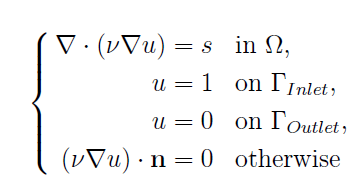
|  |
| --- |
|  |
| Poisson problem in C++ |
| Programming for Engineers and Scientist |
|  |
| **Sònia Garrido Ballart** |
| **Alba Navarro Casanova** |

|  |
| --- |
|  |

# Statement of the Problem

We are asked to implement a FE code for Poisson problem using C++ in a similar way as we did in the first assignment of the course.

The problem that we aim to solve is:



Where the source term s = 0 and the diffusivity ν = 1 are given.

The main difference with the Matlab code that we have developed in the first part of the course is that the code has to be adapted to the C++ language and we are explicitly asked to write and object oriented program. In the following section we will explain which classes are created and how the program is structured.

# Structure of the Program

As we have told to develop the code to solve the Poisson problem we are using object oriented programming. As this was really new for us we have only implemented one of the eight problems that we were solving in the first assignment. That means that our problem is only for 2D structures using triangular linear elements. The 2D problem with triangular quadratic elements and the quadratic linear and quadratic elements will be solved in a really similar way but they are not implemented. For the 3D case changes the way that we solve the linear system, this case is not implemented.

To solve the FEM problem we need input files with the coordinates of the nodes of the mesh of our problem and another with the connectivity matrix that relates the elements with the nodes of the mesh. This information is stored in text files that have to be read by our program. We also need to impose the boundary conditions, in this particular case we only have Dirichlet boundary conditions in the inlet () and on the outlet () nodes, this information is also stored in a text file and has to we read. The diffusion coefficient and the source term also have to be determined, in this case we consider and for all the nodes.

Depending on the problem that we are solving we will have different integration points and integration weights as so for the shape functions and its derivatives. We need these integration parameters and shape functions to generate the stiffness matrix K and the vector f. This is done in different ways depending on the dimensions of the element that we are studding.

The C++ program does the function of calculate the solution of the problem; we use Paraview as a postprocessor. To see the results of our unknown we need to write them in a .vkt file that can be opened in Paraview.