Introduction to HPC for CFD

Jielin Yu
HPC User Services
LSU HPC   LONI
sys-help@loni.org

Louisiana State University
Baton Rouge
October 30, 2019

Outline

- **Things to be covered in the training**
  - Introduction to Computational Fluid Dynamics (CFD)
    - What is CFD
    - Why use CFD
    - Where is CFD used
    - Why use HPC for CFD
    - CFD process
  - Available CFD software on HPC
  - Show case of running example CFD cases
    - OpenFOAM
    - ANSYS Fluent

Introduction to CFD

- **Fluid dynamics**
  - Fluid dynamics is a subdiscipline of fluid mechanics that describes the flow of fluids.
  - Fluid flow is commonly studied in one of three methods:
    - Analytical methods
    - Experimental methods
    - Numerical methods: Computational fluid dynamics (CFD)
- **What is CFD?**
    - Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical analysis and data structures to analyze and solve problems that involve fluid flows.
    - Key governing equations: Navier-Stokes equation, Continuity equation, energy equation.
    - Numerical methods: Finite Difference Method (FDM), Finite Element Method (FEM), Finite Volume Method (FVM) …

Introduction to CFD

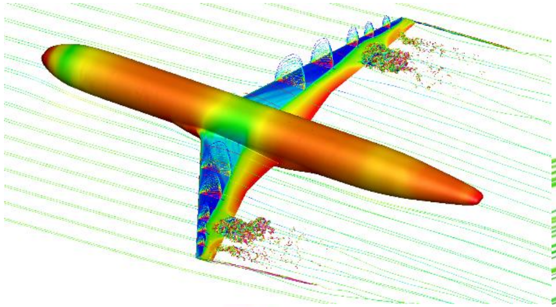
- **Why use CFD**
  - Numerical solutions
    - Analytical solutions of some Patial Differential Equations (PDEs) cannot be obtained currently. So Analytical methods are limited to simplified cases such as solving one-dimensional (1D) or 2D geometry, 1D flow, and steady flow.
  - Relatively low cost
    - Experimental methods need a lot of resources such as electricity, expensive equipment, data monitoring, and data post-processing.
    - CFD simulations are relatively inexpensive, and costs are likely to decrease as computers become more powerful.
  - Ability to simulate any conditions
    - CFD provides the ability to theoretically simulate any physical condition.
  - Comprehensive information
    - CFD allows the analyst to examine a large number of locations in the region of interest, and yields a comprehensive set of flow parameters for examination.

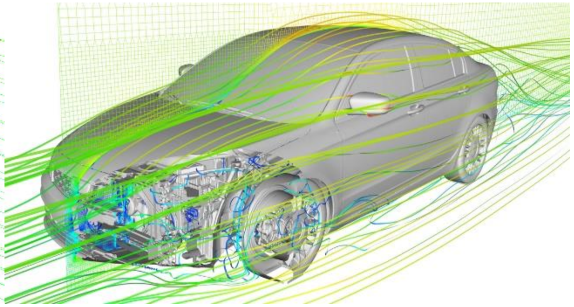
Introduction to CFD

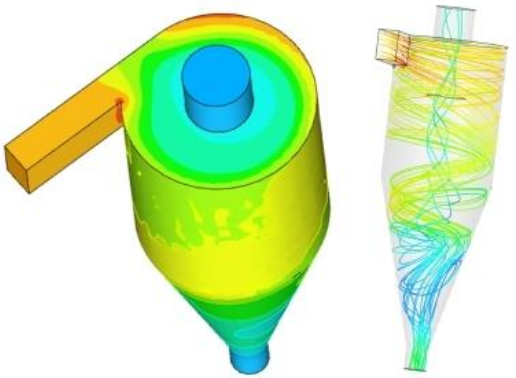
- **Why use HPC for CFD**
  - Speed
    - With the help of HPC, CFD simulations can be executed in a short period of time, which means engineering data can be introduced early in the design process.
  - Memory
    - A large CFD simulation can’t typically fit into the memory of a single machine.

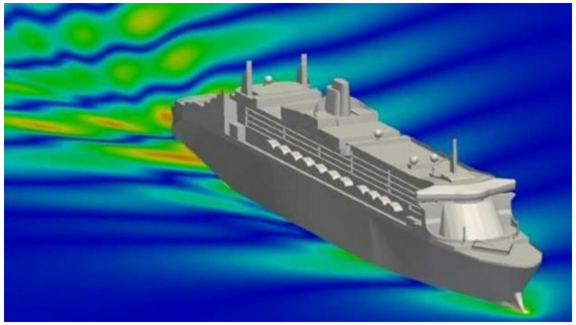
Introduction to CFD

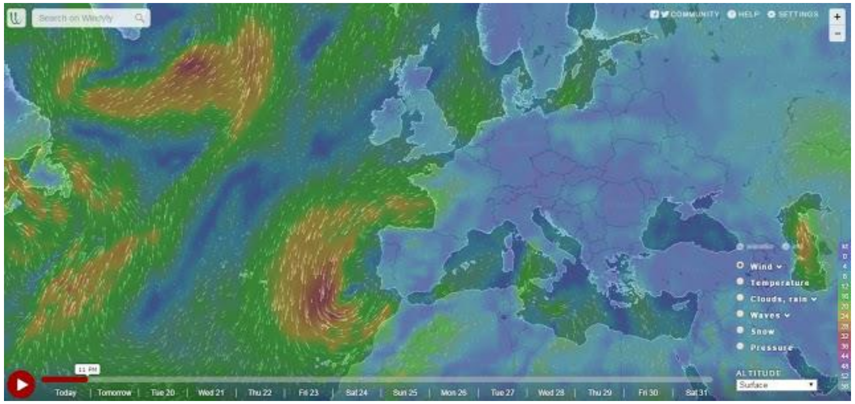
- **Where is CFD used**
  - Aerospace
  - Automotive
  - Chemical Processing
  - Hydraulics
  - Marine
  - Oil & Gas
  - Power Generation
  - Weather forecasting
  - Ocean
  - …











Introduction to CFD

- **CFD Process**

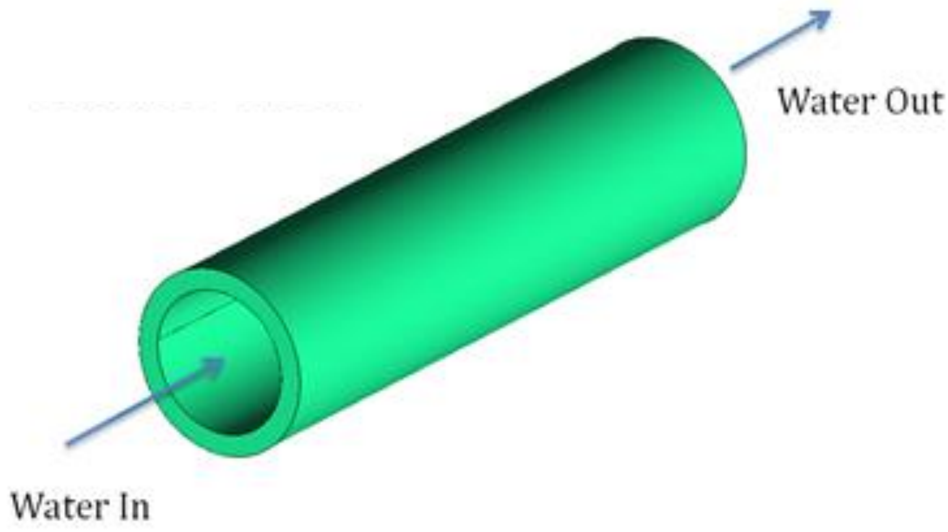
Pre processing
- Geometry building
- Meshing

Solving
- Flow conditions
- Discretization

Post processing
- Analysis
- Visualization

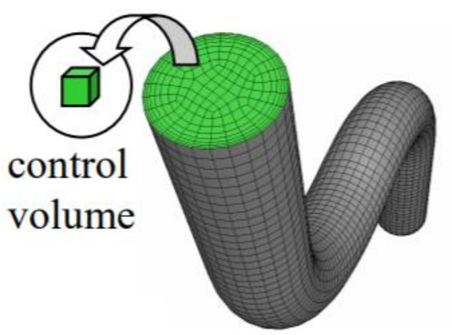
Introduction to CFD

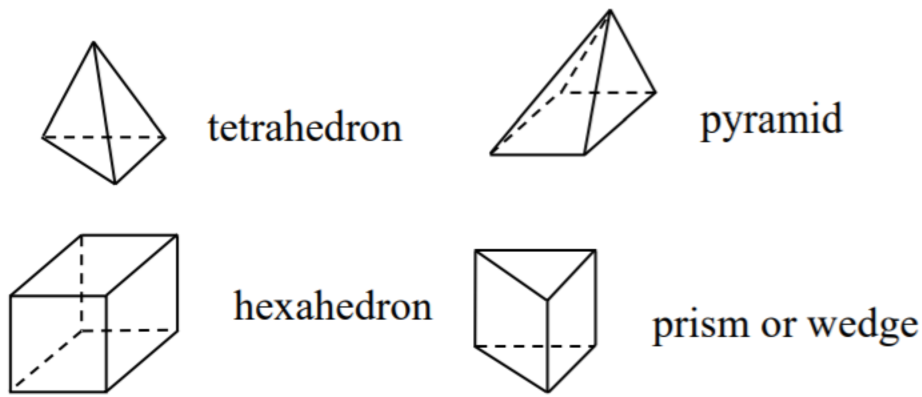
- **Pre-processing**
  - Geometry building
    - Defining the domain where the solver will solve the governing equation.
    - For example, for water flow through pipe problem, the domain of interest will be only the fluid domain (wherever there is water present) and we will be interested in solving equations only in inside part of the pipe.



Introduction to CFD

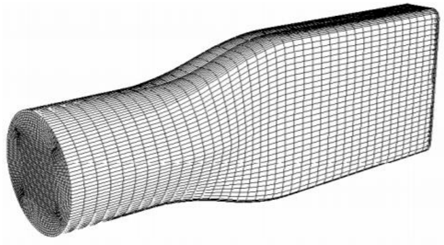
- **Pre-processing**
  - Meshing
    - Domain is discretized into a finite set of control volumes or cells. The discretized domain is called the “grid” or the “mesh”.
    - The governing equations are then discretized and solved inside each of these cells.
    - Grids can either be structured (hexahedral) or unstructured (tetrahedral).



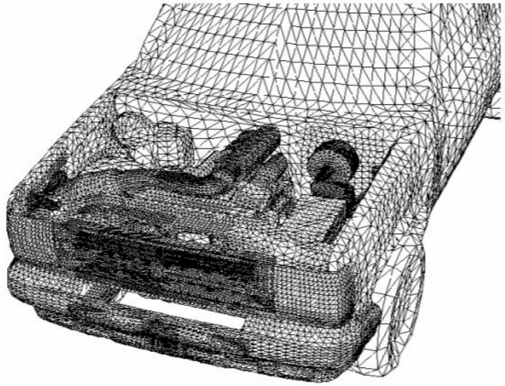


Introduction to CFD

- **Pre-processing**
  - Meshing
    - For simple geometries, quad/hex meshes can provide high-quality solutions with fewer cells than a comparable tri/tet mesh.

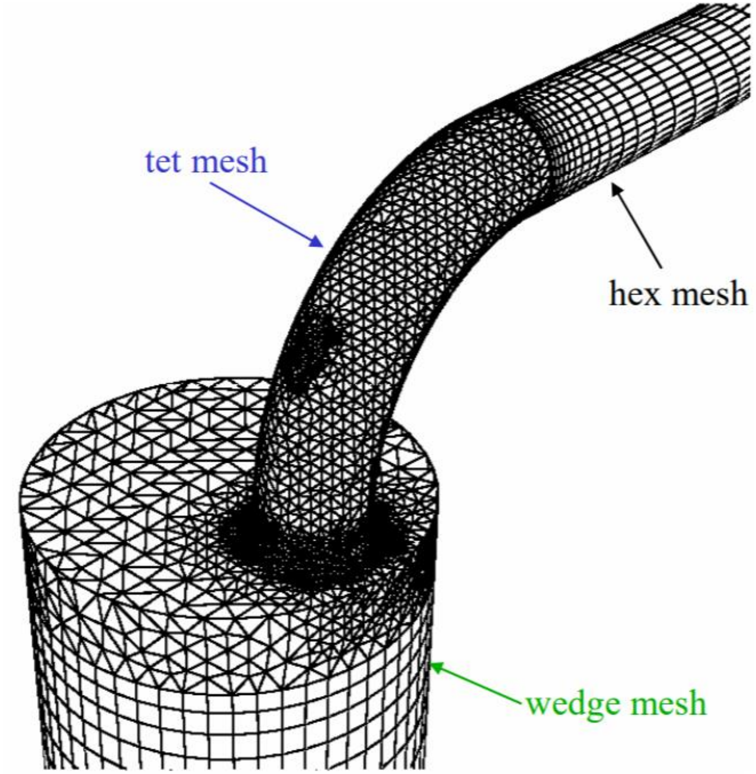


    - For complex geometries, quad/hex meshes show no numerical advantage, and you can save meshing effort by using a tri/tet mesh.



Introduction to CFD

- **Pre-processing**
  - Meshing
    - Hybrid mesh: specific regions can also be meshed with different cell types. Both efficiency and accuracy are enhanced relative to a hexahedral or tetrahedral mesh alone.



Introduction to CFD

- **Solving**
  - Physics
    - Turbulent vs. laminar
    - Incompressible vs. compressible
    - Single- vs. multi-phase
    - Thermal effects
    - Chemical reactions
    - mass transfer
    - etc
  - Initial and Boundary conditions
    - Reasonable guess of initial conditions can speed up the convergence
    - Boundary conditions: wall/periodic/inlet/outlet …
  - Discretization

Introduction to CFD

- **Post-processing**
  - Calculation
    - Flux balances
    - Wall shear stress
    - Surface and volume integrals and averages.
    - Forces and moments
  - Visualization
    - Simple X-Y plots
    - Simple 2D contours
    - 3D contour carpet plots
    - Vector plots and streamlines
    - Animations

Introduction to CFD

- **Available CFD packages**
  - Commercial
    - ANSYS Fluent
    - COMSOL
    - STAR-CCM+
  - Open-source/Free
    - OpenFOAM
    - MFiX
    - SU2

Introduction to OpenFOAM

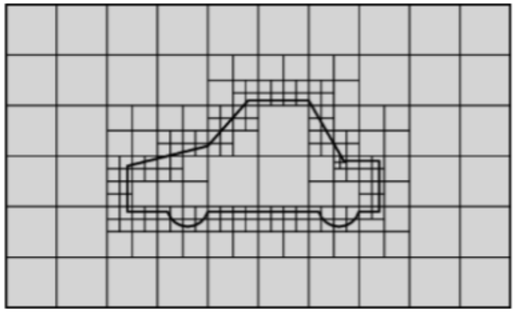
- **Open Field of Operation And Manipulation (FOAM)**
  - Free, open source CFD software package
  - C++ programming language
  - A set of libraries for continuum mechanics
- **Based on Finite Volume Method (FVM)**
- **Ideal for research oriented CFD problems:**
  - Open architecture
  - Low(Zero)-cost CFD
  - Problem-independent numerics and discretization
  - Efficient environment for complex physics problems

OpenFOAM Solvers

- **Standard solvers from: https://www.openfoam.com/documentation/user-guide/standard-solvers.php**
  - *icoFoam*: Transient solver for incompressible, laminar flow of Newtonian fluids
  - *simpleFoam*: Steady-state solver for incompressible flows with turbulence modelling
  - rhoSimpleFoam: Steady-state solver for turbulent flow of compressible fluids
  - interFoam: Solver for 2 incompressible, isothermal immiscible fluids
  - dnsFoam: Direct numerical simulation solver for boxes of isotropic turbulence
  - reactingFoam: Solver for combustion with chemical reactions
  - thermoFoam: Solver for energy transport and thermodynamics on a frozen flow field
  - DPMFoam: Transient solver for the coupled transport of a single kinematic particle cloud including the effect of the volume fraction of particles on the continuous phase

Mesh Generation

- **blockMesh**
  - For simple geometries, there is blockMesh, a multi-block mesh generator that generates meshes of hexahedra from a text configuration file.
  - Look at the OpenFOAM distribution files which contains numerous example configuration files for blockMesh to generate meshes for flows around simple geometries, e.g. a cylinder, a wedge, etc.
- **snappyHexMesh**
  - For complex geometries, meshes to surfaces from CAD
  - Can run in parallel
  - Automatic load balancing
- **Other mesh generation tools**
  - extrudeMesh
  - polyDualMesh



Mesh Conversion

- **From: http://www.openfoam.org/features/mesh-conversion.php**

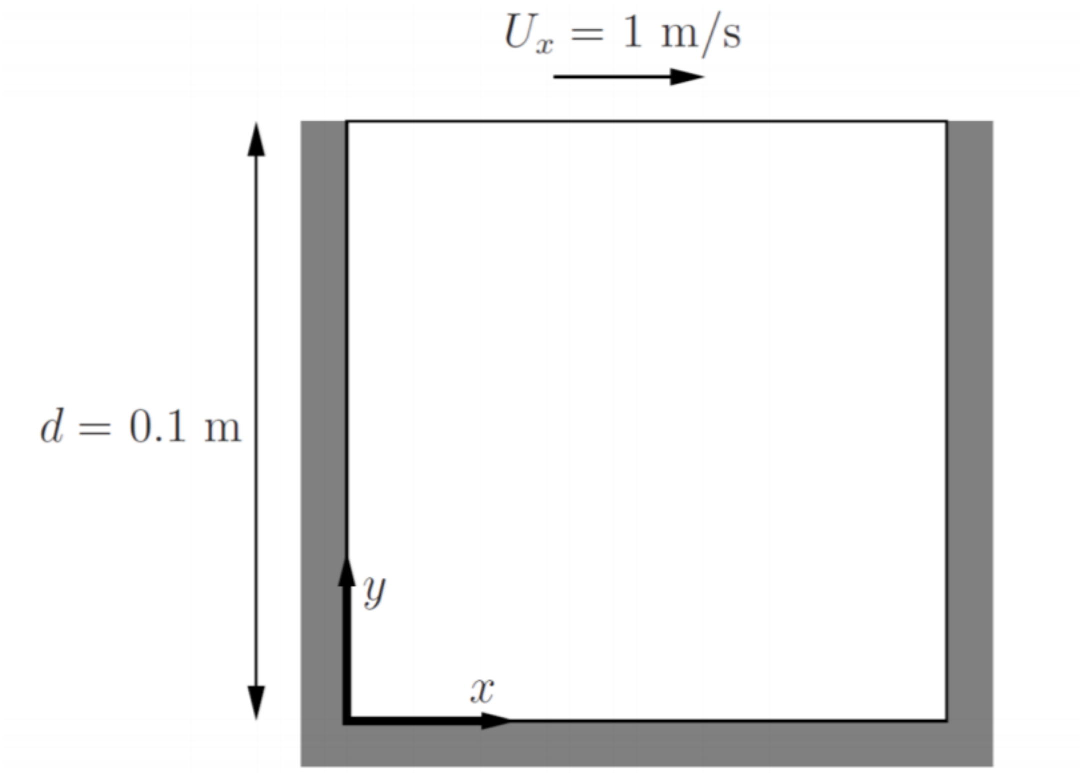
Part of the mesh converters	
ansysToFoam	Converts an *ANSYS* input mesh file, exported from *I-DEAS*, to OPENFOAM® format
cfx4ToFoam	Converts a *CFX* 4 mesh to OPENFOAM® format
datToFoam	Reads in a datToFoam mesh file and outputs a points file. Used in conjunction with blockMesh
fluent3DMeshToFoam	Converts a *Fluent* mesh to OPENFOAM® format
fluentMeshToFoam	Converts a *Fluent* mesh to OPENFOAM® format including multiple region and region boundary handling
foamMeshToFluent	Writes out the OPENFOAM® mesh in *Fluent* mesh format
foamToStarMesh	Reads an OPENFOAM® mesh and writes a *PROSTAR* (v4) bnd/cel/vrt format
foamToSurface	Reads an OPENFOAM® mesh and writes the boundaries in a surface format
gambitToFoam	Converts a *GAMBIT* mesh to OPENFOAM® format
gmshToFoam	Reads .msh file as written by Gmsh
See http://www.openfoam.org/features/mesh-conversion.php for complete list	

Running OpenFOAM on HPC/LONI clusters

- **If you're going to run it on the HPC/LONI cluster we recommend to use one of the clusters.**
  - SuperMIC: 2.3.0
  - QB2: 2.3.0
- **The package is installed as a module, so you have to load it using module before using it.**
- **However, being a free and open-source code, OpenFOAM can be modified and re-compiled by the user, if you want to run customized solver or need a particular version (e.g. OpenFOAM Ext version), please compile by yourself in home/project directory.**

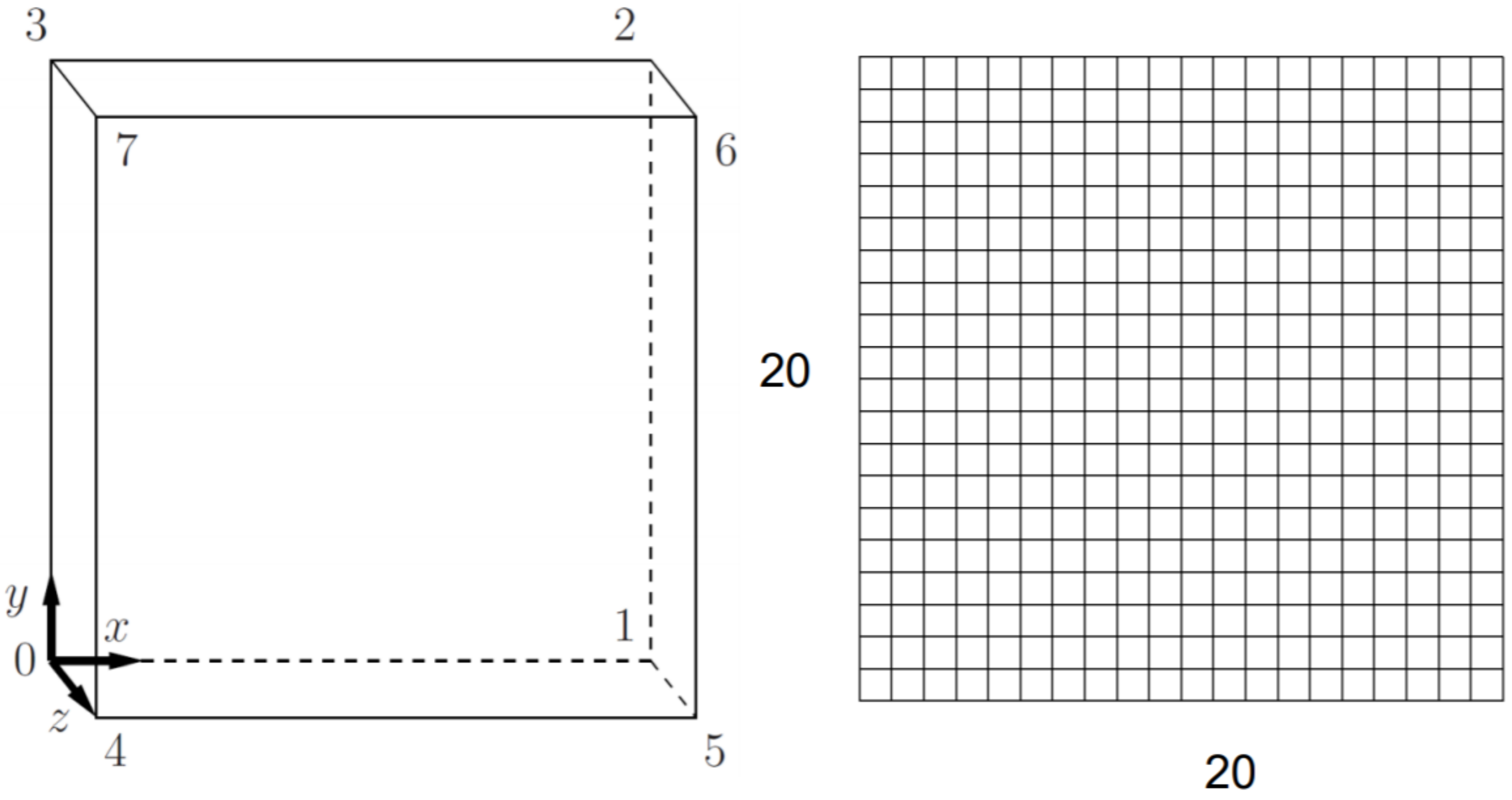
Run First OpenFOAM case

- **Lid-driven cavity flow using icoFoam**



Lid-driven cavity flow

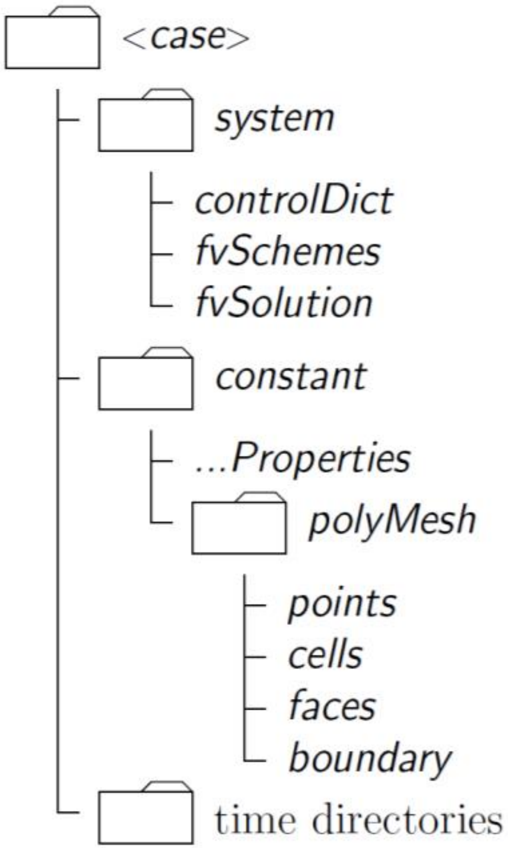
- **The cavity domain consists of a square of side length d=0.1m in the xy plane. A uniform mesh of 20x20 cells will be used initially.**



Inside case configuration

- **File structure of OpenFOAM cases**

  $ ls -R $FOAM_RUN/tutorials/incompressible/icoFoam/cavity



Inside case configuration

- **The minimum set of files required to run an OpenFOAM case**
  - constant directory:
    - description of the case mesh (geometry): e.g. ***polyMesh***
    - physical properties files: e.g. ***transportProperties***
  - system directory: solution procedure settings
    - ***controlDict***
    - ***fvSchemes***
    - ***fvSolution***
  - "time" directories: ***U, p***
    - initial conditions (I.C.)
    - boundary conditions (B.C.)
    - Future result files (typically determined by controlDict)

Inside case configuration

- **constant directory:**
  - polyMesh
    - ***blockMeshDict: mesh description, will be detailed later***
    - boundary: list of patches with BCs definition
    - faces: list of mesh faces (list of points)
    - neighbour: list of neighboring cell labels
    - owner: list of owning cell labels
    - points: list of mesh points with their coordinates
  - transportProperties

Edit blockMeshDict file (0)

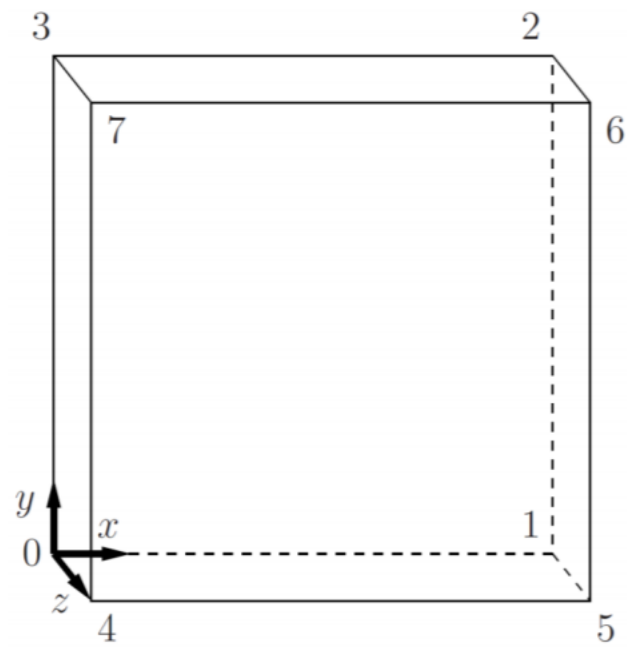
- **OpenFOAM file header:**

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.2.1                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
```

Edit blockMeshDict file (1)

- **Edit blockMeshDict file**

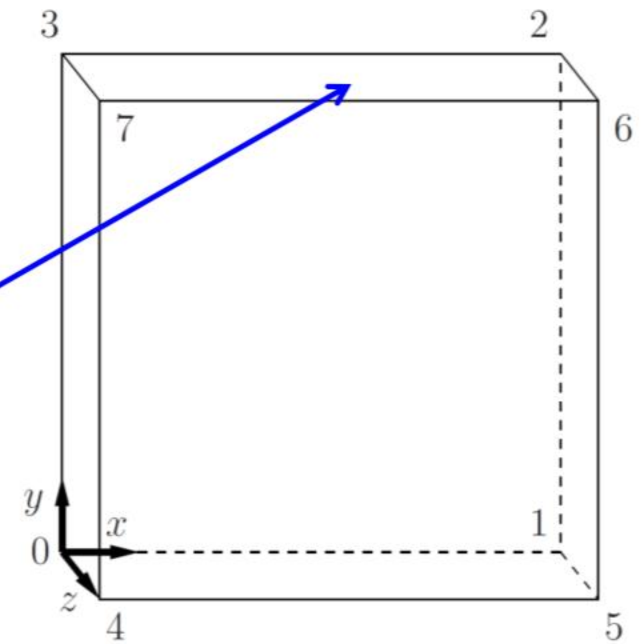
```
convertToMeters 0.1;

vertices
(
     (0 0 0)   //0
     (1 0 0)   //1
     (1 1 0)   //2
     (0 1 0)   //3
     (0 0 0.1) //4
     (1 0 0.1) //5
     (1 1 0.1) //6
     (0 1 0.1) //7
);

blocks
(
     hex (0 1 2 3 4 5 6 7) (20 20 1) simpleGrading (1 1 1)
);
```



Edit blockMeshDict file (2)

- **Edit blockMeshDict file**

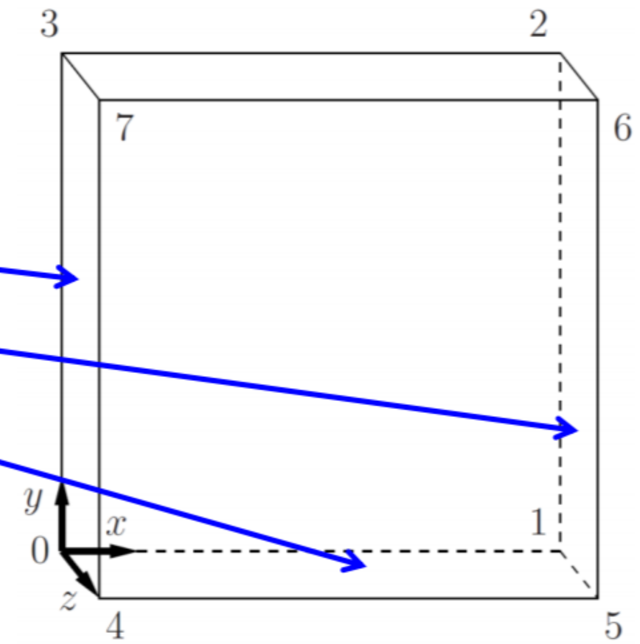
```
boundary
(
    movingWall
    {
        type wall;
        faces
        (
            (3 7 6 2)
        );
    }
```



Edit blockMeshDict file (3)

- **Edit blockMeshDict file**

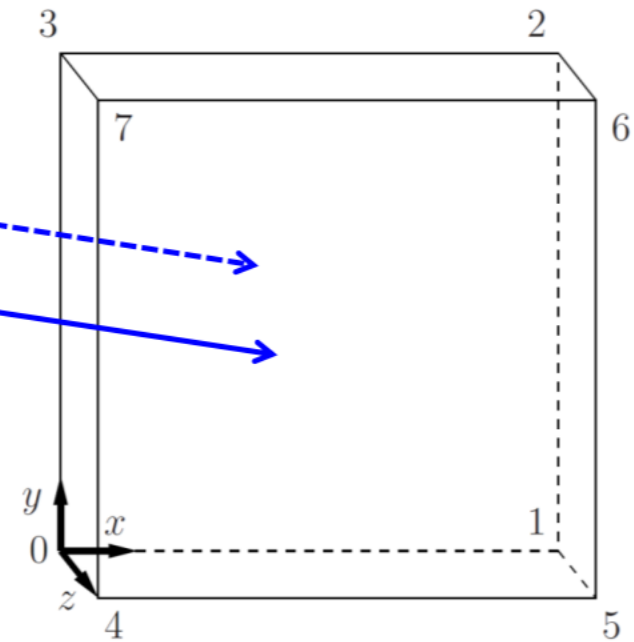
```
fixedWalls
{
    type wall;
    faces
    (
        (0 4 7 3)
        (2 6 5 1)
        (1 5 4 0)
    );
}
```



Edit blockMeshDict file (4)

- **Edit blockMeshDict file**

```
      frontAndBack
      {
          type empty;
          faces
          (
              (0 3 2 1)
              (4 5 6 7)
          );
}
);

mergePatchPairs
(
);
```



Solver settings

- **constant directory also contains:**
  - File which defines physical/material properties, transportProperties
  - Files which define some other mesh properties, e.g. dynamicMeshDict
  - Files which defines turbulent properties RASProperties
- **dimensions/units in OpenFOAM**
  - Representation of SI system
    - `//dimensions [kg m sec K mol A cd ];`
    - `dimensions [0 2 -1 0 0 0 0];`
- **Note: for incompressible solvers it is not needed to specify density. Pressure is then represented as *p/ρ***
- **transportProperties-representation of SI system**

```
transportModel Newtonian; //viscosity options:
newtonian/non-newtonian
nu nu [ 0 2 -1 0 0 0 0 ] 0.01; // kinematic viscosity
```

Solver settings

- **system directory contains:**
  - Files concerning solver parameters, as well as definition files for utility tools e.g. decomposeParDict
    - controlDict – simulation control and parameters, additional libraries to load and extra functions
    - fvSchemes – definition of discretization schemes
    - fvSolution – definitions of solver type, tolerances, relaxation factors

Solver settings-controlDict

- **controlDict: (basic time step control, how your results are written, etc.)**

```
application       icoFoam;
startFrom         startTime;
startTime         0;
stopAt            endTime;
endTime           0.5;
deltaT            0.005;
writeControl      timeStep;
writeInterval     20;
purgeWrite        0;
writeFormat       ascii;
writePrecision    6;
writeCompression  off;
timeFormat        general;
timePrecision     6;
runTimeModifiable true;
```

Solver settings-fvSchemes

- **fvSchemes:**

```
// time schemes (Euler , CrankNicholson,
backward, steadyState )
ddtSchemes
{
    default         Euler;
}
// gradient schemes (Gauss , leastSquares,
fourth, cellLimited, faceLimited )
gradSchemes
{
    default         Gauss linear;
    grad(p)         Gauss linear;
}
// convection and divergence schemes (
interpolation schemes used: linear,
skewLinear,  cubicCorrected, upwind,
linearUpwind, QUICK, TVD, SFCD, NVD)
divSchemes
{
    default         none;
    div(phi,U)      Gauss linear;
}
```

```
laplacianSchemes
{
    default         none;
    laplacian(nu,U) Gauss linear orthogonal;
    laplacian((1|A(U)),p) Gauss linear
orthogonal;
}
```

Solver settings-fvSchemes

- **fvSchemes:**

```
// interpolation schemes to calculate values on the faces (linear, cubicCorrection, midPoint , upwind, linearUpwind, skewLinear , QUICK, TVD, limitedLinear , vanLeer , MUSCL, limitedCubic, NVD, SFCD, Gamma )
interpolationSchemes
{
    default          linear;
    interpolate(HbyA) linear;
}
// schemes for surface normal gradient on the faces ( corrected, uncorrected, limited, bounded, fourth )
snGradSchemes
{
    default          orthogonal;
}
// lists the fields for which the flux is generated in the application
fluxRequired
{
    default          no;
    p;
}
```

Solver settings-solution control

- **fvSolution:**

```
solvers
{
    p
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-06;
        relTol          0;
    }

    U
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-05;
        relTol          0;
    }
}
// pressure - velocity coupling
// SIMPLE (Semi - Implicit Method for
Pressure - Linked Equations )
```

```
// PISO ( Pressure Implicit with Splitting
of Operators )
// PIMPLE ( Combination of SIMPLE and PISO
)
PISO
{
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
    pRefCell          0;
    pRefValue       0;
}
```

http://www.openfoam.org/docs/user/fvSolution.php

Solver settings-time directory

- **Time directories contain field files (e.g. U, p, k, epsilon, omega, T etc.)**
- **Fields files store field solution values on all cells and boundary conditions on the computational domain**
- ***0* time directory is initial directory containing field files with initial field values and boundary conditions.**
- **Common parts for all field files are:**
  - header
  - dimensions
  - internalField
  - boundaryField

Boundary Conditions (BCs)

- **base type (described purely in terms of geometry):**
  - patch, wall, empty, symmetry, cyclic
- **primitive type (base numerical patch condition assigned to a field variable on the patch):**
  - fixedValue, fixedGradient, zeroGradient, mixed, directionMixed, calculated
- **derived type (complex patch condition, derived from the primitive type, assigned to a field variable on the patch):**
  - Inlet/Outlet

Initial and Boundary conditions: Velocity

- **U**

```
dimensions      [0 1 -1 0 0 0 0];
internalField   uniform (0 0 0);
boundaryField
{   movingWall
    {
        type            fixedValue;
        value           uniform (1 0 0);
    }

    fixedWalls
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }

    frontAndBack
    {
        type            empty;
    }
}
```

Initial and Boundary conditions: Pressure

- **p**

```
dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0;

boundaryField
{
    movingWall
    {
        type            zeroGradient;
    }

    fixedWalls
    {
        type            zeroGradient;
    }

    frontAndBack
    {
        type            empty;
    }
}
```

Running OpenFOAM on HPC/LONI clusters

- **Start an interactive session:**

```
qsub -I -l nodes=1:ppn=20,walltime=02:00:00 –A your_allocation_name
```

- **Load OpenFOAM in module:**

```
module load openfoam/2.3.0/INTEL-140-MVAPICH2-2.0
```

Run First OpenFOAM case

➢ **Steps of running first OpenFoam case on SuperMIC/QB2:**

$ cd /work/$USER/

$ wget http://www.hpc.lsu.edu/training/weekly-materials/2019-Fall/hpccfd.tar

$ tar xf hpccfd.tar

$ cd hpccfd/OpenFOAM/cavity/

$ blockMesh #generate mesh information

$ icoFoam #running the PISO solver

$ foamToVTK #convert to VTK format, optional

$ paraFoam #post-processing, not suggested for large scale problems

system/decomposeParDict

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      decomposeParDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
numberOfSubdomains 16;
//Simple geometric decomposition in which the domain is split into
//pieces by direction, e.g. 4 pieces in x direction, 2 in y, 2 in z
method          simple;
simpleCoeffs
{
    n               ( 4 2 2 );
    delta           0.001;
}
```

Running in parallel

- **`decomposePar`** **//specify the parallel run params**
- **`mpirun --hostfile <machines> -np <nProcs>`**
  `<foamExec> <otherArgs> -parallel > log`
  - Examples on QB2 (use proper values in decomposePar):

```
mpirun --hostfile $PBS_NODEFILE -np 20 icoFoam -parallel > log
```

- **`reconstructPar`** **//merge time directories sets from each processor**

Post-processing

- **Most used post-processing software for OpenFOAM data visualization is Paraview**
- **paraFoam – script for automatic import of OpenFOAM results into Paraview**
- **OpenFOAM Data transformation in other formats: e.g. foamToVTK (also used by Paraview)**

Documentation and Help

- **OpenFOAM course: http://www.tfd.chalmers.se/~hani/kurser/OS_CFD/**
- **OpenFOAM homepage: www.openfoam.com**
- **OpenFOAM User Guide, Programmers Guide, Doxygen**
- **OpenFOAM Wiki: www.openfoamwiki.net**
- **OpenFOAM-extend:**
  - http://sourceforge.net/projects/openfoam-extend/,
  - http://www.foam-extend.org
  - http://extend-project.de
- **OpenFOAM Forum: http://www.cfd-online.com/Forums/openfoam/**
- **OpenFOAM workshop: www.openfoamworkshop.org**
- **CoCoons project: http://www.cocoons-project.org/**
- **User forum:**
  - http://www.cfd-online.com/Forums/openfoam/

Introduction to ANSYS Fluent

- **ANSYS**
  - a general purpose software, used to simulate interactions of all disciplines of physics, structural, vibration, fluid dynamics, heat transfer and electromagnetic for engineers.
- **ANSYS Fluent**
  - Commercial CFD software package
    - Detailed documentations
    - User service support
  - Based on Finite Volume Method (FVM)
  - GUI
  - User Define Functions (UDF)

Geometry&Mesh Generation

- **Geometry**
  - ANSYS Workbench
    - ANSYS DesignModeler
    - ANSYS SpaceClaim
  - Import geometry
- **Meshing**
  - ANSYS Workbench
    - ANSYS Meshing
  - Import mesh
    - GAMBIT, GeoMesh, Tgrid, etc

ANSYS Fluent features

- **Features from: https://www.ansys.com/products/fluids/ansys-fluent/ansys-fluent-features**
  - Turbulence Modeling
  - Heat Transfer & Radiation
  - Multiphase Flow
  - Reacting Flow
  - Acoustics
  - Optimize Your Design – Automatically
  - etc

Running ANSYS on HPC/LONI clusters

- **If you're going to run it on the HPC/LONI cluster we recommend to use one of the clusters.**
  - SuperMike2: 18.1 and 19.3
  - SuperMIC: 15.0, 16.0, 17.0 and 18.1
- **The package is installed as a module, so you have to load it using module before using it.**

Running ANSYS on HPC/LONI clusters

- **Load ANSYS in module:**

```
module load ansys/19.3
```

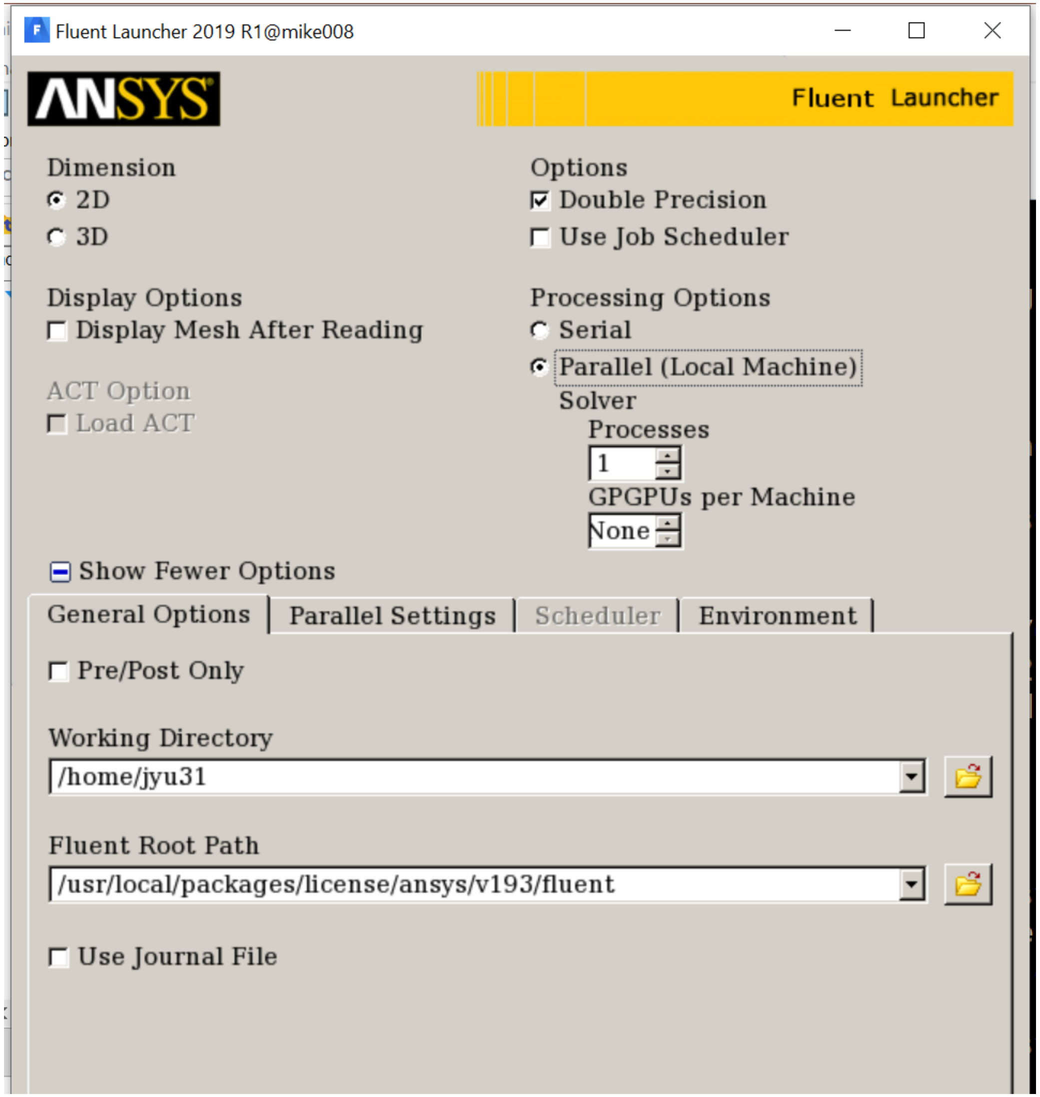
- **Start an interactive session:**

```
qsub -I -l nodes=1:ppn=20,walltime=02:00:00 –A your_allocation_name
```

- **Do *NOT* start ANSYS GUI on head node!!!**
- **Start the Ansys Fluent Launcher by typing *`fluent`* on terminal:**

```
$ cd /work/$USER/
$ wget http://www.hpc.lsu.edu/training/weekly-materials/2019-Fall/hpccfd.tar
$ tar xf hpccfd.tar
$ cd hpccfd/Fluent/
$ fluent
```

Running ANSYS on HPC/LONI clusters



Case setup (1)

- **Read the grid file, cavity.msh**

  Select File > Read > Mesh from the menu, select cavity.msh

  Select "Display Mesh After Reading" option
- **Check the grid**

  In the left panel, double click General, then click Check in the right panel under Mesh section.
- **Check the scale of the grid**

  Click Scale in the right panel under Mesh section.
- **Display the grid**

  Click Display in the right panel under Mesh section.
- **Check the Models**

  In the left panel, double click Models, then click Viscous in the right panel and click Edit.

Case setup (2)

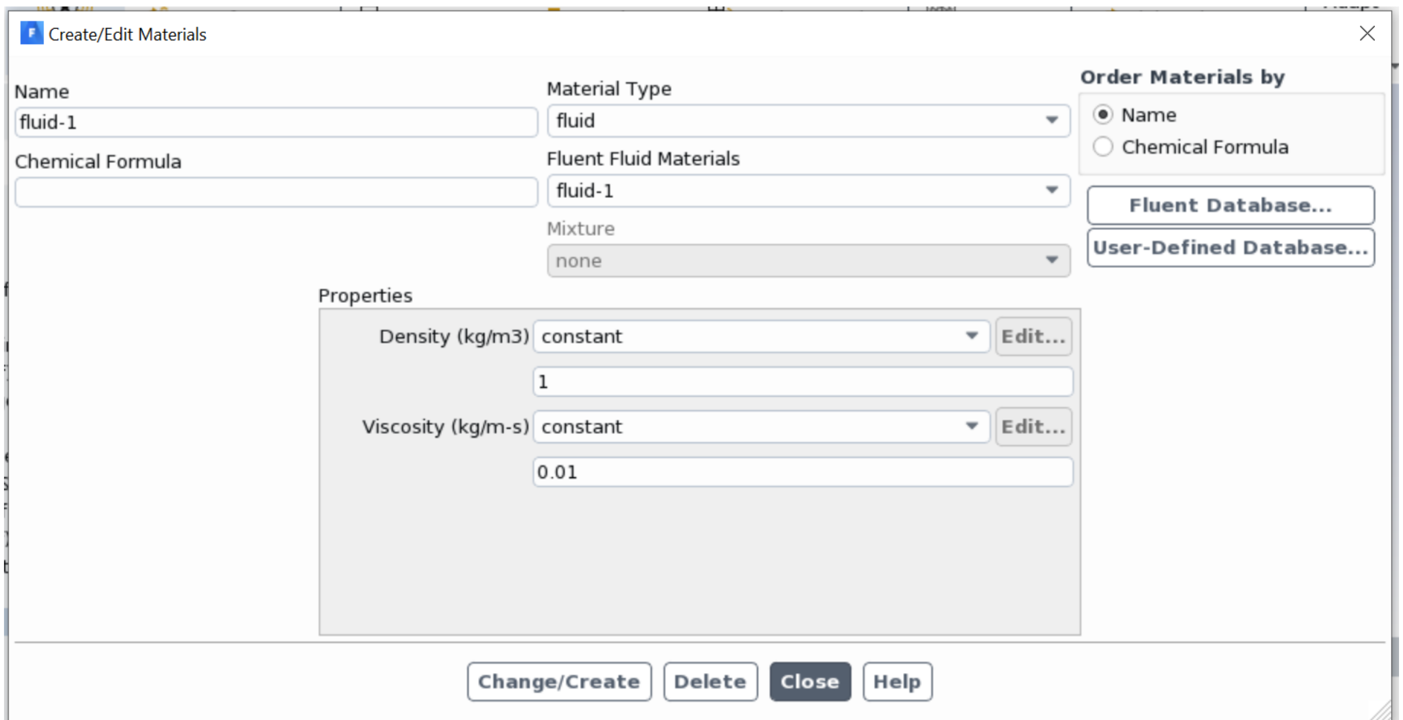
- **Create Materials**

  In the left panel, double click Material, then right click Fluid and select New

  Set the Density (kg/m3) to 1.

  Set Viscosity (kg/m-s) to 0.01.

  Click on Change/Create and close the panel.

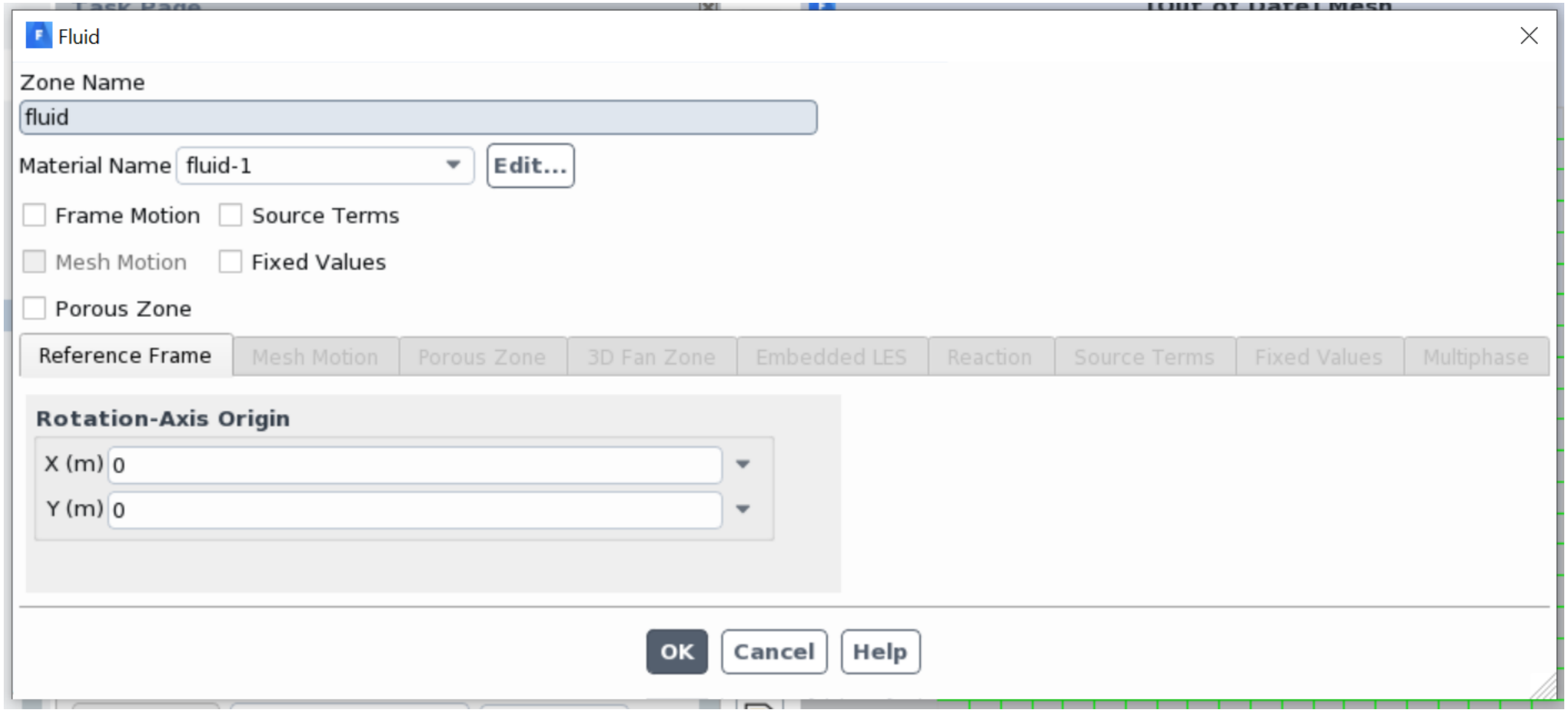


Case setup

- **Cell Zone Conditions**

  In the left panel, double click Cell Zone Conditions, then select fluid in the right panel and select Edit

  Set the Material Name to fluid-1 and click OK.



Case setup

- **Boundary Conditions**

  In the left panel, double click Boundary Conditions, then select wall-top in the right panel and select Edit

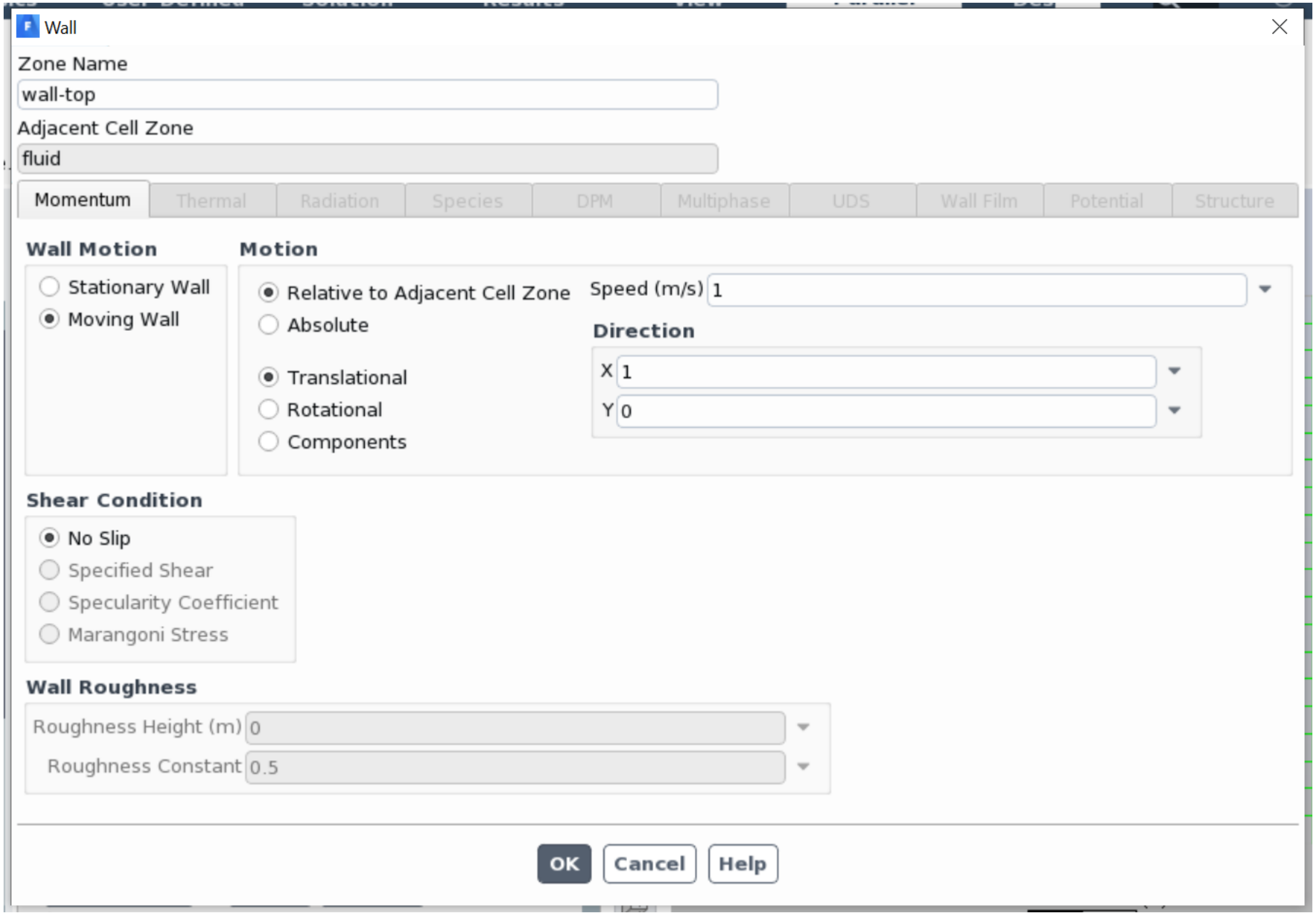
  Set the Material Name to fluid-1 and click OK.

  Under Wall Motion, select Moving Wall.

  Set the Speed (m/s) as 1 and click OK.

Case setup

- **Boundary Conditions**



Solution

- **Solution**

  In the left panel, double click Methods in Solution section
- **Save the case file**

  Select File > Write > Case from the menu
- **Initialization**

  In the left panel, double click Initialization in Solution section

  Set the Initialization Methods to Standard Initialization and click Initialize.
- **Run**

  In the left panel, double click Run Calculations in Solution section

  In the right panel:

  Set Time Step Method to User Specified.

  Set Pseudo Time Steps (s) to 0.005.

  Set Number of Iterations to 100.

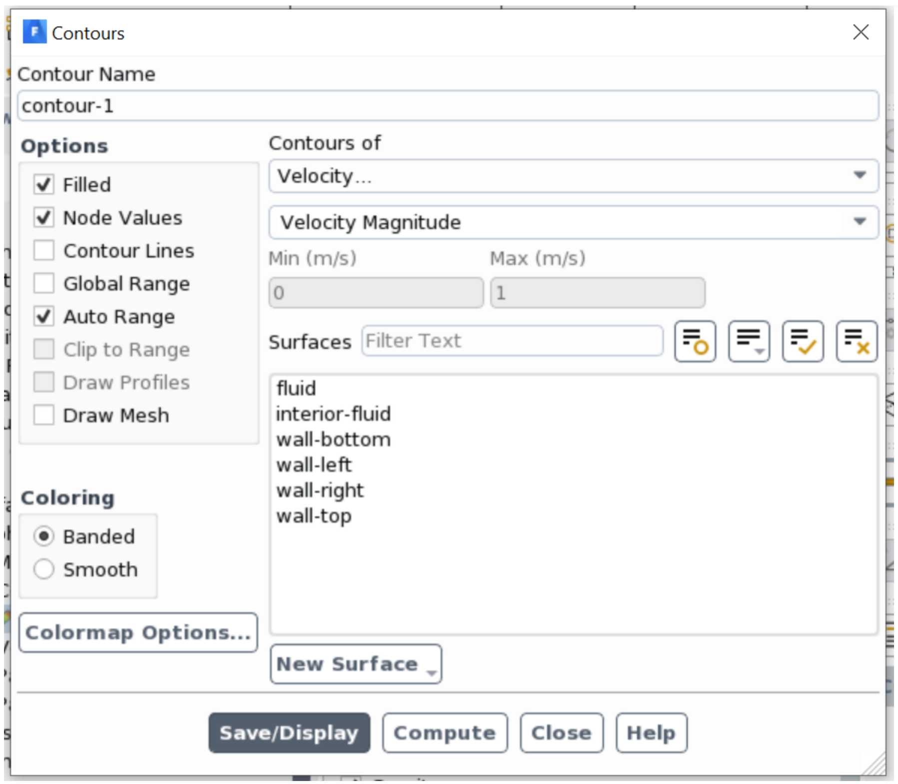
  Set Reporting Interval to 1.

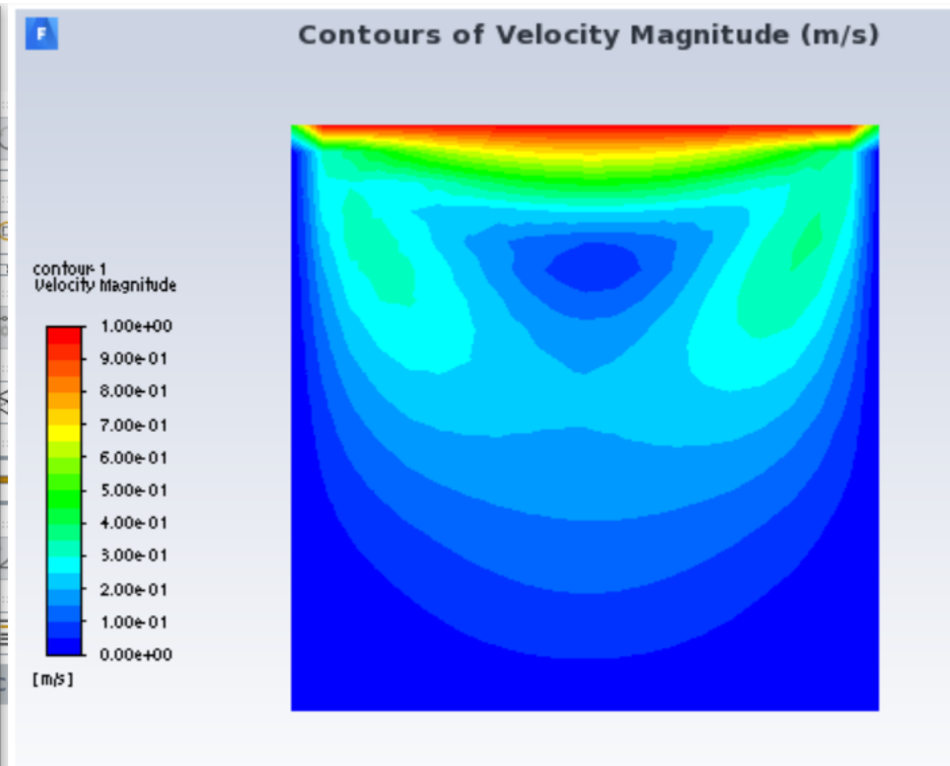
  Click Calculate.

Post-processing

➢ **Contour plot**

In the left panel, double click Contours in Results section



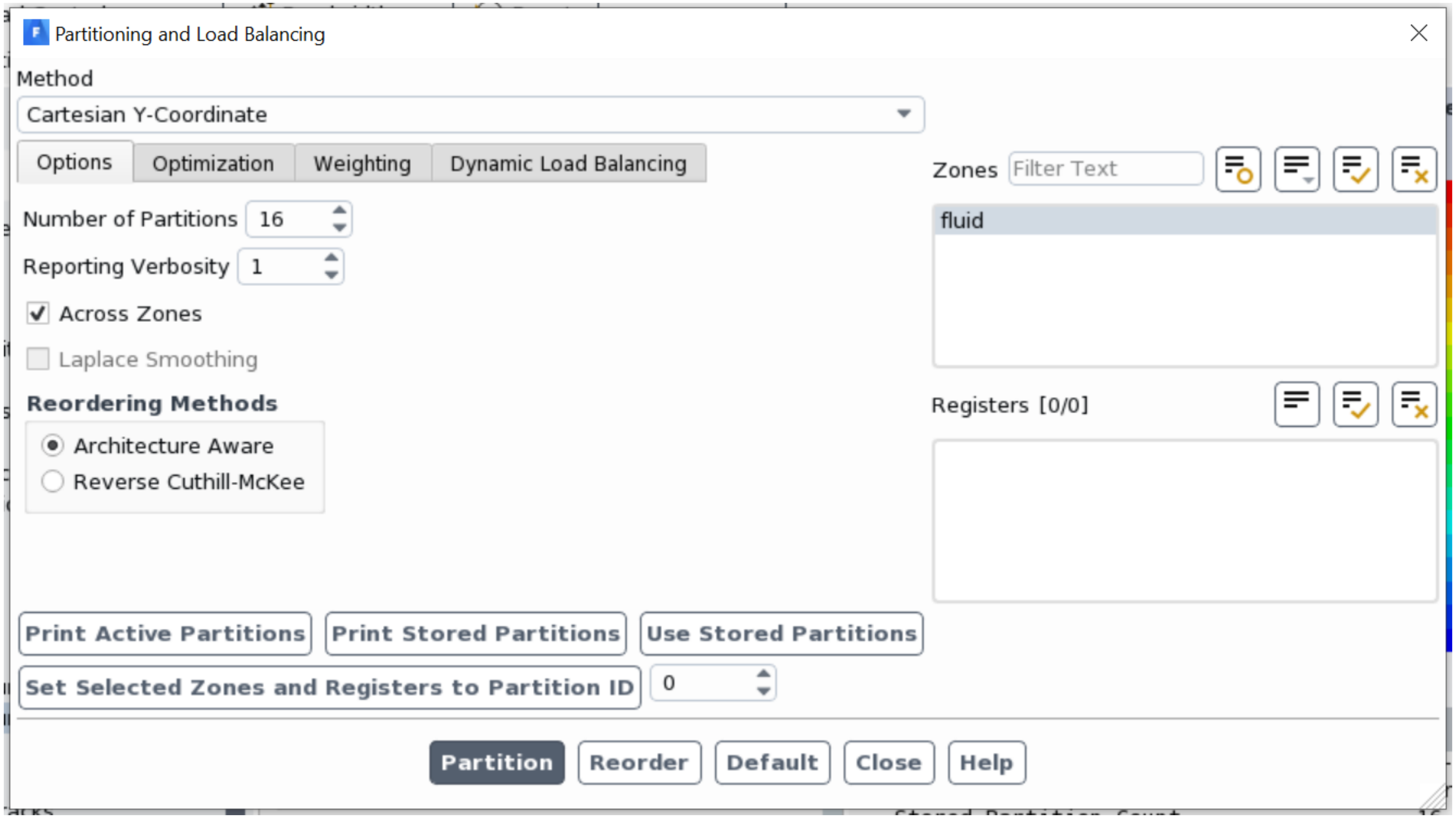


Partition the Existing Mesh

- **Partition the Existing Mesh**

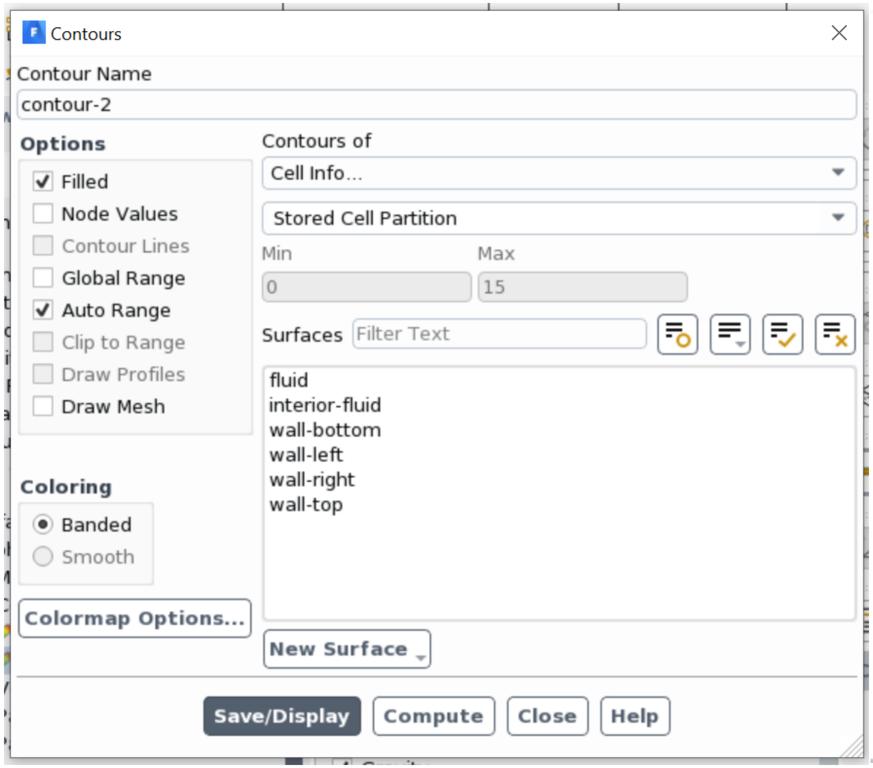
  Select Parallel tab

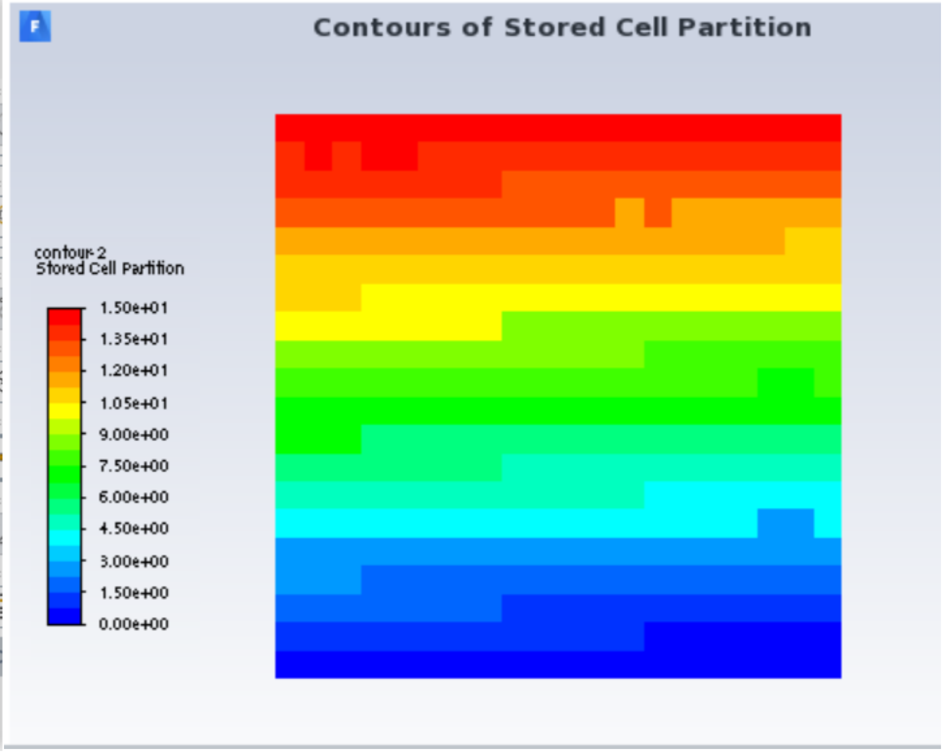
  Select Auto Partition or Partition/Load Balance under General



Graphically Examine the Partitions

- **In the left panel, double click Contours in Results section**





PBS Batch Job Script (1 nodes)

```
#!/bin/bash
#PBS -l nodes=1:ppn=16
#PBS -l walltime=1:00:00
#PBS -A your_allocation_name
#PBS -q workq
#PBS -N par_cavity

cd $PBS_O_WORKDIR
echo "start fluent run"
TIC=`date +%s.%N`
# run on 32 processes on 2 nodes, -cnf=$PBS_NODEFILE is important!!!
fluent 2ddp -g -i cavity.jou -t32 -cnf=$PBS_NODEFILE -pinfiniband -mpi=intel -ssh
TOC=`date +%s.%N`
J1_TIME=`echo "$TOC - $TIC" | bc -l`
echo "end fluent run"
echo "simulation took=$J1_TIME sec"
```

The breakdown of the fluent command

fluent $version -g -i $journal -t$nCPU -p$ic -cnf=$hostfile -mpi=$mpi -ssh

Example:

```
fluent 3ddp -g -i input.jrn -t32 -pib -cnf=$PBS_NODEFILE -mpi=intel –ssh
```

- **fluent is ANSYS FLUENT command.**
- **$version: specifies the version for ANSYS FLUENT.**
  - 2d:2-Dimension; 3d: 3-Dimension
  - dp: double precision; sp: single precision
- **-g: runs without gui or graphics.**
- **-t$nCPU: specifies number of processor for parallel computing.**
- **-p$ic: specify interconnect between nodes; <ic>={default|myri|ib}.**
- **-cnf=$hostfile: specify the hosts file**
- **-i$journal reads the specified journal file.**
- **-mpi=<mpi>: specify MPI implementation; <mpi>={pcmpi | intel | ...}**
- **For a more detailed reference use: fluent -h**

Ansys Journal File: “cavity.jrn”

- **Ansys journal file contains a list of instructions to pass to FLUENT**

```
; Read case file
rc cavity.cas
; Initialize the solution
/solve/init/init
; Calculate 10k iterations
it 10000
; Write data file
wd cavity.dat
; say OK when asking for overwrite previous file
OK
; Exit Fluent
exit
; say yes when confirming
yes
```

Summary

- **Introduction to CFD**
- **Running example case using OpenFOAM**
- **Running example case using ANSYS Fluent**

Next Week Training

- **Weekly trainings during regular semester**
  - Wednesdays “9:00am-11:00am” session, Frey 307 CSC
- **Programming/Parallel Programming workshops**
  - Usually in summer
- **Keep an eye on our webpage: www.hpc.lsu.edu**