AM119: Yet another OpenFoam tutorial

Prof. Trask

April 11, 2016

1 Todays project

Today we're going to implement a projection method for the Navier-Stokes, learn how to build a mesh, and explore the difference between high and low Reynolds number flows.

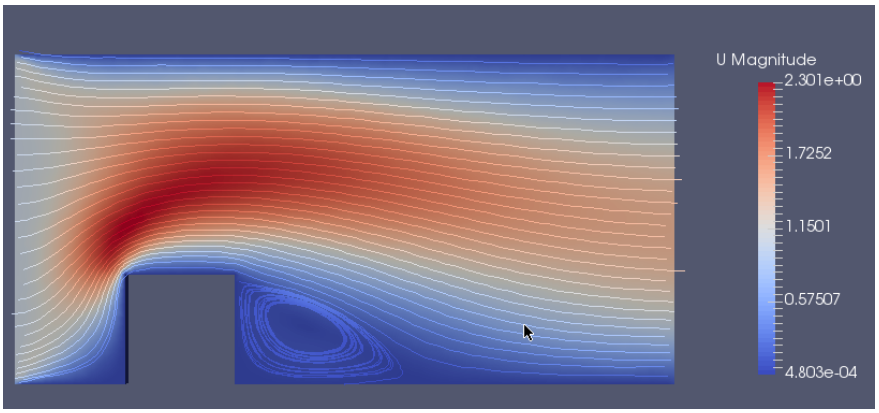


Figure 1: The final product

2 From advectionDiffusionFoam/implicitAdvectionDiffusionFoam/eulerEquationFoam to navierStokesFoam

At this point you know the drill for how to make your own solver. We're going to do it again - this time to make an incompressible Navier-Stokes solver. The steps to this will be:
In createFields.h:

- Read in the viscosity parameter
- Read in the timestep size to use in the pressure correction step.
- Initialize velocity and pressure fields

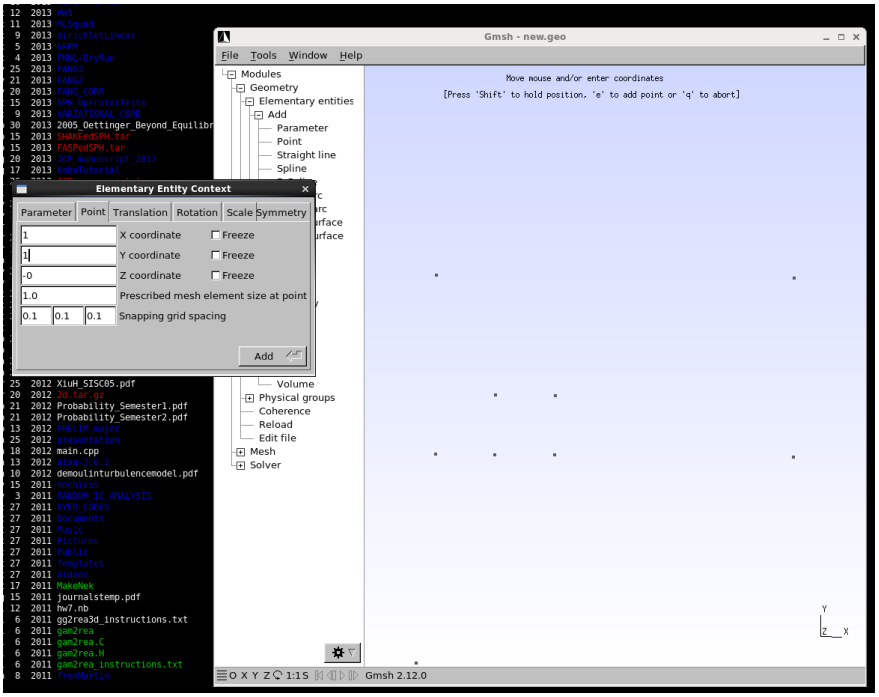
In navierStokesFoam.C:

- Solve the momentum equation - this will look like the momentum equation from your Euler equations solver but without a pressure gradient, with an implicit viscous term (like last weeks lab), and with implicit advection.
- Recalculate the flux *fvc::interpolate(U)&mesh.Sf()*
- Solve the Poisson equation. Approximate $\frac{\nabla \cdot u}{\Delta t}$ as *fvc::div(phi/dt)*.
- Update the velocity $u^{n+1} = u^* - \nabla p$ using u = *fvc::reconstruct(phi)* - *fvc::grad(p)*dt*

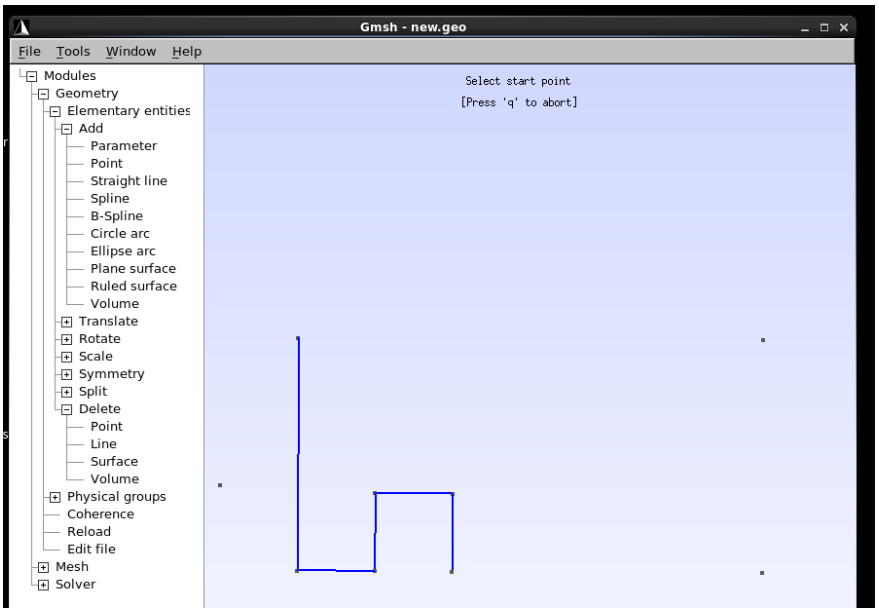
You should be able to rerun your channelFlow case now and produce an identical result.

3 Make a mesh in gMsh

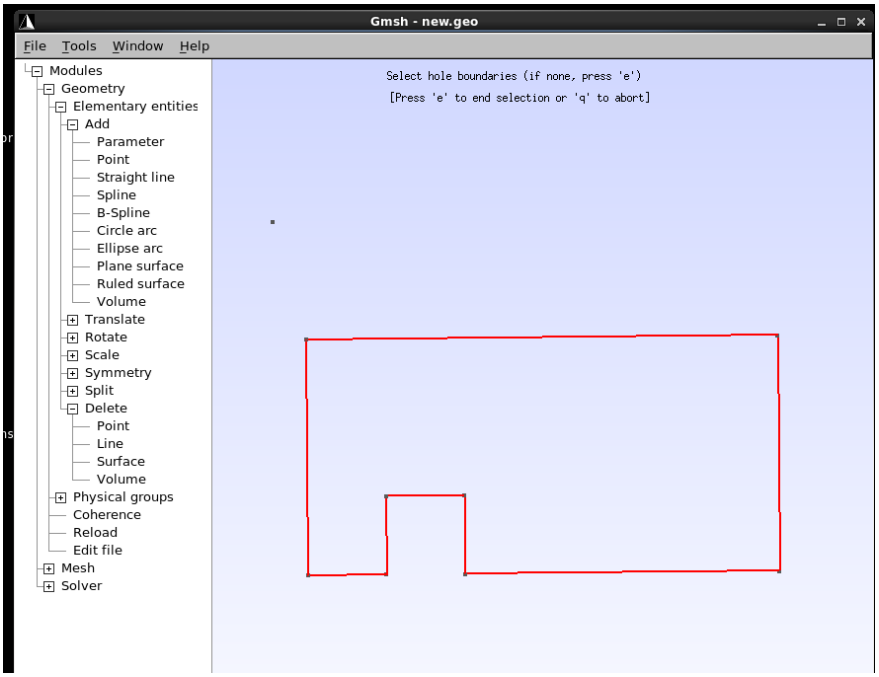
Download and install gmsh on your machine. Fire up the executable in the bin folder called gmsh, and go to *Geometry/elementary entities/add* and add points to the vertices of our domain.



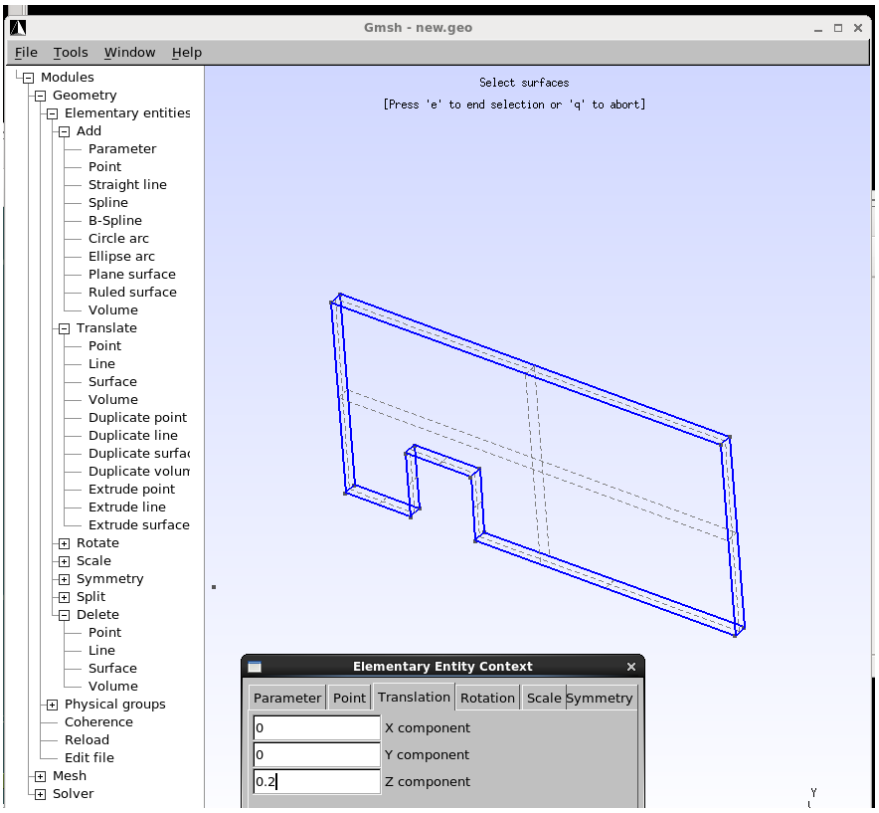
Then select the straight line option in the menu, and follow the instructions to stitch together the boundary of your domain.



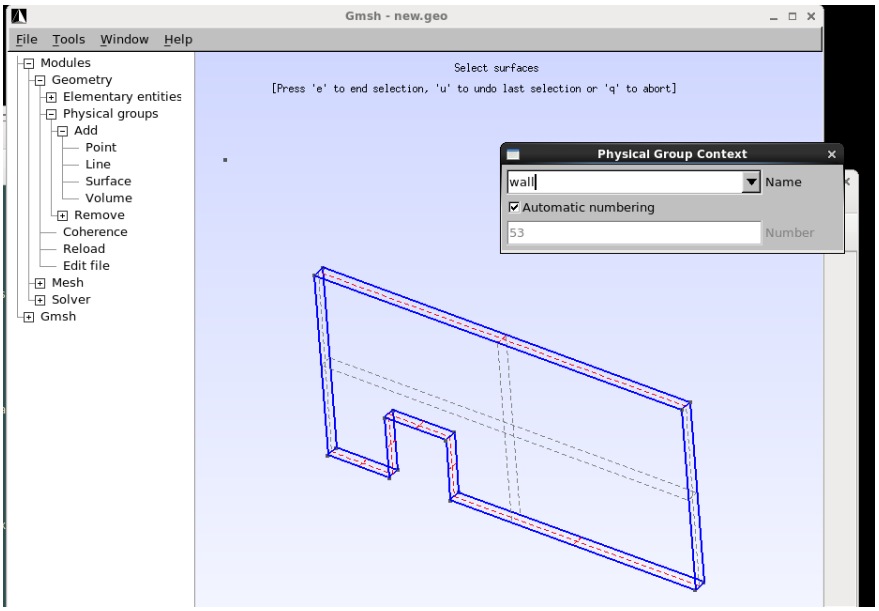
Once you've completed this step, select *Add/plane surface* to build a 2D planar surface out of your edges



Select *Geometry/Elementary entities/Translate/Extrude surface* and extrude your surface 0.2 in the $z-$direction.



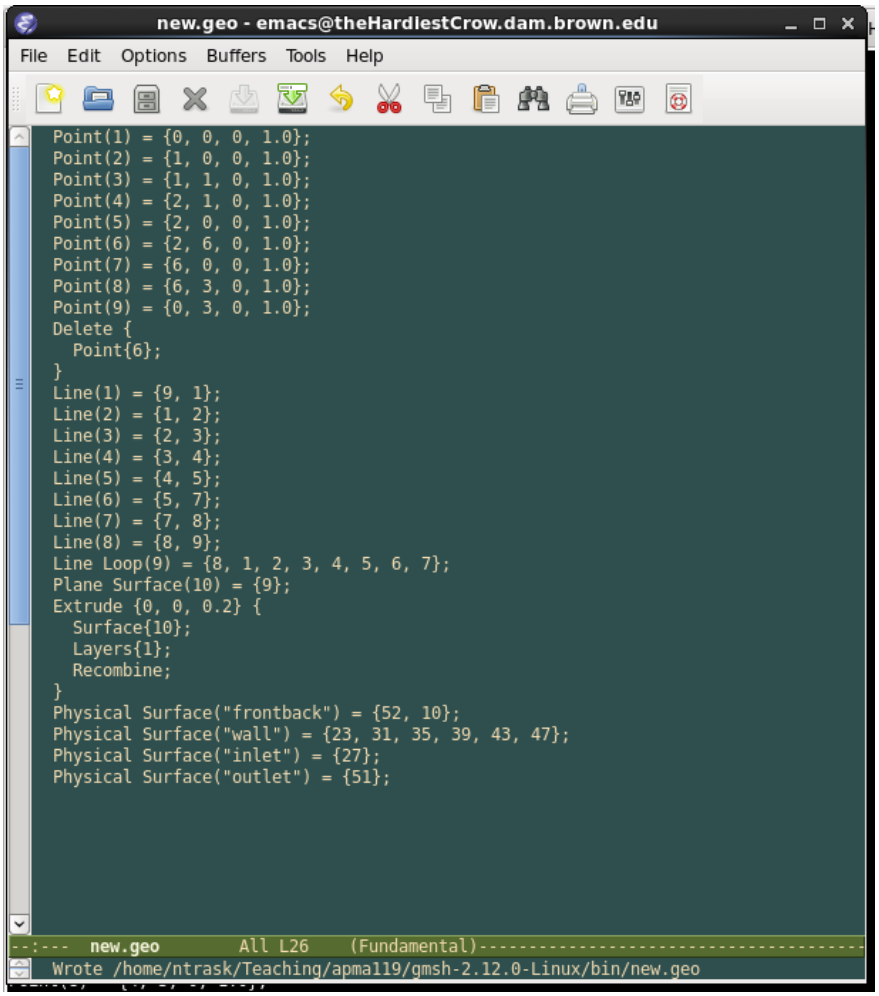
Great - at this point we've built up the geometric model of our domain. Next we need to label the patches. Go to *Geometry/Physical groups/add/surface*. Label the front and back as *frontback*, the walls as *wall*, the inlet as *inlet* and the outlet as *outlet*. Select *Geometry/Physical groups/add/volume* and label the volume of the domain.



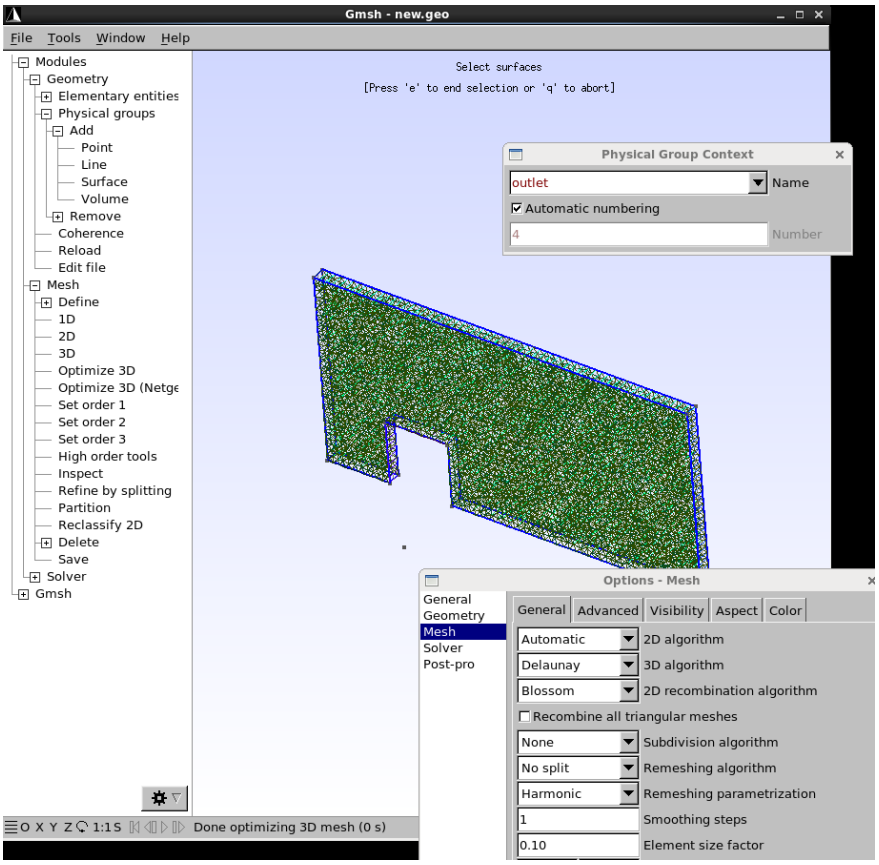
As we've been building all of this up, gmsh has been creating a .geo file in the same folder as gmsh. If you pop this file open in a text editor you should see the following description of your problem geometry.

```
Point(1) = {0, 0, 0, 1.0};
Point(2) = {1, 0, 0, 1.0};
Point(3) = {1, 1, 0, 1.0};
Point(4) = {2, 1, 0, 1.0};
Point(5) = {2, 0, 0, 1.0};
Point(6) = {2, 6, 0, 1.0};
Point(7) = {6, 0, 0, 1.0};
Point(8) = {6, 3, 0, 1.0};
Point(9) = {0, 3, 0, 1.0};
Delete {
  Point{6};
}
Line(1) = {9, 1};
Line(2) = {1, 2};
Line(3) = {2, 3};
Line(4) = {3, 4};
Line(5) = {4, 5};
Line(6) = {5, 7};
Line(7) = {7, 8};
Line(8) = {8, 9};
Line Loop(9) = {8, 1, 2, 3, 4, 5, 6, 7};
Plane Surface(10) = {9};
Extrude {0, 0, 0.2} {
  Surface{10};
}
Physical Surface("frontback") = {52, 10};
Physical Surface("wall") = {23, 31, 35, 39, 43, 47};
Physical Surface("inlet") = {27};
Physical Surface("outlet") = {51};
```

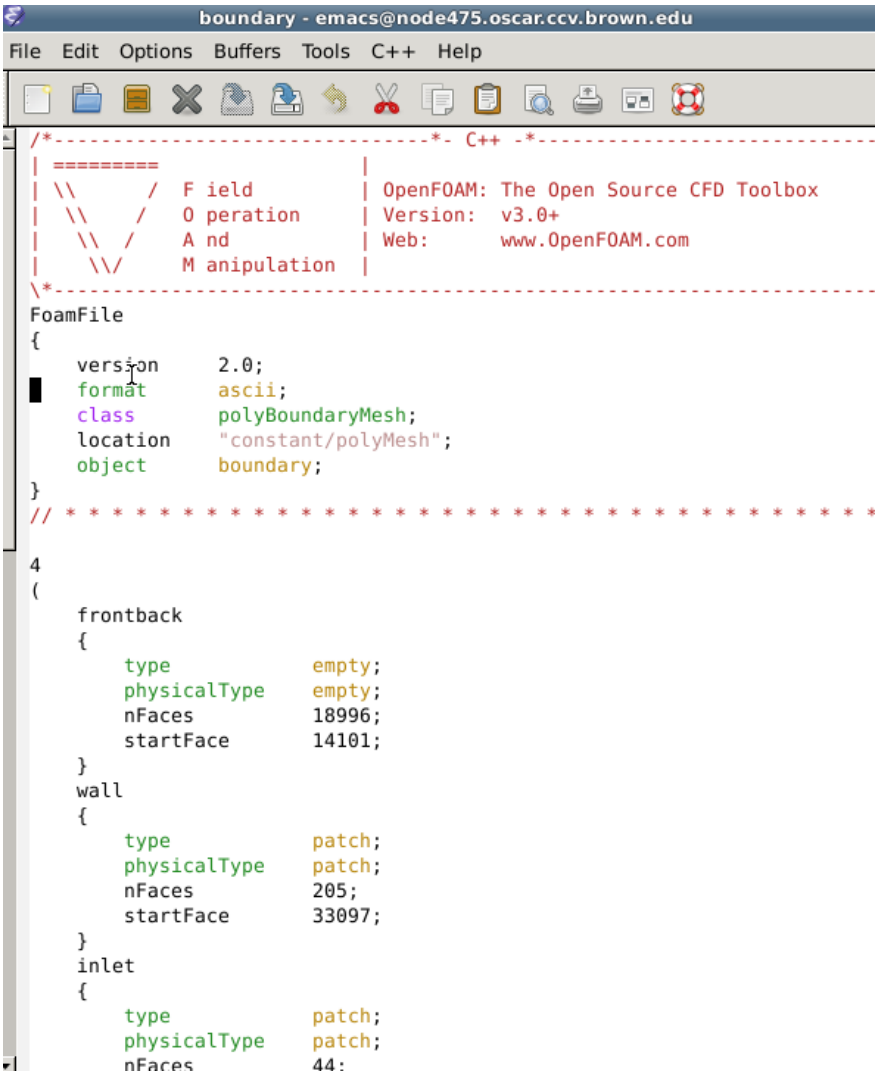
To build a 2D mesh in OpenFoam, we need to ensure that Gmsh only builds a mesh 1 element thick in the z-direction. To do that, we'll alter the .geo file as follows:



Close gmsh and reopen your altered geo file. Now we're ready to build a mesh. Click *Mesh/Define/3D* and gmsh will generate a mesh with default values. It'll look good, but be too coarse. If you select *Geometry/reload*, the mesh will be destroyed and we'll start over with our unmeshed geometry model. At the top of the screen, select *Tools/options*. Select the mesh sub-menu on the left of the window and the general tab at the top. The element size factor will alter the size of the mesh that we generate. If it's too small, our matrix system will be massive, the solvers will be slow, and we'll run our of memory on our computer. If it's too large, the error will be too big and we won't get a good answer. In general, the trick here is to start with a relatively coarse mesh, get a result, and then repeat while refining until your results don't change. We'll take a shortcut - you can trust me that a size factor of 0.1 is good.



Remesh, go to *File/save mesh* and save your mesh as a .msh file. We're almost finished. Next we need to copy the mesh onto your CCV account. This can be done from a terminal using scp or using winSCP. Once you've got the file into your CCV directory, we'll need to set up a new case directory. Since our code looks similar to the icoFoam tutorial we used before, it'll be convenient to make a copy of the channel flow case from the previous homework. Once you've made a copy of this, copy your msh file into the case directory, and run the utility *gmshToFoam yourmeshfilename.msh*. If you run *checkMesh* after this, you should see that the mesh was successfully read in. We have one final alteration to make before we can set up and run the solver. In a text editor, open up *constant/polyMesh/boundary*. You can see the list of each of the boundaries here with type listed as *patch*. For the periodic faces, we'll need to change *frontback* from *patch* to *empty*.



At this point, you can procede with setting up your case like the previous cases. In terms of the workflow of the previous solvers that you've used, these steps replaced the part where we alter *blockMeshDict* and run *blockMesh*. From here out, you should theoretically know the rest of the pieces that go into setting up and running your solver. Good luck!