Project 1: Mesh Generation

AE 623, Computational Fluid Dynamics II, Winter 2019
Runda Ji

I. Introduction

In this project, a set of unstructured, triangular meshes with different resolutions were generated for a two-dimensional channel with a bump perturbation on the bottom wall. To begin with, a baseline mesh with 104 elements was generated using Gmsh. After that, several rounds of mesh refinement were implemented. Verification tests were also conducted on each of the refined mesh.

II. Methods

Since we need to apply Gmsh to create the baseline mesh, we are going briefly introduce the format of the input and output files of Gmsh within this section.

A. Input file for Gmsh (.geo)

Gmsh need a geometry file as an input file to determine the boundary of computational domain. The input file of Gmsh, the .geo file, includes four sections: "Points", "Lines", "Curve Loop" and "Plane Surface".

1. Points

In order to determine a point (node), we need four values:

$$\text{Point}(i) = \{x_i, y_i, z_i, r_i\}; \tag{1}$$

where (x_i, y_i, z_i) are the x,y,z coordinates, respectively. While r_i represents the local grid size around that specific point.

2. Lines

The lines (segments) are defined as,

$$\text{Line}(i) = \{p_{i1}, p_{i2}\}; \tag{2}$$

where p_{i1} and p_{i2} are the two nodes of the segment.

3. Curve Loop

After each segment is defined, we may then create a curve loop that contains all these segments,

$$\text{Curve Loop}(i) = \{l_{i1}, l_{i2}, l_{i3}, \cdots, l_{in}\}; \tag{3}$$

where l_{ij} represents the j^{th} segment of the i^{th} curve loop.

4. Plane Surface

Finally, the plane surface can be defined as a set of the curve loops. Since we only have one curve loop for this specific problem,

$$\text{Plane Surface}(1) = \{1\}; \tag{4}$$

B. Output file of Gmsh (.msh)

After loading the .geo file into Gmsh and generating the mesh, we can easily obtain a .msh file that automatically created by Gmsh. This mesh file includes four sections: "MeshFormat", "Entities", "Node" and "Elements".
The two sections that we are interested in are "Nodes" and "Elements". The format for the .msh file can be easily found at the official website of Gmsh.[1] Following this format, we can transform the .msh file into our familiar .gri file.

```
$Nodes
  numEntityBlocks(unsigned long) numNodes(unsigned long)
  tagEntity(int) dimEntity(int) parametric(int; see below) numNodes(unsigned long)
    tag(int) x(double) y(double) z(double)
       <u(double; if parametric and on curve or surface)>
       <v(double; if parametric and on surface)>
    ...
  ...
$EndNodes

$Elements
  numEntityBlocks(unsigned long) numElements(unsigned long)
  tagEntity(int) dimEntity(int) typeEle(int; see below) numElements(unsigned long)
    tag(int) numVert(int) ...
    ...
  ...
$EndElements
```

Listing 1. Format of "Nodes" and "Elements" in the .msh file

III. Questions and Tasks

A. Task 1. Generate the coarse mesh

As aforementioned, in order to generate a mesh using Gmsh, we need to first create a .geo file for Gmsh to read. Specifically, in order to define the shape of boundary, all positions of the nodes on the boundary should be included in the .geo file. The distribution of these boundary nodes is depicted in figure 1. Note that on the bottom wall, the nodes are clustered in the vicinity of the bump, rather than evenly distributed.

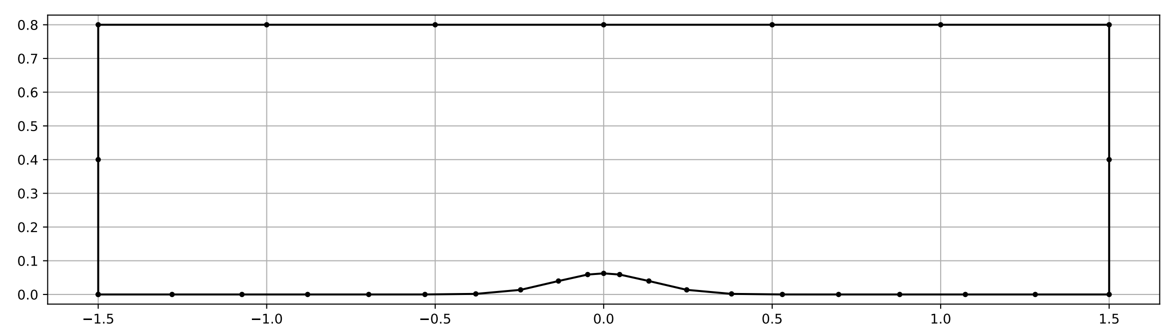

Figure 1. Nodes distribution on the boundary.

After loading the .geo file into Gmsh, we can easily obtain the .msh file that automatically generated by Gmsh. This .msh file can be transformed into .gri file according to the format we have mentioned in the previous section. The mesh generated by Gmsh is shown in figure 2. Note that the total number of elements is nElem = 104 for this baseline mesh.

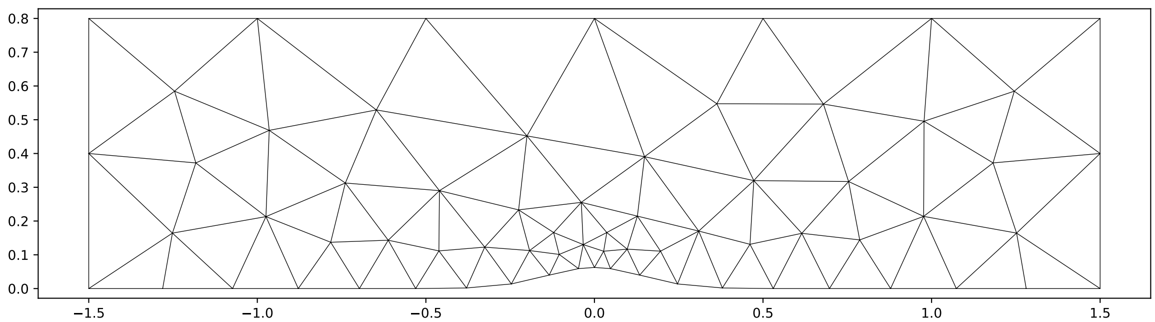

Figure 2. Baseline mesh generated by Gmsh (bump0.gri).

B. Task 2. Generate the matrices

In order to find the matrices I2E and B2E, we need to establish a sparse matrix (connection matrix) to store the data. This connection matrix $\bar{\bar{C}}$ is a (nNode $\times$ nNode) matrix. If there is no connection between node i and node j, then $C_{i,j} = 0$. Otherwise, the global edge index of the edge connecting these two nodes will be stored in that cell. To construct this connection matrix $\bar{\bar{C}}$, we need to loop over all elements. For each element, we will need to check all three edges.

If an edge has not been previously stored in matrix $\bar{\bar{C}}$, then this edge will first be append to a global edge list. At the same time, a global edge index will be tagged to that edge. After that, the global edge index corresponding to that edge will be stored in matrix $\bar{\bar{C}}$.

Otherwise, if an edge has already been stored in matrix $\bar{\bar{C}}$, which indicates that the edge is sharing by two elements, then we can say that the edge is connected to both elements.

As shown in figure 3, the test mesh includes four nodes, five edges and two elements.

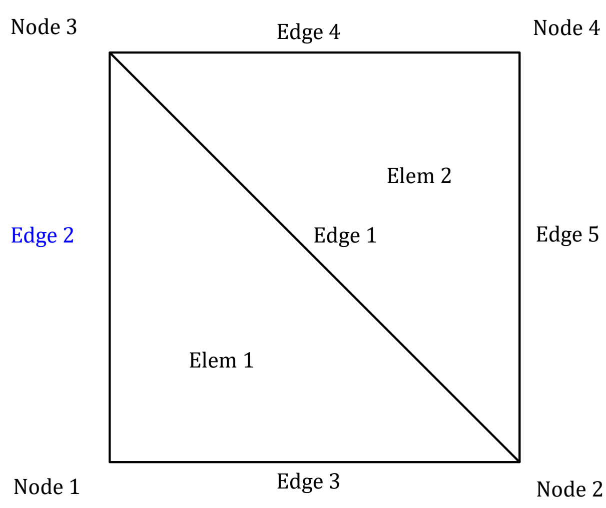

Figure 3. Schematic of the test mesh.

The matrix $\bar{\bar{C}}$ and the list of edges for this test mesh are shown as follows,

$$\bar{\bar{C}} = \begin{bmatrix} 0 & 3 & 2 & 0 \\ 3 & 0 & 1 & 5 \\ 2 & 1 & 0 & 4 \\ 0 & 5 & 4 & 0 \end{bmatrix}$$

Global edge #	n_1	n_2	e_1	e_2	t_1	t_2
Edge 1	2	3	1	2	1	2
Edge 2	3	1	2	Left	1	None
Edge 3	1	2	3	Bottom	1	None
Edge 4	4	3	1	Top	2	None
Edge 5	2	4	3	Right	2	None

Figure 4. Connection matrix and list of edges for the test mesh.

The I2E and B2E matrices for this test mesh are given by,

$$\text{I2E} = \begin{bmatrix} 1 & 1 & 2 & 2 \end{bmatrix} \tag{5}$$

$$\text{B2E} = \begin{bmatrix} 1 & 3 & \text{Bottom} \\ 2 & 3 & \text{Right} \\ 2 & 1 & \text{Top} \\ 1 & 2 & \text{Left} \end{bmatrix} \tag{6}$$

The normal vectors for interior and boundary faces In and Bn are,

$$\text{In} = \begin{bmatrix} 0.707107 & 0.707107 \end{bmatrix} \tag{7}$$

$$\text{Bn} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{8}$$

The area for the elements are,

$$\text{Area} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \tag{9}$$

C. Task 3. Verification test

To verify the mesh, we need to calculate the following vector for each element.

$$\vec{R} = \sum_{i=0}^{3} \vec{n}_i^{\text{outward}} l_i \tag{10}$$

This vector quantity should be zero, to machine precision, on each element. As we can see in figure 5, the test mesh satisfies this condition perfectly as all $|\vec{R}| = 0$. Thus for the test mesh we have, the maximum magnitude of the "residual" vector $\max\{|R|\} = 0$.

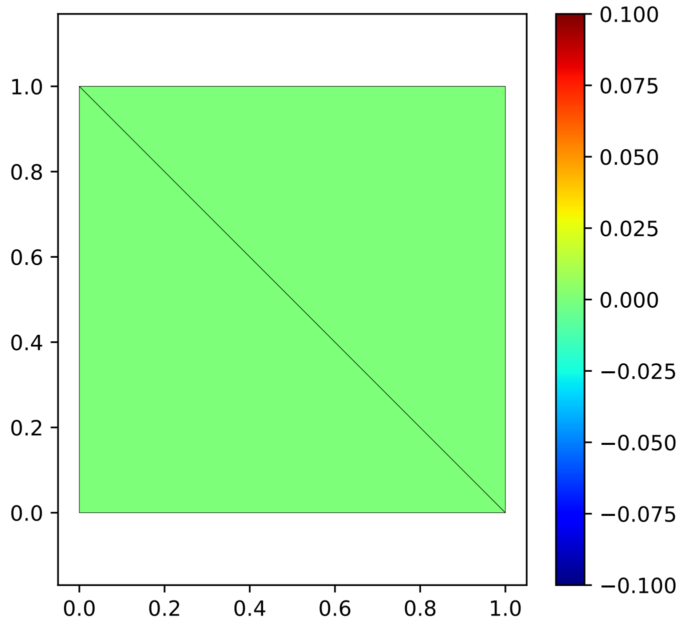

Figure 5. Verification test on the test mesh.

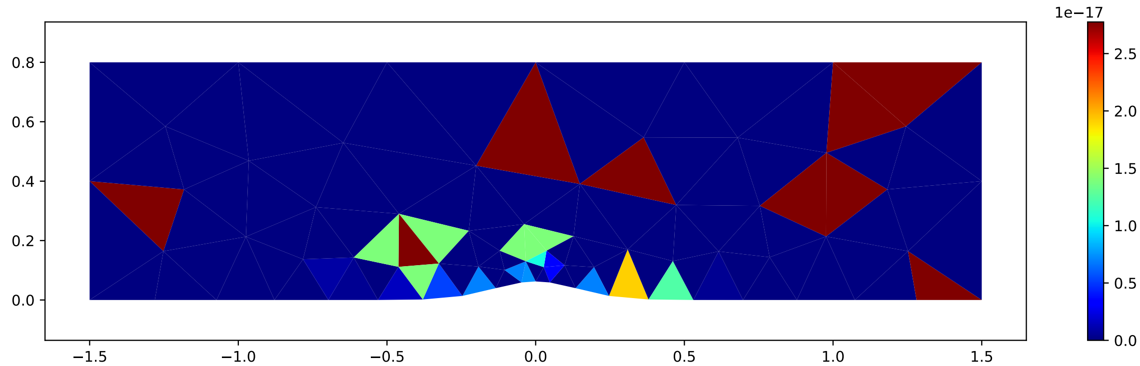

Figure 6. Verification test on the bump0 mesh.

The "residual" distribution for the baseline mesh bump0 is depicted in figure 6. Where the maximum magnitude of the "residual" vector is 2.7755575615628914^{-17}.

D. Task 4. Refinement

The meshes after the first, second, third and forth uniform refinement are depicted in figure 7.

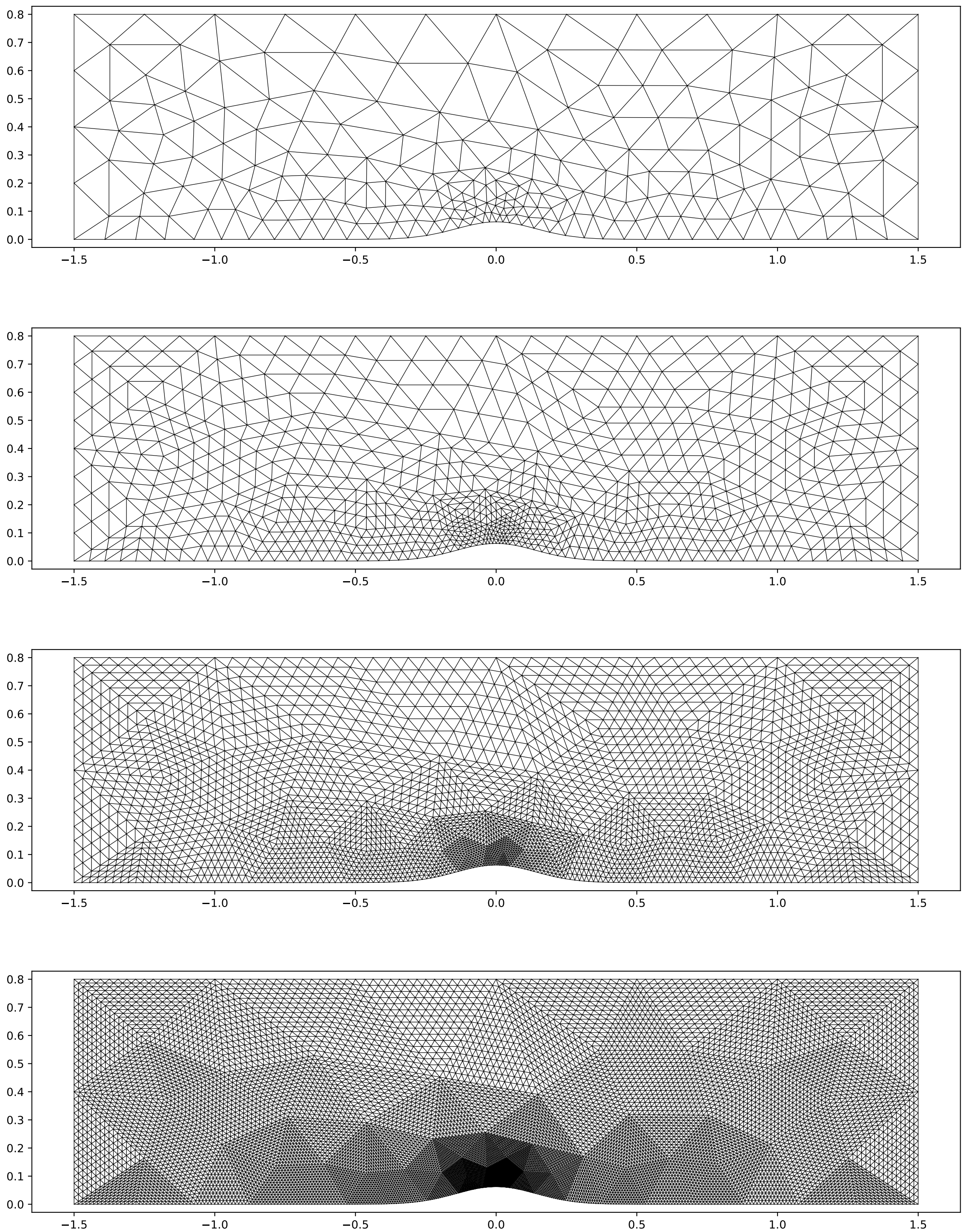

Figure 7. Mesh after uniform refinements (bump1.gri - bump4.gri).

Implementing the verification test on these refined meshes, we can see that the maximum magnitude of the "residual" vector decreases as the resolution of mesh increases.

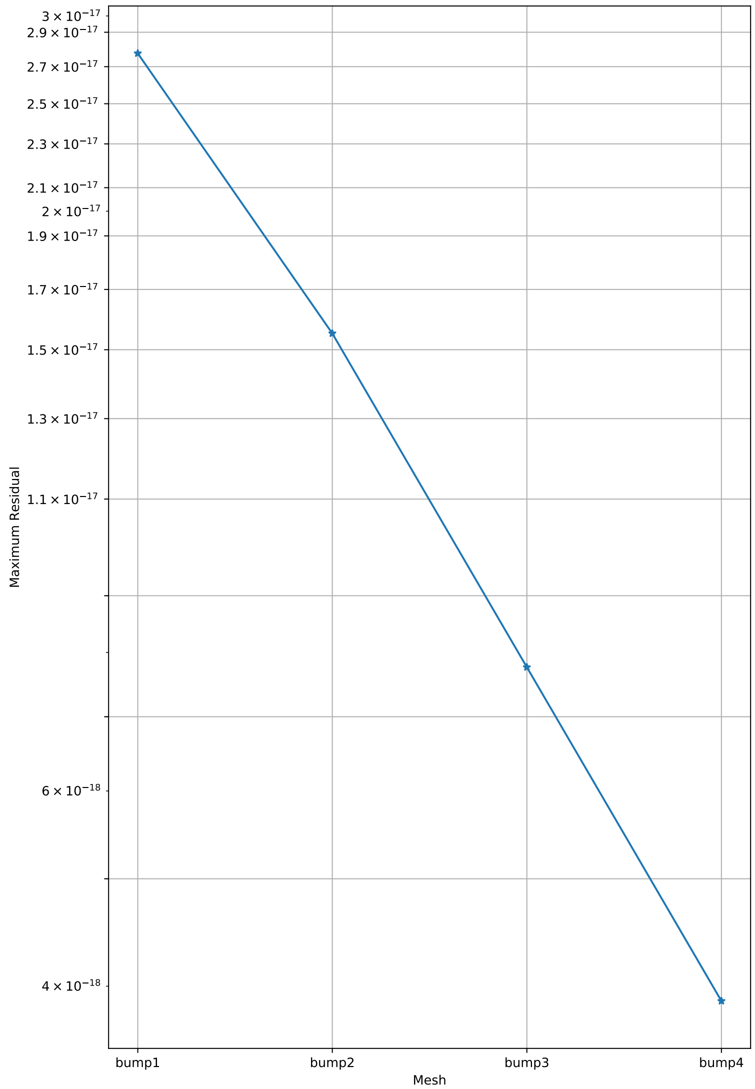

Figure 8. Residual for meshes with different resolution.

Table 1. Residual for meshes with different resolution

Mesh	$\max\{\lvert R\rvert\}$
bump1	2.7755575615628914^{-17}
bump2	1.5515838457795457^{-17}
bump3	7.757919228897728^{-18}
bump4	3.878959614448864^{-18}

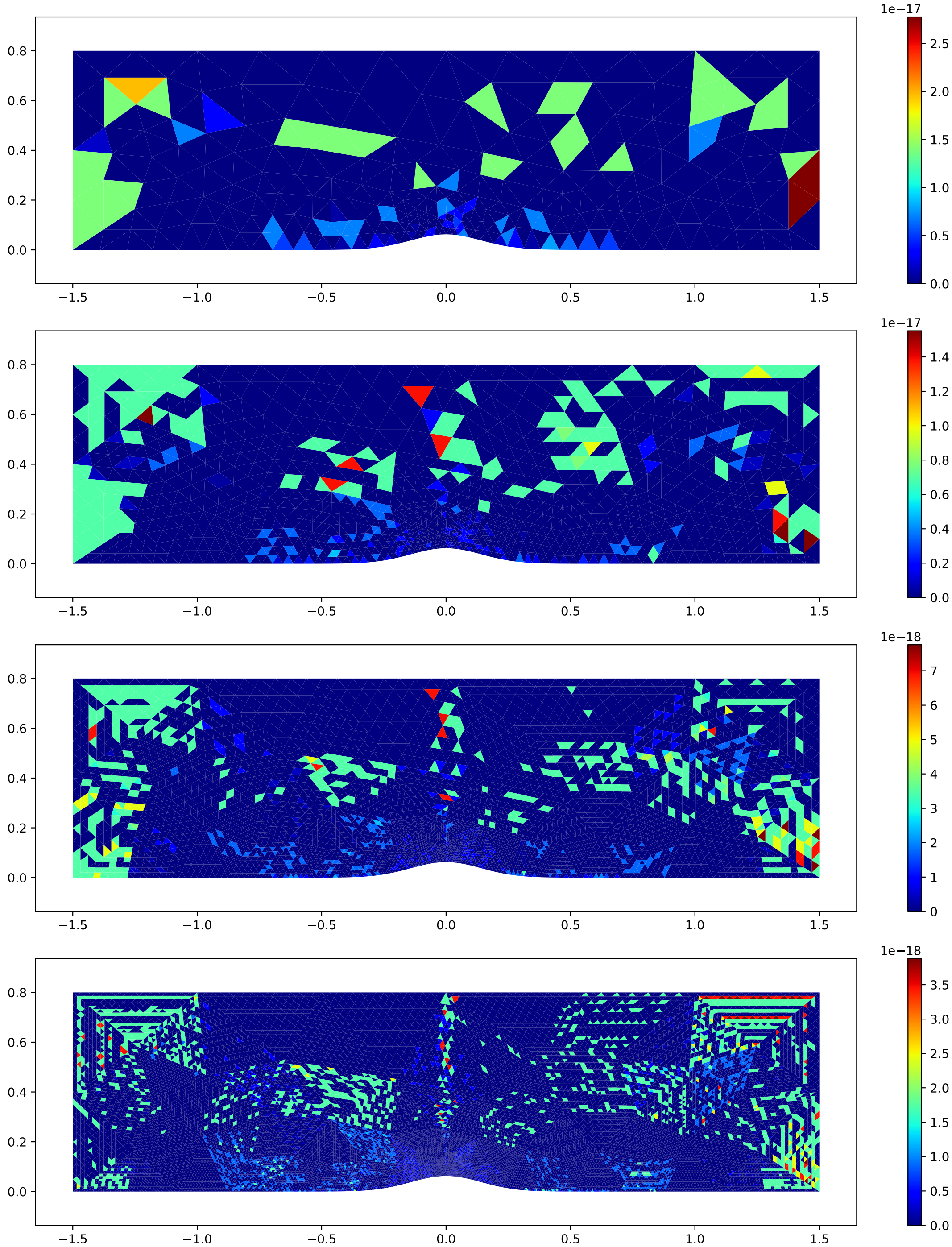

Figure 9. Residual distribution after the verification test on the refined meshes. (Top to bottom: bump1, bump2, bump3 and bump4).

References

[1]Gmsh.info. (2019). Gmsh 4.1.1. [online] Available at: http://gmsh.info/doc/texinfo/gmsh.html#MSH-file-format-_0028version-4_0029 [Accessed 20 Jan. 2019].