

Post-processing in paraFoam (ParaView-5.0.1)

- The installation instructions are for the native/built-in reader distributed with Ubuntu (not 5.0.1). Differences will be pointed out.
- Note that in some cases there may be a window popping up when starting paraFoam, containing warnings. It is in most cases just to close that window. If it appears again, just move it out of sight.

Post-processing in paraFoam

- paraFoam is the main post-processor distributed with OpenFOAM. As previously mentioned, you can also use other alternatives. However, paraFoam is quite powerful and competitive with commercial alternatives, and it is based on OpenSource software.
- paraFoam is a wrapper for the third-party product Paraview (www.paraview.org). It basically generates a file named `<case>.OpenFOAM` (native reader: `<case>.foam`), which is necessary since Paraview needs a file to be specified. paraFoam also makes sure that the OpenFOAM reader is used, and that the paraview GUI is modified according to the specifications for OpenFOAM.
- Paraview is based on VTK, the visualization toolkit (www.vtk.org)
- Results from OpenFOAM can be converted to the VTK format using `foamToVTK`. In some cases, like for lagrangian particle tracking, this may in fact be necessary.
- References: The ParaView Guide, ISBN 1-930934-21-1, VTK User's Guide, ISBN 1-930934-13-0, The Visualization Toolkit, ISBN 1-930934-12-2, Programming Python, ISBN 0-596-00085-5.

Post-processing the icoFoam/cavity case in paraFoam

In the following, we will use paraview and the OpenFOAM reader compiled with OpenFOAM. It looks slightly different if you use the OpenFOAM reader that comes with Paraview.





- We will now do some post-processing of the `icoFoam/cavity` case, with the purpose of learning the basics of Paraview.
- Go to the `icoFoam/cavity` case that we have already run:
`cd $FOAM_RUN/cavity`
- Type `paraFoam&` to post-process that case.
- The case name will appear as the object `cavity.OpenFOAM` in the Pipeline Browser.
- If you do an `ls -l` in the case directory you will see that a temporary empty file named `cavity.OpenFOAM` has been generated.
- Click `Apply` to read in the data.
- The object is displayed in the visualization window, which will be discussed in the coming slides

Visualization window

- Rotate by holding the left mouse key, and moving the mouse. It is like rolling a ball.
- Zoom by holding the right mouse key, and moving the mouse up and down.
- Move by holding the center mouse key, and moving the mouse.

Visualization window

- At the top left of the visualization window, you can see a number of icons, e.g.:

-  : Undo last visualization transformation.
-  : Redo last undone visualization transformation.
-  : Toggle 3D/2D visual transformation
-  : Adjust camera (use standard views, or adjust the views).

Visualization window

- At the top right of the visualization window, you can see four icons (in fact, perhaps also a 5th one):



- No1: Split Horizontal.
- No2: Split Vertical.
- No3: Maximize (only useful if there are multiple visualization windows). If you click on this you will see the fifth icon, which restores the view.
- No4: Close (only useful if there are multiple visualization windows).

Properties tab

- It is the Properties tab that differs between the native/built-in reader and the one which is distributed with OpenFOAM.
- **HIGHLY** recommended:
Click on the gearwheel icon next to the search bar at the top of the Properties tab. This will show useful advanced settings in the Properties tab. Some of the following descriptions require this.

Properties tab

- In `Mesh Parts`, mark the parts of the mesh that should be imported. It can be the `internalMesh`, groups of patches, or individual patches.
- In `Volume Fields`, mark the volume fields that should be imported.
- Click on `Patch Names` if you want to view them. Then you also need to tick the patches in the `Mesh Parts`.
- Click on `Include Sets` and/or `Include Zones` if you have sets or zones that you want to visualize. This is very useful in order to check if your sets and zones have been generated properly. However, in the `cavity` case, we have no sets or zones.
- Click `Apply` to read in the mesh and the volume fields.
- If the contents of your case directory has changed, click on `Refresh Times` to import the new information. Click on `Skip Zero Time` tick-box if the zero directory is incomplete (from a `paraFoam` point of view - not from an `OpenFOAM` point of view).

Properties tab

- Color by a Solid Color, or using a variable. For cell based representation (box), the variables are assumed to constant in each cell, and for point based representation (point), the variables are interpolated to the points, yielding a smooth field. For vector variables you can choose Magnitude, X, Y, and Z.
- You can Toggle Color Legend Visibility, Edit Color Map, and Rescale to Data Range.
- In Edit Colormap, you change the color scale limits, change the number of colors, change the Color Space, and change the text in the legend.
- The color legend can be moved and re-sized in the visualization window.

Properties tab

- Under `Cube Axes`, tick the box of `Show Axes` to show the physical extent of the part. Good to verify the scale of the geometry, to make sure that it is in [m]!!! Click on `Edit` to edit the appearance of the axes.
- The representation of the object can be as outline, wireframe, surface etc.
- The `Transforming` functionality can be used to translate, scale and rotate the object, which is useful when comparing results from different simulations of the same case, or when viewing the same case in different angles. The `Orientation` is the rotation in degrees about each axis, from the chosen `Origin` - note that vector variables will correspond to the original orientation.

Information tab

- Here you get some information on the mesh and the fields.

Filters

- The `Filters/Alphabetical` at the top menu lets you manage your data in different ways. Some of those can be selected by the icons:






- We will have a look at some of them in the next slides...

Calculator Filter





- With the Calculator filter you can create new fields from existing scalar and vector fields, using simple mathematical functions.
- The new fields can be used just like the original fields.
- Create a new scalar field that is only the X-component of the velocity by clicking on Scalars and choosing U_X. Specify Result Array Name as for instance U_X, and click Apply. A Calculator1 object appears in the Pipeline Browser, and the geometry is colored by U_X. The new object is based on the object that was selected when generating calculator1.
- Use the 'eye' icon in front of the objects to show/hide each object.
- Delete the calculator1 object by clicking on it in the PipelineBrowser and then clicking on delete in Object Inspector / Properties.



Contour, Clip and Slice Filters

- The Contour Filter  lets you make a contour plot or iso-surface, if the parent object is 2D or 3D, respectively. A value range can be specified for several values at the same time. Make sure not to be in the 0 time!
- The Clip Filter  lets you clip your results using plane/box/sphere/s-calar.
- The Slice Filter  lets you make plane/box/sphere cuts through your results.
 - It is possible to specify the plane in different ways.
 - Multiple cuts can be made simultaneously using Slice Offset Values.
 - Each Slice will generate a new object in the Pipeline Browser, which can be colored etc. like any other object.

Threshold, Glyph and Cell Center Filters

- The Threshold Filter  lets you show cells with solution scalar values in a specific range.
- The Glyph Filter  lets you make vector plots. It can also be used to visualize lagrangian particles.
 - Glyphs are by default generated at the grid points. If you want the vectors to be located in the center of the cells, you first need to make a new object of the cell centers using the Cell Centers filter that can be found in `Filters/Alphabetical`.
 - Mark the cell centered object, and filter it with the default values of the Glyph Filter.
 - It is possible to modify the representation of the vectors in the `Properties` tab.

Stream Tracer and Warp Filters

- The Stream Tracer filter  lets you make streamline tracers, submitted from a point, a spherical cloud, or line sources.
- The Warp By Vector filter  lets you distort your geometry according to the local value of a vector. This is mostly useful in solid mechanics. There is also a Warp By Scalar filter available in `Filters / Alphabetical`, where you can find many additional filters...

Plot Over Line Filter

- In `Filters / Alphabetical`, you find a long list of filters. (Note that a filter that is available in any of the other sub-menus can't be chosen under `Alphabetical` for some silly reason. If you have used a filter it may perhaps only be chosen under `Recent!!!`).
- The `Plot Over Line` filter lets you make detailed plots of the variables along a line.
- In the `Properties` tab, you can choose which variables to plot, and choose line colors and line styles.
- Later on we will learn how to extract line data using the OpenFOAM sample utility, and plot it in OpenSource Gnuplot, www.gnuplot.info, (or any other plotter).

Hint: You can also press `CTRL+SPACE` and search for a filter (if you know the start of its name)

Selection and Extraction

- If you want to visualize only a part of the domain you can select that part using the selection icons:



(e.g. the Select Cells Through - can also be chosen by pressing 'f')

Then you use the Extract Selection filter to get a new object to play with in the Pipeline Browser.

- Click on the dust bin to clear the current selection.
- Press the Esc key to escape a selected selection icon.

Image output

- `File/Save Screenshot` saves pixel images in a number of formats: png, bmp, tiff, ppm, jpg, pdf
- The format is chosen by the extension of the filename.
- Increase the resolution for high quality images.
- Use `ls -l` to see the difference in file size for the different formats.
- Use `display` to view, or `gimp` to view and edit all the image formats.
- Convert to eps format for inclusion in L^AT_EX documents using:
`/usr/bin/convert test.jpg test.eps`
(we specify the whole path to make sure that we use the correct one)
- View the eps file using `gv`.

Animation output

- File/Save Animation saves frame images in a number of formats: ogv, jpg, tiff, png
- Use `/usr/bin/convert` to convert the frame images into a movie:
`/usr/bin/convert -quality 90% frames*.png movie.gif`
- View the movie using `firefox file://$PWD/movie.gif`
- Increase the resolution for high quality animations.
- In View you can toggle Animation View, which lets you control your animation. You can for instance specify which time steps to use, and how to sweep the camera during the animation (try Camera and Orbit with default settings, and click on the plus-icon). Some things work better without Parallel Projection.

Save and Load states

- The current state, including created objects and current view, can be saved for later loading, in `File / Save State` and `Load State`. This is useful if the same visualization needs to be done many times.
- Saved states can be loaded directly from `paraview`, so there is no need to run `paraFoam`.
- State files are in ASCII, so it is possible to generate or modify them externally if needed.
- However, they are not fail-safe. E.g. they use full path to files. They need the same time steps. ...

Multiple cases

- Multiple OpenFOAM cases can be read into paraFoam.
- For the additional cases you must generate the <case>.OpenFOAM file by hand:

```
touch $FOAM_RUN/<case>/<case>.OpenFOAM
```
- Launch paraFoam with the first case as usual.
- Read in the additional cases in paraFoam using File/Open
- This procedure can also be used to view cases that are decomposed for parallel simulation, in order to see the domain decomposition. Then you generate a processor*.OpenFOAM file in each processor* directory. We will discuss parallel simulations later.
- This is also the procedure for multi-region cases, like conjugate heat transfer and fluid-structure interaction.

View a blockMeshDict dictionary

This is only available with the reader distributed with OpenFOAM, i.e. not the native/built-in reader.

- paraFoam can be used to view a blockMeshDict dictionary:

```
paraFoam -block
```

- Note that it will delete everything in constant/polyMesh except the blockMeshDict, so you will have to run blockMesh again!