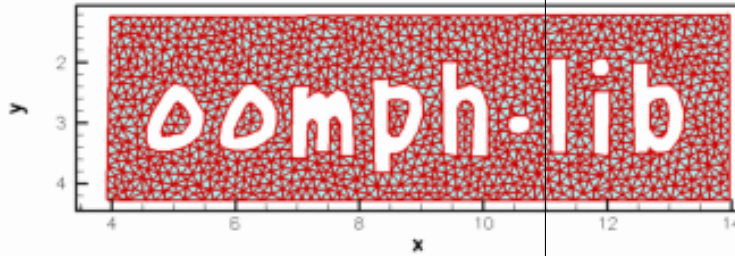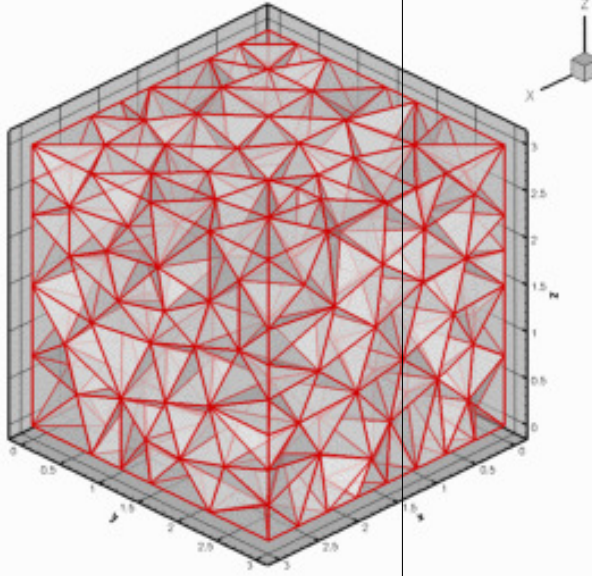# Chapter 1

# Existing (unstructured) meshes: Wrappers to third-party mesh generators

`oomph-lib` does not provide its own unstructured mesh generator but has several mesh classes that generate unstructured meshes from the output of third-party unstructured mesh generators.

**Notes:**

1. The unstructured tet and triangle meshes listed below can **not** be used with `oomph-lib's` mesh adaptation or node-update procedures. A suitably fine mesh has to be generated offline by the third-party mesh generator. If required, node-updates (in response to changes in the domain boundaries) have to be performed manually.

2. For some element types, the mesh generation process is not particularly efficient (yet!). A suitable warning message is issued in such cases.

3. Since the third-party mesh generators tend to triangulate the domain with simplex elements, curvilinear boundaries are not resolved more accurately by using higher-order elements unless some post-processing is performed.

4. The meshes have not been tested as extensively as `oomph-lib's` structured meshes, described `elsewhere.`
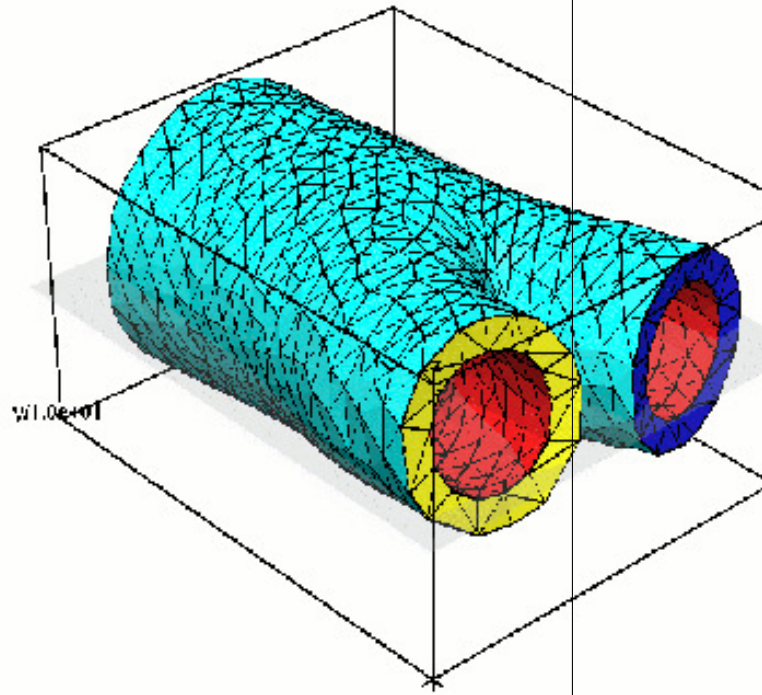
   .

## 1.1   Mesh list

| Mesh | Representative Mesh plot |
|------|--------------------------|
| **TriangleMesh**<**ELEMENT**> <br><br> • This class creates `oomph-lib` meshes based on the output from  `J.R.Shewchuk's` Delaunay mesh generator  `Triangle` <br><br> • The mesh can be used with all `Finite`↩ `Elements` that are derived from the geometric finite element `TElement<2,NNODE_1D>`. <br><br> **Example driver codes:** <br><br> • The use of  `Triangle` and the `Triangle`↩ `Mesh` class are explained in a  `separate tutorial.` <br><br> • In  `another tutorial` we demonstrate how the code  `fig2poly.cc` may be used to generate input files for  `Triangle` based on the output from the open-source drawing program  `xfig.` |  |
| **TetgenMesh**<**ELEMENT**> <br><br> • This class creates `oomph-lib` meshes based on the output from  `Hang Si's` open-source mesh generator  `Tetgen` . <br><br> • The mesh can be used with all `Finite`↩ `Elements` that are derived from the geometric finite element `TElement<3,NNODE_1D>`. <br><br> **Example driver codes:** <br><br> • The use of  `Tetgen`  and the `Tetgen`↩ `Mesh` class are explained in a  `separate tutorial.` |  |

**Generating meshes from medical scans with VMTK**

- We provide the option to generate tetgen-based meshes for physiological fluid-structure interaction problems, using the Vascular Modeling Toolkit (VMTK).

**Example driver codes and tutorials:**

- We provide a separate tutorial that shows how to generate oomph-lib meshes from medical images.

- The methodology is used in the following driver codes:

    - The inflation of a blood vessel.

    - Finite Reynolds number flow through a (rigid) iliac bifurcation.

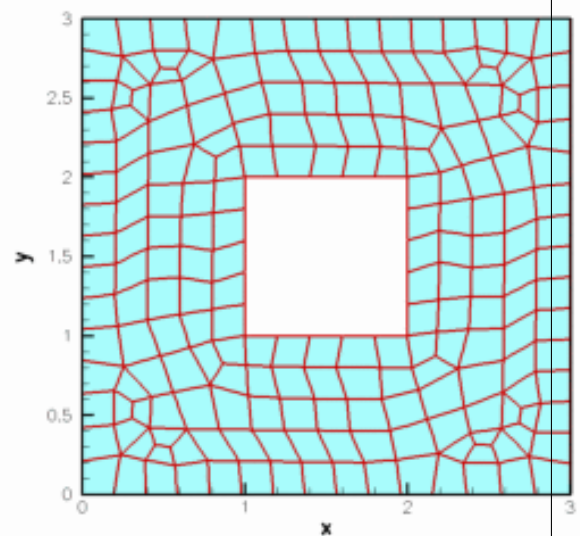    - Finite Reynolds number flow through an elastic iliac bifurcation.



**GeompackQuadMesh<ELEMENT>**

- This class creates oomph-lib meshes based on the output from Barry Joe's mesh generator Geompack++, available as freeware at http://members.shaw.ca/bjoe/.

- The mesh can be used with all Finite↩ Elements that are derived from the geometric finite element QElement<2,2>.

**Example driver codes:**

- The use of Geompack++ and the GeompackQuadMesh class are explained in a separate tutorial.

## 1.2   PDF file

A `pdf version` of this document is available. \