OpenFOAM

Enabling

Search through the available modules to see what OpenFOAM ones exist

```
$ module avail openfoam
```

This only shows versions that are compatible with other loaded modules. All versions can be found with the `spider` command along with directions on what else needs to be loaded to load them

```
$ module spider openfoam
$ module spider openfoam/8
```

Load the openfoam module (not specifying a version uses the default)

```
$ module load openfoam
```

Can use the `show` command to see what this does

```
$ module show openfoam
```

Create the top-level user OpenFOAM directory and change to it (not strictly required, but recommended by the tutorials)

```
$ mkdir -p $FOAM_RUN
$ cd $FOAM_RUN
```

Tutorial

Copy the OpenFOAM incompressible cavity example from the tutorials directory

```
$ cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity/cavity .
$ cd cavity
```

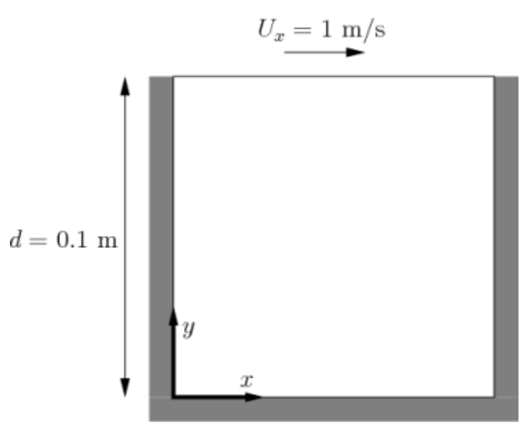

Figure 1: Lid driven cavity problem.

Generate mesh

Generate the mesh data files in *constant/polyMesh* from the higher-level description file *system/blockMeshDict*

```
$ blockMesh
```

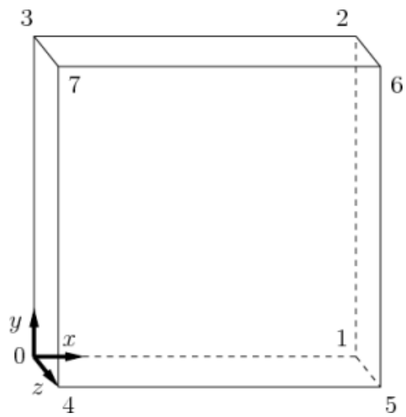

Figure 2: Lid driven cavity vertices.

Interactive run

Run the simulation (incompressible flow using the PISO algorithm) in an interactive session to test

```
$ salloc --mem-per-cpu=500M --ntasks=1 --account=def-SPONSOR
```

```
$ icoFoam
$ exit
```

Batch run

Create a *run.sh* (name not important) using `nano` to do the same

```
$ nano run.sh
```

```
#!/bin/bash
#SBATCH --time=00:05:00
#SBATCH --mem-per-cpu=500M
#SBATCH --ntasks=1
#SBATCH --account=def-SPONSOR
#SBATCH --output=icoFoam-%J.log
icoFoam
```

Submit the batch file to the cluster to run

```
$ sbatch run.sh
```

New case

Make a new *cavityMPI* case based on the *cavity* case setup

```
$ cd ..
$ foamCloneCase cavity cavityMPI
$ cd cavityMPI
```

Run in parallel

Basic four processor *system/decomposeParDict* decomposition setup copied from damBreak tutorial

```
$ cp \
  $FOAM_TUTORIALS/multiphase/interFoam/laminar/damBreak/damBreak/system/decomposeParDict \
  system/
```

Decompose the cavity across the four processors

```
$ decomposePar
```

Run the simulation in parallel using four processors

```
$ salloc --mem-per-cpu=500M --ntasks=4 --account=def-SPONSOR
```

```
$ mpirun icoFoam -parallel
$ exit
```

Collect the output back into a single file

```
$ reconstructPar
```

Prep for ParaView

Create a `NAME.foam` file to give something to open in ParaView (name doesn't matter – just identifies it as an OpenFOAM directory)

```
$ touch cavity.foam
```

Can also generate VTK input files

```
$ foamToVtk
```