Exercises for the introduction to the Grid Interface

Exercise 1 Constructing Dune grids

In this exercise you will experiment with constructing different grids. The working directory of the exercise is the same as in the previous exercise on the grid interface.

The file `grid-exercise2.cc` contains a code, which

- defines the `GridType` to one of a list of available Dune grids,
- constructs a grid through one of the three basic factory concepts:
  - `StructuredgridFactory` for equidistant grids,
  - `GmshReader` for unstructured grids,
  - `TensorGridFactory` for tensor product grids,
- potentially refines the grid once globally (disabled by default),
- fills a data structure that maps each cell to its index in the index set,
- Outputs this data structure to a vtk file which can be visualized in `paraview`.

Try to construct as many different grids as possible and look at the result in paraview. Here are some questions to guide your exploration of grid construction:

- Find out (visually) how the elements in a `YaspGrid` are ordered in the index set.
- Construct a structured grid with an unstructured grid manager.
- Load an unstructured grid from one of the `.msh` files you find in the exercise directory.
- Construct a `YaspGrid` for the domain $[-1, 1]^2$.
- Enable the global refinement in the code and observe the effect on the index set for structured and unstructured grids.
- Build a tensor product `YaspGrid` with and without global refinement. What do you observe?

Exercise 2 VISUALIZATION WITH PARAVIEW

The purpose of this exercise is to explore the visualization capabilities of ParaView.

We start by visualizing the file `2dexample.vtu`. It contains two data sets: a solution of a finite element method for the Poisson equation and the interpolation of the exact solution onto the grid. Try the following steps:

- Load the file into paraview and get it visualized.
- Get a 3D Plot by using the `Warp by Scalar` filter.
- Plot the solution along a line using the `Plot over Line` filter.
- Add a third data set using the `Calculator` filter, which contains the error of the finite element discretization.

We will now visualize the solution of the finite volume example from this morning. If you have not successfully implemented the scheme, you can produce the results by running the executables from the solution subfolder. Instead of a single file, we now load a `pvd` file that describes a time series. Try the following:

- Load the `pvd` file, get it visualized and run it.
- Use `Plot over Line`.
- Try using the filter `Warp by Scalar`. What might be the problem with it?
- Calculate the total mass by using the filter `Integrate Variables`.
- Try plotting the total mass over time and hopefully observe mass conversation. To do this, you need to select the row with your integrated variable by clicking it and then apply the filter `Plot Selection over Time`.

Now load the 3D Finite Volume Data and try to explore it with ParaView. In addition to the tools mentioned above, you might want to play around with `Clip`, `Contour` and `Threshold`.