Lab 1

Math 9830 Timo Heister, heister@clemson.edu

These are just exercises, do not hand them in as homework.

Note: it is a good idea to make of copy of the `step-xy` directory (call it `my-step-xy` for example) and keep the original one around.

Note: you will end up creating different images when doing the following tasks. You might want to copy some of those into your journal (to practice how to do that and so that you can compare them easily). Especially for question 2.

1. For tutorial step-1:
   (a) Create an image of an L-shape domain (add another function to step-1).
   (b) Refine the mesh in a) adaptively around the re-entrant corner.
   (c) Create a helper function that takes a `Triangulation` and outputs the following information: number of levels, number of cells, number of active cells. Test this with the (now three) meshes.
   (d) Output the mesh as an svg file instead of eps. Open it in a browser to display it.
   (e) Create a 3d cylinder and refine it 3 times (globally). Try outputting to gnuplot format too.
   (f) Bonus: create a 3d unit cube and create a loop that in each step a) refines globally once, b) outputs the number of active cells, c) the amount of memory in megabytes required to store this mesh (look for a function called `memory_consumption()`). Do this for 6 global refinements first. How many refinements can you fit into memory of your computer (typing `free` in the terminal tells you how much memory you have)?
2. For tutorial step-2:
   (a) How does the pattern change if you increase the polynomial degree from 1 to 2 or to 3?
   (b) How does the pattern change if you use a globally refined (say 3 times) unit square?
   (c) How many entries per row do you expect for a Q1 element (assuming four cells are around each vertex)? Check that this is true for the mesh in b) (look for `row_length(i)` and output them for each row). How does that change with a different mesh? Can you construct a mesh (without hanging nodes) that has a row with more entries?
   (d) Print all entries for row 42 for the original renumbered sparsity pattern.
   (e) Are these patterns symmetric? Why/why not?
   (f) Compute and output statistics like the the number of unknowns, bandwidth of the sparsity pattern, average number of entries per row, and fill ratio.
   (g) Bonus: figure out a way to write the sparsity pattern into a file that you can either read in using Matlab or write it as a .PPM image file.