# Assignment 03

## 1 Instructions

1. Open terminal in the parent directory containing the assignment files.
2. Go to the build directory : `$ cd build`
3. Build the executable : `$ make`
4. Execute the program : `$ ./intug path/to/INGRID.vtk path/to/OUTGRID.vtk`
5. The `OUTGRID` together with the interpolated values is written as `OUTGRID_interpolated.vtk` to the `05.outputs` directory.
6. Build the bonus task : `$ make intug_bonus`
7. Execute the program : `$ ./intug_bonus path/to/INGRID.vtk path/to/OUTGRID.vtk`
8. Observations for the main task are presented in section 2.
9. Observations for the bonus task are presented in section 3.

## 2 Main Task

Figure 1: `INGRID` and `OUTGRID` for coarse mesh.

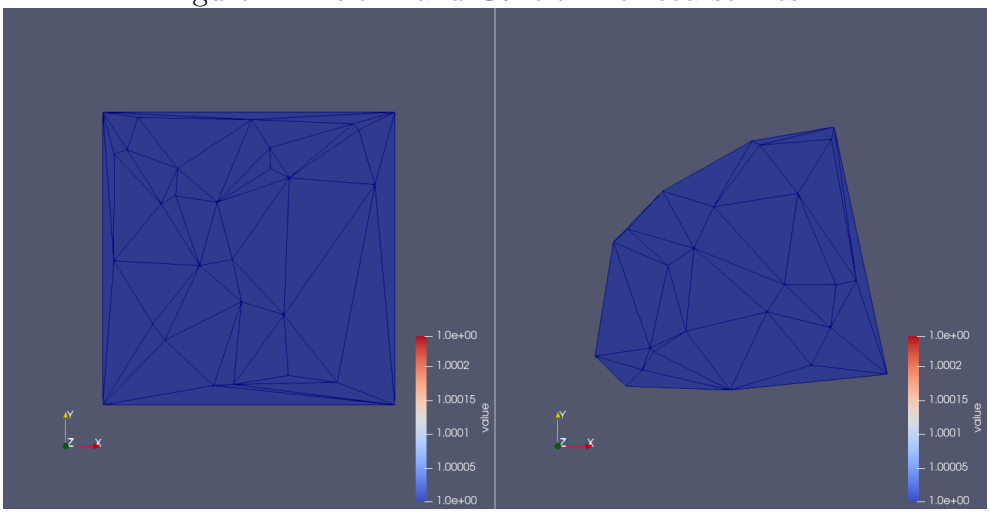


Figure 2: `INGRID` and `OUTGRID` for fine mesh.

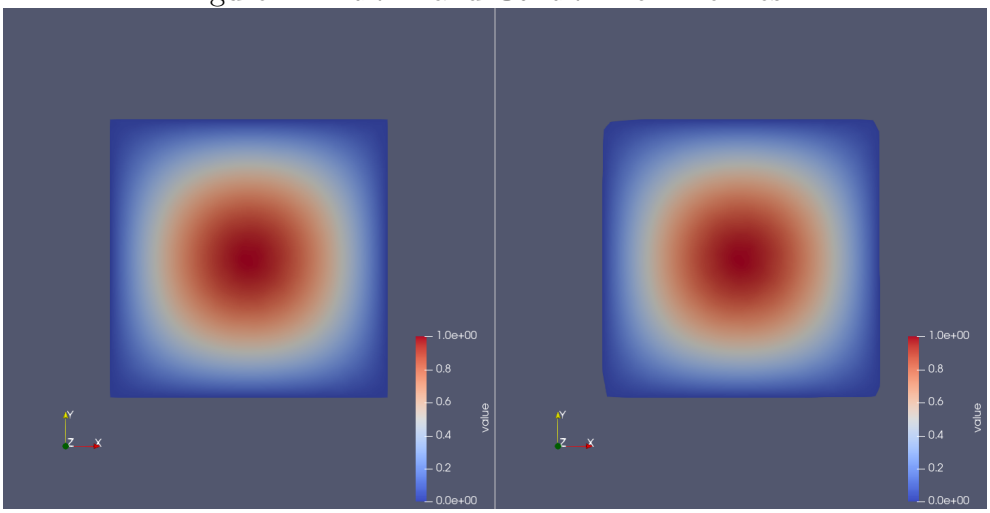



Figure 3: `INGRID` and `OUTGRID` for finest mesh.

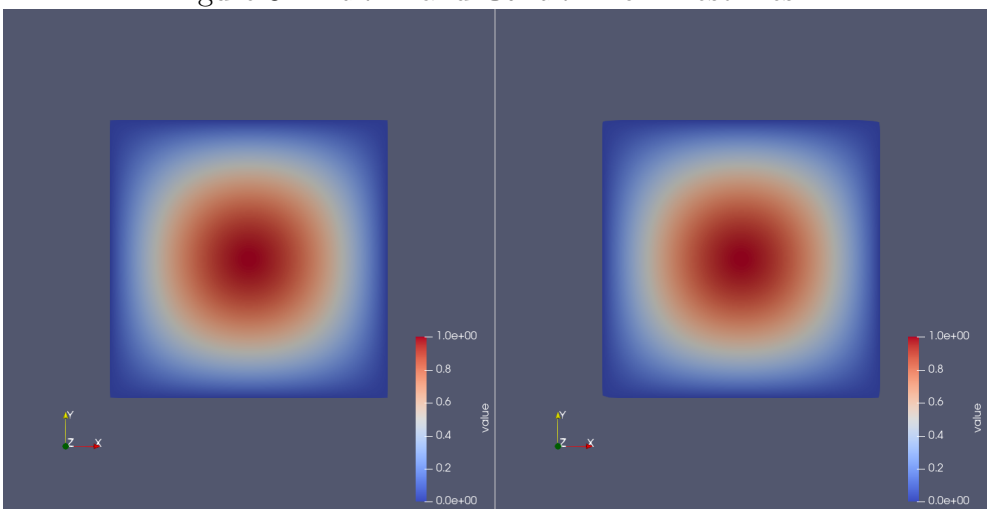



Figure 4: `INGRID` and `OUTGRID` for medium mesh.

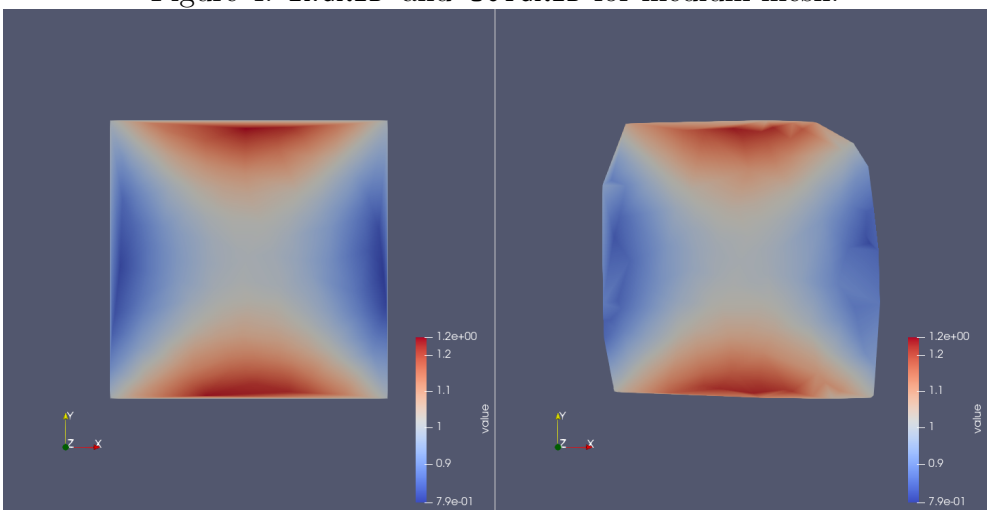
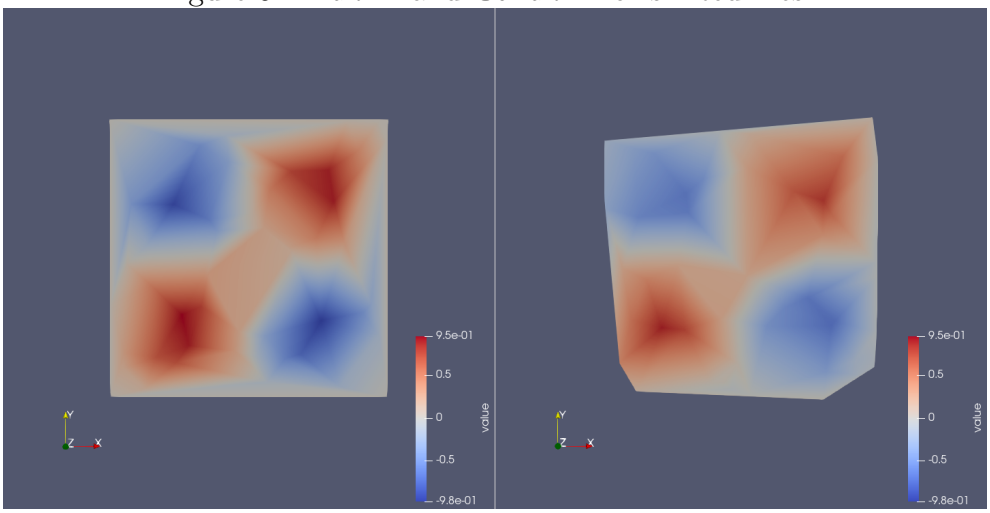
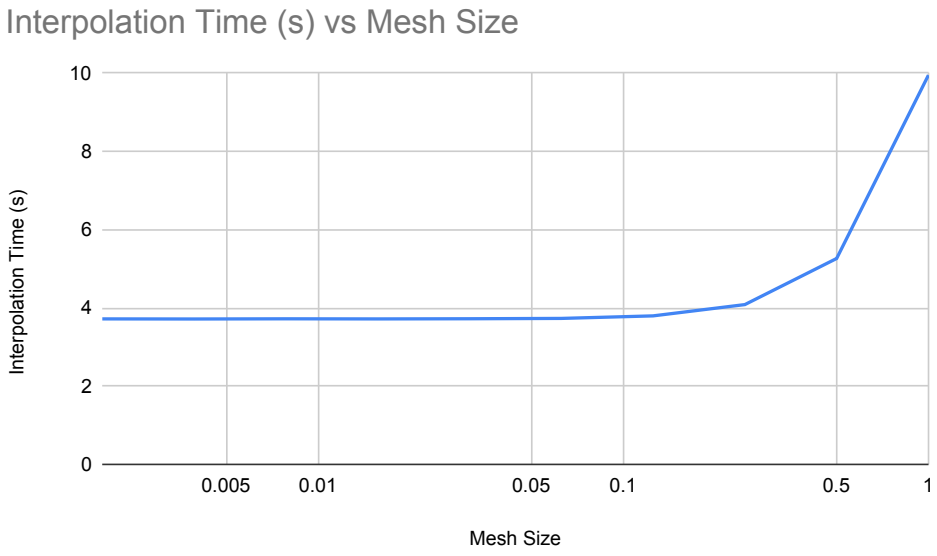


Figure 5: `INGRID` and `OUTGRID` for shifted mesh.




# 3 Bonus Task

Figure 6: Runtime vs Mesh size (logarithmic x-axis).




For larger mesh sizes, cells in the intermediate structured grid have more intersecting trianges from the input grid. Therefore, for all the points in the output grid that belong to a cell have many possible trinagles to which they can belong to. Checking locations for all such points and triangles increases the run time.

As the mesh size gets smaller, the number of intersecting triangles become less. This reduces the search spaces for the points being interpolated and improves run time. After a cetrain threshold, it is possible that the cells are so small that they lie inside the triangles of input mesh and further reduction of mesh size does not reduce the number of intersecting triangles for the cell.