

# Lab 01 - Introduction

## Numerical Solution of PDEs Using the Finite Element Method

MHPC P2.13\_seed

Martin Kronbichler kronbichler@lnm.mw.tum.de and Luca Heltai luca.heltai@sissa.it

---

### 1. Setup

- Edit file `~/bashrc` to contain the line  
`source /scratch/smr2909/enable.sh`  
and close and re-open your terminal. You can use `gedit ~/bashrc`  
to open an editor. Check that this worked by typing `echo $DEAL_II_DIR`  
You should see `/scratch/smr2909/deal.II/install` printed to the screen.
- Please note, inside `/scratch/smr2909/` there are the following folders:
  - `labs/` – a folder with exercise sheets and example programs
  - `bin/` and `apps/` – several programs (you shouldn't need to access them directly, because they will be imported into your PATH automatically)
  - `libs/`, `candi/`, `candi-build` – libraries deal.II depends on.
  - `deal.II` – source, build, and installation of deal.II.
  - `deal.II/dealii/examples/` – all tutorial programs.
- to make a copy of tutorial 1, configure, compile, and run it:

```
cp -r /scratch/smr2909/labs/lab01/step-1 ~/
cd ~/step-1
cmake .
make
./step-1
```
- IDE: open `qtcreator .`

### 2. Tasks for tutorial step-1:

1. See documentation at  
[https://www.dealii.org/8.4.0/doxygen/deal.II/step\\_1.html](https://www.dealii.org/8.4.0/doxygen/deal.II/step_1.html)
2. Compile and run inside qtcreator and look at the output.
3. Comment out the `.set_manifold(0, ...)` line in `second_grid()`. What happens now?
4. Create an image of an L-shape domain (add a function `third_grid()` to step-1) with one global refinement.
5. Now change the output format of the previous example to `vtk` and open the new file in paraview.
6. Refine the L-shaped mesh adaptively around the re-entrant corner several times but with a twist: refine all cells with the distance between the center of the cell and re-entrant corner is smaller than  $1/3$ .
7. Output mesh two as an `svg` file instead of `eps`. Open it in a browser to display it (firefox for example).

8. Create a helper function that takes a reference to a `Triangulation` and prints the following information: number of levels, number of cells, number of active cells. Test this with all of your meshes.
9. Generate a circle using `GridGenerator::hyper_ball()` in 2d: use a `SphericalManifold` everywhere, only on the boundary, or on all cells except the center cell and refine the mesh globally twice.
10. Go into `second_grid()` and remove the last line (`.set_manifold(0);`). The program will crash when you run it. Try to find out what is going on by debugging the program (“Debug” -> “Start debugging” in qtcreator) and stepping through the function `second_grid()`. You can fix this problem in a more elegant way than putting the line you removed back in. How? See the tutorial description for more info.
11. Bonus: Create a mesh that represents the surface of a torus and refine it 2 times globally. Output to vtk format and check the output. Note that your `Triangulation` needs to be of type `Triangulation<2,3>`, which we will discuss later this week.
12. Bonus: Take a look at step-49 and read the included `.msh` file in your modified step-1 program.