

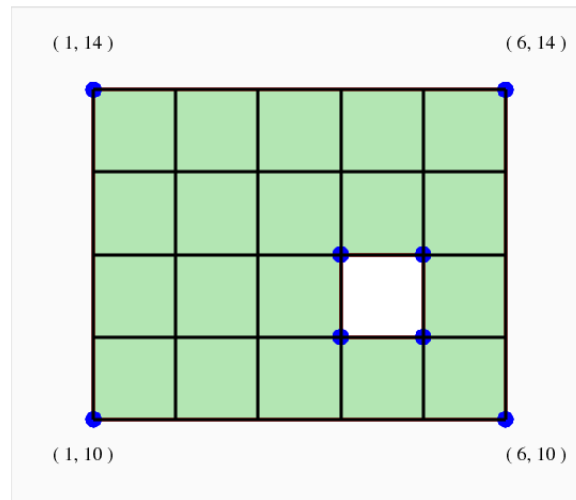
## ISC 5935 - Computational Tools for Finite Elements

### Homework #7

Assigned 22 October 2014, Due 29 October 2014

[http://people.sc.fsu.edu/~jburkardt/classes/fem\\_2014/homework7.pdf](http://people.sc.fsu.edu/~jburkardt/classes/fem_2014/homework7.pdf)

---



1. Get a copy of the **mesh2d** meshing program from [http://people.sc.fsu.edu/~jburkardt/classes/fem\\_2014/mesh2d\\_mod.tar.gz](http://people.sc.fsu.edu/~jburkardt/classes/fem_2014/mesh2d_mod.tar.gz). Using the diagram, define the boundary using a list of vertices **v**, and of edges **e**, with the proper clockwise or counterclockwise orientation. Pass this data to **mesh2d()** to create a mesh of the rectangular region. If you have done things correctly, the interior hole will not be meshed. The **mesh2d()** command includes an optional third argument, **hdata**, a data structure. One of its components is **hdata.hmax**, the maximum mesh size. I would suggest specifying this to be 0.5.

The mesh2d command returns points and triangles:

```
[ p, t ] = mesh2d ( v, e, hdata );
```

Use the **mesh2d\_to\_xml()** command to save this information as an XML file that FENICS could use.

```
mesh2d_to_xml ( 'rectangle.xml', p, t );
```

**Turn in:** your vertex and edge definitions (as MATLAB commands or an M file), a plot of your mesh, and the mesh file *rectangle.xml*.

---

2. Get a copy of the **Gmsh** meshing program from <http://geuz.org/gmsh>. Use Gmsh in the interactive mode. You are going to create and mesh a region that looks like a 5-pointed star. This will require defining 10 points and connecting them in the right way so that they **outline** a star. You don't have to be accurate, but your drawing should remind us of the right thing!. You begin with the Geometry/ElementaryEntities/Add/Point menu. Each time you click on a point on the screen and type "e" to save it. Then switch to the Line menu. Now you click on a start point, and move to the end point and type "e". Finally, when you have created the star region, move to the PlaneSurface menu and click on the boundary.

Now switch to the Mesh menu, and choose 2D. This gives you a crude mesh. Now choose Refine by Splitting. Finally, choose Save to save your mesh.

This will create a file with the extension ".msh", which we might rename *star.msh*. You should also have automatically created a file with the "geo" extension, which we might rename *star.geo*, which contains a description of your input.

Now get a copy of the **dolfin-convert** script from [http://people.sc.fsu.edu/~jburkardt/py\\_src/dolfin-convert/dolfin-convert](http://people.sc.fsu.edu/~jburkardt/py_src/dolfin-convert/dolfin-convert). Use the Python script to convert *star.msh* to *star.xml*, using a command like

```
dolfin-convert star.msh star.xml
```

This is the procedure you would need to use in order to create a mesh with Gmsh for use with Fenics.

**Turn in:** your Gmsh input file *star.geo*, and the mesh file *star.xml*.

---

3. We want to use **Fenics** to set up and solve a problem on a (slightly) unusual region, and to save the resulting mesh. We will use the **CircleMesh(center, radius, cellsize)** to create a mesh. We want the circle to be centered at (1.0, 0.0), with radius 1, and cell size of 0.25. The variable **center** is not a number, but rather a variable of type **Point()** that can be set by a command like

```
center = Point ( xvalue, yvalue )
```

Use Fenics to compute the finite element solution of

$$-\Delta u = f$$

on the circle, with  $f = (x - 1)^2 + y^2 - 1$ , and Dirichlet boundary conditions of  $u(x, y) = 0$ .

Use Fenics plot commands to display the solution and mesh. Use a **File** command to save the mesh to the file *circle.xml*.

**Turn in:** the python commands you used, and the mesh file *circle.xml*.

---