1 Introduction

This document contains all the information needed to start using this application. It was coded with C++ on Visual Studio 2013 and multiple libraries were used (e.i. muparser[1] , CGAL for the mesh[2], Armadillo[3] and Qt[4]). The application is the result of a personal project and is free to use. It is offered as-is.

The application can solve a partial differential equation in 2D in a rectangular domain with the finite element method. The general equation that can be solved is

$$\vec{a}(x,y) \cdot \vec{\nabla} u(x,y) - \vec{\nabla} \cdot \{b(x,y) \vec{\nabla} u(x,y)\} + c(x,y) u(x,y) = f(x,y),$$

with the boundary conditions :

$$u(x,y) = g(x,y)$$
$$b(x,y) \vec{\nabla} u(x,y) \cdot \vec{n} = h(x,y).$$

2 How to use it

Each section in the application UI will now be explained.

Size of elements : The mesher (CGAL) tries to create a mesh with triangular elements that are smaller than the chosen value.
Density of points : A high value results in a high density of points in each element. The default value is 5. Take note that it doesn't mean that there are 5 points in each element.
PDE parameters : The definition of each parameter can be found in the general equation in the introduction. The value can be a constant or a function e.g $x^2 + y^2$. See muparser documentation for all the available functions.
Boundary positions : The domain is a rectangle. It is defined by 4 constants $xmax$, $xmin$, $ymax$, $ymin$.
Boundary values : There is two types of boundary conditions : the Neumann boundary condition ($h(x,y)$) and the Dirichlet boundary condition ($g(x,y)$). The first one fixes the normal derivative of $u(x,y)$ and the second one fixes the value of $u(x,y)$. See the tooltip in the application for the details about the position of each boundary condition. Also, the boundary condition can be a function of x and y.
Degree of precision : This is the degree of the polynomial interpolation in each element.

2.1 Need more info ?

All relevant data is saved in the main directory after each simulation. The color plot is saved in plot.png and the solution is saved in data.csv. The source code is also available and more information can be found in there.

[1] muparser.beltoforion.de/

[2] www.cgal.org/

[3] arma.sourceforge.net/

[4] www.qt.io/