Tutorial: BwUniCluster 3.0/HoreKa
Visualization

In this tutorial we will learn how to visualize the results of our OpenFOAM simulations in ParaView, which allows us to examine the physical meaning behind our calculations. There are plenty of useful videos about this free software. This section covers various approaches to loading your data and explores useful features.

1. What is Paraview?

Paraview is a free, open-source program for data analysis and visualization. It is quite powerful and it is fully compatible with OpenFOAM.

Note: This tutorial is prepared with version 5.10 of ParaView. Slight deviations may appear for newer versions.

Note: Tecplot also recognises/imports OpenFOAM-data but it is not free of charge. However, many universities do have licences for Tecplot.

2. Preparation (postprocessing) of the OpenFOAM data

We take as a start for this tutorial the already finished parallel simulation case from the tutorial „5_hot_radiation_tutorial_parallel". The directory is presented in Fig. 1 and contains the initial state (directory "0") and the grid for the non-parallel (serial) case in directory „constant", the 8 directories with the number of each processor (before reconstruction). In the terminal we postprocess the results with the command:

$> reconstructPar (this is the recommended way of postprocessing)

This creates the postprocessing directories 100 to 935 which are the time-steps (or iterations) that have been saved during the simulation in each of the processors' directory. In these directories the **reconstructed data** are placed. **They correspond to the directories one would obtain when running the case in a serial (non-parallel) mode.** So, up to now we have the data in two different formats: serial (reconstructed) and parallel (as from the simulations, or decomposed). Both formats can directly be loaded to ParaView. A third way to prepare the data for visualization is to apply the command in the hotRadiationRoom-directory:

$> foamToVTK (not recommended)

This writes the data in the directory VTK in the VTK-format. We copy the data locally –

as shown on the left-hand side of the Figure. Directories „system” and „constant” should be copied too.

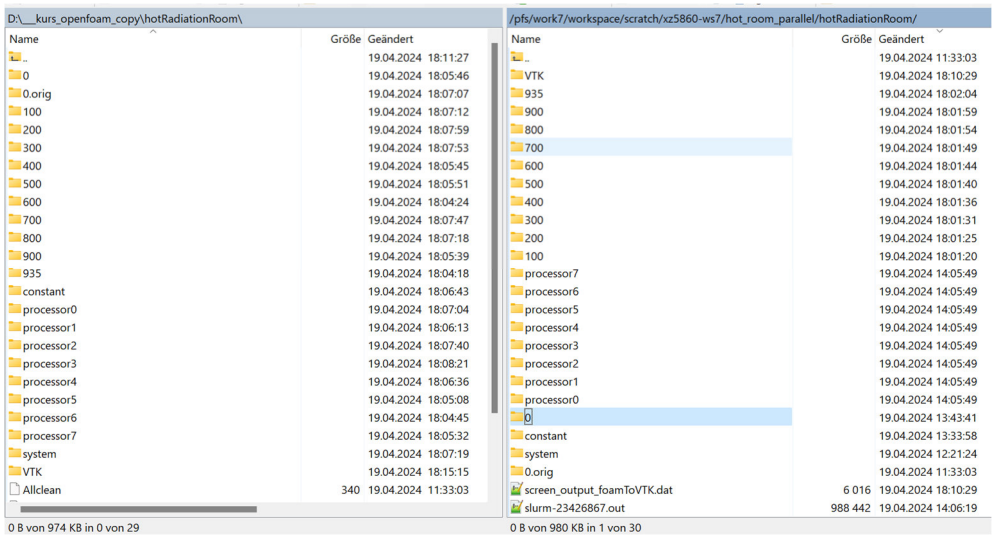

Fig. 1 The directory containing the simulated and the postprocessed results. The right-hand side directory is on the bwUniCluster, the left one – on the local laptop

3. Loading the data in ParaView and work with the reconstructed case

To load the reconstructed (analogous to serial) data we create an empty file called „foam.foam” in the local directory hotRadiationRoom. Then we start ParaView and choose “File”, “Open” and double-click on „foam.foam” to load it. In the pipeline browser the name „foam.foam” appears (as a name of the case). Next to the pipeline browser is the tab „Properties”, we click on it and then on the green button „Apply”. The first variable „p” is loaded, the screen and the variables are given in Fig. 2. Each variable appears twice – as a cell-data (the „truth” – as computed in OpenFOAM and marked with a small cell/cube symbol) and as a cell-to-point data (or simply point format). The data in the point-format are smoother, but for quick views one may wish omit them (it may take some time) which is controlled with the highlighted (circled in red) button in Fig. 2. Currently we keep the point-data and continue to work using them.

Note: in some cases, you may need to choose first another color legend. To do so, one should go to the „Properties” menu, submenu „Coloring”.

Note: ParaView automatically chooses the last possible time-step, in this case „935”. By choosing another time-step, one can see the development of the variables in time.

Using the mouse, the picture can be rotated/translated/scaled and one can see that the non-transparent (opaque) walls of the room envelope in 3D are loaded.

Selecting the temperature as variable and viewing the room from the bottom, we start exploring the icons from the „common” menu. Menus can be selected/deselected freely. By right-clicking on any one of them a selection menu appears as shown in Figure 3. Activating

again menus „Properties” and „Pipeline Browser” and clicking on the name „foam.foam” to select it, we click on the slice-icon (Fig. 3) to make a 2D cut through the room. In „Properties” we choose Y_normal and the screen looks like in Fig. 4. Then we deselect „show plane” and click „Apply”. A new element „Slice1” appears in the pipeline browser (on the left) having an eye next to it. We click on this name to select it. All selections and changes we will now make, will be only valid for this slice and not affect the 3D view of „foam.foam”.

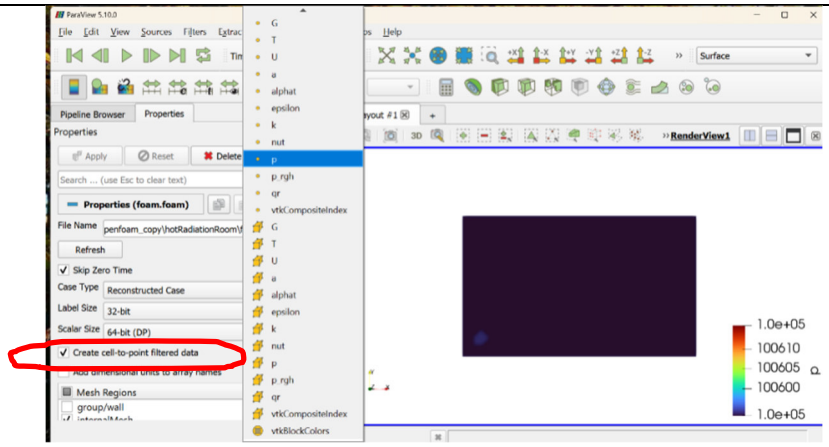

Fig. 2 The variable „p” loaded as „points” and a list of all variables loaded.

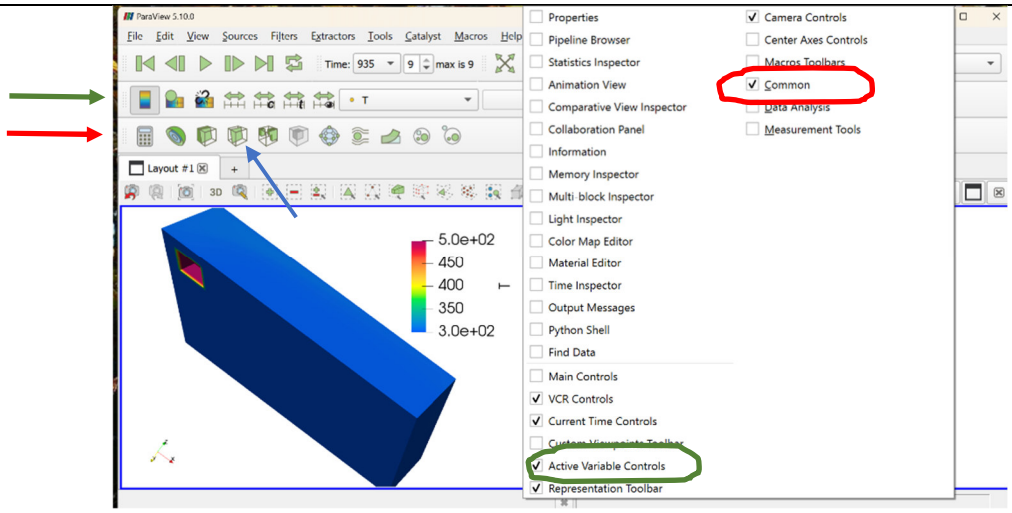

Fig. 3 The menus can be freely configured. Note that for this picture the menus „Properties” and „Pipeline Browser” are currently de-selected.

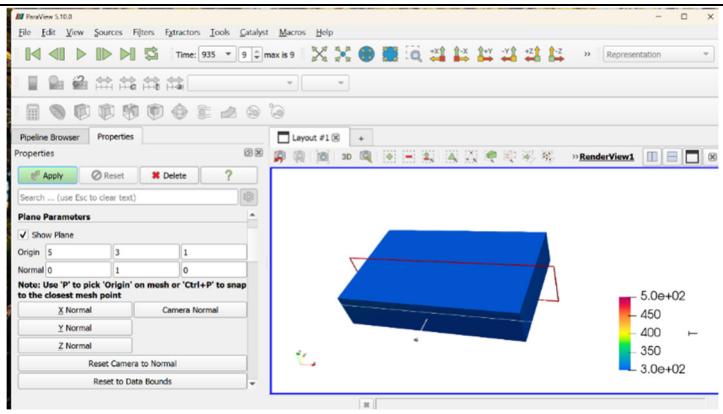

Fig. 4 The cutting plane Y_normal (in red): one can move it forth and back with the mouse

Now we select the variable “T” (temperature). The temperature appears in a uniform blue color. This is because of a much higher temperature (500K) is available somewhere else in the room (on the box surface). Therefore, we rescale the temperature range between 300 and 303K. To do this, we click on the icon with letter „c” from the menu „Active variables control” and set manually the range. The result from rescaling is shown in Figure 5. Alternatively, we can click on the icon with the eye (Rescale to Visible Data Range).

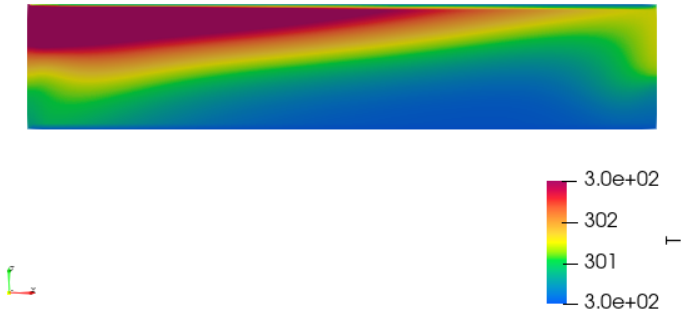

Fig. 5 The rescaled temperature (300-303K) for slice1.

To present the velocities as arrows, we highlight again “Slice1” in the „Pipeline Browser” menu and click on the „Glyph”-icon (“common”-menu). A new Glyph appears with the name „Glyph1”. Before we click on „Apply” in the „Properties” menu, we select the Glyph Type „Arrow”, select as orientation array - „U”, as scaling array - again „U” and as Vector Scale Mode „Scale by Components”. The settings are shown in Fig. 6. We still see the Temperature variable on the right – the reason is that we did not deselect it (using the eye-icon) in the „Pipeline Browser” menu. We deselect (close the eye of “Slice1”), select Gylph1 and come back to the Properties menu. We go down to submenu „Masking” and choose „Glyph mode”=Every Nth Point, Stride=4. A bit more down we select the variable that will color the arrows, e.g. „k” in submenu „Coloring”. In the submenu „Scale” we choose Scale Factor=10. After „Apply” we obtain the result shown in Fig. 7 which is still not perfect. The reason is that the velocity component down the direction along „y”-axis is quite strong and the arrows try to reflect this and therefore appear in a skewed manner. To improve this, we will use the calculator tool.

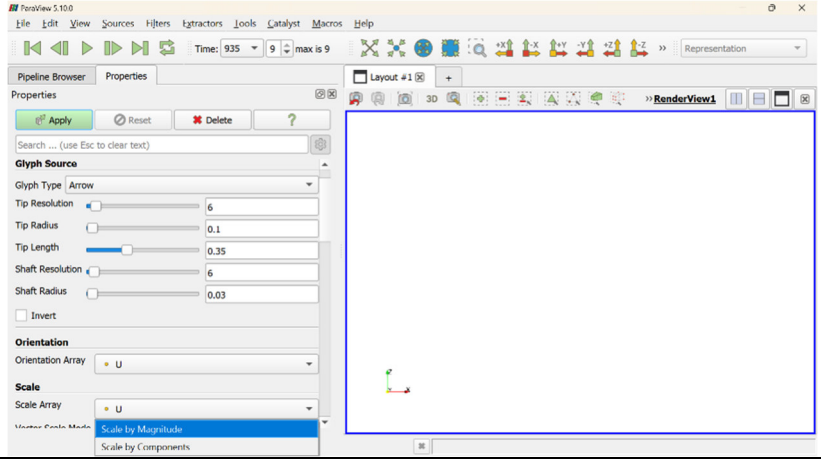

Fig. 6 The settings for presenting the velocity vectors (arrows).
Note: There are no options to manage the arrow details in a later version ParaView 5.13

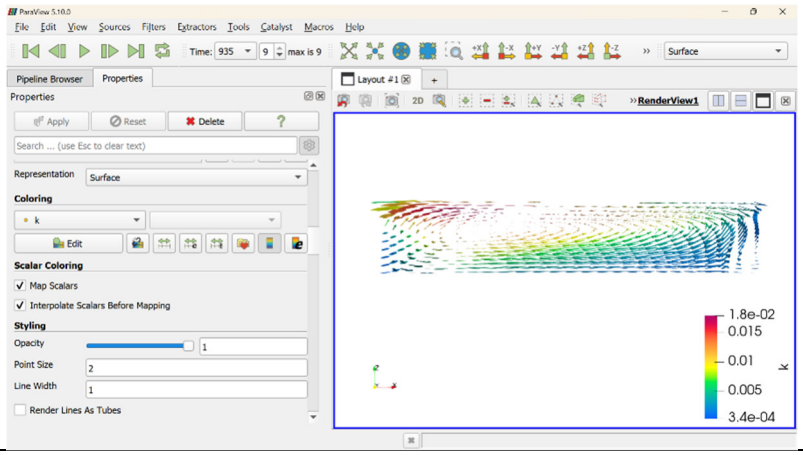

Fig. 7 The 3D velocity vectors in the plane „Slice1”

The calculator tool will help us to remove the velocity component in the direction of axis „y” – this way the arrows will appear in the plane xz and will become more „nice”. To do so, we deselect (close) the eye-icon of „Glyph1” and deselect „Glyph1” itself. We select „Slice1” (and open the eye of it) and will operate on its data. Then we click on the calculator icon from the „Common” menu. In „Properties” we change the name from „Result” to „xz-velocity”. One line below the Result name we type „U_X*iHat+U_Z*kHat” which calculates the new vector with only two components (U_X and U_Z you can find in the scalar variables of the calculator). Note, that iHat and kHat are the unity vectors in the corresponding direction. We click „Apply”. The settings for the calculator are shown in Fig. 8.

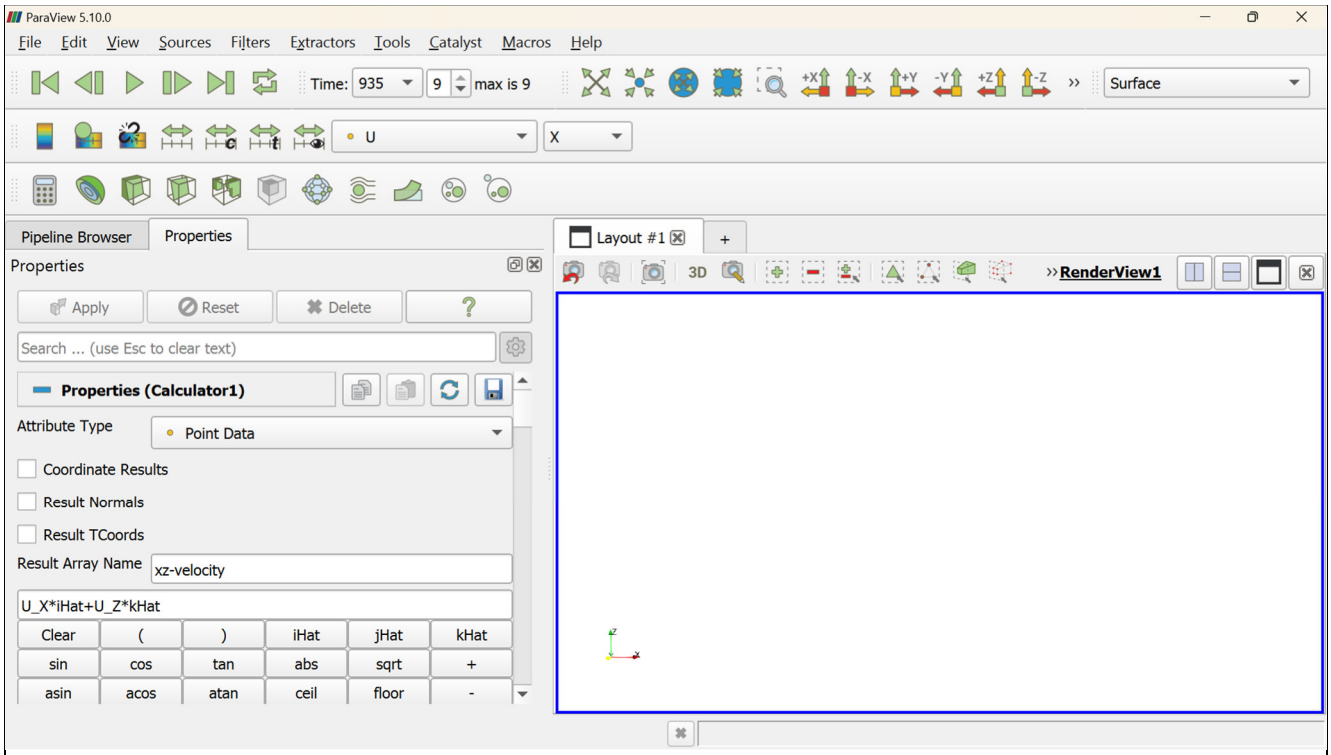

Fig. 8 The settings used in the calculator

We need to plot the new vector creating for it a new Glyph. We go to the pipeline

browser, deactivate all eye-icons and highlight the “Calculator1”. Using the „Glyph”-icon again, we create the new Glyph2 as in Fig. 9.

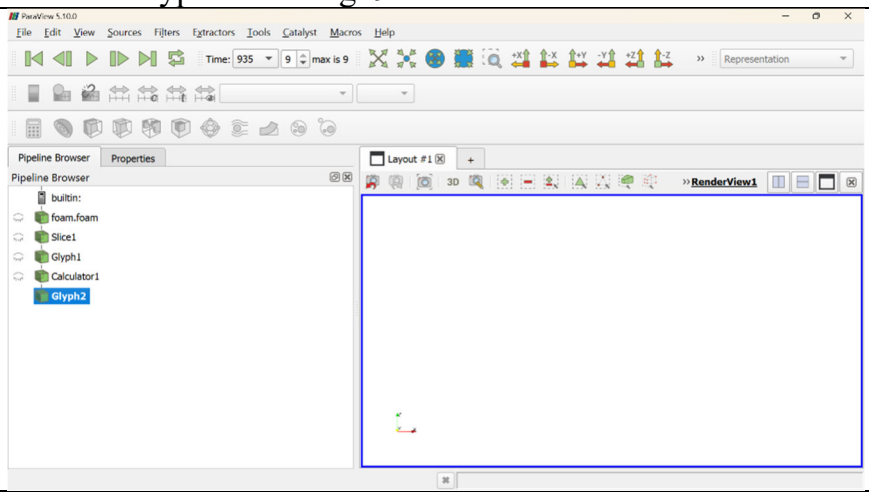

Fig. 9 Creating „Glyph2” from „Calculator1”

In the „Properties” menu, we choose the options shown in Figure 10 (unfortunately, this menu is not available in all ParaView versions). The result is also shown in Figure 10. The arrows are clearly visible and much nicer than before.

Note: It is important that the calculator is created from „Slice1” (and not from „Glyph1”) and that „Glyph2” is created from „Calculator1”, see Fig. 9.

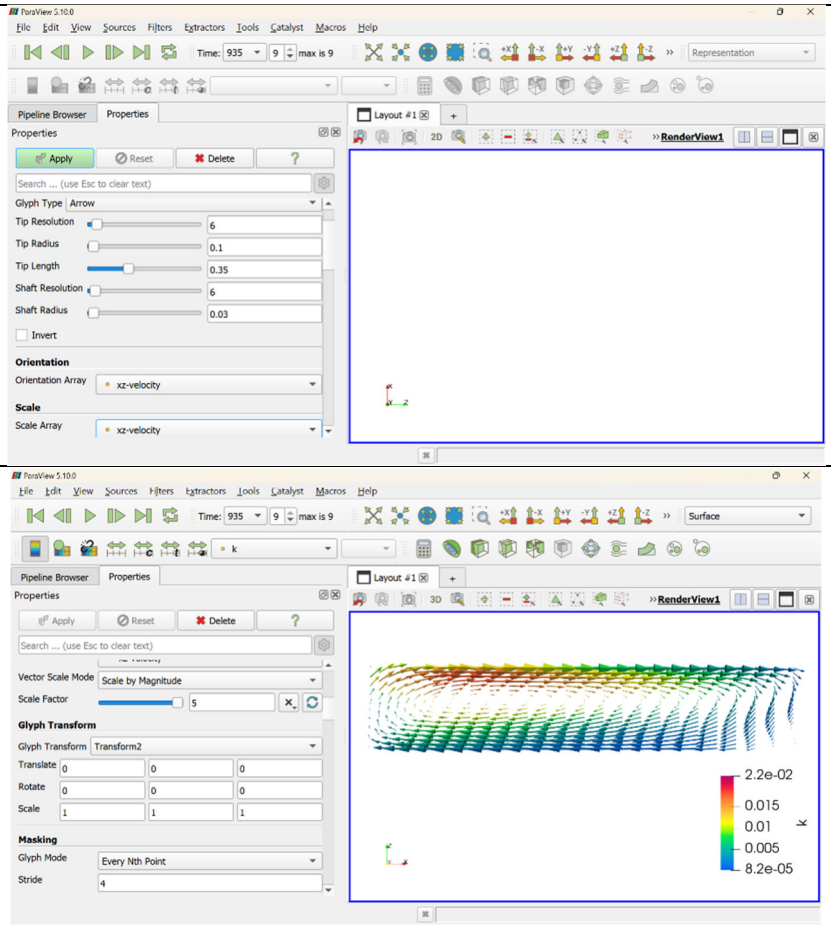

Fig. 10 Options for Glyph2 and the resulting vectors coloured by variable “k”

4. Loading the data in ParaView and work with the decomposed case

To load the decomposed case, ParaView looks in the controlDict-file and decides what to read and load from there. For this reason, we need the system-directory to be copied locally too. The good news is that with this method one can look to only part of the data. For instance, if we do not want to load the data from processors 3 and 7, we simply rename these directories to _processor3 and _processor7 (prior to start ParaView). Loading part of the domain is especially useful, if you have a large set of data and low RAM. Starting now ParaView, we click on "File", „Open" and go to the system-directory. Then we choose (down) „Files of Type = All Files" and click on file controlDict and „OK".

We are prompted to select the solver - select „OpenFOAMReader" and again OK. The case to be loaded is called now „controlDict", see Fig. 11 (top). We go to „Properties" and on the left-hand side we select „Decomposed Case" and „Apply", see Fig. 11 (middle). The picture that appears contains the 3D room but without processors 3 and 7 which we renamed – see Fig. 11 (bottom). We selected the variable "k" (instead of "p" that appears by default).

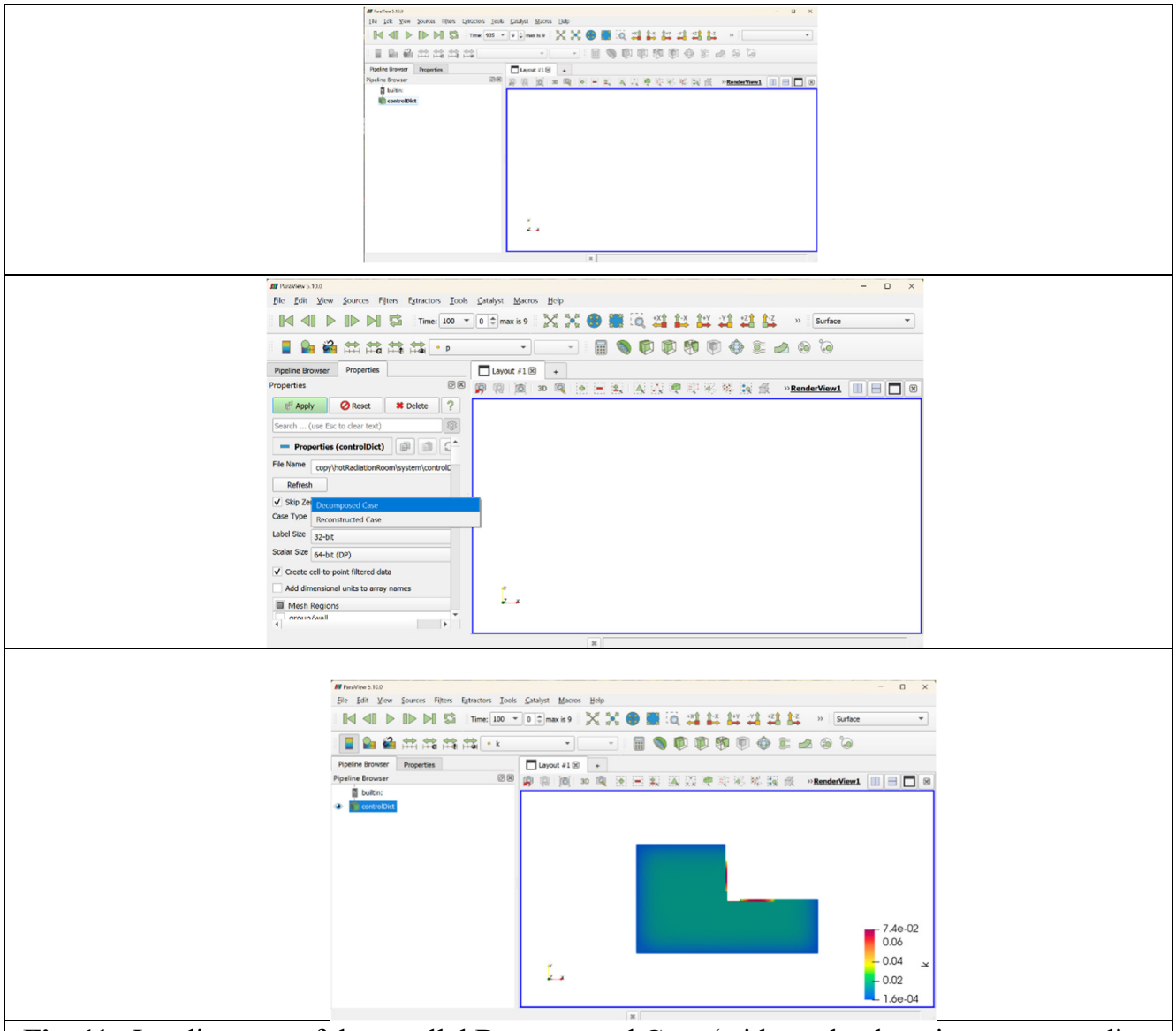

Fig. 11 Loading part of the parallel Decomposed Case (without the domains corresponding to processors 3 and 7 – to decrease the data amount)

After this, the work proceeds as in the reconstructed case from chapter 3.

5. Loading the data in ParaView and work with the VTK data

To load the VTK-data, we go to “File”, “Open” and go to directory VTK. In it, we select the „hotRadiationRoom.vtm.series” – file and OK as shown in Fig. 12. As usual, we go to the „Properties” menu and click „Apply”. The name of the case in the „Pipeline Browser” is „hotRadiationRoom.vtm.series”. By default, the VTKBlockColours are shown. By choosing any other variable, the work can continue as in chapter 3.

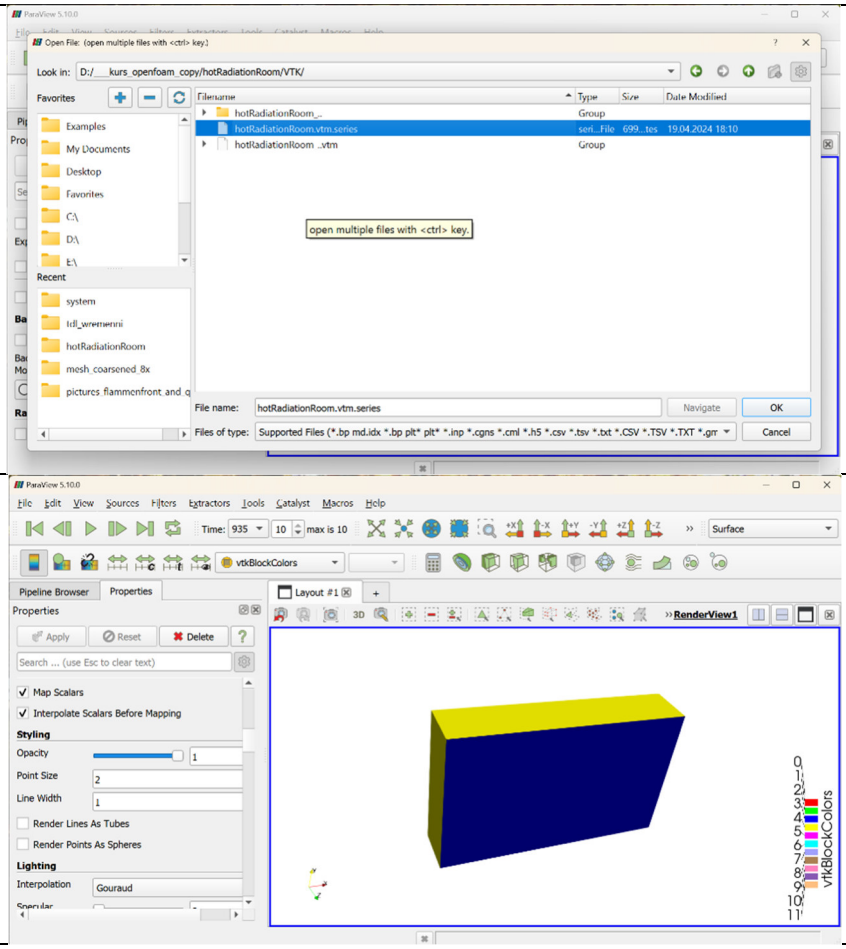

Fig. 12 Loading the VTK data