

More tutorials

- We will now learn how to use a (small) number of useful utilities and libraries. Some of them are described in the UserGuide and ProgrammersGuide, and some of them have been discussed in the Forum.
- In your home assignment you will be asked to go through all the written tutorials in the UserGuide and ProgrammersGuide, where you will find some written tutorials.
- If you find a utility, solver or library that is not described anywhere, you can include a description of it in your project tutorial. I would prefer if you then use my L^AT_EX slide template, so that I can easily use it in coming courses.
- At the end of this part we will learn how to search for additional tutorials and other information on the use of different applications and libraries.

The mapFields utility

- The mapFields utility maps the results of one case to another case. You will use this utility when you do the tutorials in the UserGuide.

- Usage:

```
mapFields <source dir> [-parallelTarget] [-consistent]
      [-sourceTime scalar] [-parallelSource] [-case dir]
      [-help] [-doc] [-srcDoc]
```

(You get this information by simply typing `mapFields -help`)

- The time used for the mapping is specified by `startFrom/startTime` in the *target* case.
- The flag `-consistent` is used if the geometry and boundary conditions are identical in both cases. This is useful when modifying the mesh density of a case. For non-consistent cases a `mapFieldsDict` dictionary must be edited, see the `icoFoam/cavityClipped` tutorial.
- The flags `-parallelSource` and `-parallelTarget` are used if any, or both, of the cases are decomposed for parallel simulations.

The sample utility

- This is used to produce graphs for publication or to extract surfaces. You will use this utility when you do the tutorials of the UserGuide, and when you do the `ercoftacConicalDiffuser` case-study.
- Usage:

```
sample [-latestTime] [-time ranges] [-parallel] [-constant]  
      [-noZero] [-case dir] [-region name] [-help] [-doc] [-srcDoc]
```
- Copy and modify `sampleDict` in the `solidDisplacementFoam/plateHole` tutorial. In this case a line with 100 sample points is defined along the `leftPatch`, between `(0 0.5 0.25);` and `(0 2 0.25);`.
- Use `axis distance;` for a more general way to set the x-axis abscissa.
- An example of a gnuplot command that uses the output and compares it to an analytical solution:

```
plot [0.5:2] '<datafile>', 1e4*(1+(0.125/(x**2)))+(0.09375/(x**4)))
```
- `sample` can also sample surfaces...

The sample utility - surfaces

- Additions for extracting surface cuts (second example commented):

```
surfaceFormat    vtk;  
  
surfaces  
(  
    outputName  
    {  
        //type            patch;  
        //patchName       movingWall;  
        //triangulate false;  
        type plane;  
        basePoint (0.05 0.05 0.005);  
        normalVector (0 0 1);  
    }  
);
```

- Visualize the surfaces in paraview.

The sample utility - interpolationScheme

- Use dummy entries to see the alternatives for interpolationScheme:

cell

cellPoint

cellPointFace

cellPointWallModified

- See the source code for exact definitions.

The sample utility - formats

- Use dummy entries to see the alternatives for formats:

- `setFormat:`

```
xmgr  
jplot  
gnuplot  
raw
```

- `surfaceFormat:`

```
foamFile  
null  
raw  
vtk
```

The sample utility - types

- Use dummy entries to see the alternatives for types:

- Sets:

uniform
midPointAndFace
face
midPoint
cloud
curve

- Surfaces:

thresholdCellFaces
cuttingPlane
isoSurfaceCell
patch
isoSurface
distanceSurface
plane

The foamCalc utility

- This utility calculates new fields from existing ones. This replaces some of the utilities in previous versions of OpenFOAM, such as magU and Ucomponents.
- Usage:
`foamCalc <calcType> <fieldName1 ... fieldNameN>`
- To get a list of available <calcType>s, write:
`foamCalc xxx` and get the following list:
`randomise, magSqr, magGrad, addSubtract, div, mag, interpolate, components`
- Examples:
`foamCalc div U`
`foamCalc components U`
- The new fields are written in the time directories.

The setFields utility

- The `setFields` utility is used to set values to the fields in specific regions. You will use this when you do the `interFoam/damBreak` tutorial in the UserGuide.

- Usage:

```
setFields [-latestTime] [-time ranges] [-parallel]  
          [-constant] [-noZero] [-case dir] [-help] [-doc] [-srcDoc]
```

- A `setFieldsDict` dictionary is used. Find an example in the `interFoam/damBreak` tutorial: `/system/setFieldsDict`.
 - The `defaultFieldValues` sets the default values of the fields.
 - A `boxToCell` bounding box is used to define a set of cells where the `fieldValues` should be different than the `defaultFieldValues`.

The funkySetFields and groovyBC utilities

- The `funkySetFields` utility is a user contributed utility available in the OpenFOAM Wiki:
<http://openfoamwiki.net>
- `funkySetFields` is a development of the `setFields` utility, and it includes the option of specifying mathematical expressions etc.
- The `groovyBC` utility is similar, but for boundaries.

The foamToVTK, checkMesh, and flattenMesh utilities

- The foamToVTK utility can be used in many different ways. Example:
- The two empty sides of a 2D mesh must have the same mesh distribution. Add 0.0001 to the z-position of one of the constant/polyMesh/points of the cavity case.
- The checkMesh utility can be used to verify this. If not, it will complain:

```
***Number of edges not aligned with or perpendicular to non-empty directions: ????  
Writing ???? points on non-aligned edges to set nonAlignedEdges
```

- The point labels are written to constant/polyMesh/sets/nonAlignedEdges
- Take the opportunity to visualize the point set in paraFoam: First open the cavity case in paraFoam, then use File/Open <case>.OpenFOAM to read in the same case again. This time mark Include Sets, mark *only* Mesh Parts/NonAlignedEdges, and visualize using box glyphs.
- Another way to view the problematic points in *paraview* (not paraFoam):

```
foamToVTK -pointSet nonAlignedEdges
```

The result appears in the VTK directory.
- The flattenMesh utility can sometimes fix the problem, like in this case.

The extrudeMesh utility

- A mesh can be extruded in a third direction, yielding a new mesh with a number of layers that are similar. This is useful when making a 2D mesh.

- A dictionary is needed to be copied to constant:

```
cp $FOAM_UTILITIES/mesh/generation/extrudeMesh/extrudeProperties constant
```

- See www.openfoamwiki.net for more details.

The transformPoints utility

- Moves, rotates and scales the mesh.

- Usage:

```
transformPoints [-translate vector] [-yawPitchRoll (yaw pitch roll)]  
  [-rotateFields] [-parallel] [-rotate (vector vector)]  
  [-rollPitchYaw (roll pitch yaw)] [-scale vector] [-case dir]  
  [-help] [-doc] [-srcDoc]
```

- Example:

```
run  
cp -r cavity cavityMoved  
transformPoints -case cavityMoved -translate "(0.1 0 0)"
```

- Have a look in paraFoam:

```
run  
touch cavityMoved/cavityMoved.OpenFOAM  
paraFoam -case cavity
```

Click Accept and then use File/Open to open the cavityMoved case at the same time.

The mergeMeshes utility

- Takes the meshes from two different cases and merges them into the master case.
- mergeMeshes reads the `system/controlDict` of both cases and uses the `startTime`, so be careful if you have a moving mesh for example. The first case that you specify will be the master, and a new time (`startTime+deltaT`) will be written in which a new polyMesh is located. Move it to the correct position (`constant/polyMesh`), and you have a case with the merged mesh.

- Example (start from clean cases):

```
run
```

```
cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity cavityMerged
```

```
cp -r $FOAM_TUTORIALS/incompressible/icoFoam/cavity cavityTransformed
```

```
blockMesh -case cavityMerged
```

```
blockMesh -case cavityTransformed
```

```
transformPoints -case cavityTransformed -translate "(0.1 0 0)"
```

```
mergeMeshes . cavityMerged . cavityTransformed
```

```
mv cavityMerged/0.005/polyMesh/* cavityMerged/constant/polyMesh
```

- Note that the two meshes will keep all their original boundary conditions, so they are not automatically coupled. To couple the meshes, use `stitchMesh`...

The stitchMesh utility

- Couples two uncoupled parts of the mesh that belong to the same case.
- You should have a patch in one part of the mesh (`masterPatch`) that fits with a corresponding patch on the other part of the mesh (`slavePatch`). If you have that, then the command is:

```
stitchMesh masterPatch slavePatch
```
- `masterPatch` and `slavePatch` are important, as the face and cell numbers will be renamed after the `masterPatch`.
- After `stitchMesh`, `masterPatch` and `slavePatch` are still present in the new `polymesh/boundary`, but they are empty so just delete them. The same thing can be done as well for the boundary conditions in the `0` folder.
- We have to re-organize the patches for this to work with our `cavityMerged` case.
- See www.openfoamwiki.net for more details.

The decomposePar utility

- `decomposePar` makes a domain decomposition for parallel computations. This is described in the UserGuide.
- Usage:

```
decomposePar [-fields] [-force] [-copyUniform] [-cellDist]
              [-filterPatches] [-ifRequired] [-case dir] [-region name]
              [-help] [-doc] [-srcDoc]
```
- A `decomposeParDict` specifies how the grid should be decomposed. An example can be found in the `interFoam/damBreak` tutorial: `system/decomposeParDict`.
- There are some different alternatives for which method to use for the decomposition. See the UserGuide. `numberOfSubdomains` specifies the number of subdomains the grid should be decomposed into. Make sure that you specify the same number of subdomains in the specific decomposition method you will use, otherwise your simulation might not run optimal.
- We will discuss parallel processing later on.

The reconstructPar utility

- `reconstructPar` is the reverse of `decomposePar`, reassembling the grid and the results.
- Usage:

```
reconstructPar [-zeroTime] [-fields "(list of fields)"]  
               [-latestTime] [-time ranges] [-constant] [-noZero]  
               [-noLagrangian] [-case dir] [-region name]  
               [-help] [-doc] [-srcDoc]
```
- This is usually done for post-processing, although it is also possible to post-process each domain separately by treating an individual processor directory as a separate case when starting `paraFoam`.
- We will discuss parallel processing later on.

functionObjects

- functionObjects are general libraries that can be attached run-time to any solver, without having to re-compile the solver.
- An example can be found in the `incompressible/pisoFoam/les/pitzDaily` tutorial.
- A functionObject is added to a solver by adding a `functions` entry in `system/controlDict`
- You can find functionObjects in the source code, in the OpenFOAM Wiki (www.openfoamwiki.net), and in the OpenFOAM-extend project (www.sourceforge.net).
- Search the tutorials for examples using:

```
grep -r functionObjectLibs $FOAM_TUTORIALS
```
- The implementations can be found in:

```
$FOAM_SRC/postProcessing/functionObjects
```

The probes and fieldAverage functionObjects

- The probes functionObject probes the development of the results during a simulation, writing to a file in the directory probes.
- The fieldAverage functionObject calculates the time-average of specified fields and writes the results in the time directories.
- Copy and modify the functions part at the end of the controlDict of the incompressible/pisoFoam/les/pitzDaily tutorial to your case and run it.
- Visualize the output file of sample in Gnuplot:

```
plot "probes/0/p" using 1:2, "probes/0/p" using 1:3
```
- Visualize the output from fieldAverage in paraFoam.

The surfaces functionObjects

The surfaces functionObject writes out surface interpolated results to disk. If the surfaceFormat is VTK, those can be viewed in paraview. Two examples (see the commented lines for the second one):

```
functions(  
  pressure{  
    type surfaces;  
    functionObjectLibs ("libsampling.so");  
    outputControl timeStep;  
    outputInterval 1;  
    surfaceFormat vtk;  
    fields ( p );  
    surfaces  
    (  
      walls  
      {  
        //type          patch;  
        //patchName     movingWall;  
        //triangulate   false;  
        type           plane;  
        normalVector (0 0 1);  
        basePoint (0 0 0.005);  
      } ); } );
```

The forceCoeffs functionObject

- The forceCoeffs functionObject prints out the lift and drag coefficients.
- See the `sonicFoam/ras` tutorial: `system/controlDict`
- Note that you must specify realistic reference values!
- See the source code for the details!

How to search for other tutorials

- Type `tut` to go to the `$FOAM_TUTORIALS` directory. Here you find many case-setups for the solvers in OpenFOAM. Unfortunately, they are not well-described. Describing these tutorials may be part of your project.
- All the solver tutorials have `Allrun` scripts that describe the use of those tutorials. We will now have a look at the `Allrun` script of the `$FOAM_TUTORIALS/incompressible/icoFoam` tutorials. This is actually what you will do manually when you do the `cavity` tutorials in the `UserGuide`. In other words, you can use the `Allrun` script as a short summary of the description in the `UserGuide`.

Run the icoFoam cavity tutorials using the Allrun script (1/5)

(Note that I did not have time to make sure that the description below is without any error after conversion to 1.6.x. The principle is however correct.)

In the icoFoam tutorial directory there is an Allrun script.

When running this script it is preferred to copy the entire
\$FOAM_TUTORIALS/incompressible/icoFoam directory to your run directory.

If you run the Allrun script for the icoFoam cavity tutorials you actually
first run the cavity case

```
#Running blockMesh on cavity:  
blockMesh  
#Running icoFoam on cavity:  
icoFoam
```

Run the icoFoam cavity tutorials using the Allrun script (2/5)

If you run the Allrun script for the icoFoam cavity tutorials you actually **then run the cavityFine case:**

```
#Cloning cavityFine case from cavity:
mkdir cavityFine
cp -r cavity/{0,system,constant} cavityFine
    [change "20 20 1" in blockMeshDict to "41 41 1"]
    [set startTime in controlDict to 0.5]
    [set endTime in controlDict to 0.7]
    [set deltaT in controlDict to 0.0025]
    [set writeControl in controlDict to runTime]
    [set writeInterval in controlDict to 0.1]
#Running blockMesh on cavityFine
blockMesh
#Running mapFields from cavity to cavityFine
mapFields -case cavity -sourceTime latestTime -consistent
#Running icoFoam on cavityFine
icoFoam
```


Run the icoFoam cavity tutorials using the Allrun script (3/5)

If you run the Allrun script for the icoFoam cavity tutorials you actually **then run the cavityGrade case:**

```
#Running blockMesh on cavityGrade
blockMesh
#Running mapFields from cavityFine to cavityGrade
mapFields -case cavityFine -sourceTime latestTime -consistent
#Running icoFoam on cavityGrade
icoFoam
```

Run the icoFoam cavity tutorials using the Allrun script (4/5)

If you run the Allrun script for the icoFoam cavity tutorials you actually **then run the cavityHighRe case:**

```
#Cloning cavityHighRe case from cavity
mkdir cavityHighRe
cp -r cavity/{0,system,constant} cavityHighRe
#Setting cavityHighRe to generate a secondary vortex
    [set startFrom in controlDict to latestTime;]
    [set endTime in controlDict to 2.0;]
    [change 0.01 in transportProperties to 0.001]
#Copying cavity/0* directory to cavityHighRe
cp -r cavity/0* cavityHighRe
#Running blockMesh on cavityHighRe
blockMesh
#Running icoFoam on cavityHighRe
icoFoam
```

Run the icoFoam cavity tutorials using the Allrun script (5/5)

If you run the Allrun script for the icoFoam cavity tutorials you actually **then run the cavityClipped case:**

```
#Running blockMesh on cavityClipped
blockMesh
#Running mapFields from cavity to cavityClipped
cp -r cavityClipped/0 cavityClipped/0.5
mapFields -case cavity -sourceTime latestTime
    [Reset the boundary condition for fixedWalls to:]
    [      type          fixedValue;                ]
    [      value         uniform (0 0 0);          ]
    [      We do this since the fixedWalls got     ]
    [      interpolated values by cutting the domain ]
#Running icoFoam on cavityClipped
icoFoam
```

Run all the tutorials using the Allrun scripts

- You can run a similar script, located in the tutorials directory, and also named `Allrun`. This script will run through all the tutorials (calls `Allrun` in each solver directory).
- You can use this script as a tutorial of how to generate the meshes, how to run the solvers, how to clone cases, how to map the results between different cases etc.

Finding tutorials for the utilities in OpenFOAM

- There are no tutorials for the utilities, but we can search for examples:

```
find $WM_PROJECT_DIR -name \*Dict | grep -v blockMeshDict | \  
    grep -v controlDict
```

You will get a list of example dictionaries for some of the utilities.

Now you should be ready to go on exploring the applications by yourself.

More tutorials can be found in

- The UserGuide
- The Programmer's guide, chapter 3
- The OpenFOAM Wiki
(in particular the Turbomachinery Working Group)
- The OpenFOAM Forum