Introduction

This is the report detailing the project done by Stefano Costa for the numerical methods for PDE course.

FreeFem is used for the programming part.

The programming part is not discussed in depth since the code has been commented in detail. It is emphasized that almost the entirety of code was **not** seen in class, and it may be of interest to review it.

In addition, the program outputs are not reported in detail, since most of them were used to construct this report (images and numerical values)

Problem

Let $\Omega = (-1,1)^2 \setminus [0,1)^2$. Consider the problem:

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega \\ u = g_i & \text{on } \partial\Omega \end{cases}$$

with:

$$g_1(x,y) = e^{-x}\sin(y)$$
$$g_2(x,y) = r^{\frac{2}{3}}\sin\left(\frac{2}{3}\left(\theta - \frac{\pi}{2}\right)\right)$$

Exact solutions

Here I will prove that when g_i is used in the boundary condition, then g_i is the exact solution. Since it already coincides with the boundary conditions, it is sufficient to show that: $\Delta g_i = 0$.
But that is a matter of computing the derivatives:

$$\frac{\partial g_1}{\partial x} = -e^{-x}\sin(y) \qquad \frac{\partial g_1}{\partial y} = e^{-x}\cos(y)$$
$$\frac{\partial^2 g_1}{\partial x^2} = e^{-x}\sin(y) \qquad \frac{\partial^2 g_1}{\partial y^2} = -e^{-x}\sin(y)$$

From which we find:

$$\Delta g_2 = \frac{\partial^2 g_1}{\partial x^2} + \frac{\partial^2 g_1}{\partial y^2} = e^{-x}\sin(y) - e^{-x}\sin(y) = 0$$

Hence it solves the problem.
As for the g_2 function, the steps are repeated by applying the chain rule. In this way we obtain that $\frac{\partial^2 g_2}{\partial x^2} = -\frac{\partial^2 g_2}{\partial y^2}$, and then conclude that $\Delta g_2 = 0$.

Mesh

For each h is here reported the image of the mesh created. The domain $\overline{\Omega}$ is partitioned into triangles by first subdividing it into squares with sides of length h and then dividing each square by its north east diagonal into two triangles.

Case $h = 2^{-3}$

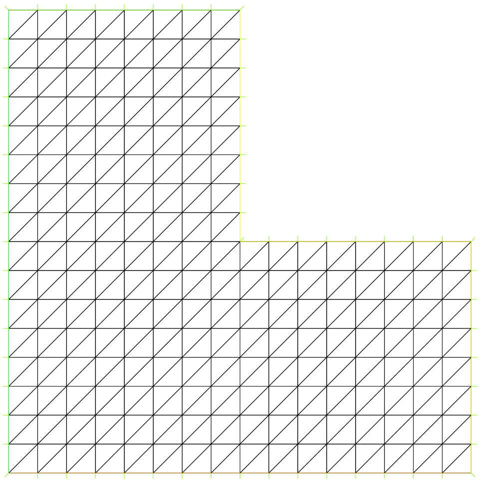

Figure 1: Mesh as stated, with $h = 2^{-3}$

Case $h = 2^{-4}$

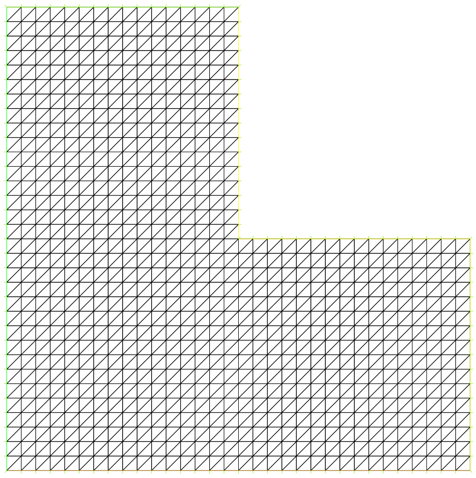

Figure 2: Mesh as stated, with $h = 2^{-4}$

Case $h = 2^{-5}$

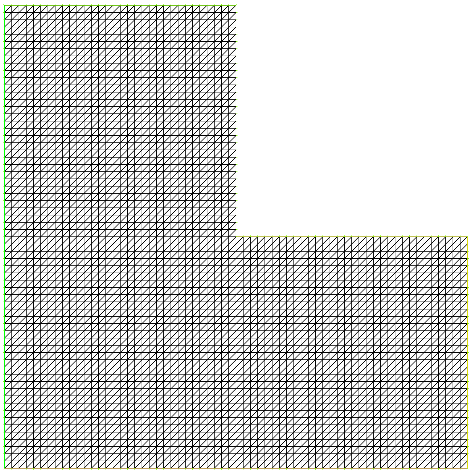

Figure 3: Mesh as stated, with $h = 2^{-5}$

Case $h = 2^{-6}$

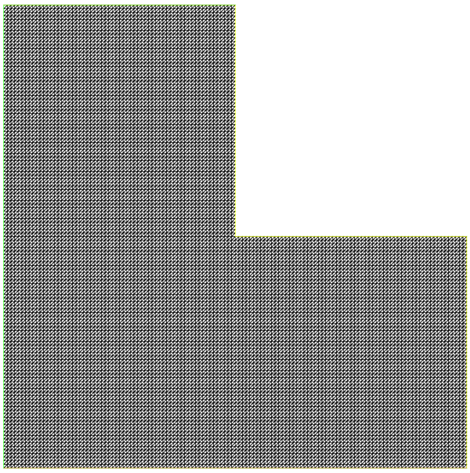

Figure 4: Mesh as stated, with $h = 2^{-6}$

Approximated solution - results

On the meshes previously showed, the approximated solution u_h is computed. The graphs of the approximate solution and the exact solution after applying the piecewise linear interpolation operator are shown below.

Using g_1

Because the function g_1 is quite smooth, the approximation of the solution is quite accurate, so little difference can be seen between the exact interpolated solution and the calculated solution.

Case $h = 2^{-3}$

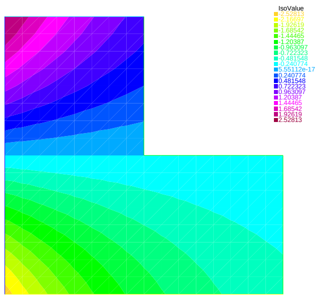

Figure 5: Approximated solution for $h = 2^{-3}$, using g_1

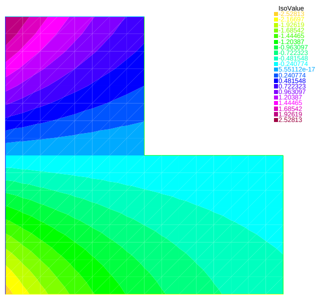

Figure 6: Exact solution interpolated for $h = 2^{-3}$, using g_1

Case $h = 2^{-4}$

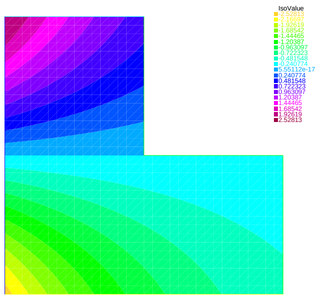

Figure 7: Approximated solution for $h = 2^{-4}$, using g_1

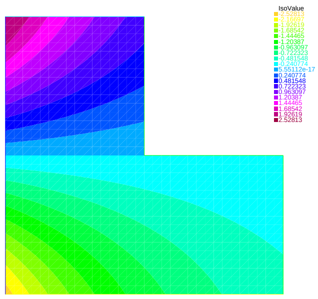

Figure 8: Exact solution interpolated for $h = 2^{-4}$, using g_1

Case $h = 2^{-5}$

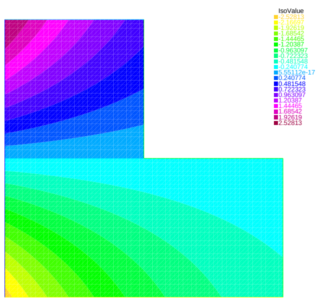

Figure 9: Approximated solution for $h = 2^{-5}$, using g_1

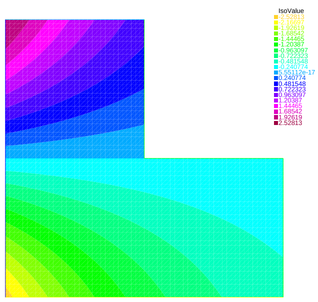

Figure 10: Exact solution interpolated for $h = 2^{-5}$, using g_1

Case $h = 2^{-6}$

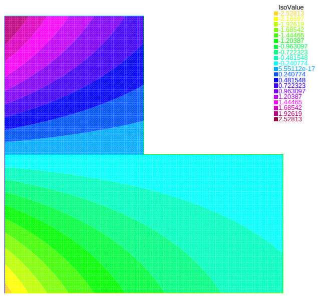

Figure 11: Approximated solution for $h = 2^{-6}$, using g_1

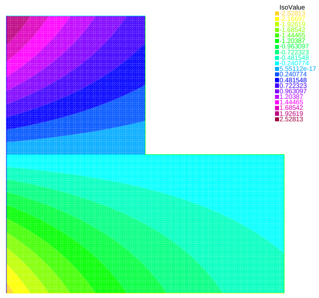

Figure 12: Exact solution interpolated for $h = 2^{-6}$, using g_1

Using g_2

Regarding the use of g_2, the situation is different. In fact, the function has a behavior that challenges the approximation in the area $y \geq 0$, $x \geq 0$ because of θ function.
During execution, with some zooming, one can also notice with the naked eye the improvements that result from refining the mesh as h varies.

Case $h = 2^{-3}$

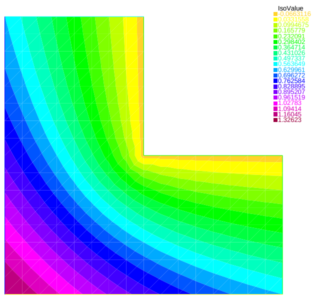

Figure 13: Approximated solution for $h = 2^{-3}$, using g_2

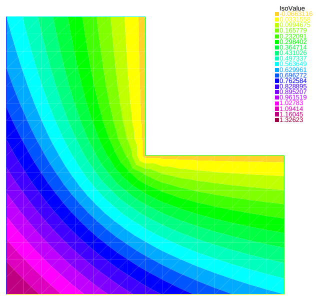

Figure 14: Exact solution interpolated for $h = 2^{-3}$, using g_2

Case $h = 2^{-4}$

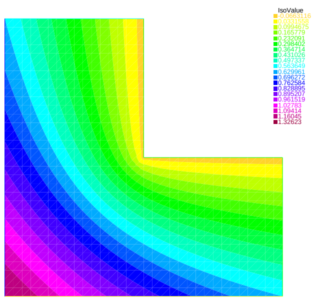

Figure 15: Approximated solution for $h = 2^{-4}$, using g_2

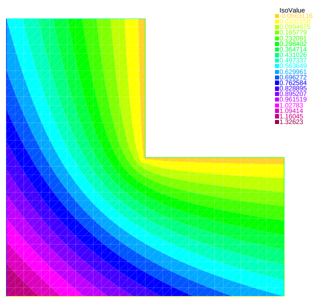

Figure 16: Exact solution interpolated for $h = 2^{-4}$, using g_2

Case $h = 2^{-5}$

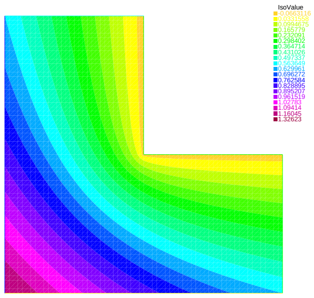

Figure 17: Approximated solution for $h = 2^{-5}$, using g_2

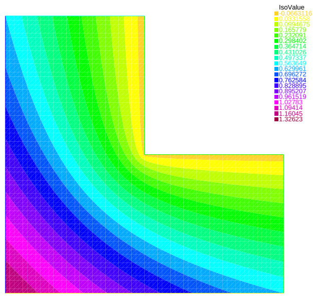

Figure 18: Exact solution interpolated for $h = 2^{-5}$, using g_2

Case $h = 2^{-6}$

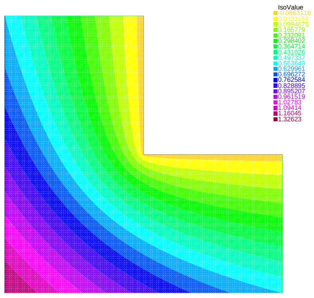

Figure 19: Approximated solution for $h = 2^{-6}$, using g_2

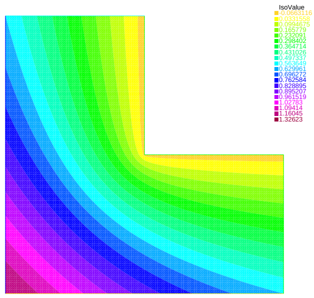

Figure 20: Exact solution interpolated for $h = 2^{-6}$, using g_2

Errors

Following are the values of the error norms obtained for different values of h (reported as different values for k).

Using g_1

k	$\|\|\prod_h^1 u - u_h\|\|_{L^2(\Omega)}$	$\|\|\nabla\left(\prod_h^1 u - u_h\right)\|\|_{L^2(\Omega)}$	$\|\|\prod_h^1 u - u_h\|\|_{L^\infty(\Omega)}$
3	0.000156889	0.000643941	0.000190367
4	4.01708e-05	0.000163942	4.82107e-05
5	1.01058e-05	4.11781e-05	1.21057e-05
6	2.53069e-06	1.03069e-05	3.02897e-06

Table 1: Evaluation of the norms using g_1

As expected from the graphical results, the errors are very low and decrease significantly as the mesh refinement increases. Calculating the convergence rate as seen in class yields:

	$\|\|\prod_h^1 u - u_h\|\|_{L^2(\Omega)}$	$\|\|\nabla\left(\prod_h^1 u - u_h\right)\|\|_{L^2(\Omega)}$	$\|\|\prod_h^1 u - u_h\|\|_{L^\infty(\Omega)}$
p	1.99759	1.99826	1.99879

Table 2: Convergence rate using g_1

Since the solution is sufficiently regular, the theory tells us that the rate of convergence is 2.

Using g_2

k	$\|\|\prod_h^1 u - u_h\|\|_{L^2(\Omega)}$	$\|\|\nabla\left(\prod_h^1 u - u_h\right)\|\|_{L^2(\Omega)}$	$\|\|\prod_h^1 u - u_h\|\|_{L^\infty(\Omega)}$
3	0.00591396	0.0435355	0.0201109
4	0.00257775	0.0277815	0.0130302
5	0.00107831	0.0175876	0.0082978
6	0.000441391	0.0111008	0.0052495

Table 3: Evaluation of the norms using g_2

Unlike the previous case and as suspected from the graphical results, large errors are obtained and accuracy slightly improves as h decreases.

Calculating the convergence rate as seen in class yields:

	$\|\|\prod_h^1 u - u_h\|\|_{L^2(\Omega)}$	$\|\|\nabla\left(\prod_h^1 u - u_h\right)\|\|_{L^2(\Omega)}$	$\|\|\prod_h^1 u - u_h\|\|_{L^\infty(\Omega)}$
p	1.28864	0.663895	0.66055

Table 4: Convergence rate using g_2

The results obtained are caused by the lower regularity of the function (as we guessed from the graphs), for which we cannot apply the theory and therefore cannot get a rate of convergence of 2, but much lower.