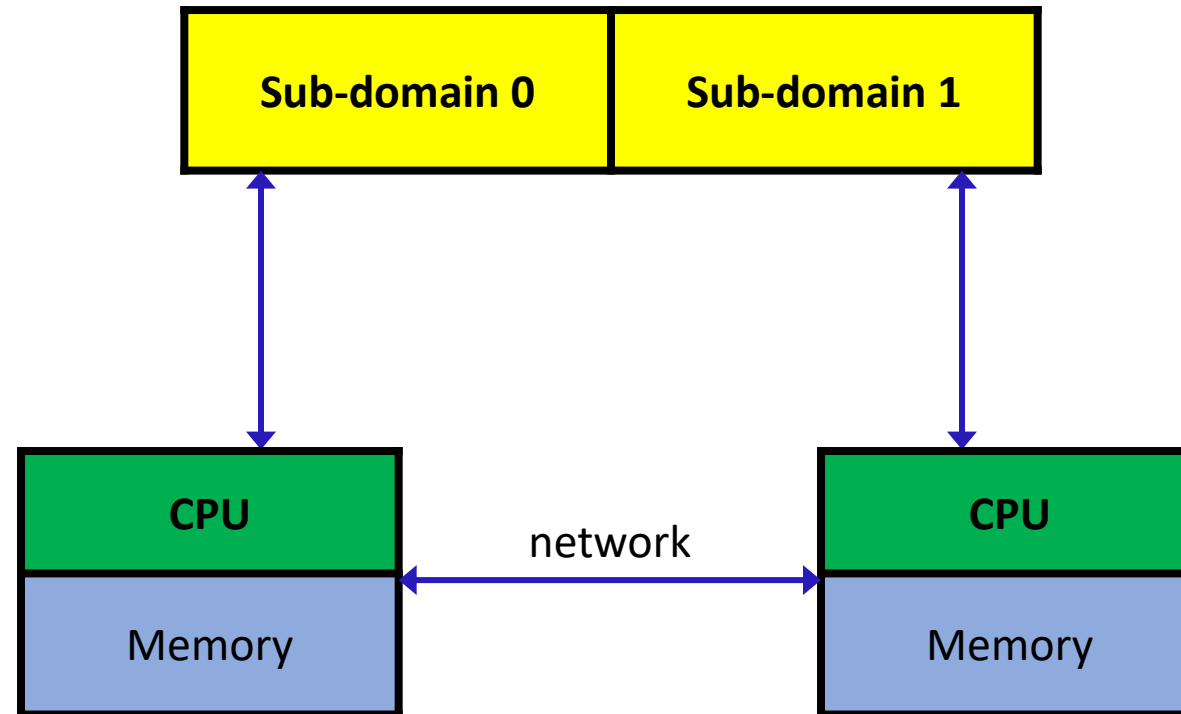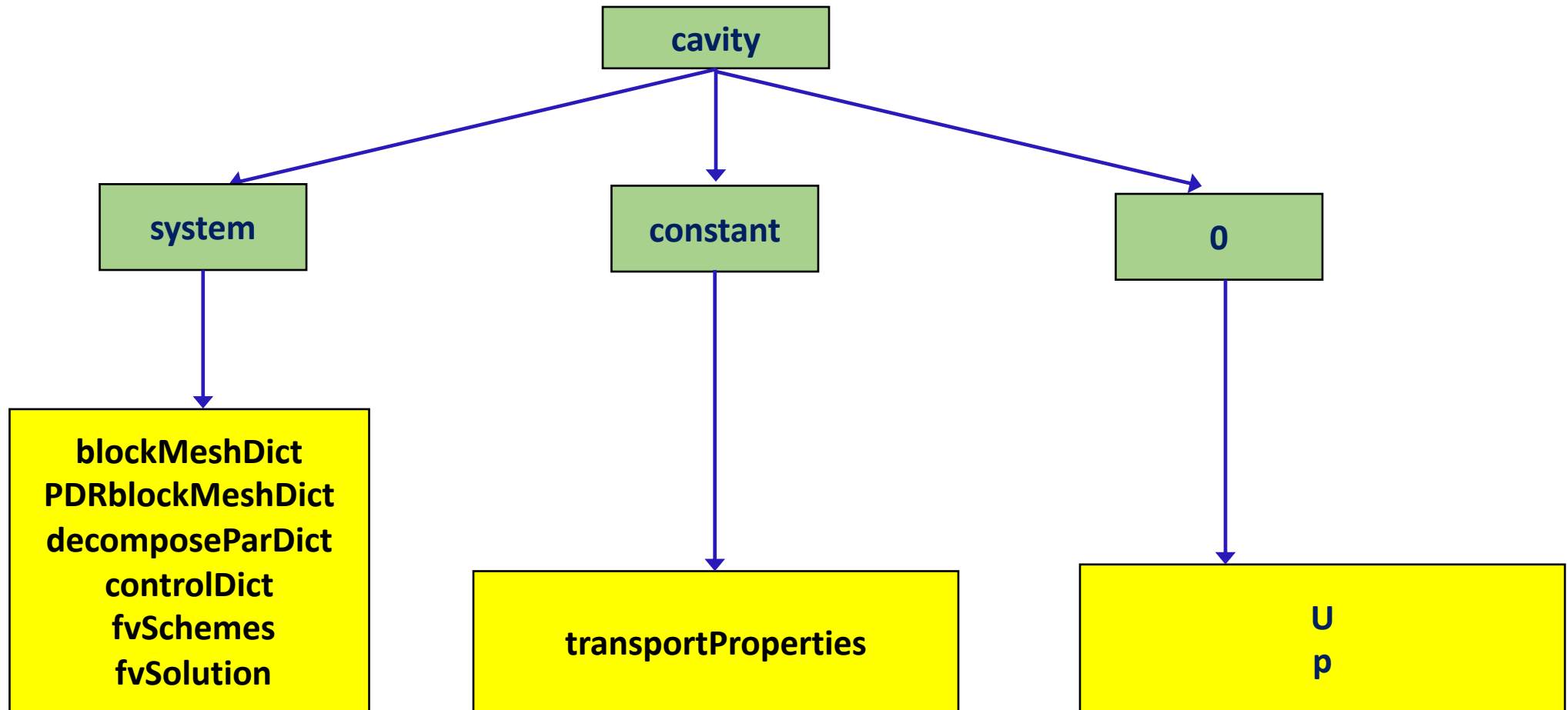
Lid driven cavity
Geometry and assumptions

- The flow is 2 dimensional

- The fluid is incompressible (density does not change)

- The flow is isothermal (no heat transfer)

moving wall
$U \neq 0$

Stationary wall
$U=0$

Fluid

Stationary wall
$U=0$

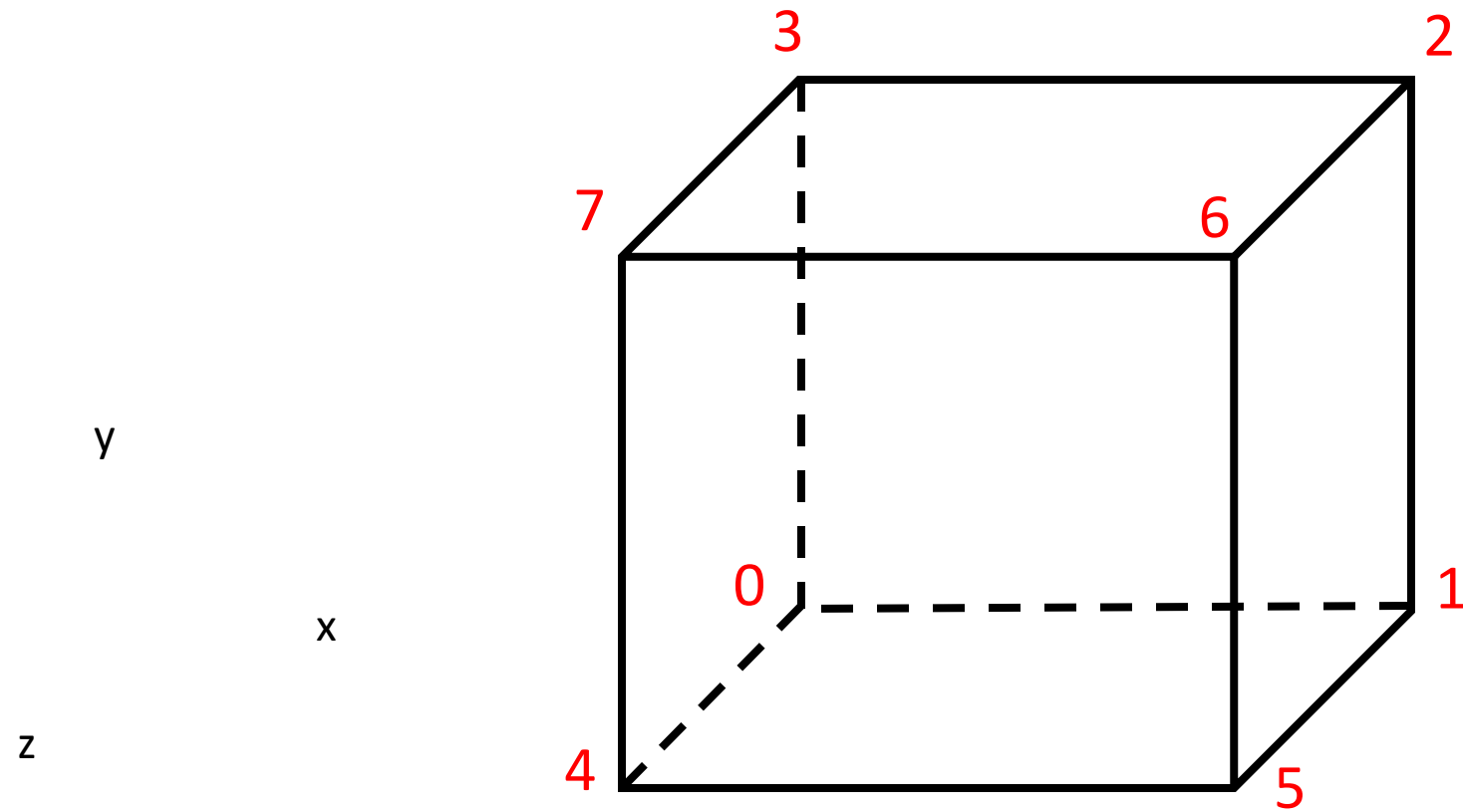Stationary wall
$U=0$

Lid driven cavity
input files

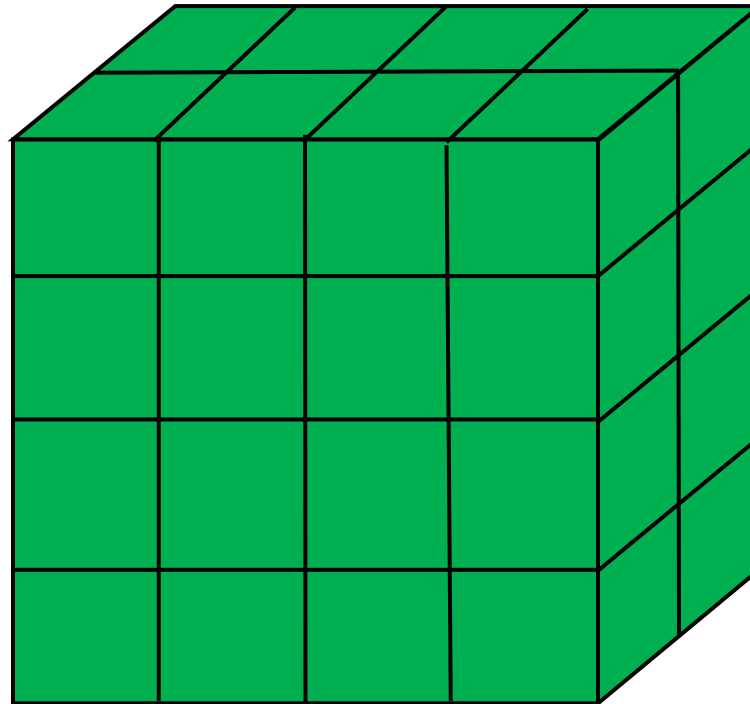- Located at tutorials/incompressible/icoFoam/cavity/cavity

Lid driven cavity
Geometry

Lid driven cavity
blockMesh utility

decomposes the domain geometry into a set of 1 or
more three dimensional, hexahedral blocks

Lid driven cavity
simulation workflow

Domain and mesh generation → **blockMesh**

Domain decomposition → **decomposePar**

Run simulation → **icoFoam**

Merge processor data into a single file → **reconstructPar**