Project 3
SF2568 Program construction in C++ for Scientific Computing

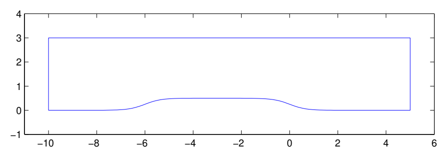
October 5, 2018

In this project you will implement a general class for modelling "4-sided" domains and structured grids on them. Consider the following skeleton of an abstract base class:

```
class Curvebase {
protected:
  double pmin; // minimal value for p
  double pmax; // maximal value for p
  bool   rev;  // orientation of the curve
  double length;
  .....
  virtual double xp(double p) = 0;
  virtual double yp(double p) = 0;
  virtual double dxp(double p) = 0;
  virtual double dyp(double p) = 0;
  double integrate(double p); //arc length integral
  .....
public:
  Curvebase();   //constructor
  double x(double s);  //arc length parametrization
  double y(double s);  //arc length parametrization
  .......
};
```

Task 1 Complete the class by writing the non-virtual functions. Add more variables or functions to the class, if you find it necessary.

Task 2 You will generate a grid on the domain in the figure below.



The corners are located at $(-10,0),(5,0),(5,3)$ and $(-10,3)$. The lower boundary is given by the function

$$f(x) = \begin{cases} \frac{1}{2}\frac{1}{1+exp(-3(x+6))}, & x \in [-10,-3) \\ \frac{1}{2}\frac{1}{1+exp(3x)}, & x \in [-3,5] \end{cases}$$

Derive classes that are needed to represent the boundary curves of the domain in the figure above from the base class. Test the classes by using it in a simple main program.

Task 3 Design a class `Domain` as outlined in the lecture. The class should contain four boundary curves of type `Curvebase` and have capability for generating a grid on the domain. Write a main program which generates the grid. Use the algebraic grid generation formula. *Hint: Be very careful about the orientation of the boundary curves!*

Task 4 Add a function to the class Domain to write the grid to a file. The simplest is to use `cout` and write to an ASCII-file. A better way is to use the functions `fopen`, `fwrite` and `fclose` (or the corresponding `ofstream` class) to output the grid in binary format. We give an example below of how they are used to write a vector x consisting of $n \times m$ doubles:

```
#include<cstdio>
FILE *fp;
fp =fopen("outfile.bin","wb");
fwrite(x,sizeof(double),m*n,fp);
fclose(fp);
```

Use the UNIX commands `man fopen` and `man fwrite` to obtain more information about these functions. The grid can be viewed in Matlab. To read a binary file, use the Matlab functions `fopen` and `fread`.

Task 5 (optional) You will generate a stretched grid to better resolve the lower boundary. Denote the reference coordinate mapped into the vertical direction (ξ or η) by s. Use the stretching

$$s = T(\sigma) = 1 + \frac{\tanh \delta(\sigma - 1)}{\tanh \delta},$$

where $\delta = 3$ and $\sigma \in [0,1]$. Choose a uniform grid with respect to σ. The x-points should remain uniform.

The programming exercises should be done individually, or in groups of two. Hand in a report containing:

- Comments and explanations that you think are necessary for understanding your program.
- A picture of the generated grid, using approximately 50×20 grid points.
- Printout of the source code.
- E-mail the source code to `hanke@nada.kth.se`.