The Durham gmsh Tutorial

Introduction

This provides a short step by step guide to meshing an aerofoil using the ***gmsh*** package. The aim being to introduce you the important features of the program in the shortest possible space of time. The mesh that this tutorial produces will require further work to be used as a basis for serious calculations but this tutorial contains all the information that you need to do this.

gmsh is a free (as in freedom) software package available from http://geuz.org/gmsh/ Although there are Windows and Mac versions available since OpenFOAM only runs on GNU/Linux this tutorial assumes that you have gmsh installed on a GNU/Linux system.   Once you have logged into your system you can start the program from the command line using the command gmsh**.**

Hint: GNU/Linux is an operating system that programs such as gmsh run upon. It is widely used from supercomputers to mobile devices and is installable on most laptops and desktops. My recommendation is a version called Ubuntu available from http://www.ubuntu.com/ at zero cost. The tutorials were carried on Ubuntu 12.04 but any other distribution should work just fine.

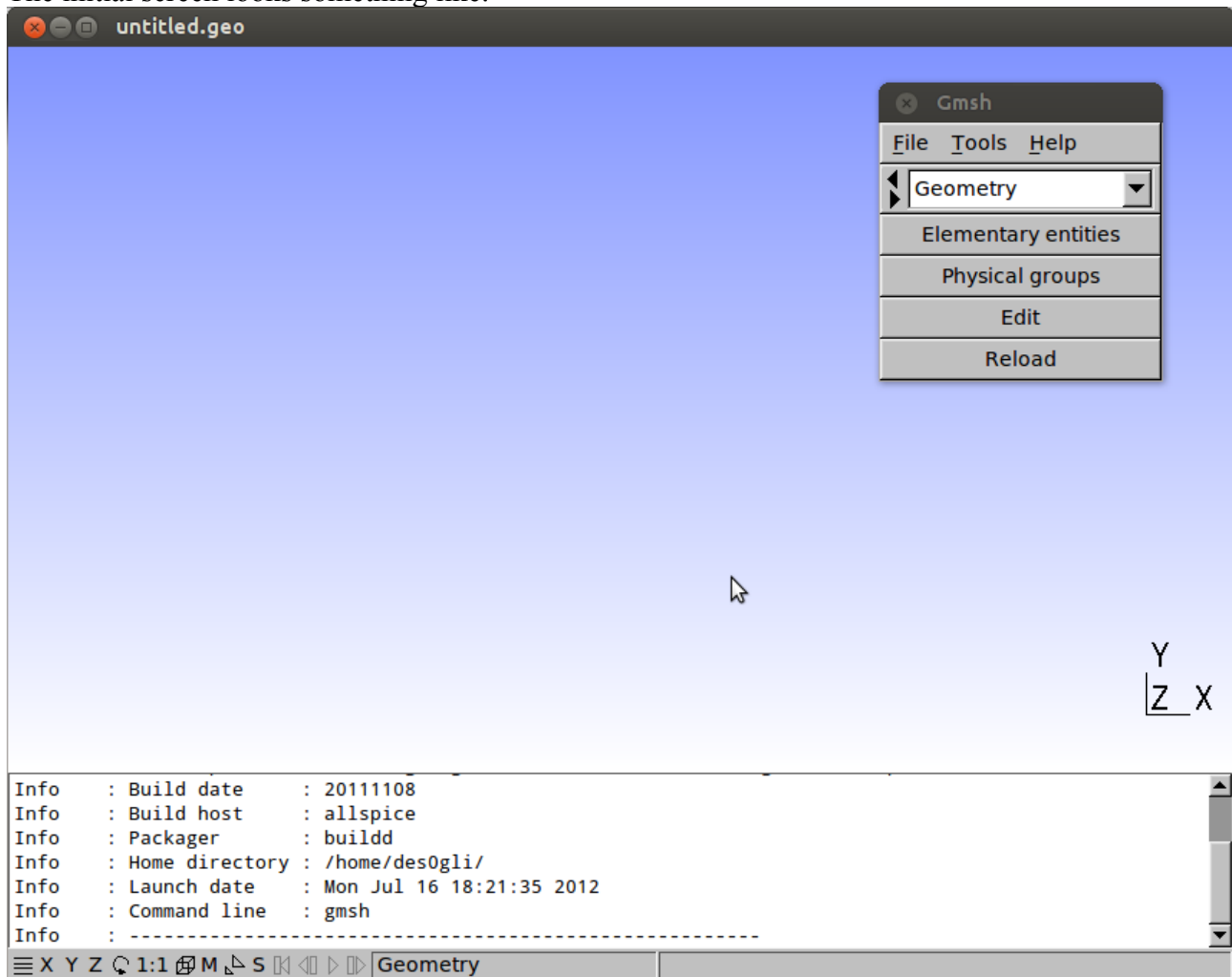The initial screen looks something like:

Figure 1: The gmsh screen.

There are two windows displayed, a viewing area where the results of meshing operations are

displayed and a window containing menus for operating the program. In practice it is often easer to edit the gmsh file directly (it is just plain text) and this is what we will do in this tutorial for a number of steps.

> **Hint:** A good text editor to use is called *gedit* and in an Ubuntu 12.04 installation as the default text editor. The gmsh interface has a reload button which instantly updates the screen with your edits.

Importing the Geometry

- Obtain the NACA5012 coordinates file (this should be on the same page/folder where you obtained this document)

- This gives you a set of coordinates for the aerofoil – we can input these into gmsh one by one, but that is a little time consuming. Currently there is no way of importing a point cloud into gmsh and drawing a line through it so we need to do this ourselves.

- The data file consists of x,y and z coordinates for each point that defines the aerofoil. There is one line per point and the points are tab separated so the file looks like:

```
0.99591    0.00228    0.00000

0.98332    0.00530    0.00000
```

- We need to convert this to the gmsh format for point again is one point per line and looks something like:

```
Point(1000) = { 0.99975300, 0.00129500, 0.00000000, 0.01};
```

  The fourth number is a characteristic length which controls some details of the meshing.

- There are a number of ways of getting these points in the *gmsh* format, you could write a script in Python or MATLAB or use a spreadsheet with the concatenate function to create an appropriate block of text. An example script in python is provided for download with this document.

- To convert the points to gmsh format, download the `dat2gmsh.py` file and the `naca5012xyz.dat` files and copy these files into your working directory.

- Open the terminal and then in the working directory type:

```
python dat2gmsh.py naca5012xyz.dat
```

- The script then converts the points in the `naca5012xyz.dat` file into a *gmsh* set of points and then put a spline through them. The output looks something like Figure 2, to obtain something like this figure run the command:
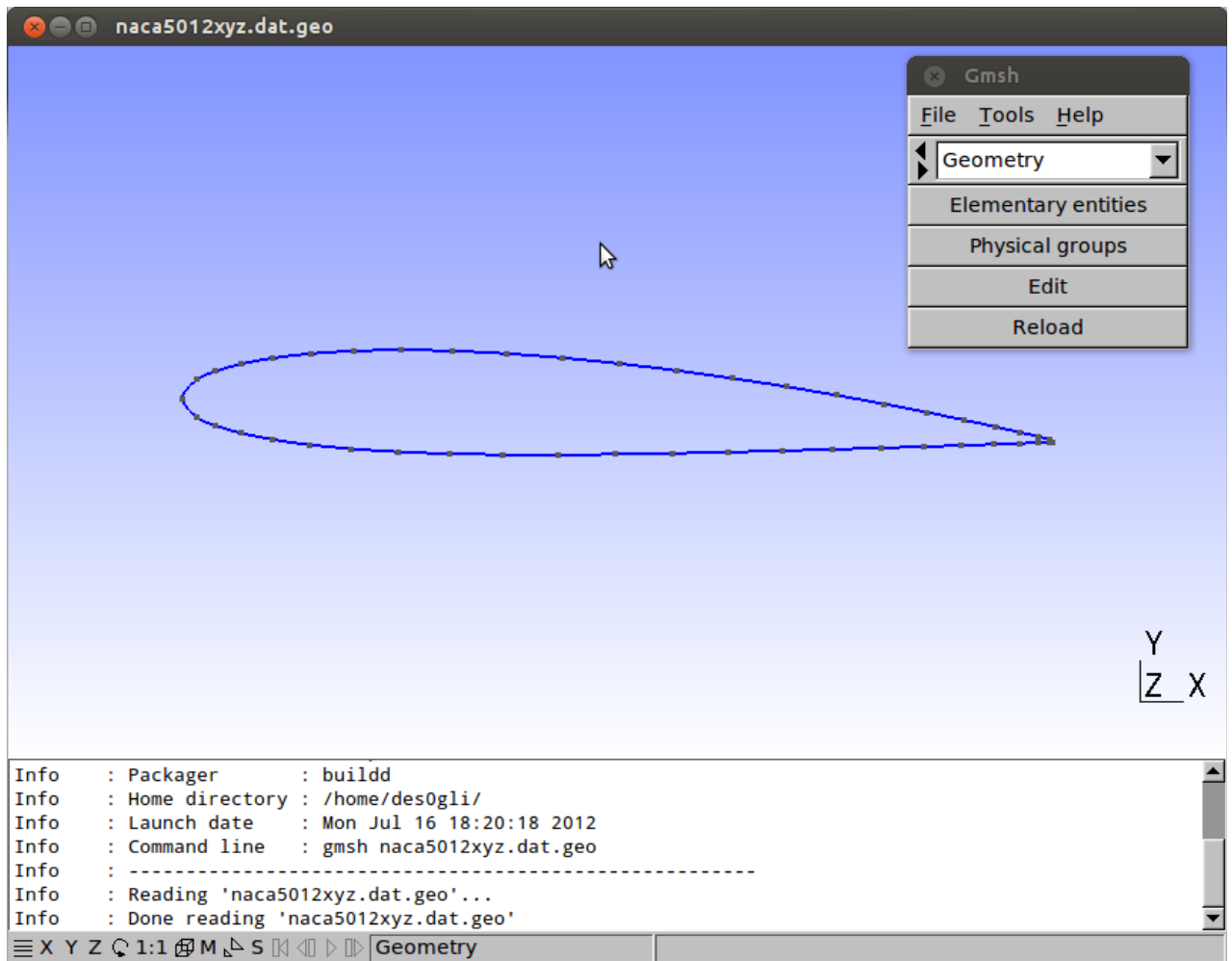
```
gmsh naca5012xyz.dat
```

Figure 2: Points Converted to gmsh format

- If you zoom in closely at the trailing edge you will note that the aerofoil has an open trailing edge. This is simply a consequence of how the points are arranged in the input file. What needs to be done next is to edit the *gmsh* text file so a trailing edge is added. A commented *gmsh* file called `naca5012_step1.geo` is provided with this download. This adds an additional point and then puts a spline through the complete set of points.

Hint: A second way to get this geometry is to use the graphical user interface. Generate an additional point (Geometry → Elementary entities → Add → New → Point) and connecting the points with a B-spline (Geometry → Elementary entities → Add → New → B Spline) *gmsh* will then automatically change the .geo file when you hit save.

Setting up the Boundaries

There are a number of strategies that can be used to mesh aerofoils, for example a "C" mesh can be used or a circular mesh. Here we simply place the aerofoil in a large box. The aerofoil coordinates have a unit length for chord and to make the calculation simple, we put in four points four axial chord lengths above and below the aerofoil and five axial chord lengths forward and back.

- To do this you then need to add various points to the geo file, such as:

```
Point(1100) = { 5, 4, 0, 0.1};
```

- The points, then need to made into lines:

```
Line(1) = {1100,1101};
```

- Creating a surface in *gmsh* is quite straightforward – even if the surface has holes in it. We need to inform *gmsh* that the lines represent boundaries using the Line Loop command. Then the surface is created using the Plan Surface command.

```
Line Loop (1) = {1,2,3,4};   and   Plane Surface(1) = {1,2};
```

- A full example is found in `naca5012_step2.geo` the result of which should look like Figure 3 below. Only a selection of the edits made are detailed above – full details are in the file.
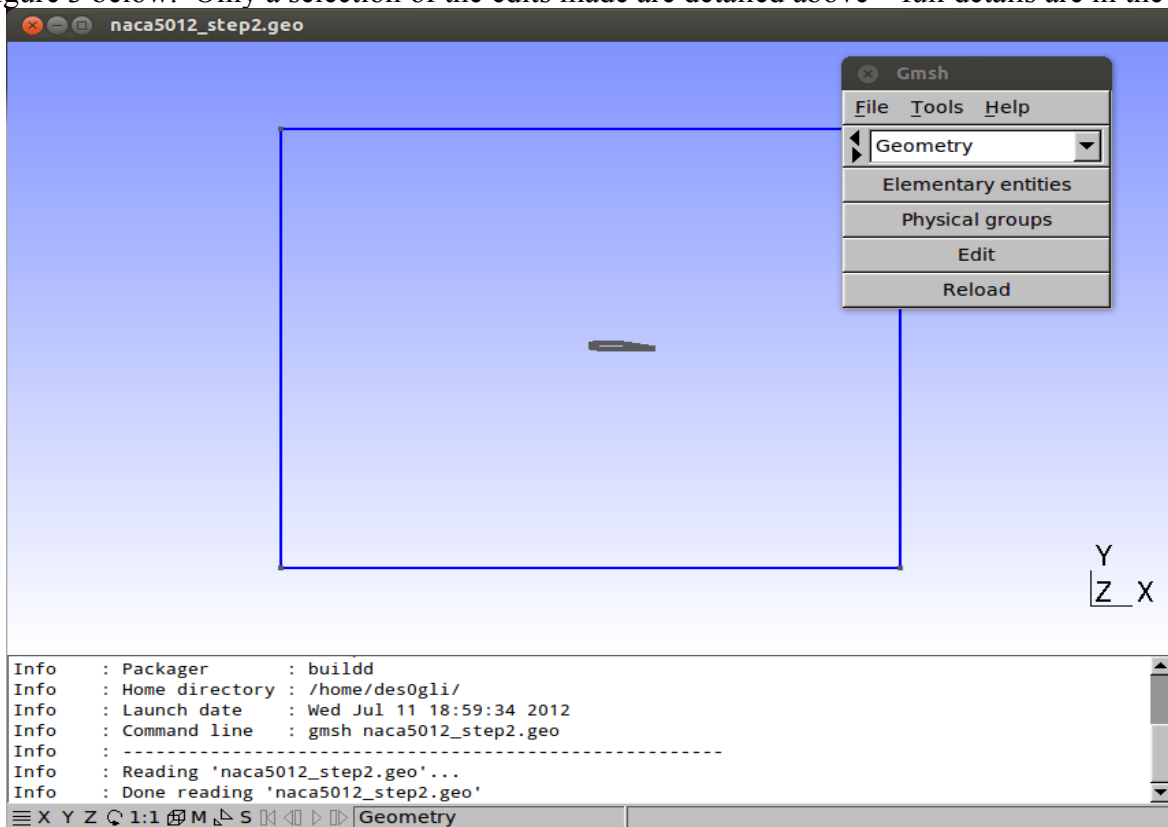

Figure 3: The Geometry Fully Defined

Extruding into 3D and Labelling the Surfaces

We intend to use this mesh with the OpenFOAM solver which requires three dimensional elements even if the mesh is 2D. We use an Extrude command to extrude the surface into a volume and then a couple of commands to ensure that the mesh will only be one element thick.

```
Extrude {0, 0, 1} {

Surface{1};

Layers{1};

Recombine;

}
```

In order to use the various faces in OpenFOAM later we need to highlight them as Physical Surfaces and give them a name. The choice of names is important as these will be used when we run OpenFOAM. The various edges are given names as follows:

```
    Physical Surface("back") = {1027};

    Physical Surface("front") = {1};

    Physical Surface("top") = {1022};

    Physical Surface("exit") = {1010};

    Physical Surface("bottom") = {1014};

    Physical Surface("inlet") = {1018};

    Physical Surface("aerofoil") = {1026};
```

Finding the number associated with a particular surface is a bit of a pain. The menu commands Physical Groups → Add → Surface will add the surfaces to the file in question but then you need to add the names manually afterwards. A similar process is required for the Volume:

```
    Physical Volume("internal") = {1};
```

This doesn't introduce any changes in the graphical interface but these steps are important for later. The additional changes can be seen in `naca5012_step3.geo`

Meshing the aerofoil

This is really straightforward just press "3" (to mesh in 3D) and ***gmsh*** will work its magic....

Hint: This tutorial just touches the surface of what you can do with gmsh, vast amounts of information about the program can be found online:

http://geuz.org/gmsh/

A lot of the steps in this tutorial follow on from "aeroslacker" on CFD-Online:

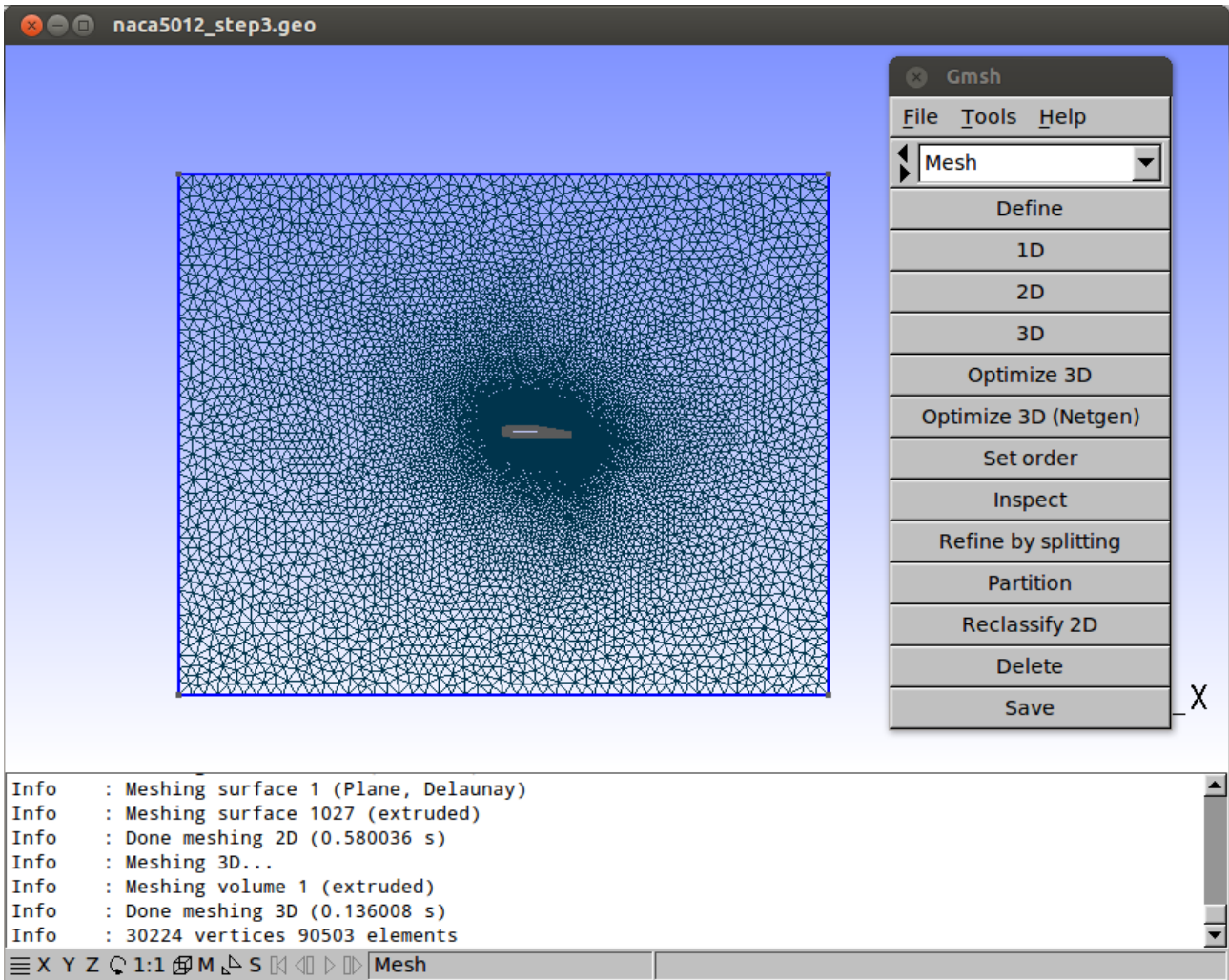http://openfoamwiki.net/index.php/2D_Mesh_Tutorial_using_GMSH

Figure 4: The Meshed Aerofoil

You will note that the mesh has more points near the surface and fewer near the boundaries. This is controlled by the characteristic length of the various points which we set to be 0.005 for the aerofoil surfaces and 0.2 for the edges. You can see this in the file `naca5012_step3.geo` where we have also used variables for these values so that we can change them with a single edit.

Exporting the mesh

The final step here is to export the mesh. To this just push the Save button and a .msh format mesh file will be produced. This will then need to be copied to the OpenFOAM directory to be converted to that format.

Grant Ingram

Wednesday 14 November 2012