ME5204 Finite Element Analysis

Assignment 1

Pratham Sunkad

ME21B145

Contents

Pratham Sunkad
ME21B145

1 Problem Statement

To Generate a triangular mesh on the IITM Map using GMSH. To read mesh information in .msh format and convert it to .vtk format using python code (without use of library functions in my case). Visualize all of this using Paraview.

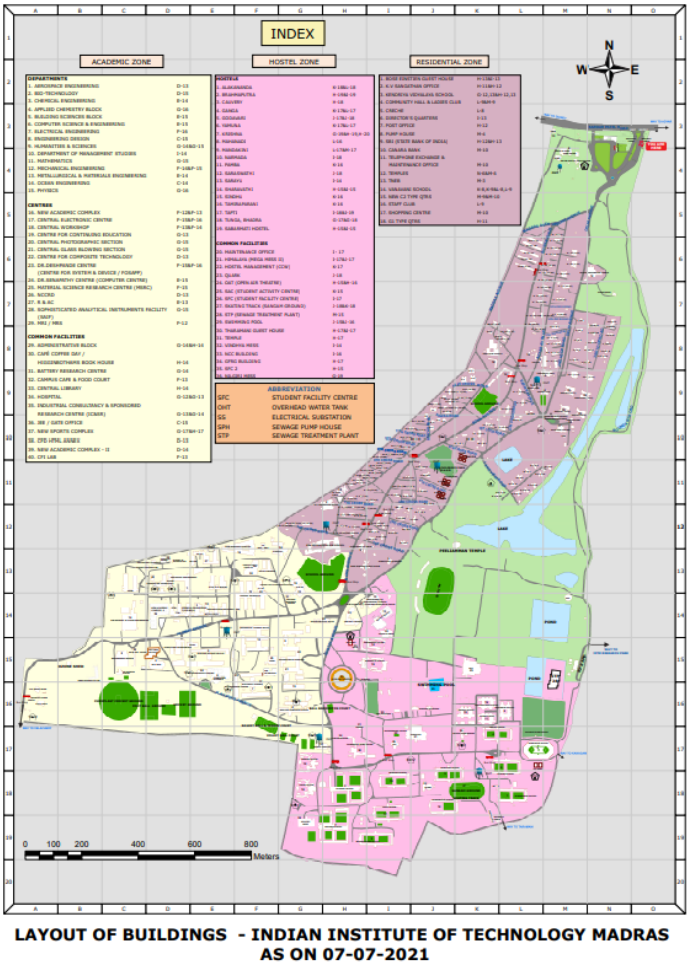


Figure 1: Indian Institute of Technology Madras Map.

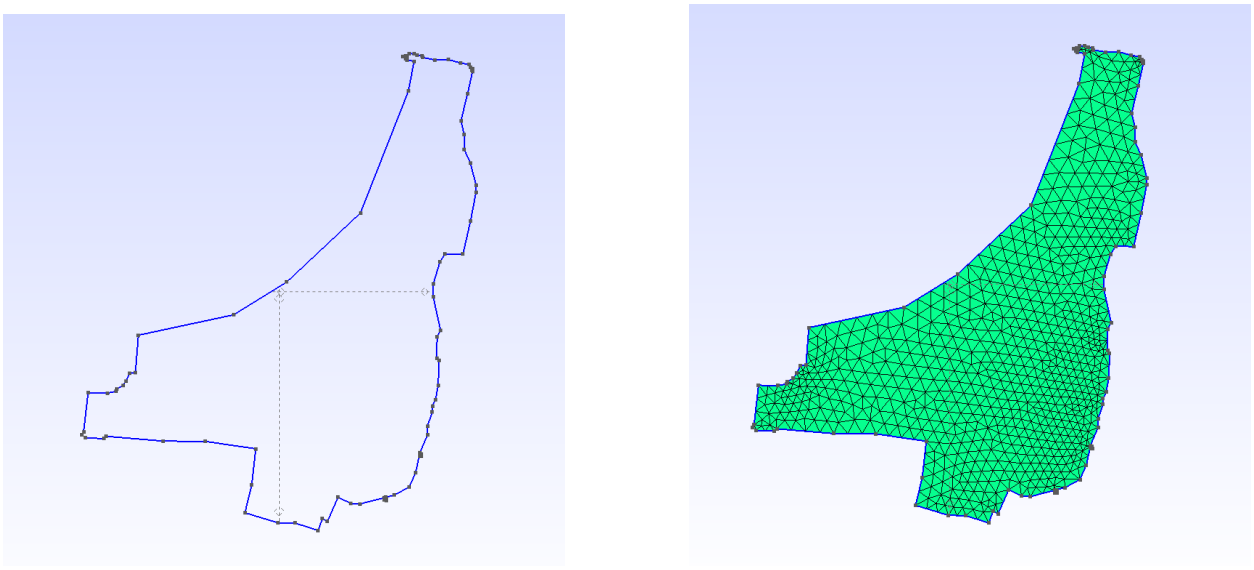
2 Solution

2.1 Points Generation

2.1.1 Using Gmsh

Using Gmsh we can plot points and join them using lines. This forms a closed loop which can then be converted to a plane surface. Meshing can be done after this. Here are the steps followed:

- Open the image of the IITM Map on gmsh
- Manually sample points along the boundary of this map, sample more points where the geometry deviates from a straight line structure
- Join these points in order using Line function. This will form a closed boundary of the map



(a) 88 Points sampled along the boundary (b) Triangular mesh

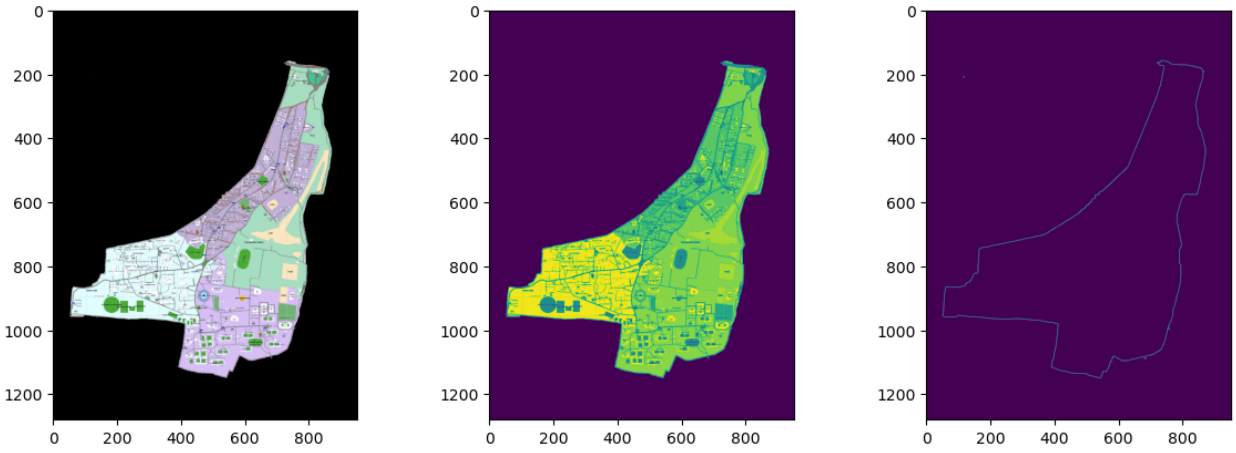
Figure 2: Mesh Generation steps

2.1.2 Attempt using opencv

OPENCV is a python package used for image processing. It has several features like edge detection, gray-scale conversion, background removal etc.

Ideal Roadmap :

- Remove the background of the image and let the map be the only coloured object in the image.
- Convert the image to gray-scale and then make the image binary (having only 2 colours). This will be ideal for edge detection for detecting the map boundary
- Perform Canny edge detection using `cv2.canny(image)` to detect the boundary
- Sample points along the boundary of the image



(a) Image 1 Caption (b) Image 2 Caption (c) Image 3 Caption

Figure 3: Overall caption for all three images

Convert .geo file to .gmsh file using Gmsh. Choose ASCII 2 format.

2.2 The .gmsh file

The '.gmsh' file format is a common format for representing mesh data. It typically consists of several sections, including nodes and elements, which are essential for defining the geometry and topology of the mesh. Below, we explain the structure of a '.gmsh' file using an example.

2.2.1 Nodes Section

The 'Nodes' section contains a list of all the points in the mesh. Each node is defined by an ID and its coordinates in 3D space. The format is as follows:

```
$Nodes
<number_of_nodes>
<node_id> <x_coordinate> <y_coordinate> <z_coordinate>
...
$EndNodes
```

For example:

```
$Nodes
1494
1 742.7 1103.3 0
2 730.9 1041.3 0
3 630.7 787.6 0
4 476.9 643.9 0
5 365.8 575.6 0
6 167.1 533.2 0
7 161 455.2 0
```

...

$EndNodes

In this example:

- The first line '$Nodes' indicates the start of the nodes section.
- The number '1494' specifies the total number of nodes.
- Each subsequent line lists a node's ID followed by its 'x', 'y', and 'z' coordinates.
- The section ends with the line '$EndNodes'.

2.2.2 Elements Section

The 'Elements' section defines the connectivity between nodes, forming the mesh elements (Boundary points, Boundary Points, Triangles). The general format is:

```
$Elements
<number_of_elements>
<element_id> <element_type> <tag1> <tag2> ... <node1> <node2> ... <nodeN>
...
$EndElements
```

For example:

$Elements

3219

1 15 2 0 1 1

2 15 2 0 2 2

```
3 15 2 0 3 3
4 15 2 0 4 4
...
$EndElements
```

Here:

- The first line '$Elements' marks the beginning of the elements section.
- '3219' indicates the total number of elements.
- Each line describes an element:
  - The first number is the element ID.
  - The second number specifies the element type (e.g., 1 for a line, 2 for a triangle).
  - The remaining numbers indicate node IDs that form the element.
- The section concludes with the line '$EndElements'.

2.3 The .vtk file

The '.vtk' file format is used for storing mesh data in a format compatible with visualization tools like ParaView. The file typically consists of several sections, including headers, points, and cells (elements). Below, we explain the structure of a '.vtk' file using an example.

2.3.1 Points Section

The 'POINTS' section lists all the nodes (points) in the mesh. Each point is defined by its coordinates in 3D space. The format is as follows:

```
POINTS <number_of_points> <data_type>
<x_coordinate> <y_coordinate> <z_coordinate>
...
```

For example:

```
POINTS 1494 double
742.7 1103.3 0
730.9 1041.3 0
630.7 787.6 0
476.9 643.9 0
365.8 575.6 0
167.1 533.2 0
161 455.2 0
...
```

In this section:

- 'POINTS 1494 double' indicates that there are 1494 points, and their coordinates are stored as double-precision floating-point numbers.
- Each subsequent line specifies the 'x', 'y', and 'z' coordinates of a point.

2.3.2 Cells Section

The 'CELLS' section defines the elements by listing the nodes that form each element. The general format is:

```
CELLS <number_of_cells> <total_number_of_indices>
```

```
<size> <node1_id> <node2_id> ... <nodeN_id>
```

...

For example:

```
CELLS 3219 12553
1 0
1 1
1 2
1 3
...
```

Here:

- 'CELLS 3219 12553' indicates that there are 3219 cells, and the total number of indices (including the size of each cell) is 12553.
- Each line describes a cell:
  - The first number is the number of points that make up the cell (e.g., '1' for a point, '2' for a line).
  - The remaining numbers are the node IDs that form the cell.

2.3.3 Example Summary

The '.vtk' file format provides a structured way to store mesh data, including the coordinates of points, the connectivity of those points to form cells, and the types of those cells. This information is essential for visualizing the mesh in tools like ParaView.

2.4 Conversion of .gmsh to .vtk and reading mesh information

The conversion from '.gmsh' to '.vtk' involves translating mesh data from the GMSH file format to the VTK file format to facilitate visualization with tools such as ParaView. The process includes the following steps:

2.4.1 Parsing the .gmsh File

The '.gmsh' file is read to extract mesh information. Key sections include:

- **Nodes:** Contains node IDs and their coordinates.
- **Elements:** Defines elements by specifying their type and connectivity (i.e., which nodes form each element).

2.4.2 Formatting for .vtk

The extracted data is then formatted for the '.vtk' file format:

- **POINTS:** Lists node coordinates.
- **CELLS:** Specifies cell connectivity. Each cell type (e.g., vertex, line, triangle) and its corresponding node indices are included.
- **CELL_TYPES:** Enumerates the type of each cell using VTK conventions.

2.4.3 Writing the .vtk File

A new '.vtk' file is created with the following sections:

- **HEADER:** Metadata about the file.
- **POINTS:** Node coordinates in a 'double' precision format.

- **CELLS:** Connectivity data specifying which nodes make up each cell.
- **CELL_TYPES:** Type of each cell, such as tetrahedron or hexahedron.

2.5 Paraview - Result

The .vtk file is rendered using Paraview

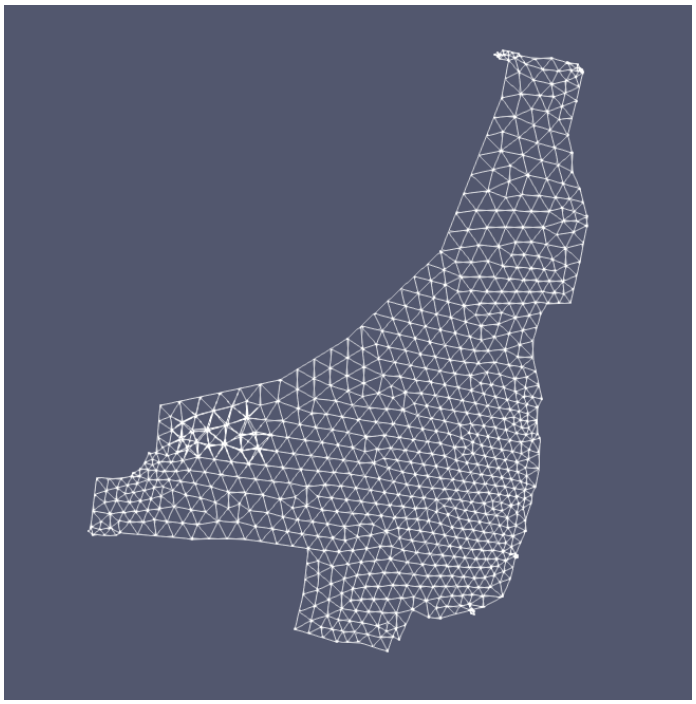


Figure 4: Paraview wireframe model.

****