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`, 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 is in fact 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

- 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.
- The object can be controlled using the `Object Inspector`, which we will discuss in the coming slides...

paraFoam, Properties tag

- In the `Region Status`, check what parts of the mesh should be imported. Note that there is a problem reading patches defined as `empty`, so don't check them.
- In the `Vol Field Status`, check what volume fields should be imported.
- Click on `Show Patch Names` if you want to view them.
- Click `Apply` to read in the mesh and the volume fields.
- If the contents of your case directory has changed, click on `Update GUI` and `Apply` to import the new information.

paraFoam, Display tag

- Control visibility (can also be done by clicking on the 'eye' in the `Pipeline Browser`).
- Color the object by a constant color, or using a variable. For cell based representation, the variables are assumed to constant in each cell, and for point based representation, the variables are interpolated to the points, yielding a smooth field. The colormap can be edited, and the color range can be re-scaled. In `Edit Colormap` (and in the `view` menu), a `Color Legend` can be added. The color legend can be moved and re-scaled with the mouse.
- The representation of the object can be as outline, wireframe, surface etc.
- The cartesian extent of the object is shown by clicking on `Show cube axes`. This representation can be edited.
- The object can be translated, which is useful when comparing results from different simulations of the same case, or when viewing the same case in different angles.

paraFoam, Information tag

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

paraFoam, 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.

paraFoam, Visualization window

- At the top left of the visualization window, you can see five icons:
  - No1: Undo last visualization transformation.
  - No2: Redo last undone visualization transformation.
  - No3: Edit View Options (background color, parallel projection, lights, toggle axes - if 'interactive', you can move them with the mouse).
  - No4: Lookmark (if you want to save a view).
  - No5: Adjust Camera (use standard views, or adjust the views).

paraFoam, Visualization window

- At the top right of the visualization window, you can see four icons:
  - No1: Split Horizontal.
  - No2: Split Vertical.
  - No3: Maximize (only useful if there are multiple visualization windows).
  - No4: Close (only useful if there are multiple visualization windows).

paraFoam, Filters

- The `Filters/Alphabetical` menu lets you manage your data in different ways. It also explains some of the icons.
- We will have a look at some of them in the next slides...

paraFoam, Slice Filter

- The `Slice Filter` lets you make a plane cut through your results.
- It is possible to specify the plane in different ways.
- Mutliple 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.
- Unfortunately, only a plane cut is available.

paraFoam, Contour Filter

- The `Contour Filter` lets you make a contour plot or iso-surface, if the parent object is 2D or 3D, respectively.

paraFoam, Glyph and Cell Center Filters

- 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.
- 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` tag.

paraFoam, Stream Tracer and Tube Filters

- The `Stream Tracer` filter lets you make streamline tracers, submitted from a point, a spherical cloud, or line sources.
- The `Tube` filter makes it possible to give the streamlines a nice tubular appearance.

paraFoam, Warp (scalar) Filter

- The `Warp (scalar)` filter lets you distort your geometry according to the local value of a scalar. This is mostly useful in solid mechanics.

paraFoam, Plot Over Line Filter

- The `Plot Over Line` filter lets you make detailed plots of the variables along a line.
- In the `Display` tab, you can choose which variables to plot, and choose line colors and line styles.
- Right-click on the graph and click on `Properties` to edit the appearance of the graph.
- 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).

paraFoam, 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.

paraFoam, 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 `gqview` to view all the images except the `pdf` image, which you can view using `xpdf` or `acroread`.
- Convert to `eps` format for inclusion in LaTeX documents using:
  `/usr/bin/convert test.jpg test.eps`
  (There is another `convert` in the default path that overrides this `convert`, so we have to specify the whole path)
- View the `eps` file using `ggv`.

paraFoam, Animation output

- Here I describe the basic procedure. Unfortunately there are a couple of error messages while doing this, and I did not have the time to resolve them. Help me resolve the problem, and have a better chance of a higher grade.

- `File/Save Animation` saves frame images in a number of formats: jpg, tiff, png

- Use `/usr/bin/convert` to convert the frame images into a movie:
  `/usr/bin/convert -quality 90% animation*.jpg movie.mpeg`

- View the movie using `mplayer movie.mpeg -loop 0`
  (Since we were not able to convert the movie to `mpeg` format, you can at least use `gqview` to view each frame)

- Increase the resolution for high quality animations.

paraFoam, Save and Load states

- The current state, including created objects and current view, can be saved for later loading. 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.

paraFoam, 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.