Part 1. Basic Operations in ParaView

TASK 0 – GETTING FAMILIAR WITH PARAVIEW
Refer to the getting started.pdf for a quick introduction to Paraview’s user interface.

TASK 1 – SURFACE RENDERING & VISUALIZATION (30 MIN)

In this task, we will visualize and analyze the output of a simulation for the flow of air around a heated and spinning disk. Once you load the model you will see a geometry of a cylinder with a hollowed out portion in one end. The hollow area in the middle is where the heated disk was and the mesh you are seeing is the air around the disk (cylinder shape being the boundary of the simulation).

Step 1. Data Loading.
Load the disk model disk.ex2. This model has approximately 8500 points & 7500 cells. Since it stores the output from a simulation, it has many attributes of which the more prominent ones are pressure, temperature, velocity each with different scales. Try to look up the information for more information.
Try to get a feel of the data by visualization pressure, temperature & velocity in different modes (surface, surface with edges, wireframe).

Step 2. Creating a visualization Pipeline
Next we will create a small meaningful visualization pipeline using filters. Filters are designed to retrieve quantitative values from the data. The filter compute data on mesh, extract elements from the mesh, or plot data. Before proceeding with this step, make sure that the disk object is selected in the pipeline browser and you are viewing it as a wireframe.

(1). Add a contour filter and in its properties change the ‘Contour by’ to ‘Temperature’ and the value range to 400. This would create an isosurface with a value 400 Kelvin.

Rather than showing the mesh surface in the wireframe, which often interferes with the view of what is inside, we will replace it with a cutaway surface. For this we will create an intermediate layer ‘Extract surface’ filter and then a ‘clipping’ filter.
(2). Add a ‘Extract surface’ filter (Filter->Alphabet->Extract Surface) and apply the default properties. Make sure the original model (disk.ex2) is selected when you are adding this new filter.

(3). Select this new filter and add a ‘Clip’ Filter (Filter->Alphabet->Clip). In its properties panel try playing around with different cutting planes. Once you are done apply those changes.

(4). Next we will analyze the velocity field generated by the movement of the air over the heated rotating disk. Select the original model (disk.ex2) and add a ‘Stream Tracer’ filter (Filter->Alphabet->Stream Tracer) and apply the default properties. Hide all the other filters except this Stream Tracer. These lines represent the flow through the volume. Notice the spinning motion around the center line of the cylinder. There is also a vertical motion in the center and near the edges.
(5). Select the stream tracer and add a ‘tube’ filter (Filter->Alphabet->Tube) and accept its default properties. You can now see the streamlines more clearly. As you visualize the streamlines from the side, you should be able to see the circular convection as air heats, rises, cools and falls.

(6). Next, select the Stream Tracer and add ‘Glyph’ filter (Filter->Alphabet->Glyph). Go in the properties and change Glyph Type to Cone, Vectors to V, Scale Mode to Vector and press the refresh button (this will rescale the values). Once done apply these changes. Also change the color of the glyphs with Temp attribute.

Now the streamlines have pointers, which point in the direction of velocity and their size is proportional to the magnitude of the velocity. Notice also the hierarchy in the pipeline browser. These days a lot of visualization software follow a similar approach. You could try adding more filters for different kinds of visualizations if you have time.

With such a rich visualization, we should now be able to answer questions like:

1. Where is the air moving the fastest? Near the disk or away from it? At the center of the disk or near its edges?
2. Which way is the plate spinning?
3. At the surface of the disk, is the air moving towards the center of away from it?
Part 2. Volume Visualization in ParaView

TASK 1 – VISUALIZATION OF CT DATA WITH MPR (MULTI-PLANAR REFORMATTING) (20 MINS)

In this task, you will visualize a CT-scan of the abdominal region of a human. You will also visualize a pre-segmented “mask” that shows where in the volume the liver is located. You will use ParaView to compare the mask with the original volume. Such visualization can be, for example, be used by a physician to verify that the mask, which might have been generated with an automatic segmentation algorithm, is correct.

Step 1. Data Loading.
The CT scan file is ctscan.vtk. The data is stored as signed 16-bit data (short) representing Hounsfield units. The segmented liver is liver.vtk. It is represented as a binary 8-bit (unsigned char) volume where the liver voxels have the value 255 and the background voxels have the value 0.

After loading data, check your information panel, the two data sets should be shown as below:
Step 2. Create Slices From ctscan.vtk data
Create three slices using “Filters->Common->Slice”, and make them to be “X Normal”, “Y Normal”, “Z Normal”, respectively.

Note: Before creating slice, please make sure the object, from which you want to create slice, is highlighted in the “Pipeline Browser”, otherwise you may get undesirable result.

Step 3. Create a histogram from the ctscan.vtk data, and adjust
(1) Click the ‘ctscan.vtk’ node in ‘Pipeline Browser’ make it highlighted, and in Menus, choose ‘Filters->Alphabetical->histogram’ to create a histogram.

(2) Open a new tab in rendering window, and choose ‘Histogram View’.
(3) Click the ‘histogram1’ in ‘Pipeline Browser’ to make it highlighted. And in ‘Properties’ panel, adjust ‘Bin Count’ to 100. And click ‘Apply’ button.

(4) Analyze in what range most of data lies, and go back to 3D rendering view, and adjust scale to a range which you think is better than default one. See if the volume visualization is getting better.
Task 2 – Color and Opacity Map for Volume Rendering (15 Mins)

In this task, we will reuse the ctscan.vtk models in task 1, and explore to adjust the color and opacity map to have a better understanding of different parts of the model, for example:

When you have found the location of liver in the ctscan.vtk, please show and hide liver.vtk to see if the liver.vtk is a good segmentation or not.

Hint: It usually makes things easier if you scale the color map to the range which you are interested in first, and then adjust the color transfer function in a finer degree.
Opacity transfer function editor

Control points

Color transfer function editor

Data value for selected control pt

Label showing values at selected control pt

Mapping Data

Data: 185.8

- Use log scale when mapping data to colors
- Enable opacity mapping for surfaces
- Automatically rescale color map range to fit data

[Image of a color map editor interface]
TASK 3. PYTHON PROGRAMMING FOR VOLUME RENDERING (20 MINS)

In many cases, for example, when there is a large set of data which need to be visualized, the visualization pipeline is desired to be automated. ParaView provides Python Shell for such kind of work.

Step 1. Starting the Python Interpreter

Start “Python Shell” by clicking “Tools->Python Shell”, and a window will pop up as below:

Step 2. Use the code below to load data and visualize it as a volume.

```python
# Read the model file, replace <path> to the folder where your file is located in.
>>> brain = OpenDataFile('<path>/brain.vtk')

>>> Show()

>>> Render()

# Set the representation to ‘Volume’
>>> dp_brain = GetDisplayProperties(brain)

>>> dp_brain.Representation=’Volume’

# Use the brain’s data value as LookupTable for color transfer function
>>> brain_data = brain.PointData['brain']

>>> dp_brain.LookupTable = MakeBlueToRedLT(brain_data.GetRange()[0], brain_data.GetRange()[1])

>>> dp_brain.ColorArrayName = ‘brain’

>>> Render()
```
Step 3. (Optional) Create slice.

```python
>>> Hide(brain) # hide the volume
>>> slice = Slice(brain)
>>> Show(slice)
>>> Render()
```

Step 4. Use the code below to save screenshot.

```python
>>> SaveScreenShot('path>/screenshot.png')
```

Note:

1. Make sure you have write access to the folder in which you would like to store the file, for example, in Windows, directly writing under C:/ drive is usually not allowed. In this case, try to write it into your “Desktop” folder, which is typically ‘C:/Users/<Your Name>/’.
2. ‘\’ symbol is not allowed for the path name. Please replace all ‘\’ in your path name to ‘/’. 