

Meshing

Fluidity training workshop

Applied Modelling and Computation Group

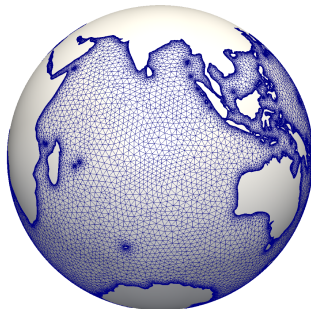
Alexandros Avdis

Department of Earth Science and Engineering, Imperial College London

5-7 November 2014

Tutorial overview

- ▶ What is a mesh.
- ▶ What is Gmsh.
- ▶ Viewing and meshing a 3-D geometry.
- ▶ Generating and meshing a 2-D geometry.
- ▶ Meshing realistic domains.



but, before we start...

- ▶ This is a hands-on session, we need to do a bit of house-keeping.
- ▶ Open a new terminal (Ctrl+Alt+t) :
 - We will be using this terminal throughout this tutorial.
 - \$ **text** signifies commands to be typed into the terminal.
 - \$ **mkdir mesh**
to create a directory for this tutorial
 - \$ **cd mesh**
to “enter” the newly created directory
- ▶ Fetch & open a copy of the present slides:
 - From GitHub (under meshing):
<http://fluidityproject.github.io/training/>

A bit more house-keeping.

- ▶ Keep the slides open in a window.
 - You can copy-paste lengthy commands & statements.
 - You will be asked to click on links in the slides.
 - Feel free to progress at a faster pace than the speaker.
 - Feel free to put your hand up and ask for help!
- ▶ The slides, as well as a more detailed tutorial, are freely available.
 - Slides: via [Figshare](#) or [GitHub](#)
 - Tutorial document: via [Figshare](#)

What is a mesh?

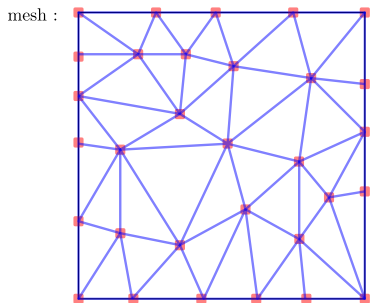
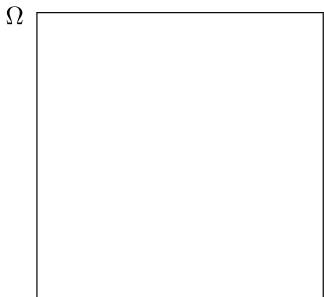
A mesh can be qualitatively thought of as the tessellation of a domain Ω into a set of non-overlapping sub-domains ω_i :

$$\begin{aligned}\Omega &= \cup \{ \omega_i \mid i = 1, 2, \dots, ele \} \\ \emptyset &= \cap \{ \omega_i \mid i = 1, 2, \dots, ele \}\end{aligned}\tag{1}$$

where ele is the number of elements in the tessellation.

What is a mesh?

A mesh can be qualitatively thought of as the tessellation of a domain Ω into a set of non-overlapping sub-domains ω_i :



What is Gmsh?

- ▶ It is the role of the mesh (or grid) generator to scatter the points and generate the mesh, whilst ensuring high quality elements.
- ▶ Gmsh is a “3D finite element grid generator with a build-in CAD engine and post-processor. Its design goal is to provide a fast, light and user-friendly meshing tool with parametric input and advanced visualization capabilities.”¹. Furthermore, Gmsh can be used as a 1–, 2– and 3– dimensional mesh generator for use with the Fluidity CFD code.
- ▶ Not the only mesh generator that can be used with Fluidity.
 - ▶ Fluidity can also read meshes in ExodusII format.
- ▶ Distributed under the GNU General Public License, available for Linux, Windows and Mac OS.

¹from <http://www.geuz.org/gmsh/>

Fetching an example

- ▶ Lets fetch a Gmsh file and open it.
- ▶ Download a simple example file:
 - A torus!
 - Point your browser to [here](#)
 - Look at the file contents, it is a geometry description.
 - Click on “Download”.
- ▶ Go back to your terminal
 - \$ `mv $HOME/Downloads/torus.geo .`
to move the downloaded file into the working directory

Starting Gmsh

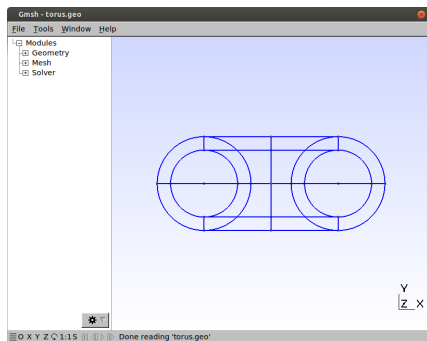
- ▶ Open the file, with Gmsh:

\$ **gmsh** **torus.geo** &

The ampersand is important.

- ▶ The Gmsh window is composed of two panels:

- A menu panel, a tree-like structure (left).
- The Graphic area (larger, right).
- A status bar at the bottom.



Navigating menus.

- ▶ Gmsh's architecture is centred around four modules, this is reflected in the menu panel.
- ▶ The menu panel can be used to switch between the different modules.
 1. Geometry : For defining domain geometry.
 2. Mesh : For building the mesh.
 3. Solver.
 4. Post-Processing : More visible in older versions of Gmsh.

Manipulating the view

- ▶ Panning : Hold right button down and move cursor.
- ▶ Zooming : Scroll or hold middle button down and move cursor.
- ▶ Rotating : Hold left button down and move cursor.
- ▶ **Practical** : Try modifying the view.

Meshing and saving the mesh

- ▶ Mesh the torus:
 - Mesh > 3D
 - Once again, try modifying the view.
- ▶ To save the mesh click on File (menu window) and select Save Mesh.
 - This creates a file “torus.msh”, storing the mesh.
- ▶ More information on the torus is available in the tutorial document fetched earlier.
 - Including instructions how to build the geometry.

The various files

- ▶ torus.geo : Stores the geometry as an ASCII script file.
 - You have already seen the contents, when downloading.

\$ **cat torus.geo**

To list the file lines (catenate) in your terminal

- Possible to write this file from scratch with a text editor.
- Possible to use the Gmsh GUI to create this file.

- ▶ torus.msh : Stores the mesh
 - Also contains tags (numerical ID) on element vertices, edges and faces that we can use to assign boundary conditions.
 - Can be ASCII or binary.

\$ **cat torus.msh**

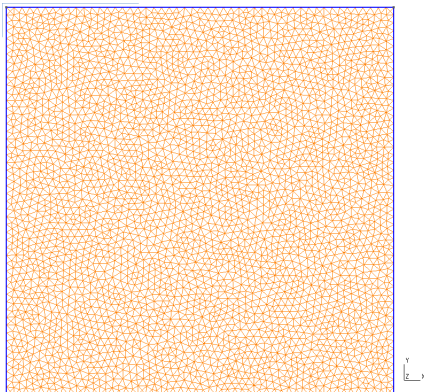
The file we generated here is ASCII

Mesh file formats

- ▶ Fluidity can read meshes in gmsh format.
- ▶ Fluidity can also read meshes in triangle format.
 - The gmsh2triangle utility is part of the Fluidity distribution and can convert gmsh to triangle format:
\$ gmsh2triangle torus.msh
Generates torus.ele, torus.face, torus.node
\$ ls -l
to list the files.
\$ gmsh2triangle
Shows usage and options.
- ▶ Fluidity can also read meshes in ExodusII format.
- ▶ Work is on-going on PETSc DMplex capability.

Gyre example: Aim

- ▶ To generate a 2-D mesh on a $1,000\text{km} \times 1,000\text{km}$ square.
- ▶ Using the Gmsh GUI.
- ▶ Typical element size (edge length): 20km .
- ▶ Mesh will be used in subsequent simulations/examples.



Gyre example: Getting started

- ▶ Close your existing Gmsh instance.
- ▶ Download the solution file, to help you & tutors trace mistakes.
 - Download from [here](#)
 - \$ `mv $HOME/Downloads/gyre-example.geo .`
- ▶ Open a new Gmsh instance, on a new file:
 - \$ `gmsh gyre.geo &`
The ampersand is important.

We proceed by defining:

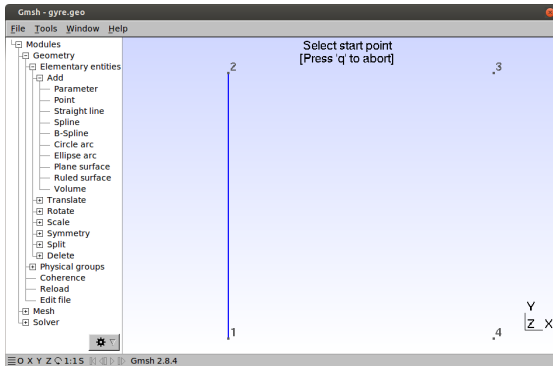
1. Points
2. Lines
3. Surfaces
4. in 3D cases, volumes

Gyre example: Creating points

- ▶ Geometry > Elementary Entities > Add > Point
- ▶ The **Contextual Geometry Definitions** window will appear.
- ▶ Enter the point coordinates and click “Add”.
 - Do not move cursor outside **Contextual Geometry Definitions** window while entering coordinates! Hold shift down if you have to.
 - Always look at the instructions shown in the graphic window.
 - Point 1: [0.0 , 0.0 , 0.0], Prescribed mesh element size **2e4**
 - Point 2: [0.0 , 1.e6, 0.0], Prescribed mesh element size **2e4**
 - Point 3: [1.e6, 1.e6, 0.0], Prescribed mesh element size **2e4**
 - Point 4: [1.e6, 0.0 , 0.0], Prescribed mesh element size **2e4**
- ▶ Press ‘q’ and close Contextual Geometry Definitions window.

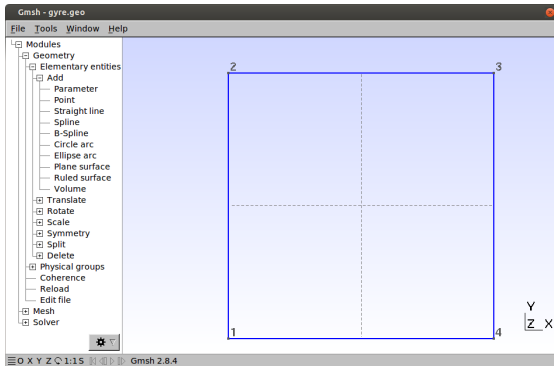
Gyre example: Creating lines

- ▶ Geometry > Elementary Entities > Add > Straight Line
 - Draw a line between two points by selecting the points.
 - Join points (1,2), (2,3), (3,4), (4,1)
 - Once all lines are drawn, press 'q'
 - Always look at the instructions shown in the graphic panel.



Gyre example: Declaring a plane surface.

- ▶ Geometry > Elementary Entities > Add > Plane Surface
- ▶ Click on any of the sides, all sides will be highlighted.
- ▶ Press 'e' then 'q'.
- ▶ Gmsh will highlight the surface with grey, dash-lines.



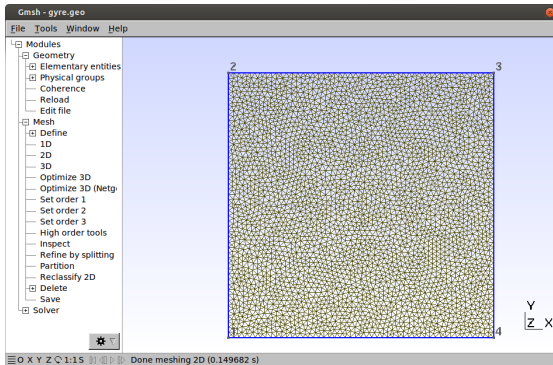
Gyre example: Declaring physical groups.

In order to specify regions and boundaries in Fluidity, they must first be defined as “Physical Groups” in Gmsh:

- ▶ Assign “Physical Line” ID’s to the domain boundaries.
 - Geometry > Physical Groups > Add > Line
 - Select **bottom** side and press ‘e’.
 - Select **right** side and press ‘e’.
 - Select **top** side and press ‘e’.
 - Select **left** side and press ‘e’.
 - Once you have done all sides press ‘q’.
- ▶ Assign “Physical Surface” ID to the plane surface.
 - Geometry > Physical Groups > Add > Surface
 - Select the highlighted surface: Click on the grey dash lines.
 - Press ‘e’ then ‘q’.

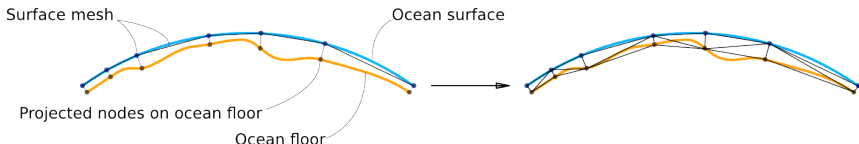
Gyre example: Producing a mesh

- ▶ Produce a 2-D mesh: Mesh > 2D
- ▶ Save the mesh: Click on File and select “Save Mesh”.
- ▶ Convert to triangle format:
\$ `gmsh2triangle -2 gyre.msh`



Meshes for realistic ocean domains & Fluidity

- ▶ Mesh is constructed on a reference surface.
 - A spherical shell, Earth's surface geoid.
 - A flat surface, a cartographic projection datum (e.g. a UTM zone).
- ▶ User can choose to set-up 2D or 3D simulations
 - 2D, suitable for regional shallow water (not currently on-the-sphere).
 - In 3D no need to do three-dimensional mesh generation.
 - Mesh is “vertically extruded” within Fluidity.



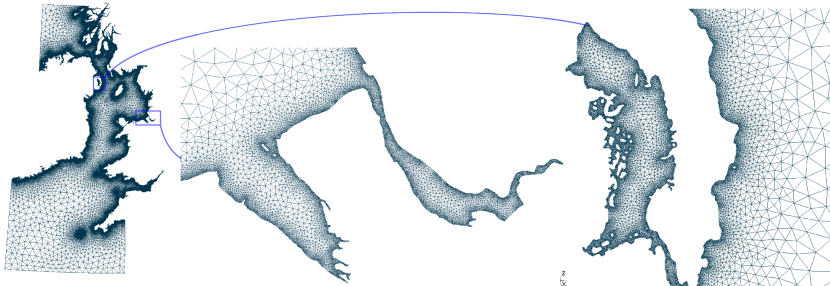
Meshing realistic ocean domains: Key points

- ▶ A Gmsh user typically has to specify two essential parts:
 - Domain shape.
 - Characteristic element size.
- ▶ The geometry is very complex, boundaries are fractal-like
- ▶ Ideal characteristic element size can also be dependent on many parameters, for example:
 - Depth
 - Ocean floor topography
 - Explicit requirement in order to resolve tidal turbine array, etc.

Meshing realistic ocean domains: Challenges

A simple CAD engine is insufficient:

- ▶ Shorelines are geometrically very complex.
- ▶ The Gmsh GSHHS plug-in fits a spline through the GSHHS points, leading to intersecting shorelines (support for Gmsh GSHHS plugin now dropped).
- ▶ Drawing arbitrary lines as open boundaries –e.g. contour at a given depth– not easily done.



More challenges

The spatial variation of the mesh element size could be complex:





- ▶ Mesh element size must usually be fine near the coastlines to capture their structure.
- ▶ Mesh element size must usually be fine in areas of steep ocean floor topography and in shallow areas: Not easily done with current approach.

Meshing realistic domains – Our proposed approach

Use Geographical Information Systems to extract domain boundaries and prescribe mesh metric size.

- ▶ Existing GIS software capable of reading databases in popular formats.
- ▶ Capability of extracting contours from field-type databases is usually available. (→domain boundaries)
- ▶ Capability of generating field-type databases is usually available. (→mesh size metric)

Further reading

-  AMCG, **The present slides**, accessible via [GitHub](#) and [Figshare](#).
-  C. Geuzaine and J.-F. Remacle, **Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities**. International Journal for Numerical Methods in Engineering, Volume 79, Issue 11, pages 1309-1331, 2009.
-  C. Geuzaine and J.-F. Remacle, **Gmsh Reference Manual**.. Available at <http://geuz.org/gmsh/#Documentation>.
-  A. Avdis and S.L.Mouradian, **A Gmsh tutorial**, accessible via [Figshare](#)

Questions?

AMCG:

<http://amcg.ese.ic.ac.uk/>

Fluidity:

<http://fluidityproject.github.io/>

Fluidity code on GitHub:

<https://github.com/FluidityProject/fluidity>

<https://github.com/FluidityProject>

Fluidity wiki:

<https://github.com/FluidityProject/fluidity/wiki>